Tender
5/9/2023

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 will be part of the contract

❗ The text/question contains requirements to be met

① The question is weighted and included in the evaluation

ⓘ The question is asked for information only

✓ The question is marked for special follow-up

⚠ Updated section or question

☰ The text is included in the qualification

🌐 The text will be published in the contract catalogue

ESPD The text/question contains ESPD requirements

⚖ The question is weighted and included in the evaluation

🔒 The question is answered by the buyer

⚠ The answer does not meet the requirement in the question

Tenderers

Supplier	Tender	Corporate ID	Qual.
Slovenia	Better	8505047000	

Contents

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

May 31st 10:00-12:00AM or alternatively, 1:00-3:00PM

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

Better is a technology solution provider offering an open data, Digital Health Platform, named Better Platform. Better Platform is a high-performance Digital Health Platform (DHP) providing three distinct core capabilities:

- a unified, personalized application experience: Better Portal
- the ability to rapidly deliver new applications using low-code tools: Better Studio
- a vendor-neutral data core enabling a longitudinal care record: Core Data Services

Better Portal is a web-based clinical portal application for care teams (e.g. doctors, pharmacists, nurses) and citizens. Better Portal includes User Management, User Dashboards, Patient Lists, Patient Dashboards, Patient Banners, Views, Actions and Auditing.
Better Platform Studio is a client-side application made to interact with EHR Server and other Better infrastructure components. It serves as a visual representation of a server logic – a GUI to make interaction visual, more straightforward and apparent.

The studio provides the users with the capability to create user entry forms and query stored data. It includes four tools:

- The Form Builder
- AQL builder
- Archetype Designer
- Administration tools

Better Platform Core Services store, manage, query, retrieve and exchange structured and unstructured electronic health record data in real-time, based on the latest release of openEHR specifications. Better Platform Core Services consist of the following services: Core Data Services, Demographics/MPI and Terminology.

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

Better is a registered company with offices in 3 countries, Slovenia, United Kingdom and Germany.
Company headquaters are located in Ljubljana, Slovenia. To deliver our solutions to more than 20 markets globally, we work through a wide network of local partners, which in case of Sweden are the following:

- TietoEvry
- CGI
- EY
- Stratetiq
- Strikersoft
- ServiceWell
- Atea
- Microsoft
- Medanets

Also, we hereby confirm that Better has the capacity to deliver our solutions and services via existing National Swedish Framework agreements.

c. Describe the overall architecture of your Solution.

Text field

Better Platform is a high-performance Digital Health Platform designed for heavy user interaction, low-code application building, and storing, managing, querying, retrieving, exchanging structured and unstructured electronic health record data. Logical architecture is presented on the diagram provided in the external link here - : https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EbFO8cJ67gBJiAU-OprKvDABbU3gLCC2WHwlxFh8DZTa3w?e=PHrDf7

Better Platform is a service-oriented (SOA) platform for deploying or integrating Electronic Health Records (EHRs) and EHR applications locally, regionally, or internationally. It is built on a decoupled SOA, with microservices that enable independent horizontal and vertical service scaling capabilities. As healthcare evolves and new software modules are added to implement specialized information generated by distinct specialties in hospitals, Better Platform is designed for stepwise evolution over time. This ensures that the system can adapt to changing needs and requirements, while maintaining data integrity, high performance, and good scalability. One of the key strengths of the Better Platform is its support for multi-tenancy, clustering, and sharding as well as replica servers for historic data and data views.

Open APIs

Better Platform™ provides a set of Open APIs for secure access to health data, making application development easier. API technologies are continually changing; currently, developers demand REST APIs carrying a JSON payload. The full specifications of the APIs to connect applications to the platform are freely available.
The EHR Server exposes an OpenEHR REST API coarse-grained interface supporting check-out, commit and query operations that implement runtime archetype and template processing logic and provides API with the ability to create, modify and query EHRs. The API layer is

stateless by nature. EHR server exposes the following REST API groups:
- EHR: management of EHRs.
- Composition: management of compositions
- Query: executing AQL queries
- View: retrieval of EHR data based on pre-stored queries (similar to views in relational databases)
- Presentation: generic composition as document displays of EHR data
- Template: management of OpenEHR operational templates

HL7 FHIR APIs are also provided as an alternative way to access data in EHR Server.

Event-Based / Asynchronous Coupling

Better Platform can react to specific compositions as they are stored. This is very similar to triggers in relational databases. Each event registered by the server has an AQL query associated with it. A non-empty result of an AQL query triggers an event. Each event also has a 'phase' – either synchronous or asynchronous. Synchronous events are executed as a part of an incoming composition. They have access to composition's data and can thus add additional compositions to be committed. They can also veto specific parts of a composition from being committed. Asynchronous events are executed after a composition has been successfully committed, and a result returned to the calling application. The executing event will receive all data that is the result of the AQL associated with it. For composition updates, previous values and new values are available to the event.
To guarantee the delivery and support high throughput, Better Platform uses the Apache Kafka event streaming platform (ESP) to provide data streams to other platform components or 3rd party systems.
All Better Platform services can publish events to the ESP. Services like Data views, Decision Support, and others subscribe to these events. The ESP service is designed so that producers and consumers are fully decoupled and agnostic of each other, which is a crucial design element to achieve high throughput, scalability, and resilience.
This approach enables Better Platform to easily scale horizontally and vertically in order to meet any environment's needs, whether it's high numbers of concurrent users, a massive dataset to analyse, or both at the same time.

Standards-based

Better Platform fully supports openEHR standards for health data persistence. We offer a rich set of standard interfaces to support integration with other systems and components, including HL7 v2.x and HL7 FHIR and IHE XDS and ATNA. We also support core terminologies such as SNOMED CT, LOINC, and ICD-10. In addition to the interoperability services outlined above, the Better Platform also contains an embedded version of the Rhapsody integration engine from Lyniate. Rhapsody is used by over 1.500 customers to integrate with all major healthcare systems across a range of domains, including EHRs, Labs, eMPI, Population Health, and Care Coordination. The Rhapsody platform supports all major standards and formats, among them HL7 (v2 and v3), HL7® FHIR, CCDA, NCPDP, X12, IHE, DICOM, JSON, and XML, in addition to connecting data using non-standard or proprietary interfaces. Rhapsody is an optional module with a consumption-based pricing mode.

Access and Security Services

Better Platform™ provides a strong, flexible, consistent, and high-performance security infrastructure while minimizing its application performance burden. This security architecture is based on authentication, authorization, auditing, and database encryption.

- Authentication. Better Platform™ supports basic, OAuth2 and multi-factor authentication. Several options are available including Keycloak, ADFS, Azure B2C
- Authorization. Using comprehensive authorizations, you can easily assign and manage role- and application-based resource access privileges either as role-based access control (RBAC) or

attribute-based access control (ABAC)
• Auditing The audit server collects audit logs from all platform services for traceability, dispute resolution, security, privacy, and compliance with policies (external and internal). It fully supports the IHE ATNA standard.
• Database encryption. Better Platform™ encrypts data-at-rest and data-in-motion. To protect entire databases, it offers block-level encryption.

By default, role-based access control is used for platform APIs. Additionally, many cases require more granular access control based on specific attributes giving specific persons a right to access a data set. Attribute-based access control (ABAC) allows writing access rules based on relations between parties and attributes defined within the access context.

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

Task Planning

For comprehensive management and more complex clinical pathways, Better Platform is planning to implement Better Platform Tasks: Better Platform Tasks is designed process engine for health care - a suite of tools for the modelling, execution and monitoring of clinical and medico-administrative processes. It is based on the OpenEHR open standard called OpenEHR Task Planning Model Specification and aims to be a fully conforming implementation of that standard.

The Better Platform Tasks suite consists of three main-level systems:

- the Better Platform Task Engine, a clustered server executing, controlling and monitoring the execution of a clinical process
- the Archetype Designer, a tool for modelling the clinical processes
- the Better Platform Task Planning Cockpit, a visual tool for the execution, adjustment and monitoring of medical task plans.

In addition to these two systems, there are at least two other systems, external to the Better Platform Tasks stack, that need to participate in order for a medical process to be successfully executed: the Better EHR Server and the Controlling Application (an application or a system that uses all the above–mentioned systems to achieve a specific clinical or medico-administrative goal. It activates the plan(s), receives and sends notifications to the Task Engine and integrates the task planning sub-system with its own business logic.).
Due to the complexity of the implementation of the task planning specifications, at Better we have decided to go towards the apporach of Simple tasks and BPM, described below. We are open to selling the source code of the task planning solution to the interested party.

Simple Tasks

Better Platform addresses the workflow and intervention requirements by using openEHR entry types, Entry type INSTRUCTION, its subpart ACTIVITY to specify interventions in the future, and the Entry subtype ACTION to record what has happened. Several important features are provided in this model, including:
• a single, flexible way of modelling all interventions, whether they be single drug medication orders or complex hospital-based therapies.
• a way of knowing the state of any intervention, in terms of the states in a standard state machine, shown below; this allows a patient's EHR to be queried in a standard way to return "all active medications", "all suspended interventions" etc.
• a way of mapping particular care process flow steps to the standard state machine states, enabling health professionals to define and view interventions in terms they understand.
• support for automated workflow without requiring it.

The approach has no logic and is suitable for simple plans where users can choose what tasks to create, add their own, change the timing of jobs, etc. They can also be used for tasks not related to patients. The features are more suitable for cases where we need to create tasks in a more ad-hoc manner. Tasks are created directly by the application built on top of the Better Platform. Tasks can be created automatically based on the timing specified in the INSTRUCTION timing or by the application itself or a combination where tasks are created from INSTRUCTION timing and then further adjusted (added, removed, moved) by the application. Task API provides a simple means of complementing the openEHR INSTRUCTION/ ACTION flow. The tasks ("shadow" actions) can be queried by time, assignee, patient, etc.

CMN / BPM

Better Platform also provides the capability to implement Web 2.0 visual design capabilities, and is agnostic to modelling tools (like Camunda, ProcessMaker or other), allowing for user led design of clinical pathways and order sets, that are further propagated into activities and tasks across applications, screens, and messages. It includes the integration of OpenEHR based clinical forms: This requirement is needed to allow the web application to connect the user with the clinical forms based in OpenEHR archetypes. In each Task of the workflow in the BPMN diagram, we can create forms and link variables to them. For each form, there is a variable called that contains the URL for the external form based on OpenEHR archetypes. The web application receives that variable and routes the user to the URL.

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

Reducing clinical variation and duplicative testing, ensuring patient safety, and avoiding complications that may result in expensive hospital readmissions are top priorities for providers in the modern regulatory and reimbursement environment. Alert and alarm fatigue consistently make industry lists of health IT hazards and patient safety since, together with clinical burnout, there are common by-products of missing real-time decision-making processes.

A Better Platform Decision support module is designed to be decoupled and agnostic of different decision notations, a key design element to achieve high scalability and flexibility. Decision support can be used either with DMN, GDL or expression language if they use clinical data stored in a clinical data repository, demographic server, or terminology server. All relevant data for decision support can come using the repositories API (as queries) or to provide real-time decision support using the event streaming and event processing components of the Platform.

Current version of the EHR Server supports triggering upon committing of new data into the system. Each trigger is configured with an AQL which filters when event should fire. Fired events can be delivered over simple POST REST call or delivered via a Kafka queue.

Both DMN and GDL support executing CDS guidelines over a REST API as well as be configured as a listener on the Better event streaming pipeline with ability to react to incoming events (from EHR and Demographics) and either write an alert back to the EHR or deliver an alert to other systems.

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

In terms of delivery/deployment models, Better platform offers two main options: local installation (also known as on-premises) and cloud installation (also known as Software-as-a-Service or managed service). In on-premises setup, Better platform runs on customer's own hardware infrastructure while in cloud installation we can offer a deployment into a cloud system/provider of customer's choice, that can be managed by Better, or its partner.

In addition to these two main options, Better also supports hybrid deployment models that combine both local and cloud-based components. This can offer the best of both worlds, allowing organizations to leverage their existing on-premises infrastructure while also taking advantage of the benefits of cloud computing.

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

For our SaaS deployments, Better are typically the lead contractor who deliver and manage the service, but we can also work alongside one of partners where appropriate. Depending on the situation the partner can act as lead contractor, or sub-contractor. Should a SaaS offering be required we would be happy to discuss the contracting arrangements with you in more detail.

Our SaaS offering is single tenant to align with many of the governance and compliance requirements, including GDPR. This means we can deploy the service within a jurisdiction and public cloud provider that is most appropriate for the customer.
For example, the project we recently delivered across all of London was delivered as a SaaS solution, deployed in local data centres in the UK by a global cloud provider selected by the customer, which in this case was Microsoft Azure.
For deploying a SaaS offering into Sweden we would adopt a similar approach as in the other markets where the customers have the choice of global vendors or local data centres and local providers, where this could be deployed .

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

All third-party software included in Better products is at sole responsibility of Better in regards to licensing, usability and support.

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

Any code generated on top of the platform is the sole ownership of the developer, which may be sold as a commercial product or provided as an open source solution at developer's sole discretion.

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

Text field

Better Platform support classic deployments on virtual machines – as shown on the first diagram or a Kubernetes deploy, which makes is easier to switch cloud providers if needed. Diagram of a typical VM setup: https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EUfctWdUkglDmnEORuY1RDcB0zegYUXjJb1s4qO5uo0_rA?e=pyzlPh

Better Platform runs on Microsoft Azure and Amazon Web Services. The integration with Microsoft Azure supports native Azure services such as databases, storage accounts and application gateways. In addition, the platform fully supports Azure monitoring and security management through Azure Monitor, Azure Security Center, and Azure Sentinel.

Better Platform can also be deployed in Kubernetes, which allows for easier deployment into differing cloud environment. All major cloud providers (Azure, AWS and GCP) provide native Kubernetes services. Use of Kubernetes also provides auto-scaling capabilities of the platform. Typical Kubernetes deployment: https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EerHmU4FFDVNtDWD9luxqBIBNfcI75kNZ5gS16639p_sjg?e=taSeui

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

Text field

Better Platform secures that all innovation is built on the same model, so scaling an innovation in a particular use case is accelerated by reusability of the models below. Also, a single instance of the platform can than serve multiple applications at the same time.

Better Platform can be scaled vertically (by adding more memory and CPUs to each node) or horizontally by adding more cluster nodes. Sharding also provides scalability, as new shards can be added as needed to accommodate growing data volumes. The minimal suggested configuration is two nodes with 8GB and 2 VCPUs. Storage requirements depend on the type and amount of traffic. The rule of thumb is approximately 3KB per composition for database storage and approximately 2KB per composition in the index. In a typical configuration, the platform can achieve several 1,000 TPS (transactions per second) per node. Using a clustered deployment, an uptime above 99.99% is possible. Better EHR Server can use several separate relational databases for data storage.

This allows setting the system up in such a way to use a high-performance database for the most recent data. Older data can be moved to a slower database system. The databases themselves can be different, for example it is possible to use Oracle for the primary database and use PostgreSQL for archived data. This system also prevents the main production database to grow to a size which is difficult to manage. This setup is completely transparent to the user as AQLs and other means of data lookup perform the same way as before (just slower for lower performance databases), but no special work is needed to retrieve archived data. The process of moving data is available on the administrative REST API and requires no additional tooling.

Clustering is another critical feature of the Better Platform, which allows for horizontal scalability by distributing the workload across multiple nodes or servers. This ensures that the platform can handle increased data volumes, concurrent users, and transactional loads without compromising on performance. Clustering also provides fault tolerance, as if one node fails, the workload is automatically shifted to other nodes, preventing downtime and ensuring high availability. Better Platform can be deployed in a single node setup (for development, testing, local development). For production environments, clustering provides high-availability, resilience, and high throughput, as well as zero down-time with rolling cluster upgrades. Clustering: https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EVV8m7OaozBKklQOM2w1SjwBs3sXO6sUASvFTyqC1SpBuA?e=8nsXdf

In addition to clusters, sharding involves dividing a large dataset into smaller, more manageable pieces or shards that are distributed across multiple nodes or servers. This allows for efficient data storage and retrieval, as each shard can be processed independently, reducing latency and improving performance.

Better Platform offers tools for massive data intake for timely use in the incorporation of new areas of knowledge. Better batch update API offers bulk data import. Due to flexible nature of the infrastructure architecture, solution can be scaled up during the bulk data import to improve mass import performance and scaled down once import is completed. An example of the biggest implementation up to date: Moscow City – Live implementation for 12.000.000 citizens, for 85 hospitals with 150 branches and 408 units; 170 outpatient clinics with nearly 500 branches and 665 units, EMIAS system with 32.000 + professionals

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

1. National case: Slovenia

o Implementation period: 9 months

o Hospitals: 35
o Total beds: 9,200 beds
o Primary care centers: 57
o Patients: 2,100,000
o Doctors: 8,600
o Visits per year: 15,000,000

An eHealth interoperability backbone represents part of the Slovenian national eHealth project with an important mission to improve the quality of health services with effective electronic solutions. It is a crucial component of the national information infrastructure that enables integration of various information systems. Implementation has brought out many important challenges that are linked to questions as: how to enable an exchange of main clinical documents; how to secure this data exchange; how to identify data about a specific patient in different health institutions; how to control data access; how to answer simple population questions (e.g. what is the percentage of patients with high BMI?).

The idea of the eHealth project in Slovenia entails the inclusion of stakeholders into a functional network, reconstruction of the healthcare system business model, as well as integration and harmonisation of many specialised health and care providers at different levels. The project's aim is to integrate all of the country's existing fragmented Health Information Systems and offer a complete solution benefitting all stakeholders.
Our solution has involved creating an environment (open platform) to support health and care services in Slovenia, enabled document level sharing and mobilisation of documents already produced by legacy systems and ensure the capacity to put data into an open, fine-grained, structured format that is technology and vendor-neutral. We have provided a health data repository to eHealth applications such as eRefferal, ePrescription, eVaccination and national clinical registries. The backbone was developed using widely accepted industry standards such as IHE and openEHR.

Today, more than 98% of all national shared data in Slovenia, is stored within the Better Platform.

The implementation process is described below:

Phase I

The interoperability backbone was first developed as IHE based cross-enterprise enabling document level sharing, with the ability to mobilize documents already produced by legacy systems.
• Discharge summary – documenting the key information listed above to be used by other carers when a patient leaves a hospital.
• Clinical notes – documenting key information, listed ambulatory care and other clinical documentation.
Together with document exchange features, it also provides standardised patient and provider identification, or at least mechanisms of negotiating/linking across existing identification systems and patient data privacy mechanisms to ensure acceptable and reliable privacy of health data based on patient consent.

Phase II

In the second phase of the project, the backbone was upgraded with a clinical data repository for storing structured EHR records at the national level based on openEHR. As such, it enhanced the capabilities of the backbone with the ability to share core clinical information sufficient to practice emergency medicine and a reasonable amount of non-emergency treatment:

• Basic data – demographics, blood type,
• Problem list – recognised diagnoses & conditions,
• Referrals – documenting transfers of responsibility and requests for care,
• Prescriptions – documenting prescribed medications,

• Medications list – current medications, including dosage, regimes, etc.,
• Allergies and reactions – list of patient-specific contra-indications to drugs, food, etc.,
• Immunisation records – documenting the record of immunisation or vaccination episode,
• Other related data.

2. Regional case: OneLondon

o Implementation period – 7 months
o Integrated Care Systems (ICS): 5
o NHS Trusts : 40
o General practices: 1,400
o Local authorities: 33

On behalf of the NHS in London, NHS South West London has worked in partnership with Better to deploy a shared care planning application and platform in only seven months.

The solution will ensure that patients' wishes for urgent care situations are considered at the point of care. The new technology, provided by Better's market-leading openEHR, low-code platform, also enables the rapid future development of care plans and digital services to support people with a variety of health and wellbeing needs and conditions, such as learning disabilities, mental health problems, frailty, and asthma.
Underpinned by open health data and low-code tools, the shared care plan reduces duplication, improves accessibility, and facilitates the creation of shared care plans for a variety of clinical pathways starting with the Urgent Care Plan.

The Pan London Strategic Commissioning Group (SCG) selected Better as its digital development partner in December 2021. The partnership between the NHS in London, Better, and multiple vendors of electronic patient records has resulted in a dynamic, integrated care planning solution to deliver joined-up care. Better selected partners Atos, ReStart, CareIS, XYCare, and FreshEHR to support with the successful deployment of the solution.

By adopting an openEHR data platform across the region, health and care providers can access and contribute information in real-time, contribute towards a single care plan, and make use of a standard set of tools that can be deployed at the point of care. Health data, which can be accessed through other platforms, such as patient-facing apps and personal health records, is also being used to support research, and to encourage citizens to take greater ownership of their care by allowing them to create and amend care plans.
Link to more technical descrption: Technology Behind OL
The Urgent Care Plan, implemented across five integrated care systems, 40 NHS trusts and 1,400 general practices, enables all Londoners to have their care and support preferences shared digitally with health and care professionals across the capital.

Hospital case: The Christie

o Implementation period – 4 months
o Total patients served: 3,200,000
o Per year: 60,000 patients
o Local authorities: 33
o Clinical trials: 650 annually
o ePROMS/clinical forms: 683

The Christie NHS Foundation Trust is the largest single site cancer centre in Europe treating more than 60,000 patients a year. Due to its pioneering treatments for cancer, it is the first UK centre to be accredited as a comprehensive cancer centre.
The Christie is adopting a platform-based approach and moving to a data-centric model with openEHR at the core. Better's low code tools, an integral part of Better's digital health platform, are used to support the modernisation of The Christie's Electronic Health Record and the transformation of clinical, patient, and cancer research services. The Christie team developed a

suite of 24 ePROMs forms using Better low-code tools in just 4 months. They are now starting to work on reengineering over 600 forms, which are in their current EHR platform. Effective management of data is critical for healthcare providers to deliver the best possible care to their patients. Below, we explore the challenges faced by The Christie NHS Foundation Trust in managing their data systems and how they overcame these challenges.

Challenges Faced: The Christie NHS Foundation Trust faced several challenges with their data management system, including patchy systems that could not be accessed over the internet, lack of integration at the interface level, siloed data, unstructured data, low-quality data unusable for research, and the same information stored differently across systems. Duplicate data entries were also a significant issue, causing confusion and inefficiencies. These challenges made it difficult for staff to access and utilize data effectively.

Solutions Implemented: To overcome these challenges, The Christie NHS Foundation Trust implemented the Better platform, together with a low-code forms environment with a modern user interface suitable for mobile devices as well. This allowed the trust to move towards a more integrated system that allowed for seamless data management across all departments. The trust worked on unlocking their data to be structured in an open format accessible in real-time. This made it easier for different departments to access and share information, leading to improved collaboration and decision-making.

Results Achieved: The solutions implemented by The Christie NHS Foundation Trust had a significant impact on their data management system. The modern user interface and low-code forms environment allowed staff to access and input data quickly and efficiently, reducing the amount of time spent on administrative tasks. The open format of the data also made it easier for different departments to access and share information, leading to improved collaboration and decision-making. Through the data collected until now, we managed to achieve:
o Saved 66% of time, taken to conduct telephone reviews
o Saved 28% of total review time
o 97% of patients found the developed ePROMS easy to complete
o 99% of patients found the ePROMS easy to understand
o 85% less unnecessary outpatient appointments

Conclusion: In conclusion, The Christie NHS Foundation Trust faced significant challenges in managing their data systems. However, by implementing a low-code forms environment with a modern user interface and unlocking their data to be structured in an open format accessible in real-time with the power of Better Platform, the trust was able to underpin their EHR transformation journey with open data. This allowed for more efficient data management across all departments, leading to improved collaboration, decision-making, and ultimately, better patient outcomes. At the moment, The Christie is going through the implementation of Better Meds, a medication management solution from Better, also based on open standards.

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

Text field

Better's implementation methodology uses best practices developed over many years of experience. It facilitates a successful implementation and a roadmap to reaping a return on investment on the project for many years to come.
Our methodology employs best practices of both, the traditional and an agile approach to the digital transformation journey while making informed and guided adjustments to fit the customer's specific situation. To set a structure to the straightforward parts of a project, we use traditional methodology for high-level planning. We use agile methodology for detailed planning and unclear specifics of a project. This approach allows us to provide a clear structure with agile flexibility.
Better Platform methodology is scalable, structured, and uses a phased approach (iterative) consisting of pre-defined inputs (openEHR models, forms, views, widgets), activities, and outputs, which deliver a solution that meets the customer's objectives.

Better's implementation methodology centres on five sequential project phases with infrastructure,
integration, and project management activities spanning across all five phases. 5 phases of Better Platform implementation methodology:

Phase 1: Initiation and planning (Initial scoping, resources identification, project planning, organising the project, project kick-off meeting)
Phase 2: Analysis and design (Requirements review, Gap analysis and resolution, infrastructure profile)
Phase 3: Build/implement (Infrastructure verification and system set-up, data source modelling and template refinemnets)
Phase 4: Train and acceptance (End-user training, UAT, readiness testing and assesment, go/no-go preview)
Phase 5: Deploy and leverage (Production data import, Cut-over to Better Platform/go-live and early life support, handover to support, handover to account manager, project closure meeting)

Below is a deployment fact sheet:
https://bettercare365-my.sharepoint.com/:i:/g/personal/jovanp_better_care/ERU8qz1vQPZMh1iyWNNZTggBavCPrs8_Nq1901wC-WGr-Q?e=jHibS3

i. Describe your software lifecycle strategy and release cadence.

Text field

Major release available once per year. The previous version of the Better platform will remain fully functional and supported, up to the performance levels, for at least two years.

Better Platform support lifecycle provides:

• Program updates, fixes, security alerts, critical patch updates and/or enhancements.
• Routine upgrades provided in the normal course of Software development. Updates may be provided either as a patch to an existing release, or as a complete new release.
• Upgrade scripts (availability may vary by program).
• Major product and technology releases, if and when made available at Better's discretion, but normally one to two times a year, which may include general maintenance releases, selected functionality releases and documentation updates.
• Maintenance versions means generally commercially released code corrections, patches, and updates of the Software
• Efforts to Correct the Software: Better shall make commercially reasonable efforts to correct bugs or other errors in the Software. Client acknowledges that Better is not required to correct every bug, error, or problem with the Software that it reports to Better or of which Better is otherwise made aware.

It may become necessary as a part of Better Platform product lifecycle to non-support certain application releases (including any embedded third-party applications for which support has been retired by the manufacturer or vendor of such applications for which, in Better's good faith determination, it is no longer practicable for Better to support), therefore Better reserves that right. However, application releases that are expressly identified within Better's lifetime support policy will be governed by the terms of the lifetime support policy. Non-support information is subject to change.

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

Text field

Healthcare data is diverse, dispersed, and frequently changing, growing at a rate that far exceeds our ability to consume it. This creates data overload, making it difficult for users to see the information they need for a specific patient at a particular time. For the future, Better Platform is to extend the capabilities to specific bundled features and modules together with content modules addressing the future needs of the users and customers.

Core Roadmap

• Faster response times, stable work and system resilience
• Decoupled demographic, operational and administrative data repository
• System resilience with tech improvements (updates, technical debt).
• Enhance extraction of data and event-based data streaming
• Provide standards-based terminology validation
• Integration capabilities with HL7 FHIR based systems
• Enrich content creation with new capabilities/building blocks
• Support rules-driven data to identify patterns and insights that trigger personalised interventions or escalation, in line with Dasboard server capabilities
• Support workflow and collaboration

Studio Roadmap

• Reduce onboarding time
• Create data presentations with ease
• Enrich content creation with new capabilities/building blocks
• Encourage collaboration and reusability

- Create bundle out of different content
- Connect to your data sources with minimal configuration
- Customize the style of your apps/forms and enhance their usability
- From idea to application in days with app builder

Architecture roadmap

- Data Views;

to extract value from healthcare data and delivers information to enable better decisions based on aggregated data in community-wide, longitudinal electronic health records. Based on automated updates from the longitudinal health record, Data Views provide actionable insights within existing clinical and care management workflows. Medical practitioners are faced with information overload, making it time-consuming to retrieve and be aware of important clinical information. The vision of Better Platform is to provides content specific packages of data to express data as condition summaries to enhance the decision-making process. For example, diabetes, the ED physician can see labs for haemoglobin A1c, glucose, and LDL over time, along with a list of medications and comorbidities – all in a single view and are specific to a combination of clinician speciality, patient condition and care setting.

- Decision Support;

Vision is to be decoupled and agnostic of different decision notations, a key design element to achieve high scalability and flexibility. Decision logic can make full use of data stored in the EHR Server, Demographics or Terminology Service. Thus supporting different ways to manage decision logic such as : DMN, GDL and an expression language.

- Workflow;

Beyond simple apps, healthcare applications require the ability to manage business processes and worksflows. For that reason it is envisioned for Better Platform to support different ways of Workflow support: Simple Tasks - for simple, discrete tasks which are time based (example – administering medications). Simple Tasks are based on the openEHR Instruction Action concept. Business Process Modeling Notation where processes can be integrated with forms built in Studio and data from EHR Server. In addition, we are planning to add support for Case Management Model and Notation in future releases. CMMN is a graphical notation used for capturing work methods that are based on the handling of “cases” requiring various activities that may be performed in an unpredictable order in response to evolving situations.

- Collaboration;

with Collaboration Service allowing clinicians, care teams, support staff, patients and family members to collaborate on treatment and care activity within the healthcare delivery organisation’s acute, ambulatory and virtual care. The service supports messages related to a specific patient, associated with a particular topic (eg. assessment, diagnosis, therapy, a specific task or for personal communication with no specific context). It also supports different channels: text (chat, comment), voice, video, notifications, alerts about events or updates on specific topics of interest.

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

Better Platform multi-tenancy refers to the ability of the platform to serve multiple clients or tenants, such as hospitals, clinics, or healthcare organizations, within a single instance of the platform. This enables efficient resource utilization and cost optimization, as multiple tenants can share the same underlying infrastructure while maintaining data segregation and security. The platform supports unlimited number of tenants. The tenants are logically completely separate, including the possibility for storing composition data into separate databases. Separation includes composition and ehr data as well as OpenEHR templates, views and events.
All our backend components support multi-tenancy (multi-domain) setup. No data is shared, including SSO-authenticated users, configuration (templates, views, forms, event triggers etc.), EHRs, compositions, FHIR resources, terminologies etc.
Multi-tenancy can be achieved in the same way both on small single-node or large clustered installations.

Backend components ensure the generated data IDs are unique, not only within but also between different domains or tenants.
SSO-authenticated users only have access to a specific single tenant.
Only CDR users defined for Basic authentication can access multiple domains (tenants) and cross-querying is also enabled given sufficient users’ rights.

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

All Better Platform components are designed to support multi-tenant model.
Each tenant (dubbed "domain" within the Better Platform) has a logically separated pool of resources, isolated from the other tenants. All tenants share the same database space by default, but there is a possibility to create individual data-source definitions and assign them to one or more tenants, thus physically separating database instances for tenant or tenant groups.
When a user is created, it is assigned a default tenant affiliation and can operate only within the assigned tenant resource space.
If there are local users defined within the EHR Server (basic authentication based), they may have access to multiple tenants (for development purposes) and can cross-query, at the same time - see 2.1.3.1(c). Such local user can have assigned a role of "superuser", that gives it top level administrative access across all tenants, which is usually employed internally, by the system administrator or administrative backend, to manage tenants creation and resources.

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

Permissions

Permissions model for the Better Platform modules is based on roles. Each user can have multiple roles assigned, that allow him performing specific actions and accesing individual resources, for example:

• accessing composition within the CDR
• creating new compositions
• creating new EHR records
• querying data
• accessing Better Platform views
• etc.

Users can be also assigned elevated admin roles to be able to manage specific segment of the individual tenant resources, such as forms, templates, views, events and users. There is also a role that gives user a "master administrator" role, that includes all other priviliges in a single role.

User context

Users (requester) context can be established in different ways - Better Platform supports two major protocols of authentication/authorisation:
1. basic authentication
2. token based OAuth2 implemented via OpenID-connect protocol

Basic authentication

In first case, the user definition is stored and managed locally, within the CDR itself. This approach is beneficial for development stages, where there is no exposure to the publicly available interfaces, no production-grade data and overall low risk of breach.
Each local user can have assigned multiple roles for different tenants, if so needed, but typically works within a single tenant.
With each individual request the user credentials are verified and a security context established, that resolves the ability of whether the user has all permissions necessary for the request to be fully executed.

OAuth with OpenID-connect protocol

For staging and production grade environments the user and permissions management is handled separately, typically using an Identity provider solution. Bundled with the Better Platform is the open source product Keycloak (keycloak.org), that can server either as a base for establishing the users set, or can act as a proxy to another, already astablished user management system, supporting various protocols (LDAP, AD and similar).This approach is used in SingleSignOn (SSO) environments, where user logs-in only once, exposing its credentials this single time, but is then issued a Token, that contains all information necessary to establish the security context each time, with any of the Better Platform components or services. Tokens are digitally signed, with a short validity timespan.

Determining resources ownership

All the necessary evaluation whether the user has permissions to interact with any CDR related content is executed before the said resource/data is accessed, based on the information stored in the security context.
When using OAuth based security context, a single user can only access a single tenant. There is no option to access resources cross-tenants.
Data is internally stored in separate tenants, logically or physically and their provenience and ownership can be uniquely resolved and established.

Metadata verification

Information about user affiliation to specific tenant/domain is stored separated from the clinical

data, either within the Identity Provider Service and passed down with each request to access data (through security context in forms of access tokens or similar) or directly on EHR Server, in a local database.
Security data is checked and validated for each request individually.
Each document stored within the tenant has the tenant "system_id" stored in the openEHR RM. It is reflected in the composition ID, it is used within the VERSIONED_OBJECT for the composition and EHR class for the EHR.
This way any tenant based operation is checked and content matched to destination. If any of the mentioned steps/requirements is not met, an error is triggered and transaction refused.

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

Filtering the CDR content can be based on composition meta data (such as EHR ownership, identifiers, template used for composition validation etc.) or archetype properties, used to define the template, and lastly, by the request context itself, which contains Oauth token and other custom properties as well.

The basic permissions are evaluated as described in 2.1.3.1(c), but more control is needed over the access control and data filtering. To achieve this, Better Platform introduces "ABAC server" (ABAC = Attribute Based Access Control). It is a policy-based access request evaluation engine, based on PDL (Policy Definition Language). Policies are bound to various API endpoints and are evaluated for each request individually. A policy evaluation is a boolean-based operation, that results in an "allowed" or "denied" output. If the policy determines the user has access to a resource (typically composition or demographic server resources), the request is passed to the target system and executed uninterrupted. If the policy denies access to a resource, the value returned on the HTTP level with a status code of 403 - access forbidden.
In case of a query based API call, policies are applied not to stop the request but in order to filter data from the query result-set.

The following input can be used within the policies:
• any information available from the composition, as defined by the openEHR Reference Model Specification, eg. EHR Class properties, COMPOSITION class properties, VERSIONED_OBJECT class properties etc.
• user roles, as provided by the Oauth token
• any other Oauth token claim, mapped into the requst/ABAC security context
• any additional external information, provided by 3rd party service or other data source, locally handled by ABAC

ABAC Server can host a number of properties and relations between them, needed to evaluate the access policies. Managing of these properties can be achieved via an ABAC REST API interface or ABAC plugins, that implement any sync/push/pull operations with external data sources.

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

See point 2.1.3.2(a) for details, same principle applies.

A practical example:

Workflow

Better Platform EHR Server is configured in such way, to apply a specific ABAC Policy for any query executed against the /query endpoint. The ABAC setup also extracts claim based information from the security token and includes that in the ABAC request context.
ABAC evaluates the policy and decides whether specific data should be filtered from the query result-set.

Situation

There are two patients, with separate EHR records, represented by EHR IDs of "1" and "2".
There are two templates within the test system, named "Vital Signs" and "Mental Health Summary". First one focuses on Vital Signs data (i.e. blood pressure), second one contains a "Diagnosis" evaluation.
Three compositions exist, patient "1" has a document based on template "Vital Signs" (document "11", patient "2" has two documents, each based one of the two templates (documents "21" and "22").

Link: https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EViaW4jo7DVLjkzcza637KcB5XcgRqrmgpK9dBVBSDNvgw?e=WHGeIe

Policies
There is a single policy in place, that drives query access to the clinical data stored in all three compositions. There might be other policies in place, for reading the individual compositions, written in similar fashion, but we will focus on querying alone. The goal is to "protect" and control the access to the "Mental Health Summary" document, containing sensitive data.
This policy is bound to every query-related API call.

Example policy: https://bettercare365-my.sharepoint.com/:i:/g/personal/jovanp_better_care/EQkGb-O_dG5JsqMXhVPPU90BS2GCiuKnNSOoTKCx9WsFuw?e=UYbH4K

Where does the information come from?
• Role ACCESS_SENSITIVE_DATA can be assigned to the user in the Identity Provider system.
• Type of the document is read directly from the composition itself.
• Users' medical specialty is read from the claim in the token and injected in the request context. It was provided by the 3rd party system.
• Consent information is gathered from an external system via API/syncing and stored locally within the custom built plugin data space.

Data access resolution

Required state: for patients data to be accessible: patient must give consent, a record of that must exist in the consents database, matching the patient identifier (in our case - EHR ID).
Required state:
Any of the two conditions must be met, to access a document or its data:
• user must be granted a special role ACCESS_SENSITIVE_DATA
• users' medical specialty must be "psychiatrist" AND there must be an existing relation between this medical specialty and document type (Mental Health Summary"

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

See point 2.1.3.2(a) , about the input available for ABAC policy evaluation.
Any information stored within the composition and its underlying openEHR Reference Model can be accessed natively within the native Policy Definition Language syntax.

Example of a policy statement, that filters compositions based on a specific template from the result-set:

NONE_OF(hasAnyValue('COMPOSITION:archetype_details/template_id/value', 'Mental Health Summary'))
The syntax “'COMPOSITION:archetype_details/template_id/value'” is used to access any Reference Model defined composition property. Same way you can access any other openEHR RM Class and their properties, for evaluation.
Folder information as well as Tags are stored in EHR Class and can be accessed the same way, Feeder Audit as well.

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

See 2.1.3.2(a) for generic ABAC functionality.

Points 2.1.3.2(b) and (c) describe how data is filtered from the query result-set (queries are not blocked!). There is an additional way to define policies that evaluate direct access to the documents (compositions) on individual basis.
Any policy can be bound to specific API endpoint/method in order to enforce its requirements and drive the access control on the resource.

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 do support multiple bulk operations, namely:

Bulk data import from CSV files

A CSV file containing clinical data and a reference to EHR data belongs to (specific EHR ID or external Subject ID) can be ingested via API, where EHRs missing would be automatically created as needed, data stored as individual compositions. The output is a status response for each record, with the composition ID or any possible errors on per-record basis.

Bulk composition create/update/delete action

For larger amount of composition based operations we do support what is called a "contribution", which is a bundle of individual entries, that define the requested action and the clinical data payload as well. Used for transferring large amounts of data from one system to another, for ingesting legacy data and supporting large-scale migrations.

Bulk composition retrieval

Usually the compositions are accessed individually, but for larger number of documents we do support two separate ways of multiple documents retrieval with a single request:
• based on a list of composition IDs
• based on an AQL query, that returns a list of composition IDs

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

See point 2.1.3.3(a).

When importing data via CSV files or using a contribution based request, each individual line/record/entry is pipelined through the same process as an individual request would be submitted to.

The input parameters for individual API request are validated against providing enough information to be able to process the request, but once we are dealing with individual entities, same rules apply as any usual request.

When importing data via CSV:

• non-existing EHRs are automatically created and annotated with subject ID if necessary
• any error encountered will affect only the individual entry, the rest is normally processed

When using a contribution payload:

• EHRs need to be available for the payload
• any validation error is individually reported but whole transaction is rolled back and request rejected

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

Audit server allows for the construction of immutable auditable logs for the purposes of traceability, dispute resolution, security, privacy, and compliance with policy (external and internal) and laws. It does not refer to ‘system logging’ that is normally produced for system and application debugging. The auditing module is an integral part of the solution. It captures all (user) events / actions and stores the relevant information to a private dedicated database, which is not accessible by regular users (thus preventing unauthorized access and / or modification). The auditing system allows front-end applications to show the action’s results data only after the audit data has been successfully written to the audit database (preventing use cases when user could see patient’s data without being logged).

Using the auditing REST API will enable users to control auditing and also run queries against the audit data for specific applications. It is also possible to clear auditing data using the API. Each log event includes a header ID, target resources, time stamp of the recorded event, request parameters, and response parameters. Users can view events logged by the Audit service by using the Console or API. You can view events, copy the details of individual events, as well as analyze events or store them separately. Data from events can be used to perform diagnostics, track resource usage, monitor compliance, and collect security-related events.

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

Besides the Audit server, as described in 2.1.3.4(a), the system logging is based on Log4J module, that allows for a flexible configuration of different logging options.

Better Platform supports togging in following services/formats:
- XML files
- ATNA server
- ELK stack - Logstash 1 and 2
- audit events details (JSON format) streamed to Kafka or rabbitMQ

All of these options can be used individually or in parallel.

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

CE marking - Dose Calculator, which is a standalone product within Better Meds, a medication management solution operating on top of Better Platform, is compliant with the regulation Medical Device Directive 93/42/EEC, Annex VII, and is currently sitting within 4 years transitional period.

MDR - N/A due to 4 years transitional period according to Article 120. During that period, the Authority is not allowed to make any changes to it.

Medical Devices Directive (MDD) 93/42/EE - MDD requirements have been implemented in the Quality management system. (Internal document: Software Development Life Cycle – Medical Devices who follows recommendations and requirements of IEC 62304:2006+AMD1:2015 and ISO 13485:2016 standards)

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

Text field

See above.

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

Better's new-generation interfaces are being designed in an inclusive way. That means all design decisions are weighted to support general population user experience as well as most colour-blind and sight-impaired users, but do not yet support specific difficult cases of impairment. We do that by using large enough text sizes, high-contrast for information display and a colour scheme and information display which is already colour-blind ready.

We also use our own language writing guidelines to improve clarity and reduce ambiguity of conveyed messages.

Better's Current Design System (on which Portal is based), is highly adapted to issues stemming from sight-impairment and colour-blindness by default.

o Colours and contrasts are chosen in a way that prevents misinterpretation by most colour-blindness types. For instance, the design system doesn't use the colour green (positive actions) as green appears the same shade as red (destructive actions and alerts) to most types of colour blindness.
o The design system also requires the use of colour to be complemented with other visual strategies (like typography, shape, icons) when differentiating between information.
o While some interface elements can appear in low-contrast shades, information of significance is exclusively presented in higher contrasts both in shades and colour.

In the future we plan to bring themes to complement the default theme. One such theme will be a high-contrast theme which will address problems of older display devices which display a limited range of colour shades. This functionality will also allow creation of colour schemes adapted to specific types of colour blindness. It is worth noting that colour-blind pallets are subject to cultural and psychological challenges of health care professionals hiding their disabilities. Instead, we have opted for the before-mentioned colour-blind ready default colour scheme.

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

Better provides a set of open-source libraries to simplify working with OpenEHR data. These have been available to the whole OpenEHR community and wider and are published on github: https://github.com/better-care

These include:
- OpenEHR reference model implementation in Kotlin – usable for Java and Kotlin projects. This RM implementation is fully aligned with OpenEHR specification, including package names. Current RM version: 1.1.0
- Web Templates – open-source version of Better's web templates, for simpler creation of OpenEHR compositions.
- ABAC server – attribute-based access control service usable for many different cases.
- Various testing sets – for web templates, openEHR REST API, etc.

All projects are released under the Apache License, version 2.0 (https://www.apache.org/licenses/LICENSE-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

Document Viewer

Medical teams are under constant pressure to improve patient flow, reduce patient risk and ensure timely admissions, transfers and discharges. In addition, health and care provision is increasingly taking place across different care settings as well as outside the traditional healthcare settings such as patients' homes. In order to get reliable and relevant patient history information, such as relevant past results and summary data, previous encounters & episodes, latest encounter overall, or latest encounter within different specialties or GPs, healthcare teams need to consult different applications, paper records and talk to patient and relatives. This means they are constantly losing time and are always at risk of making poor decisions on incomplete information that causes delays and jeopardises patient safety with potential duplication of procedures, diagnostics and even medications.

Better offers a dedicated clinical module to help medical teams get relevant and consistent information about patient's history by providing the following functionalities:

- Chronological view of all patient documentation
- Documentation grouped by location and speciality
- Quick search
- Filtering options (date, status, document type)
- Free text, PDF and attachments

Having access to a comprehensive patient history overview with the ability to filter and search for key details, (chronic) conditions, medications, etc. is a crucial tool for medical teams when they are admitting, transferring or discharging patients, so they have all the crucial information quickly at hand to make sure therapies are not overlooked or discontinued, allergies or other important facts ignored.

Clinical dashboard

Seeing patients that need urgent care is an obvious priority but clinical teams often have to do physical patient rounds to be able to identify those patients. They need a comprehensive and real-time overview of patients to help them effectively manage, review and treat priority patients as well as ensure a timely response to potentially harmful events. In turn, this will lead to improved patient outcomes, effective treatments and reduction in the antimicrobial resistance. Clinical teams must walk around the hospital or collect reports from different systems to get a comprehensive picture on the patients on their wards. As a result, time is wasted and any patients with deteriorating condition are identified with a delay. Having patient information scattered across different systems and paper notes causes time delays, inaccuracies and, delays the time to see priority patients. As a result, micro-organisms become resistant to antimicrobials, including antibiotics, and can become superbugs that don't respond to any drugs. With Better Clinical dashboard all patient data is readily available, enabling clinical teams to manage multiple patients safely and seamlessly, providing holistic approach for treating patients with multiple chronic conditions, and enabling early identification of deterioration.

Lab viewer + order

Medical teams need up-to-date information on latest laboratory results. Often the results are ordered by different care teams throughout a patient's journey. Everyone involved needs a comprehensive view of what has been ordered and if and when new results are available. Different lab results are still often received in a paper form, although the core processes are all digitised. Yet, comparison of different lab results is done manually. In order to get the latest lab result data, medical teams are still too often relying on paper records. It is also difficult, if not impossible, to always get the status of pending results, timely notifications of new results, and overview of which lab examinations have been scheduled for which patients. It is often unclear what care professional, if anyone at all, has taken any actions on the basis of the received lab result.
Crucial time is lost for manual comparisons of key lab results while examining these results leads to the increased clinicians' cognitive load. Consequently, additional coordination and communication is needed when deciding on the follow-up actions based on the received results. Transparency is also reduced with the potential for overlapping activities. It is therefore crucial that medical teams have a single and comprehensive view of latest and historical lab results, with the ability to easily compare trends over a select period of time, as well as filter and search based on biological material, status and be able to simply take ownership of follow-up actions. Better offers a dedicated clinical module to help medical teams get relevant and consistent information about patient's individual lab result or timeline view by providing the following functionalities:

- Store all lab results in a structured way within the Better Platform
- New results "inbox"
- Sparkline trends on each individual lab result
- Visually emphasised abnormal values
- Lab sign off
- Quick chronologic view
- Detailed Timeline view of lab results
- Scroll through time

The module also helps with collaboration and coordination by offering up-to-date information on each of the orders/results from the initial order, interim steps and until final result is received and acted upon.

BetterMeds - Medication Management

Medication errors remain a major global challenge with more than 2.6 million people dying every year from incorrect medical care. Reasons are many and varied: tasks done manually, use of paper charts and illegible hand-writing, similar medication names, complex medication combinations. Additional risks come from lack of digitalisation, poor implementation of IT systems, unusable systems as well as lack of traceability for a patient's medication due to insufficient interoperability among healthcare institutions that treat the patient.
Getting accurate patient medication history is time consuming and unreliable. Healthcare professionals often rely on patients to provide the information which at times is incomplete or

incorrect. In other cases, they need to rely on paper charts that are sometimes illegible to read. Finally, they would be checking the medication history with other care providers that treated the patient in the past task– another time-consuming. In the areas that implemented digital systems, more often than not the data is kept in silos, embedded in different systems, making it hard for effective sharing of information.

Disparate data sets have the potential for issues when a patient is transferred across care settings. There is a significant chance of an error being introduced into the person's prescriptions, which could have very serious consequences.

Better Meds, is a comprehensive electronic prescribing and medication management application that provides a consolidated medicines record cantered around the patient. Authorised healthcare providers have immediate access anytime and anywhere to vital patient information within a hospital or across the whole region. It brings together medicines records from all care settings: primary care, secondary care, mental health, care homes, and hospices. The crucial medicines information is available at the point of care, enabling care teams to make better decisions.

The user-friendly EPMA system combines different functions to provide all clinical staff with an integrated view of a patient’s medication history, simplifying workflows and increasing safety. Some of the other features include:

- Easy integration options to connect with other decision-support systems
- Provides robust audit trail for the entire medicines use process
- Clinical governance through clinical decision support software
- Cost reduction
- Medication configuration
- Inpatient Prescribing
- Medication Reconciliation
- Outpatient Prescribing
- Drug Dictionary Configuration Tool

Allergies

Adverse reactions to drugs occur frequently and, just in the UK, were responsible for approximately 62,000 hospital admissions annually between 1996 and 2000. Instant access to accurate patient allergen information helps healthcare providers optimize medication therapy, decrease the incidence of adverse drug reactions, reduce drug cost, decrease inpatient length of stay, and ultimately improve overall patient care. Incorporating allergy information into medication management systems also increases ADR reporting.

Drug allergies are commonly over-diagnosed, misdiagnosed, and self-diagnosed. There is considerable variation in the way drug allergies are recorded in electronic health records. Most drug reactions are likely to go unreported and/or unrecognized by healthcare professionals and, even when recognized and reported, not all reactions are accurately recorded.

Inaccurate recording of information results in increased costs, prolonged hospital stays, avoidable hospital admissions and, in some instances, it may introduce new problems as patients are denied treatments that they are erroneously believed to be allergic to.

Better Allergies application provides healthcare organisations with structured clinical data on allergen, status of the allergy and reaction type. This enables healthcare providers to use allergy data as an input for decision-making across clinical applications and processes. Hence, it ensures increased patient safety through allergy-related warnings triggered via clinical decision support. Instantly familiar, users can swiftly enter, edit, or validate allergens, status and reaction types. The date of the latest update is also available as well as who made the entry. The application also includes free text comment for a particular allergen and a general comment for relevant remarks on a patient level.

The application is made for smooth integration, and fully configurable to different needs. It is based on international interoperability (openEHR and FHIR HL7) and terminology (SNOMED CT) standards. It can be embedded in the existing clinical portal or offered as a module within the Better Portal.

Antimicrobial stewardship (AMS)

The World Health Organization calls antimicrobial resistance one of the top 10 public health threats that causes 700,000 deaths worldwide annually. To combat the crisis, it is crucial that clinical teams - on local or regional level - have a comprehensive and real-time overview of patients on antimicrobials to help them effectively manage, review and treat those patients as well as ensure a timely response to potentially harmful events. In turn, this will lead to improved patient outcomes, effective treatments and reduction in the antimicrobial resistance.
Pharmacists or Surveillance assistants (e.g. pharmaceutical technicians or nurses) must walk around the hospital or collect reports from different medication information systems to identify patients to review. This information is then entered onto a spreadsheet, which is prone to human errors and cannot be easily translated when analysing data or performing data transformations. As a result, pharmacists and other authorised clinical staff must check different information systems or paper notes to assess the appropriateness of antimicrobial prescription.
Having the AMS data scattered across different systems and paper notes causes time delays, inaccuracies and, crucially, it leads to prescription mistakes. An estimated 20% to 50% of acute care hospitals worldwide are not prescribing antibiotics appropriately due to lack of overview. As a result, micro-organisms become resistant to antimicrobials, including antibiotics, and can become superbugs that don’t respond to any drugs.

Better Antimicrobial Stewardship is a dedicated clinical application that allows clinical pharmacists and other clinical stakeholders to quickly identify, and consequently prioritize, monitor and review patients on antimicrobials across one or multiple locations within acute care settings. Moreover, users can identify the cumulative number of days the patients are on antimicrobials, the number of active antimicrobials, whether patients are on high-risk or broad-spectrum antimicrobial, and whether patients are on Gentamicin. Users can also see the details, such as name, route and indication for all the active antimicrobials.

The user-friendly application has a number of benefits:

- Decrease in time to find patients on antimicrobials that may be at risk.
- Eliminates the need for Surveillance assistant.
- Increase the number of reviewed patients per day by significantly decreasing the time to find patients on antimicrobials that may be at risk.
- Eliminate the need of contacting multiple professionals to find patients.
- Eliminate the need to identify patients in other information systems.
- Increase the accessibility of relevant information in the context of antimicrobial stewardship.

Care planning

Across the world people living with serious medical illnesses are growing in number due to demographic changes. In Catalunya, the population aged 65 and over is estimated to rise to almost 1.8 million by 2030, representing over 22% of the total population. This will, inevitably, increase costs and pressure on healthcare services. For many people living longer with deteriorating health, the quality of life recedes regardless of availability of modern treatments for their particular ailments. Therefore, elderly with chronic conditions are increasingly looking to plan their care at the end of their life. Having a unified digital care plan available to all care providers and the patient will ensure patient’s wishes are always followed.
If in place, care plans are often provided in paper forms and kept in folders in patient’s homes and across different locations. This increases the possibility of forms being lost or misplaced, information being inaccurate and incomplete, and not appropriately shared amongst care providers.
The result is increased costs due to unnecessary hospital admissions, while health and care providers are not confident they are caring for the patient in accordance with his/her wishes, in particular in terms of their preferred place of care or death. In case of an emergency, ambulance staff cannot always ensure that the patient’s end-of-life wishes are followed, especially regarding the DNACPR.
Better’s application Dynamic End of Life digital care plan enables patients and care teams to gain easy and real time read-write access to a single source of information focused on patients’ needs and wishes. The care planning solution enables interconnective, accurate and up to date

information that can be contemporaneously shared between multiple providers. The solution is simple to use with a single sign on approach that plugs into applications that healthcare organisations already use so there is no need for additional log in or sign up details. Patients and their care teams can access and update the plan at anytime and anywhere from any device, be it a tablet, a desktop or a mobile. Any updated information is re-used and repopulated across the care plan regardless of which facing app has been used to enter the data.
The patient will be given a unified digital document to complete his/her EoL record and update it whenever needed, either on his/her own or with the help of the authorized person. This means the patient and his/her family can rest assured the document does not
Finally, the single EoL care plan record will also help reduce costs due to reduction in unnecessary hospital admissions.

Clinical Registry Framework

Healthcare providers are setting up multiple clinical registries to collect, curate and disseminate information on large numbers of similar patients. The data in the registry comes from the information healthcare providers collect while providing care and is added to information on other patients with similar conditions. New clinical registries are constantly added, as well as new data sources, with workflows constantly changing, while existing ones are extended and revised. Increasingly, different healthcare providers join up the registries (especially in the field of rare diseases) to cover bigger population and to distinguish complex relationships and to support the development of new therapies, facilitate further research, and shape the health care decisions made by patients and providers. Patient reported outcomes are being collected to enhance clinician generated patient information.
Typically, different departments use different clinical registry solutions even within a single healthcare organization, and even more so across different care providers within a region, country and globally. Furthermore, there are different IT solutions used for primary data records and collections. A lot of useful patient related data is accumulated on a daily basis across multiple studies but with different data sources, approach to persistence, governance and end users. To collect PROMs, for example, typically different and incompatible solutions are used with different and incompatible proprietary data models.
Researchers and clinicians are therefore often limited to a single clinical registry data, with difficulty to combine data with other registries and PROM applications, and inability to extend the existing clinical registries without losing some of the data or backward compatibility. In addition, getting the data from different sources often results in losing details on data provenance and creating challenges with storing the data in a consistent, reusable and sustainable way over a longer period of time. Lack of standardisation on persistence layer is also limiting the reuse of data, as well as data repurposing and consolidation among different registries.
Clinical data is too important not to share. A dedicated open data based clinical registry framework is necessary to be able to build clinical data registries that support data acquisition & curation, seamless integrations on a data level, constant modifications and dissemination of data.

Better’s clinical registry framework offers out of the box support for:

- Unlimited variation and scalability of the clinical concepts and attributes: The wider the range of data you can combine, the more meaningful and valuable the insights and picture it can paint. Better’s registry framework supports collection of data from the clinicians as well as patients.
- Future proof persistence layer: In order to keep clinical registry data relevant for decades and independent of data sources, a vendor neutral persistence layer is needed which can be easily adapted without the loss of any collected data.
- Adjustable workflows and registry creation: research operations workflows can be configurable by research staff to handle variability across studies, sites, and research domains. Non-technical personnel can configure existing and new studies, forms, PROMs and research assets quickly and costs effective.
- Complex and complete data provenance: Ability to store full data provenance will support any needed forensic activities such as data-dependency analysis, error/compromise detection and recovery, auditing, and compliance analysis.

• Security: The framework supports consents, RBAC and ABAC, and is fully GDPR compliant . Privileging and security are deeply embedded in the system architecture and does not depend just on the user interface layer, however, it is enforced server side and in a way that privileging models can be configurable for each install/user environment, with granular user-specific permissions calculated and enforced.
• Seamlessly share, combine and repurpose: Once acquired, clinical data registry framework provides the capability to reorganize and transform data for different uses. From cohort discovery to intervention analysis, all research is cumulative.
• Longevity: The registry framework is built on openEHR persistence layer which guarantees full data governance by non-technical users and support for complex and collaborative research activities over decades and complete independence of the applications that were used to generate/store clinical data.

Clinical Portal

Clinical Portal is a highly configurable web-based clinical app that empowers each member of your care team with instant access to relevant patient information presented in a comprehensive way.By consolidating information from disparate systems, it provides clinicians with actionable insights that help improve patient care. Some of the key features of Better Portal are:

- SSO and one-click away ecosystem
- Adaptability to care settings plus individual user’s roles and needs
- Care plan and assesment modules
- Configurable patient lists, patient banners, clinical tracking boards, and patient-specific summary view modules
- Clinical documentation viewer
- Web based and responsive design across devices

Better Portal works in Safari, Chrome, Edge, Firefox.

Pricing model for components:

Similarly to pricing model for the platform, also the clinical modules have both perpetual license and subscription license pricing model. There are different options how the components license value is estimated, in some cases it is based on number of patients, as an upgrade to the platform license, in others it is based on the size of the hospital/trust and in the others we might offer even open source license.

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

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)

We believe that a collection of healthcare standards and architectural approaches should be combined to establish an open, interoperable healthcare system. This collection of standards should include semantic standards, such as ICDx and SNOMED CT, architectural standards, such as openEHR, interoperability standards, such as HL7 v2, CDA, FHIR and IHE, and quality standards, such as the ISO set of standards.
When building a health information technology platform using a model based architectural approach, it is important the core interoperability standards (e.g HL7, FHIR and IHE) can be seamlessly mapped to the chosen architectural standard (e.g openEHR).
To support this, Better has developed an extensive collection of integration services that support the mapping of openEHR information to various other standards or external systems. We also provide a collection of tools that support the implementation of mappings, queries, and validators.

The architecture to support this is summarised in the diagram below:

https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/Ecb8FIhUerZMm6SnQz3__Y8BF_FIPS1As14GJ2FnlLEDmQ?e=yrZ7kl

Below you may find more details and descriptions about the mentioned diagram:

Software Development Kits (SDKs)

Better Platform provides several SDKs to facilitate the creation of applications. These include:

Better Platform Development Framework – A Java based SDK providing a collection of tools that support application development, including:

o AQL Builder - To ease writing openEHR AQL queries, a helper class AqlBuilder is available. It allows easier use of parameters, select aliases, limiting, grouping, filtering, and sorting of result sets, terminology resolution and more.
o AQL Model - As an AQL query is usually represented as a string, it can be difficult to analyse or modify it programmatically. To aid with this, Better Platform Development Framework contains tools to convert an AQL between a string and an object model representation. Additionally, the object model can also be serialized to and from JSON
o EHR Rest Client – The Rest Client makes it easier to access to the most used rest methods on the EHR server using RestTemplate. It fully supports EhrMapper and provides the ability to automatically convert query results to the desired type on the client-side.

Web Template – Web Templates provide a simple way to work with openEHR compositions. Web templates are generated from operational templates and provide a simplified web-oriented view of an operational template. They provide enough information to allow clients to display data entry forms as well as provide a facility to convert data retrieved from such forms into canonical openEHR compositions (and vice-versa). Their use is by no means limited to web-based use. Data conversions and data import are areas where they can be useful as well. For HTML based GUI interfaces, they are a basis for the form designer and JavaScript based form rendering which is part of Better EHR framework. One of the major benefits of this technology is that new templates and forms can be deployed without having to redeploy server-side application as no recompiling is needed. Better have open sourced this SDK, which is available at this GitHub link.

Template Data Schema - EHR Server provides a template specific XML Schema generator. Client application can then use this schema to create a template specific xml document (TDD) and submit it to EHR Server. TDS/TDD approach may be a convenient way to access template specific data from the non-Java-based client or for integration with 3rd party systems which understand xml schemas.

EHR Mapper - It is often desired to be able to create custom data classes in Java yourself and be able to map them to OpenEHR compositions. This can be achieved with EhrMapper. EhrMapper converts the custom types to OpenEHR RM values or OpenEHR RM values to the custom types using @EhrMapped annotation. EhrMapped annotation must be used on fields of the custom data class.

R SDK - R is a popular programming language for statistical computing, which is widely used by research and analytics teams. Better Platform provides an R SDK that facilitates the querying (via AQL) to fill a data frame, which means you can use data from the Better EHR Server to perform some statistics, analytics, or predictions. Furthermore, the R SDK can also be used to create a composition to store data back in the EHR Server — for example, the likelihood of developing cancer.

Publish/Subscribe APIs

Better Platform can react to specific openEHR compositions as they are stored. This is very similar to triggers in relational databases. Each event registered by the openEHR server has an AQL query associated with it. A non-empty result of an AQL query triggers an event. Each event also has a 'phase' – either synchronous or asynchronous. Synchronous events are executed as a part of an incoming composition. They have access to composition's data and can thus add additional compositions to be committed. They can also veto specific parts of a composition from being committed. Asynchronous events are executed after a composition has been successfully committed, and a result returned to the calling application. The executing event will receive all data that is the result of the AQL associated with it. For composition updates, previous values and new values are available to the event.
To guarantee the delivery and support high throughput, Better Platform uses the Apache Kafka event streaming platform (ESP) to provide data streams to other platform components or 3rd party systems.
All Better Platform services can publish events to the ESP. Services like Data views, Decision

Support, and others subscribe to these events. The ESP service is designed so that producers and consumers are fully decoupled and agnostic of each other, which is a crucial design element to achieve high throughput, scalability, and resilience.
This approach enables Better Platform to easily scale horizontally and vertically in order to meet any environment's needs, whether it's high numbers of concurrent users, a massive dataset to analyse, or both at the same time.

HL7 FHIR

Within the Better Platform there are two main areas for FHIR adoption.
• Patient Demographics and Operational Data Repository – The Better Platform provides data repositories to store patient demographic information and operational information, including appointments, admissions, and encounters. These data repositories have been built using a FHIR server with FHIR storage and native FHIR R4 APIs.
• FHIR Connect – All clinical data within the Better Platform is stored within the openEHR-based Clinical Data Repository which exposes native openEHR APIs and SDKs (as outlined above). FHIR Connect is component that has been designed to support bi-directional mapping between FHIR and openEHR. Further information on FHIR Connect is outlined below.

FHIR Connect

FHIR Connect is part of the Better Platform that has been designed to support bi-directional mapping between FHIR and openEHR. Traditionally, mapping between FHIR and openEHR has been performed inside a third-party integration engine or embedded within propriety software routines. In contrast, we have designed FHIR Connect around a vendor-neutral, re-usable mapping specification which we have open sourced. By adopting this approach our goal is to allow a community to emerge that will create, share and re-use mappings — in a similar way to what has happened in the openEHR community with re-usable archetypes.
To achieve this, we have designed FHIR Connect with a dual mapping approach that allows a core set of re-usable mappings to be established, with an additional group of contextual mappings — aligned to a specific use case — to be configured locally by implementers.
Link: https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EbkYjfwwiCVPvOrn6XV8bL8BmzZ4w5W7Guag5BGRnOS6_Q?e=2JppMc

The first set of mappings — called Model Mappings — are between openEHR archetypes and FHIR resources. This represents a mapping between the two smallest comparable units of semantic information by both standards and as a result, allows these mappings to be globally re-usable where the given archetypes and FHIR version are deployed.

The second set of mappings — called Contextual Mappings — allows the context of a specific use case to be handled. In openEHR this context is always represented by a template. On the FHIR side, this context could be a variety of FHIR artefacts including a resource, profile, bundle, or implementation guide.
At runtime, a mapping engine will evaluate the context of each request and apply the appropriate mapping logic. We have designed FHIR Connect in a modular way to it can be plugged into a variety of use cases either as a facade, message broker or a sync agent.

API Standards

openEHR APIs
All clinical data within the Better Platform is held within an openEHR-based clinical data repository, called the Better EHR server. The Better EHR server exposes an openEHR REST API coarse-grained interface supporting check-out, commit, and query operations that implement runtime archetype and template processing logic. It also provides an API with the ability to create, modify, and query EHRs. The API layer is stateless by nature. EHR server exposes the following REST API groups:
• EHR for management of EHRs,

- composition for management of compositions,
- query for executing AQL queries,
- view for retrieval of EHR data based on pre-stored queries (similar to views in relational databases),
- presentation for generic compositions as document displays of the EHR data, and
- template for the management of openEHR operational templates.

openEHR Views
EHR Server views are similar to views in SQL. They are mapped to URLs on the REST API. In many cases they are used to divide work among frontend programmers as view consumers and advanced openEHR analysts/architects who prepare the view definitions.
Some views are very simple - i.e. AQL result-set converted to JSON. Others can be more complex, for example a Javascript view can combine results of multiple AQLs (executed in parallel, if needed) and produce a new JSON object with the exact format required by the client application.
EHR Server views support processing pipe-lines which allow passing view data from one step to the next, each performing a part of the processing. Most views though only use a single processing step.
Views can be managed in a web-based admin console.

HL7 v2, v3 and CDA
Better Platform uses the FHIR standard to store demographic and operational information, including encounters and appointments. Typically, the source of this information is from a Patient Administration or Hospital Information System and is communicated using HL7 v2 messages. Additionally, there are several use cases where information communicated in HL7 v2 or CDA format needs to be converted to openEHR. For example, orders or results from a Radiology Information System.
The Better Integration Engine can listen to incoming HL7 messages from any of these sources and transform them into messages stored in the FHIR demographics or openEHR server. Integration services provide the capability for content-based-transform, message mapping and transformation to openEHR compositions. Users can create a map file that translates a source document format to the destination format, including mapping fields, business rules or algorithmic conversions. Plugins provide message processing and forwarding to target systems. HL7 V2 message or HL7 CDA mapping and transformation to openEHR compositions are supported out of the box.

IHE Integration
The Better Platform is validated for IHE Integration profiles, enabling the platform to act as an IHE validated document repository (XDS.Repo) and document registry (XDS.Registry), providing audit repository and logs (ATNA), consistent time functionality (CT), cross-enterprise user assertion (XUA) and cross-community access and peer-to-peer querying and document retrieval (IHE XCA).

OpenAPI
As illustrated below, OpenAPI 3 is used for Better API documentation. This makes integration with API management solutions easy and straightforward.
https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EarVSCv4JcFMntC64e0swhABrR3A-imc64okKc7MQ_QM1A?e=ayGZPg

ODBC
As outlined below in Extract, Transform and Load (ETL), data held in the Better EHR Server can be exported to a reporting or analytics environment where ODBC-based clients (e.g. Microsoft PowerBI) can connect to and interact with the data.
Integration with other standards and proprietary systems
In addition to the interoperability services outlined above, the Better Platform also contains an embedded version of the Rhapsody integration engine from Lyniate. Rhapsody is used by over 1.500 customers to integrate with all major healthcare systems across a range of domains,

including EHRs, Labs, eMPI, Population Health, and Care Coordination. The Rhapsody platform supports all major standards and formats, among them HL7 (v2 and v3), HL7® FHIR, CCDA, NCPDP, X12, IHE, DICOM, JSON, and XML, in addition to connecting data using non-standard or proprietary interfaces. Rhapsody is an optional module with a consumption-based pricing mode.

Medical Imaging Standards

IHE Integration
The Better Platform is validated for IHE Integration profiles, enabling the platform to act as an IHE validated document repository (XDS.Repo) and document registry (XDS.Registry), providing audit repository and logs (ATNA), consistent time functionality (CT), cross-enterprise user assertion (XUA) and cross-community access and peer-to-peer querying and document retrieval (IHE XCA).
While the Better Platform is formally validated for XDS it is not currently validated for XDS-I, which is the extension to the IHE XDS protocol specifically for imaging. However, the XDS-I extension uses the same set of actors as XDS-I with the imaging specific use-cases being performed by an external system (such as the PACS), so we would expect to be able to extend our current capability to support this.
If required, Better would be happy to discuss how we could your needs around integration in this area.

DICOM
openEHR stores structured clinical data and as a result is not designed to store medical images. However, DICOM metadata — such as the DICOM Study and DICOM series level attributes — can be stored within openEHR, usually within the “OBSERVATION.imaging_exam_result” archetype.

OMOP and Other Research Standards

OMOP
There are currently several community initiatives relating to mapping openEHR, FHIR and OMOP. Microsoft have developed a FHIR to OMOP converter which we can support using FHIR Connect outlined above.
Furthermore, the openEHR community are discussing different approaches on the best way to maps these two standards. For example, via annotating the openEHR archetypes/templates with the equivalent OMOP class.
Better fully support these initiatives and plan to support a mapping capability once there is an agreed upon approach.
If there is a short term need to deliver OMOP mapping then the ETL capability outlined under Extract, Transform and Load (ETL) would provide a mechanism to support this.

Other Research Standards
R is a popular programming language for statistical computing, which is widely used by research and analytics teams.
Better Platform provides an R SDK that facilitates the querying (via AQL) to fill a data frame, which means you can use data from the Better EHR Server to perform some statistics, analytics or predictions. Furthermore the R SDK can also be used to create a composition to store data back in the EHR Server — for example, the likelihood of developing cancer.
Integration with existing EHR systems in Sweden

Currently we have a working integration with Medanets system. Worth mentioning, TietoEvry’s Lifecare has been built fully on Better Platform, and as such is automatically integrated with our stack, moreover the use Better Studio for low-code development and have built own form renderer. Since the platform is based on openEHR, it is quite straight forward to integrate with any native openEHR system but also FHIR based system due to FHIR Connect component, as well as HL7 systems (mainly for labs).

Extract, Transform and Load (ETL)

Better ETL is a component within the Better Platform that provides the ability to extract openEHR data into flat structures. Each extract is defined with an openEHR Archetype Query Language (AQL) query which flattens openEHR structures into either relational database tables or MongoDB collections. These are then suitable for use with BI tools or as a data service for more complex dynamic dashboards. Extractions are asynchronous from the Better Platform's EHR server delivering new data as an event stream via Kafka, providing almost real-time updates. A web-based tool is provided that allows the user to easily specify the AQL queries that will be used to extract data from the primary EHR Server.
https://bettercare365-my.sharepoint.com/:b:/g/personal/jovanp_better_care/EQ_sC3aw8ZFMhN_VHBj8_c4BDVx2SYVvLF9IUESWUmr_aw?e=80a3AC

All data extracts to the read-only EHR server is replicated in a real-time basis with no preparation required. Once the data to be extracted has been identified using AQL the data needs to be mapped to its destination. A web-based tool is provided that allows the user to map individual AQL columns to the destination database. Once the ETL job(s) have been configured they can either run on a scheduled basis, or on a real-time basis, as well as a one-off task.

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

Following the openEHR specification, Better Platform can look up terminology codes for the AQL terminology function or validate them on storing of Composition. Better Platform, by design, supports the integration of terminology servers using HL7 FHIR based on CodeSystem and ValueSet resources, which must be enabled. In AQLs, it is often helpful to resolve a set of terminology codes via a terminology REST API call. This way, you can include localization or query/filter by custom terminology properties, not just descriptions.v AQL function TERMINOLOGY forwards its parameter as a REST API call to the terminology service.

Better Platform also allows for external terminologies to design applications using Better Studio. For example, form builder provides the capability of externally defined variables. If a variable is externally defined, the form will not calculate the value. Instead, a parent application must pass the variable in Form Renderer as input. There are two existing cases of externally defined variables: ehrID and terminologyURL.

For the purpose of managing terminologies, Better Platform provides users and customers a component - The Better Platform Terminology server. It is a software component for managing and practically deploying standardized terminologies. The Better Platform Terminology server provides HL7 FHIR-based terminology services API, which works and uses the clinical language and other code systems for a clinical domain. The Better Platform Terminology server provides consistent programming interfaces, APIs and functions for managing and operating the terminology for a clinical domain. It may include vocabulary lists, value sets, taxonomies, concepts, and relationships. Terminology services can be used to support the following:

- Operational systems at runtime with dynamic interpretation and encapsulation of codes.
- The development and maintenance of systems by providing a repository of concepts and terms and a means of extending this repository coherently and co-operatively.
- The consistency and data integrity with terminology-based validation.

The Better Platform Terminology server is an HL7 FHIR R4-compliant solution that holds and disseminates assured international terminologies and classifications (such as SNOMED-CT and ICD-10) and national terminologies. We have developed the Better Platform Terminology to deliver assured content in FHIR compliant, machine-readable formats, regardless of which terminology is accessed. The Better Platform Terminology includes code systems, value sets and concept maps for many national and internationally assured terminologies and classifications. Currently loaded are (but not limited to):

- SNOMED CT - current version and current-1 at a minimum
- ICD10
- LOINC
- FHIR - including all published CodeSystems, ValueSets and ConceptMaps

The Better Platform Terminology server includes capabilities of retrieving and validating the data via the API to applications. The Better Platform Terminology server API makes it easy for external applications to reference and store concept entities, for example, as part of a patient record system. A flexible interface has been developed so that individual modules may 'export' their services via the API to external applications, so additional functionality can be made available very quickly.
Nevertheless Better Platform can also interact and integrate with 3rd party (external) Terminology service has already been tested and proven with Ontoserver (https://ontoserver.csiro.au/site/our-solutions/ontoserver/) as the world-leading clinical terminology server implementing the FHIR terminology services API and supporting syndication–based content distribution.

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

Text field

With Better Platform we follow one of the basic prinicples of openEHR – the complete complete separation of EHR and demographic information, such that an EHR taken in isolation contains little or no clue as to the identity of the patient it belongs to. The security benefits are described below.
In more complete EHR systems, numerous other services (particularly security-related). For demographic, administrative and operational data (non-clinical data) Better Platform is providing a decoupled data repository. This repository is not build using openEHR demographic model, but it is HL7 FHIR-based data repository with FHIR-based API that information required for performing healthcare services on a hospital level, enterprise level, and region or national level. Nevertheless it performs two functions: standardisation of demographic information structures and versioning.

- ·Managing a Master Record of a Patient and a Person; providing the capability for creating a “single source of truth” for patient identity and demographic information. The repository holds demographic information on every patient who receives healthcare services, ensuring that patient data is correct and consistent throughout the region or enterprise, regardless of the updates and upgrades to the ecosystem. Demographic data is represented as HL7 FHIR resources. A Patient resource describes patient demographic information and any updates to it. In addition, it can be used to communicate Patient information to other systems (e.g. other registries, clinical, ancillary and financial systems).
·
- Managing a Master Record of a Provider and Service Catalogue; With resources such as Practitioner, PractitionerRole, Organization, Location, and HealthcareService enabling the registry of individuals related to providing health care services, creating a complete picture of where, how and by whom the care services are offered to a patient.
·
- Managing Other Administrative Records, focusing on storing information about clinical activities, which occur in different ways. Episode Of Care typically covers long-term care, whereas Encounters cover short-term care. The Appointment resources permit the planning of encounters to appear and follow on with other clinical activities.

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

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?

re i

We have implemented the full openEHR AQL Specification ver 1.1.0 (https://specifications.openehr.org/releases/QUERY/latest/AQL.html), with the following exceptions:

• TERMINOLOGY() function supports only 'expand' functionality
https://specifications.openehr.org/releases/QUERY/latest/AQL.html#_terminology
• From the String functions only CONCAT() is implemented at the moment
https://specifications.openehr.org/releases/QUERY/latest/AQL.html#_string_functions
• From the Numeric functions none of them is supported yet
https://specifications.openehr.org/releases/QUERY/latest/AQL.html#_numeric_functions
Better AQL implementations has additional functions, such as:
• Results can be grouped using the GROUP BY statement.
• Grouped results can be filtered using the HAVING statement.
• Text search allows matching with wildcards: * and ?. For example:
SELECT c FROM COMPOSITION c WHERE c/name/value LIKE 'Body*'
• Filter query data based on tags, for example:
SELECT c FROM COMPOSITION c WHERE c TAGGED BY 'tag::value'
• Retrieve existing tags on composition/data elements, for example:
SELECT tags(c) FROM COMPOSITION c
• Retrieve and filter tags on composition/data elements, for example:
SELECT tags(c, 'tag1::value1::/') FROM COMPOSITION c
SELECT tags(c, 'tag1') FROM COMPOSITION c
SELECT tags(path = c, predicates = ['tag1', '::value2::/']) FROM COMPOSITION c
• Server function current_state() returns the current state of an instruction activity as per The Standard Instruction State Machine of the OpenEHR standard
• Server function instruction_aggregate_state() returns the current instruction aggregate state, so named because it is computed from the states of all of its activities as defined in the OpenEHR standard
• Server function linked_action_compositions() returns composition of a linked action, connected by its instruction details to an instructions activity contained in composition of id compositionUid.
• Server function careflow_step() returns the current careflow step of an instruction activity, function careflow_step_code() returns the current careflow step code. Example:
SELECT careflow_step_code(path = i/activities[at0001]), careflow_step(path = i/activities[at0001]) FROM INSTRUCTION i[openEHR-EHR-INSTRUCTION.medication.v1] WHERE careflow_step_code(path = i/activities[at0001]) = 'at0107'
• Server function squash() does group data objects based on a common property, example:
query:
select i/activities[at0001]/description[at0009]/items[at0121]/value/value as activity, i/activities[at0001]/description[at0009]/items[openEHR-EHR-CLUSTER.timing_nondaily.v1]/items[at0003]/value/value as dayOfWeek from EHR e CONTAINS INSTRUCTION i[openEHR-EHR-INSTRUCTION.service_request.v1]
would return
[{ "activity": "Measure blood pressure", "dayOfWeek": "Monday" }, { "activity": "Measure blood pressure", "dayOfWeek": "Wednesday" }, { "activity": "Measure blood pressure", "dayOfWeek": "Friday" }, { "activity": "Measure blood sugar", "dayOfWeek": "Tuesday" }, { "activity": "Measure blood sugar", "dayOfWeek": "Thursday" }]
but query
select i/activities[at0001]/description[at0009]/items[at0121]/value/value as activity, squash(i/activities[at0001]/description[at0009]/items[openEHR-EHR-CLUSTER.timing_nondaily.v1]/items[at0003]/value/value) as dayOfWeek from EHR e CONTAINS INSTRUCTION i[openEHR-EHR-INSTRUCTION.service_request.v1]
returns
[{ "activity": "Measure blood pressure", "dayOfWeek": ["Monday", "Wednesday", "Friday"] }, { "activity": "Measure blood sugar", "dayOfWeek": ["Tuesday", "Thursday"] }]
• Server predicate has_link() is used to filter result set by composition link target, type or meaning attribute value.
• Server function last_version_uid() is used to retrieve the latest composition version UID.
• Server predicate is_last_version() is used to filter result set by the last composition version.
• We do support partial matches in archetype names predicates using wildcards, for example:
SELECT o FROM EHR e CONTAINS ACTION a[openEHR-EHR-ACTION.procedure.v1] CONTAINS OBSERVATION o[openEHR-EHR-CLUSTER.device-invasive*]
• We do support partial matches in names in AQl path segment using wildcards, for example:

SELECT s/items[openEHR-EHR-OBSERVATION.body_temperature.v2,'Body temperature*']
FROM EHR e CONTAINS SECTION s
• Support for joining data from multiple compositions:
SELECT e/ehr_id/value, alg, med FROM EHR e CONTAINS (COMPOSITION c1 CONTAINS EVALUATION alg[openEHR-EHR-EVALUATION.adverse_reaction-allergy.v1] AND COMPOSITION c2 CONTAINS INSTRUCTION med[openEHR-EHR-INSTRUCTION.medication.v1]) LIMIT 10
• Full text search capability, using Lucene query syntax
• Support for UNION functionality over multiple queries
re ii.
All mentioned RM parts are readily accessible with AQL, with the exception of the FOLDER and its substructures.

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

See 2.1.4(f)(ii.)

Querying capability allow for including TAGs as results/filtering criteria, but does not support FOLDER structures to be used the same way.
There is a dedicated API endpoint /folder for FOLDER management.

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

Archetype Designer is a tools, part of the Better Platform, that allows storing any artefact used or created for model design, on the GIT based repository or similarstorage and/or resource versioning service:

- Google Drive
- Dropbox
- GitHub
- Git
- BitBucket
- Box

We do provide governance guides for the resources versioning (templates, file sets, queries, views, terminologies), using sem-ver and archiving strategies.
Forms built using Better EHR Studio - Form Builder are internally versioned. EHR Server supports use of multiple form versions at the same time.

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

Better is implementing several processes, tools and components to ensure the compliance of openEHR components of Better Platform with openEHR standards and specifications. Conformance with openEHR standards is done in several ways. As part of Better R&D activities we are incorporating a set of unit and integration testing to validate individual components and platform modules as a whole with a focus to determine conformance to the published openEHR specifications we are testing: API conformance, Data Validation conformance. Next to internal monitoring, we have implemented tests defined and published as part of the openEHR community (openEHR Conformance framework). The framework (and included tests) are used and documented (audits) as part of the Better Platform code. Additionally, Better is also providing additional test examples for conformance using standardised openEHR REST APIs and WebTemplate tests on different CDRs. Tests are available as open source (apache licence): https://github.com/better-care/web-template-tests/blob/master/README.md and https://github.com/better-care/openehr-rest-tests

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

Better Platform fully supports multi-lingual features of OpenEHR templates as well as adding additional support for parts where translations directly in templates are not possible (such as constrained node names). Such translations are available as additional annotations on OPTs and are fully supported by web templates and other technologies used for GUI development.

Archetype designer – tool for creating archetypes and templates is fully multilingual (and can easily be translated into Swedish). Better Studio also has this capability although we do not have any translations at the moment, but are happy to provide it in the future.

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 are fully supportive of combing our solution with components from other openEHR vendors. We have experience of doing this from two different perspectives.

- Application Perspective – We have worked closely with a number of application providers who nativley support openEHR, where we collaborate together to build an ecosystem around the deployed openEHR platform. For example, in a UK project for Suffolk and North East Essex Integrated Care System (SNEE) we worked with a person health record (PHR) provided by a company called COHESION. COHESION have created an application with a range of person centred functionality, including care planning. The care planning functionality is natively integrated into the Better Platform using openEHR APIs.

- Data Perspective – As part of their Connected Health Cloud, EY have been developing and testing a federated approach to combining multiple openEHR systems together. This involves sending a single query to a 'global federation' service, which is responsible for querying all the openEHR nodes in the network, aggregating the results and providing a single response back to the user. As part of the testing completed for this work, EY have successfully demonstrated how the Better Platform can participate within this architecure alongside an openEHR repository from another provider.

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

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)

As part of the EHR Studio we provide a tool for creating and managing AQL queries called AQL builder. Easy to use interface allows users to create AQL queries in a matter of seconds with just a few clicks, regardless of OpenEHR expertise. Instead of writing queries from scratch AQL builder provide list of all the available OpenEHR templates saved on the platform as long as some other commonly used reference models that enables creating queries by double clicking on the data user want to query. There is no limitation on how many templates user can use in a query which makes querying and joining data in a single response a straightforward task (i).

Since OpenEHR path can be hard to read and debug we provide easy to read query formats by default which hides the complexity of archetypes and its nodes behind human readable names (for instance t[openEHR-EHR-OBSERVATION.condition_screening.v0] would be transformed into OBSERVATION t#Psychic_screening_questionnaire). User could always opt-out to see plain archetype path values if they have required level of knowledge to read them. AQL builder is also equipped with state of the art autocomplete, syntax highlighting and comprehensive error checking to support user with assembling any query they need. This simplifies writing and validating queries, be it a simple SELECT or a nested complex query containing JOIN, GROUP BY or WHERE statements (i).

Most of the queries requires one to many parameterised values to filter or limit the search. User can easily parameterise query with placeholders starting with colon (:dateFrom). AQL builder will recognise this spots as dynamic inputs and will show special inputs next to editor where test data could be provided for easy testing and executing of the queries (ii).

In addition to its query creation, AQL Builder provides a powerful presentation layer in a table form. When working with OpenEHR structured data, it can be easy to become overwhelmed by the sheer amount of information available. That's why we've designed AQL Builder's presentation layer with three levels of detail, allowing users to focus on the information that is vital for their needs. Table view provides an overview of the data, displaying key values based on the experience we gathers on the market over the years, but also allows a deeper level of granularity, which provides access to all available data points. Since table is universally recognised presentation format for data it can be exported in widely supported coma separated value (.csv) format (iii).

AQL builder automatically stores every executed query in users personal history for easy reference and retrieval of past queries. User can also save whole or just part of their queries for future use. Additionally user can save any query as Platform Views which serves as a secure querying mechanism to query data from any application or client. To further enhance the capabilities of Platform Views and support post/pre-processing of query results, users can create JavaScript Views. These views come equipped with a unique API that enables users to pre/post process queries and results tailored to their needs. Since writing and debugging JavaScript views can be a challenging task, AQL Builder provides a suite of tools to assist users in the process of making and executing it (iv).

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

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

re i.
Our model design tool Archetype Designer fully supports nested/embedded templates.

re ii.
Better Platform was tested with Archetype Designer (Better) and Template Designer tool from Ocean Informatics.

re iii.
Our modelling tool Archetype Designer supports ADL2 and ADL2 templates.

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. Developing GUI:s
GUI is mainly developed using our Better Studio components, a drag and drop form/app development tool, with an option to extend any developments with a custom script code, and it also comes with a design system for providing consistent user experience. The user also has an option to create custom widgets as predefined components that may be used on various screens. The low-code development tool is more described in section: 2.1.4. h) ii)
ii. Data management
EHR Server has an administrative dashboard, that allows users with special privileges to visually access all EHR Server resources and manage them via simple interface.
All the functionality is also exposed via Administrative API, available to user with special privileges.
Better Studio has an AQL Designer tool, that allows for point-and-click query design, based on templates and openEHR RM classes, together with query validation and visual result-set presentation.
Data export is handled by our ETL tool, that uses AQL to retrieve data from the CDR and includes a mapping engine, for data export to a relational database for BI, research or other analytical/data warehouse purposes.

iii. Import, export, and migration of data, metadata and system configuration, in open well documented formats
System resources can be fully managed via the Administrative API.
EHR Server dashboard exposes a GUI to access the same resources, allows for download/upload operations, for backup and distribution purposes.
Clinical data can be accessed via /query API, allowing for CSV serialisation. Same functionality is exposed using the Better Studio - AQL Builder tool.
For large amount of data we do support streaming query request.
Better Platform supports importing of the clinical data in CSV format
Data export is handled by our ETL tool, that uses AQL to retrieve data from the CDR and includes a mapping engine, for data export to a relational database for BI, research or other analytical/data warehouse purposes.

iv. SMART on FHIR integration
We have developed two integrations based on SoFHIR, namely with CERNER and EPIC solutions, which served their data.
Once we created the accounts and applications with their data resources access, we integrated that with the Better Platform. Past the authorisation phase, we retrieved the data from FHIR endpoints and launched our Portal application and to display the data within the app.

v. Mapping and conversion support other standards such as HL7/FHIR
As described in 2.1.4 c) we have several integration / mapping components that allow seamless integration between different exhcange mechanisms and openEHR data repository.

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

i) - While we are providing Better Platform form renderer, our partners have developed also their own renderers in order to achive the same look and feel as the overall application (usually and EMR) built on top of the Better Platform.

ii) - When you gain access to EHR Studio, you also get the access to our JavaScript library, known as the Form Renderer. This rendering library provides a way to render the forms and views produced by EHR Studio. Form renderer is a Web Component meaning it can be seamlessly integrate into any modern JavaScript codebase (i).

EHR Studio provides a Form builder module, which is a powerful low-code/no-code tool enabling users to create forms based on the OpenEHR templates, making it easy to capture clinical data in structured format that can be later shared among healthcare providers and institutions.
Assembling a form is as easy as selecting a OpenEHR templates and drag and dropping the fields or group of fields to canvas which will present the form with some additional metadata needed for building a form.
Form builder also supports defining a custom logic and dependencies between fields in a concise and human readable format which can be assembled and read by less technical people.
In addition to basic form input fields, data can also be captured through interactive images. Interactive images captures data through pressure points binded on the actual data from the model.
Alongside template we also provide supporting elements that can be used to further optimize capturing and/or presentation of the data for the end users (from simple inputs to more complex components for data visualisations - e.g.: Score & Scale, parameter values, graphs,..).
Data connectivity is enabled through API connector embedded into Form builder, to easily connect with relevant data sources. When connected, datasource can be used easily within form fields to provide data bindings (ie. term bindings) or used for presentation of the already recoded data.
Where there is need for capturing data there is mostly likely also need for presenting those data to patients or clinicians so Form builder also provides option to create Summaries which are way of presenting the data captured by the form (ii).

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

Better’s standard for log collection and monitoring is use of Prometheus, Grafana and Loki (depending on the solution additional products can be added). This tool set fully supports complex monitoring scenarios and alerting.

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

Better Platform™ provides a strong, flexible, consistent, and high-performance security infrastructure while minimizing its application performance burden. This security architecture is based on authentication, authorization, auditing, and database encryption.

• Authentication. Better Platform™ supports basic, OAuth2 and multi-factor authentication. Several options are available including Keycloak, ADFS, Azure B2C
• Authorization. Using comprehensive authorizations, you can easily assign and manage role- and application-based resource access privileges either as role-based access control (RBAC) or attribute-based access control (ABAC)
• Auditing. The audit server collects audit logs from all platform services for traceability, dispute resolution, security, privacy, and compliance with policies (external and internal). It fully supports the IHE ATNA standard.
• Database encryption. Better Platform™ encrypts data-at-rest and data-in-motion. To protect entire databases, it offers block-level encryption.

By default, role-based access control is used for platform APIs. Additionally, many cases require more granular access control based on specific attributes giving specific persons a right to access a data set. Attribute-based access control (ABAC) allows writing access rules based on relations between parties and attributes defined within the access context.

Better has completed key assessments and updates to satisfy the GDPR requirements and all initiatives have been executed with the goal of providing transparency to data subjects regarding the care with which their personal data is treated. Better's products and services meet the principles of privacy by design and default as outlined in Article 25 of the General Data Protection Regulation (GDPR). Adherence to these standards means that our products have appropriate privacy and security features embedded within their design, and Better has the ability to fully support the data subject rights called as outlined by GDPR.

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

Text field

ISO 9001 and ISO 27001

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

Better platform supports consents, RBAC and ABAC, and is fully GDPR compliant . Privileging and security are deeply embedded in the system architecture and does not depend just on the user interface layer, however, it is enforced server side and in a way that privileging models can be configurable for each install/user environment, with granular user-specific permissions calculated and enforced.
Better provides an implementation of an ABAC (Attribute-Based Access Control) server which other backend components consult, if configured so, before proceeding with execution of API requests, or to filter out sensitive data from query results. Access policies are (re)configurable on a running system without downtime using a proprietary domain-specific language.
ABAC Server needs to be aware of all the parties involved (clinicians, patients, perhaps users or organisations) and relationships between them (consents given, organizational structure etc.) which is usually achieved by writing a synchronization plug-in for the ABAC Server.
The Better tools and products are in continuous evolution and ABAC Server has a roadmap that will support ever evolving and more complex security needs.

EHR Server (our implementation of openEHR CDR) supports both Basic authentication with users defined within the EHR Server (or EHR Server cluster), and SSO using OpenID Connect. In the former case, user's roles on various domains (tenants) are also defined within the system, while in the latter case, the user's sole domain (tenant) and roles are defined within the SSO system (encoded into the JWT).
ABAC is an orthogonal functionality described in section 3.1.1. This can also be applied to implement access based on care relationship or emergency access.
Each EHR can, as per openEHR standard, carry a "subject ID", and the Patient resources (FHIR) which can be stored within the Demographics Server can have as many identifiers as needed. This can be used to relate to entities within an MPI.

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

To mitigate the risks associated with the development, maintenance, acquisitions, and sunsetting of our systems and components, but also to satisfy the ISO standards processes, we have defined a list of all software components and their dependencies that make up a product or service. This is an active list, regularly maintained, which together with additional security scanning tools such as SonarQube and Trivy allow us to quickly identify any vulnerabilities or risks associated with the software components and take appropriate actions to mitigate them.
In addition to this list, we also implement a comprehensive set of security and risk management policies and procedures, which are based on industry best practices and standards such as ISO 27001 and Cyber Essentials. We also have ISO 9001 standard for operational quality. These policies and procedures cover the entire software development lifecycle, including design, development, testing, deployment, maintenance, and sunsetting.

To ensure compliance with our security and risk management policies and procedures, we have a dedicated team of professionals who conduct regular audits, assessments, and reviews but also hire 3rd party companies to do external audits and assesments. We document all risks and mitigating actions in a centralized risk management system, which allows us to track and manage risks across the organization. Our risk management system also enables us to enforce compliance and take corrective actions when necessary.

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

Better provides a wide range of Development programs that are designed to help a wide range of our solution users. With development programs Better customers and partners get not only the access to latest technologies, but possibility to interact with our clinical and development teams in process of their own capacity building and portfolio generation. The end result is faster, more efficient and sustainable offering on the partner end, supported by Better. Better currently offers several development programs that fit the needs of different healthcare, ducation, technology and service provider needs. Development program specifics are agreed upon consultations with the interested parties, but nevertheless offer both on-line, pre-recorded content, remote or in-person training packages.

Besides the development programs, Better offers an option of giving interested parties access to a sandbox environment, through which users can learn at their own pace, using available tools, documentation, best practice guides and other materials, which are part of an extensive knowledge base.

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

Text field

Our product documentation provides a full set of documents including but not limited to:
- user manuals
- installation guides
- opeartional guides
- system administration guides
- cookbooks
- and other technical documentation that provides in-depth information about our products and services.

Our user manuals provide step-by-step instructions on how to use our products and services, including screenshots and diagrams to help our customers visualize the processes. Installation guides provide detailed information on how to install and configure our products and services, while our system administration guides provide guidance on how to manage and maintain our products and services. To complement our product documentation, we also provide a range of training materials, including online courses, webinars, and in-person training sessions. These materials are designed to help our customers get up to speed quickly and efficiently, ensuring that they can use our products and services effectively.

Finally, we also provide video tutorials and online knowledge bases, which allow our customers to access information on-demand. These resources provide a quick and easy way for our customers to get the information they need, when they need it.

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

Text field

Our go-to-market model is primarily based on working closely with partners, as we believe that this approach allows us to reach a wider range of customers and provide them with the best possible service. We work closely with our partners to ensure that they have the necessary knowledge and expertise to implement, configure, and develop our products and services effectively.
However, we also understand that in some cases, customers may require direct assistance from us during the implementation, configuration, and development phases. In these cases, we work closely with the customer to understand their specific requirements and provide them with the necessary support and assistance.
In situations where there is a strong IT team at the customer's side, we are happy to work directly with the client on these activities. Our team of experts can provide guidance and support to ensure that the implementation, configuration, and development are done in a timely and effective manner.

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.