

3. Company information

3.1. General		Answer
3.1.1	Company Name	Tieto Sweden AB in partnership with Marand Inzeniring d.o.o.
3.1.2.	Company Main office Location	Tieto- Stockholm Marand- Ljubljana Slovenia
3.1.3.	Company location in Sweden (cities)	Stockholm, Göteborg, Malmö, Linköping, Jönköping, Falun, Gävle, Östersund, Karlstad, Kalmar and 8 more cities
3.1.4.	Number of Employees (total)	Tieto Group 13 000, Marand 148
3.1.5	Number of Employees in Sweden	Tieto Sweden 2500
3.1.6.	Web address to company product site	http://www.marand.com/Think!EHR/ https://dev.ehrscape.com https://www.tieto.com/industries/healthcare-and-welfare

3.2. Contact		Answer
3.2.1	Name of sales contact	Tieto: Nima Fooladianpour (Senior Sales Manager) Marand: Miroslav Koncar (Business Development Director)
3.2.2.	E-mail of sales contact	Nima.fooladianpour@tieto.com Miroslav.koncar@marand.com
3.2.3	Phone number of sales contact	Tieto: +46 72 536 50 73 Marand: +385 91 4410719
3.2.4.	Name of technical contact	Tieto: Markoo Pyy (Head of Product Management) Samo Drnovsek (Head of professional services and presales)
3.2.5	E-mail of technical contact	Marko.o.pyy@tieto.com Samo.drnovsek@marand.com
3.2.6.	Phone number of technical contact	Tieto: +358 (40) 7623454 Marand: +385 51 420294

3.3. Partner		Answer
3.3.1	Does the company have any sales partners in Sweden? (Y (names)/N)	Yes, Tieto and Marand has signed a partnership agreement covering sales, distribution and services for the Swedish market.

4. Product information

4.1. General		Answer
4.1.1	Name of the product?	Think!EHR Platform
4.1.2.	Current version of product	4.4
4.1.3.	Number / size of installations	<p>Think!EHR Platform is deployed in different implementation scenarios where below are mentioned only the relevant ones:</p> <ul style="list-style-type: none"> • City of Moscow where Think!EHR Platform is used as Common data platform for all clinical applications for the scope of: 428 polyclinic institutions, 76 healthcare centers, 12 million patients, 161 million visits/year, 130.000 system users, 45.000 physicians, 1B documents, 25 TB of data. • Slovenian Interoperability backbone as clinical data repository for all eHealth solutions including, Patient summary, eReferral, ePrescription and health registries and covering: 30 hospitals, 90 healthcare centers, 1.800 GPs and specialists, 12.000 physicians, 9.500 hospital beds, 2.1 million patients, 15 million visits/year. • Think!Clinical - Electronic Health Record solution to manage clinical processes and data at CHILDREN'S HOSPITAL, University Medical Center, Ljubljana (EMRAM stage 6), Slovenia covering 500 employees, 220 beds, 85.000 outpatient visits. • Centralised and structured vendor-neutral data repository for primary care and hospital EHR for AYALA HEALTH Philippines Primary Care and Hospital Provider and covering 10 hospitals, 200 healthcare clinics, 5.000 hospital beds. • Inpeco (Turin, Italy) building Open Platform based Personal Health Record.

4.1. General	Answer
	<ul style="list-style-type: none"> • Medtronic offering Patient Engagement solution as a digitally enabled multidisciplinary Perioperative solution. • Salford Royal Foundation Trust offering Platform to develop innovative apps and services for Innovative App Ecosystem. • Plymouth NHS trust using openEP (Electronic eprescribing solution) on top of Think!EHR Platform. • Genomics England for Centralised and structured vendor-neutral Phenotype data repository (Sequence 100,000 genomes, Model and collect 278 data points for cancer, Model and collect 1,000 data points for rare diseases 13 regions, 6 NHS Trusts: Great Ormond Street, UCLH, The Royal Free London, Barts Health, Moorfields Eye Hospital, London North West Healthcare. • Eurotransplant International Foundation; Clinical and analytical infrastructure providing a centralised, vendor-neutral EHR operational 27/7/365, About 2,500 users, covering 134.6 million inhabitants, 25,000 registered donors, 14,560 patients on the active organ waiting list, 10,808 new registrations per year, 7,145 organ transplants per year. • Others: Diagnostiek Voor U, NL (Lab services provider), Unimed, BR (Insurance company), PasientSky, NO (Primary care EHR provider), Institute of Oncology, SLO (Cancer registry, Cervical cancer screening program, Melanoma registry).
4.1.4. Describe the product update strategy (ex. number of major/minor update/year)	<p>Software Update License & Support consists of:</p> <ul style="list-style-type: none"> • 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 our discretion, which may include general maintenance releases,

4.1. General	Answer
	selected functionality releases and documentation updates.

4.2. Support	Answer
<p>4.2.1 Availability of support? (24/7, 8/5 or other)</p>	<p>Think!EHR Platform standard Software Technical Support Policies apply to technical support for Think!EHR Platform product. Technical support is provided for issues (including problems you identify) that are demonstrable in the currently supported release(s) of the Think!EHR Platform licensed program, running unaltered, and on a certified hardware, database and operating system configuration, as specified in Client's order or program documentation.</p> <p>Software Support consists of:</p> <ul style="list-style-type: none"> • Assistance with service requests 24 hours per day, 7 days a week. Access to the customer support systems specified in the Web-Based Customer Support Systems section below (24 x 7 web-based customer support systems), including the ability to log service requests online, unless stated otherwise. • An online knowledge base of information and online FAQ that supplements the Documentation and provides up-to-date information on the Software. <p>Think!EHR Platform support services also provides the provision of guidance and troubleshooting to Client in connection with questions and issues arising from the following Client activities with respect to the Software:</p> <ul style="list-style-type: none"> • Installation and Downloads: Support for installation includes providing guidance and troubleshooting in connection with Client's downloading and installing of the Software. • Basic Configuration Issues: Support for configuration includes troubleshooting of Client's configuration settings for existing installations on Supported Platforms (as defined below) to ensure proper operation and connectivity. • Usage Issues: Think!EHR Platform qualified personnel will answer Client's "how to"

4.2. Support	Answer
	<p>questions related to standard and intended Software usage.</p> <ul style="list-style-type: none"> • New Version Issue: Support for issues regarding replacing a Previous Version with a New Version of the Software. • Efforts to Correct the Software: we shall make commercially reasonable efforts to correct bugs or other errors in the Software. Client acknowledges that we are not required to correct every bug, error, or problem with the Software that it reports to us or of which we are otherwise made aware. <p>We will provide Standard 2nd and 3rd Level technical support services to the Client. (For avoidance of doubt): First Line Support must be provided by Client or 3rd party by Client choice. Standard support term and conditions includes availability depending on the level of severity:</p> <ul style="list-style-type: none"> • High severity bug or issue (Severity 1): Availability 24/7 • Medium severity bug or issue (Severity 2): Availability 8/5 (Business hours) • Low severity bug or issue (Severity 3): Availability 8/5 (Business hours) • Procedural issue or question (Severity 4): Availability 8/5 (Business hours)
4.2.2. Availability of on-site installation support? (Free or billed)	<p>As part of standard support, we offer:</p> <ul style="list-style-type: none"> • Installation and Downloads: Support for installation includes providing guidance and troubleshooting in connection with Client's downloading and installing of the Software. • Basic Configuration Issues: Support for configuration includes troubleshooting Client's configuration settings for existing installations on Supported Platforms (as defined below) to ensure proper operation and connectivity. <p>On-site installation support is part of Professional services offered jointly by partners.-</p>
4.2.3. Availability of Health (best practice) checks?	<p>As part of the Professional services we also offer Health checks of installed solution based on best practices and other deployments. Health check is primarily focused on connections to underlying DB, Think!EHR Platform configuration, Clustered</p>

4.2. Support		Answer
		deployment, Audit configuration and underlying infrastructure (if provided based on the requirements).
4.3. Licensing		Answer
4.3.1	Describe the license model for the product (CPU, user, other)	<p>Think!EHR Platform License entitles customer (end user) with the non-exclusive, non-transferable license rights to use the software, and not the software itself. License model is based on EHR ID. EHR ID refers to the identification of an instance of patient record in the Think!EHR Platform. It serves as the unique identifier for the patient in the system.</p> <p>Think!EHR Platform license is available either as a Full Use license which can be used for unlimited number of applications, or Application Specific License (ASL), in which case the installation is allowed to run single application. Furthermore, one ASL license applies for one implementation instance of Think!EHR Platform, only. Application shall mean content rounded unit of Software that has single installation package which is provided "as is" (single installation package excludes bug fixes and patches).</p> <p>Both type of license environments (that is, Full use or Application specific license), are available as:</p> <ul style="list-style-type: none"> - Perpetual license with which we grant customer (end user) a permanent, non-exclusive, non-transferable license to use Think!EHR Platform. - Subscription license with which we grant customer (end user) a non-exclusive, non-transferable subscription based license to use Think!EHR Platform for the time limited period.
4.3.2.	Does the license model have options for setting up development and QA-environments (not for real patient care) that differs from production environment licenses?	Development and Testing: For the purpose of development and testing, your license for the Think!EHR Platform includes the right to run the software on an unlicensed computer.
4.3.3.	Describe support agreement alternatives for the product	Think!EHR Platform maintenance and support terms define standard support. For Think!EHR

4.3. Licensing		Answer
		Platform we also provides developers support which limited to developers questions only, not providing support for Severity 1 issues, responding only during working hours and other terms of Standard support. We also provide mission critical support with specific terms and conditions agreed between all parties.

4.4. Procurement & pricing		Answer
4.4.1	Is the product offered through Swedish public sector framework agreements ("Kammarollegiet" procurement contract) (E.g. via an existing Swedish partner)	Tieto is one of the preferred vendors of Kammarkollegiet. Tieto has currently several valid agreements with Kammarkollegiet e.g. Informationsförsörjning.
4.4.2.	If possible, please provide approximate price examples for some scenarios. Are there alternative price models regarding initial and recurring costs?	For case described in the RFI documentation we would offer customer (end user) a Perpetual Think!EHR Platform Full Use License. With this we would grant customer (end user) permanent, non-exclusive, non-transferable license to use Think!EHR Platform with the following provisions: <ul style="list-style-type: none"> - License can be used for unlimited number of applications - License is limited to the number of unique EHR IDs which is the number of unique patients for which data is stored. - Allows for clustered deployment as well as disaster recovery provided the number of unique EHRs does not exceed the licensed volume - License does not include the rights to use software to host applications for other entities. - Standard Maintenance and support which is paid annually.
4.4.3.	How does your business model provide compensation if promised functions (e.g. like described in 5.1.7) would be specified in a contract but would not available in time as promised?	Tieto and Marand always does it utmost to deliver on commitments. In an unlikely event that we will not be able to deliver agreed functionality due to internal or external reasons we will inform the customer and honour the contract. Actions will be taken to deliver the functionality as soon as possible.

5. Functional requirements

5.1 Basic framework	Answer
<p>5.1.1 What parts of the the openEHR Reference Model Specification are fully implemented, and according what version of the specification?</p>	<p>Think!EHR Platform out of the box support following openEHR Reference model categories:</p> <ul style="list-style-type: none"> • EHR (RELEASE 1.0.3) • Common (RM RELEASE 1.0.4) • Data Structures (RELEASE 1.0.3) • Data Types (RELEASE 1.0.3) <p>Other RM specifications which are not fully supported with Think!EHR Platform are not available as openEHR compliant implementation but are implemented with alternative approach:</p> <ul style="list-style-type: none"> • Demographic; One of the basic principles of openEHR is the 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. Following the openEHR standard, Think!EHR Server does not store patient demographic information (such as name, address, etc). Instead, it offers 2 ways of working with patients: <ul style="list-style-type: none"> ○ Store patient id from an external demographic server into the EHR subject id field (Think!EHRService.createSubjectEhr). ○ Store EHR id into the external demographic server (We mean demographic server as something that can find patients, and can easily just be a part of the client application storing patient demographic data into a relational database or any other type of persistent storage). • Support; Think!EHR Server supports validation terminology codes for imported compositions. Local terminologies (defined inside the template) will be validated internally. Validation of external terminologies, however, requires connecting to Terminology Adapter which

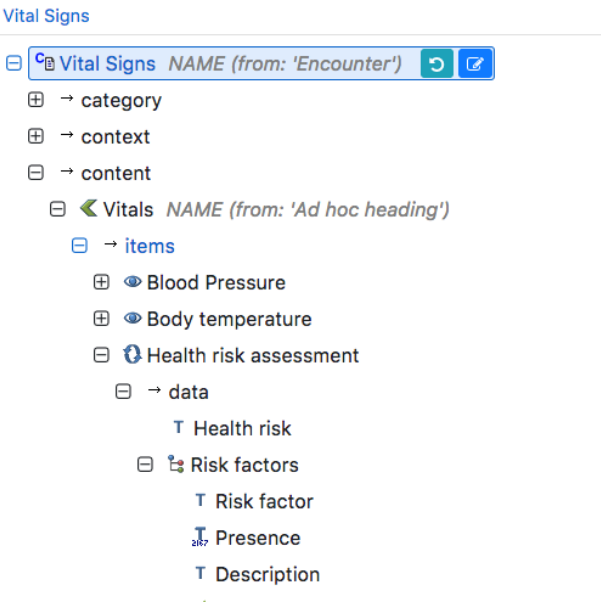
5.1 Basic framework	Answer
	<p>is part of the Think!EHR Platform. Terminology Adapter provides an admin REST API allowing uploading new CSV terminologies via the REST API as well as deleting existing ones.</p> <ul style="list-style-type: none"> • Integration: Think!EHR Platform exposes a clinical data repository API to store, manage, query, retrieve and exchange structured electronic health record data. Integration with systems and components that are not capable of interaction with openEHR based Clinical Data Repository is supported via; <ul style="list-style-type: none"> ○ Importing data from source as TDDs, i.e. XML conforming to the TDS XML Schema (Data Integration Intermediate Form) ○ Integration engine providing Adapter (content-based-transform) for message mapping and transformation to openEHR composition ○ Adapter Accelerator framework providing support for: <ul style="list-style-type: none"> ○ IHE MPI (EhrID PatientID Feed to PIX), ○ XDS.Repository (document based transformation processing), ○ XDS.OnDemand document source (generated documents) ○ Think!EHR Platform provides Adapter Accelerators as basic framework for System Integrators to address specific integration scenarios.
<p>5.1.2. What parts of the the openEHR REST API Specification are fully implemented? What formats (e.g. JSON and XML) are supported? Are any other (non standard) REST APIs implemented?</p>	<p>Think!EHR Platform fully implements following REST API Specification:</p> <ul style="list-style-type: none"> - EHR - Query - Definitions <p>REST API provides the ability to create, modify and query EHRs. Think!EHR Platform allows EHR applications to perform the following functions:</p>

5.1 Basic framework	Answer
	<ul style="list-style-type: none"> • Think!EHRService; provides base functionality for managing openEHR data • Think!EHRExtService; provides extended services such as physical delete of compositions or EHRs • TemplateService; allows manipulation and administration of openEHR templates • TemplateSchemaService and TemplateDocumentConversionService; services enable users to create specific Template Data Schemas (TDSs) and submit Template Data Documents (TDDs) as well as convert canonical openEHR compositions to TDDs • DomainService; provides management of domains (for multi-tenancy) • UserService; provides user management functionality • EventService; allows managing event triggering on EHR data submit • FormService; allows managing forms created by EhrExplorer form designer tool • QueryService; advances querying API. Used for asynchronous/stream queries. Such queries can be run for very large result sets as neither client nor server need to store complete result set in memory. • RegisteredQueryService; management of registered queries • CompositionBuilderService; provides support for composition builder on the server. <p>List of REST APIs provided by the platform can be checked and tested at: https://dev.ehrscape.com/api-explorer.html</p> <p>In Think!EHR Platform RESTful web service, requests made to a resource's URI will elicit a response that may be in JSON or XML formats. Think!EHR framework also provides a generator which can generate Web Template JSON Definition, a platform independent, structured representation of a specific template in JSON</p>

5.1 Basic framework	Answer
	<p>format. JSON template was designed to make it easier to build user interfaces.</p> <p>Think!EHR Platform provides a template specific XML Schema generator. Client applications can then use this schema to create a template specific xml document (TDD) and submit it to Think!EHR Platform. 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.</p> <p>TemplateSchemaService provides API for generating TDS for a specific template. Resulting TDS references openEHR Structure and Base XML Schemas. For more information see openEHR XML Schemas (openEHR XML-schemas - Release 1.0.2). Structure and Base XML Schemas are also available via the TemplateSchemaService API.</p> <p>In addition Think!EHR Platform also provides a set of REST API enabling server administration using Admin APIs.</p>
<p>5.1.3 Is the openEHR Archetype Query Language specification (at least version 1.0, Trial Draft) fully implemented?</p> <p>Are there any additional capabilities, e.g. full text search, FOLDER-based filtering etc?</p>	<p>EHRs stored in Think!EHR Platform are queried with portable openEHR AQL queries written in the archetype-enabled query language (AQL).</p> <p>Think!EHR Platform supports latest Stable release of openEHR AQL specification (RELEASE 1.0.2)</p> <p>The Think!EHR Platform AQL processor can query a single EHR for point of care purposes, or the entire EHR population in the server for health analytics, population health, research or epidemiological studies. A query can return whole Compositions or fine-grained health data items, ensuring efficient use of computing resources on both client and server.</p> <p>Think!EHRService provides method query for executing AQLs. This method can be used to execute single patient queries as well as population queries.</p> <p>In. addition to openEHR compliant AQL queries, Think!EHR Platform provides additional query features/capabilities such as:</p> <ul style="list-style-type: none"> Using select aliases; For elements in SELECT part of the AQL it is possible to use aliases: SELECT c/name/value AS composition_name. These aliases are useful for several purposes:

5.1 Basic framework	Answer
	<ul style="list-style-type: none"> ○ to make AQLs clearer (it is sometimes hard to know which information each column contains) ○ they are available via extended querying API ○ they can be used in ORDER BY portion of the AQL • Limiting result sets, which is often useful to use paging when retrieving query results. This is supported through keywords: <ul style="list-style-type: none"> ○ OFFSET: sets the index of the first row that will be retrieved ○ FETCH: sets the number of rows to retrieve. Also there are 2 alternatives to FETCH such as LIMIT or TOP • Sorting result sets using ORDER BY statement. • Queries with LIKE which allows matching with wildcards: * and ?. • Queries with TAGGED BY which will match all compositions with tag key key and tag value value anywhere in the composition. Further it can support different options like: <ul style="list-style-type: none"> ○ Server function 'tags' returns a list of all tags for compositions (tag key, tag value and AQL path describing which part of composition is tagged), referenced by the AQL path in the given parameter. ○ Server function 'current_state' which returns the current state of an instruction activity as per The Standard Instruction State Machine of the OpenEHR standard. Possible values are: INITIAL, PLANNED, SCHEDULED, ACTIVE, POSTPONED, CANCELLED, SUSPENDED, ABORTED, COMPLETED, and EXPIRED ○ Server function 'instruction_aggregate_state' which

5.1 Basic framework	Answer
	<p>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. Possible values are: PRE-ACTIVE, ACTIVE, INACTIVE, and TERMINATED.</p> <ul style="list-style-type: none"> • Aggregate expressions where several aggregate column expressions are supported: AVG, MIN/MAX, COUNT, COUNT DISTINCT, • Join AQLs supporting the need to explore data from several different compositions in the same result row. For example: retrieval of patient’s allergies as well as medications in a single query even though they are stored in different compositions. This type of query can be created by having EHR at the top of the containment and having two containment branches joined by a logical AND. • Special AQL functions enabling querying of versioned compositions <ul style="list-style-type: none"> ○ Find an EHR by external subject id ○ Find version and commit data ○ Find version and commit data for all versions and not only the active one <p>Think!EHR Platform also enables configurable full text search capabilities based on indexing capabilities.</p>
5.1.4	<p>How is validation of EHR content done based on RM, archetypes and templates by the system? What types and versions of template-mechanisms are used for validation?</p> <p>OpenEHR document is generally referred to as an openEHR canonical Composition. Such documents are generic structures composed out of objects from the openEHR Reference Model (RM). Think!EHR uses reference model based on schema files. Even though Compositions are generic Think!EHR Platform validates them against an openEHR template before they are committed to storage. Full validation of compositions is conducted before they are stored in the EHR - so any required fields must be present and within their validation limits, or the call will not succeed. This is not a requirement for draft compositions</p>

5.1 Basic framework	Answer
	<p>when they are stored as draft - they can be invalid. An error is generated if this validation fails and composition is rejected by the server.</p> <p>For example, a template that specifies an openEHR Composition contains a Section and this section contains Observation of a certain type (certain archetype). On the image below, template specifies that Composition should be named 'Vital Signs' and should contain a section named 'Vitals'. This section can further contain Observations (Blood Pressure and Body temperature) and an Evaluation (Health risk assessment) and so on.</p>  <p>Think!EHR Platform also provides Terminology validation which allows validation of external terminologies against the terminology adapter on composition writes.</p>
5.1.5	<p>Is GDL (Guideline Definition Language) (at least version 1.0, TRIAL DRAFT) supported? Are any other clinical decision support mechanisms available?</p> <p>Think!EHR Platform also supports GDL version 1.0 TRIAL (http://openehr.org/releases/CDS/latest/docs/GDL/GDL.html#latest_issue) as a lightweight CDS service to support execution of clinical guidelines. Use of this service can be checked and tested at: https://dev.ehrscape.com/api-explorer.html</p>
5.1.6	<p>What parts of the new "Task Planning Model Specification" are implemented?</p> <p>At the time of answering to RFI we are in the phase of implementation of openEHR Task planning model specification. It will be a core part of Think!EHR Platform process server. First prototype version is expected to be available in by end of Q2 2018.</p>

5.1 Basic framework	Answer
<p>5.1.7 What parts in the 5.1.x questions above that are not implemented right now will be available in September 2018?</p>	<p>Think!EHR Platform implements parts of openEHR specifications as define above. The functionalities covered by openEHR RM that are not implemented are supported with alternative approach (defined in question 5.1.1). Further development of Think!EHR Platform is defined via the roadmap and key features expected to be delivered by the end of 2018 are:</p> <ul style="list-style-type: none"> • IoT BLE integration • Federation & sharking • Attribute access control • Tenant Encryption • App Builder tools • Multiple Storage Container • Data Quality Analytics • SNOMED CT + AQL • Reactive streams <p>As for Think!EHR Process server which will implement openEHR Task Planning Model Specification will provide first production version of the Think!EHR Process by Q1 2019. Production version of Think!EHR Process server will then provide all missing parts of the specification. Also it will include a toolset supporting the use of Task planning specification definition, query, maintenance and other features.</p>
<p>5.1.8 Describe available terminology service usage/integrations. Is the terminology service addressable from AQL queries? Is there a FHIR Terminology Service interface?</p>	<p>Think!EHR Platform supports validation of terminology codes for imported compositions. Local terminologies (defined inside the template) will be validated internally. Validation of external terminologies, however, requires connecting to Terminology Adapter which is part of the Think!EHR Platform. Terminology Adapter provides an admin REST API allowing uploading new CSV terminologies via the REST API as well as deleting existing ones.</p> <p>Existing version of Think!EHR Platform does not support terminology based queries. This feature is planned in our roadmap, and will be made available by end of Q3 2018.</p>

5.1 Basic framework	Answer
	Think!EHR Platform does not provide FHIR Terminology service interface.

5.2 Tests & performance	Answer:
-------------------------	---------

5.2.1	<p>Please provide information and results from AQL query performance tests done for the product. (Have you for example run any of the "ORBDA" example tests?)</p>	<p>Think!EHR Platform has passed several performance testing. Stress testing is based on following workload:</p> <ul style="list-style-type: none"> • Replay of 3 months of EMR traffic from Marand's own production environment: <ul style="list-style-type: none"> ○ 40 million executed queries ○ 1 million new compositions created • Simulated HIE traffic based on patient-summary template • 2 million new patient-summary compositions created using the REST API <p>Stress test on 2 nodes cluster provided following results:</p> <table border="1"> <thead> <tr> <th rowspan="2">VCPUs / node</th> <th rowspan="2">OS RAM / node</th> <th rowspan="2">Think!EHR RAM/node</th> <th colspan="2">Query</th> <th colspan="2">Write</th> </tr> <tr> <th>TPS</th> <th>Times (Avg/95p)</th> <th>TPS</th> <th>Times (Avg/95p)</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>4GB</td> <td>2GB</td> <td>202</td> <td>20ms/55ms</td> <td>5</td> <td>69ms/172ms</td> </tr> <tr> <td>4</td> <td>8GB</td> <td>4GB</td> <td>724</td> <td>15ms/42ms</td> <td>17</td> <td>62ms/127ms</td> </tr> <tr> <td>4</td> <td>16GB</td> <td>8GB</td> <td>758</td> <td>14ms/38ms</td> <td>18</td> <td>59ms/107ms</td> </tr> <tr> <td>8</td> <td>16GB</td> <td>8GB</td> <td>1112</td> <td>11ms/31ms</td> <td>27</td> <td>40ms/69ms</td> </tr> </tbody> </table> <p>Adding a 3rd node increases overall performance by about 30%:</p> <table border="1"> <thead> <tr> <th rowspan="2">VCPUs / Node</th> <th rowspan="2">OS RAM / node</th> <th rowspan="2">Think!EHR RAM/node</th> <th colspan="2">Query</th> <th colspan="2">Write</th> </tr> <tr> <th>TPS</th> <th>Times (Avg/95p)</th> <th>TPS</th> <th>Times (Avg/95p)</th> </tr> </thead> <tbody> <tr> <td>8</td> <td>16GB</td> <td>8GB</td> <td>1501</td> <td>12ms/37ms</td> <td>36</td> <td>46ms/91ms</td> </tr> </tbody> </table>	VCPUs / node	OS RAM / node	Think!EHR RAM/node	Query		Write		TPS	Times (Avg/95p)	TPS	Times (Avg/95p)	2	4GB	2GB	202	20ms/55ms	5	69ms/172ms	4	8GB	4GB	724	15ms/42ms	17	62ms/127ms	4	16GB	8GB	758	14ms/38ms	18	59ms/107ms	8	16GB	8GB	1112	11ms/31ms	27	40ms/69ms	VCPUs / Node	OS RAM / node	Think!EHR RAM/node	Query		Write		TPS	Times (Avg/95p)	TPS	Times (Avg/95p)	8	16GB	8GB	1501	12ms/37ms	36	46ms/91ms
VCPUs / node	OS RAM / node	Think!EHR RAM/node				Query		Write																																																			
			TPS	Times (Avg/95p)	TPS	Times (Avg/95p)																																																					
2	4GB	2GB	202	20ms/55ms	5	69ms/172ms																																																					
4	8GB	4GB	724	15ms/42ms	17	62ms/127ms																																																					
4	16GB	8GB	758	14ms/38ms	18	59ms/107ms																																																					
8	16GB	8GB	1112	11ms/31ms	27	40ms/69ms																																																					
VCPUs / Node	OS RAM / node	Think!EHR RAM/node	Query		Write																																																						
			TPS	Times (Avg/95p)	TPS	Times (Avg/95p)																																																					
8	16GB	8GB	1501	12ms/37ms	36	46ms/91ms																																																					

5.2.2	<p>Please provide information regarding other performance tests done or normal loads in significant real installations.</p>	<p>We perform continuous performance tests on regular sized installations (50million compositions) as well as twice yearly large size installations (1billion compositions). Last large Scale Think!EHR Performance Testing was carried out on the Oracle Bare Metal cloud infrastructure (https://cloud.oracle.com/en_US/bare-metal) between 3-Feb-2017 and 12-Feb-2017.</p> <p>A. PostgreSQL v9.6</p> <p><u>Setup</u></p> <ul style="list-style-type: none"> - 3 x application servers, one of them also running database - PostgreSQL v9.6 - 1 x load generating server
-------	---	---

5.2 Tests & performance	Answer:
	<p><u>Machine specifications:</u></p> <ul style="list-style-type: none"> - 512GB RAM - 36 Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz CPU cores - 10Gb network interfaces <p><u>App servers:</u></p> <ul style="list-style-type: none"> - 2 of them with 4x2.9TB NVME storage modules organized in software RAID (for a maximum capacity of 11TB) - 1 of them with 9x2.9TB NVME storage modules organized in software RAID (for a maximum capacity of 26TB) <p><u>Initial dataset size:</u> We used Marand's own production dataset and enlarged it to:</p> <ul style="list-style-type: none"> - 50 million EHRs - 2.1 billion compositions <p><u>Data size on disk:</u></p> <ul style="list-style-type: none"> - 5.7TB database size - 2.08TB index size (4.16TB with 1 replica) split across 3 app servers <p><u>Think!EHR Configuration</u></p> <ul style="list-style-type: none"> - server Java memory (-Xmx): 128GB - number of replicas: 1 - clustering with unicast <p><u>Workload</u></p> <ol style="list-style-type: none"> 1. Replay of 3 months of EMR traffic from Marand's own production environment: <ul style="list-style-type: none"> o 40 million executed queries o 1 million new compositions created 2. Simulated HIE traffic based on patient-summary template <ul style="list-style-type: none"> o 2 million new patient-summary compositions created using the REST API <p>Note – both sets of traffic were running against the Think!EHR cluster simultaneously.</p>

5.2 Tests & performance	Answer:
	<p><u>Results</u></p> <ul style="list-style-type: none"> - 3 months of production traffic replayed in 4.5 hours - 2 million patient summary compositions created in 3.5 hours - EMR Call speeds: <ul style="list-style-type: none"> o 95th percentile of read calls is at 106msec o 95th percentile of write calls is at 374msec - HIE call speeds: <ul style="list-style-type: none"> o 95th percentile at 277msec <p>B. OracleDB v12.1</p> <p><u>Setup</u></p> <ul style="list-style-type: none"> - 1 x database server running OracleDB v12.1 - 3 x application servers - 1 x load generating server <p><u>Machine specifications:</u></p> <ul style="list-style-type: none"> - 512GB RAM - 36 Intel(R) Xeon(R) CPU E5-2699 v3 @ 2.30GHz CPU cores - 10Gb network interfaces <p><u>App servers:</u> each have 4x2.9TB NVME storage modules organized in software RAID (for a maximum capacity of 11TB)</p> <p><u>Oracle DB server</u> has 8x2.9TB NVME storage modules in RAID with three-way mirroring (for a maximum capacity of 7TB)</p> <p><u>Initial dataset size:</u> We used Marand's own production dataset and enlarged it to:</p> <ul style="list-style-type: none"> - 20 million EHRs - 800 million compositions <p>Data size on disk:</p> <ul style="list-style-type: none"> - 5.4TB database size

5.2 Tests & performance	Answer:
	<ul style="list-style-type: none"> - 850GB index size (1.7TB with 1 replica) split across 3 app servers <p><u>Think!EHR Configuration</u></p> <ul style="list-style-type: none"> - server Java memory (-Xmx): 128GB - number of replicas: 1 - clustering with unicast <p><u>Workload</u></p> <ol style="list-style-type: none"> 3. Replay of 3 months of EMR traffic from Marand's own production environment: <ul style="list-style-type: none"> o 40 million executed queries o 1 million new compositions created 4. Simulated HIE traffic based on patient-summary template <ul style="list-style-type: none"> o 2 million new patient-summary compositions created using the REST API <p>Note – both sets of traffic were running against the Think!EHR cluster simultaneously.</p> <p><u>Results</u></p> <ul style="list-style-type: none"> - 3 months of production traffic replayed in 7.5 hours - 2 million patient summary compositions created in 8 hours - EMR Call speeds: <ul style="list-style-type: none"> o 95th percentile of read calls is at 25msec o 95th percentile of write calls is at 90msec - HIE call speeds: <ul style="list-style-type: none"> o 95th percentile at 417msec
5.2.3	<p>The test cases/scripts in chapter 6 ("Conformance Schedule") of the "openEHR EHR Platform Conformance" documents are not finished, but when looking at the list of test descriptions, are there any of the listed capabilities your</p> <p>Think!EHR Platform is health data platform based on clinical data repository providing functionalities to store, manage, query, retrieve and exchange structured electronic health record data based on the latest release of openEHR specifications. As such it is conformant with For other components Think!EHR Platform provides capabilities which are listed in Conformance schedule:</p>

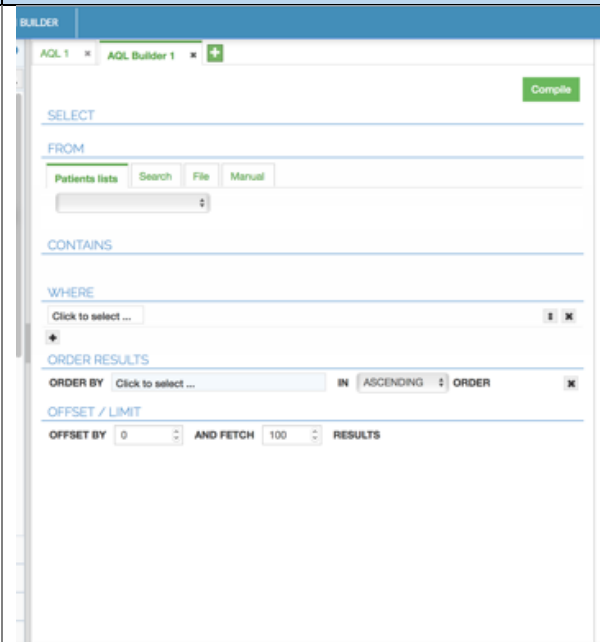
5.2 Tests & performance	Answer:
<p>system has not yet implemented in some api-accessible form?</p>	<ul style="list-style-type: none"> - Definitions Component <ul style="list-style-type: none"> o ADL 1.4 Archetype provisioning o ADL 1.4 OPT provisioning o ADL 2 Archetype provisioning o ADL 2 OPT provisioning o Query provisioning - EHR Persistence Component <ul style="list-style-type: none"> o EHR Operations o EHR Status o Composition Operations o Directory Operations o Change sets o Versioning o Archetype Validation - Querying Component <ul style="list-style-type: none"> o AQL Basic o AQL Advanced o AQL & Terminology - Admin Product Component <ul style="list-style-type: none"> o Activity Report o Physical Deletion o EHR Dump/Load o EHR Archive o Demographic Archive - Messaging Component <ul style="list-style-type: none"> o TDD - REST API Component <ul style="list-style-type: none"> o DEFINITION API o EHR API o QUERY API o ADMIN API o MESSAGE AP - Security and Privacy <ul style="list-style-type: none"> o Signing <p>From the perspective of openEHR EHR Platform conformance, Think!EHR Platform is not conformant with following capabilities since it does not implement specific parts of openEHR RM:</p>

5.2 Tests & performance		Answer:
		<ul style="list-style-type: none"> - Demographic Persistence since the Platform does not implement openEHR Demographic IM - EHR Extract capability as part of Messaging integration components.

5.3 Tooling & configuration		Answer
5.3.1	Does the product contain an application development environment that enables applications, registries etc. to be built on the repository using openEHR data. Please describe.	<p>Yes, Think!EHR Platform contains the application development framework that allows for applications and registries to be built on top of openEHR data. It does so by providing the facility for low code, fast and agile environment where simple applications can be built in matter of hours, by end users as well.</p> <p>Registries should collect standardized data sets and the also should be highly flexible so researchers and clinicians can capture, view, analyze, and report on the specific information that they want quickly, all without requiring their IT department's assistance. Think!EHR Platform supports scenario, where in-house staff can go from idea to working code in a matter of hours. This has been proven as a framework for building clinical registries where clinicians and analytical team are able have been able to build the web forms for aggregation of patient data for their separate research projects.</p> <ol style="list-style-type: none"> 1. Model; The production of high-quality, maintainable knowledge artefacts relies on Think!EHR Platform Archetype designer, a comprehensive tool support, allowing users to concentrate on defining the clinical models, semantics, knowing that lifecycle management, versioning, review processes and publishing are taken care of by the system. 2. Design; Use a form builder with user-friendly drag-and-drop functionality to design data entry forms (including validation) from the template. <ul style="list-style-type: none"> • Define dependencies between different attributes as part of the template.

5.3 Tooling & configuration		Answer
		<ul style="list-style-type: none"> • Connect selected attributes to specific terminologies available in Terminology server. • Add clinical decision support rules to evaluate data on insert to trigger alarms. <p>3. Deploy: Developed forms, can be easily now published to the server making it immediately available to applications, with a simple copy-and-paste code snippet to help you embed your form into your application. No matter where you put your form, the form results are collected and stored automatically.</p> <p>4. Query; Empowering users to build queries using simple drag-and-drop features. Query builder feature allows researchers to create and run complex, multi-dimensional queries—without any need for assistance from IT resources or any knowledge of database design or how to construct query statements.</p>
5.3.2	Is there a graphical drag and drop form generator (or similar functionality) available that makes it easy to create HTML5-based data entry forms (including client side validation and basic constraint checking) based on openEHR templates.	Think!EHR Platform provides a Form Builder tool which enables users to quickly build a well-designed, mobile-friendly and accessible web form with validation and semantics in place by simply using "drag-and-drop" and add it to a page. The data via this form are then saved to Think!EHR Server as openEHR based composition.
5.3.3	Is there a function to render compositions as human-readable documents (resolving at/id-codes and hiding "technical" attributes)	Think!EHR Platform also includes Think!EHR Angular2+ HTML Form Renderer runs in modern browsers. Form renderer works with forms created in the EHR Explorer. It renders interactive forms to client devices, typically web browsers, to collect information from users. After an interactive form is rendered, a user can enter data into form fields and click a submit button located on the form to send information back to the Forms service. Consider form renderer, where numerous items of data of several different types may need to be prepared for display in a form, rendered as HTML, edited using a convenient interface, returned to the server, validated and cleaned up, and then saved or passed on for further processing. Renderers

5.3 Tooling & configuration	Answer
	<p>functionality can simplify and automate vast portions of this work, and can also do it more according to EHR specifications than most programmers would be able to do in code they wrote themselves. Form renderer has three distinct parts of the work involved in forms:</p> <ul style="list-style-type: none"> • preparing and restructuring data to make it ready for rendering • creating HTML forms for the data • receiving and processing submitted forms and data from the client <p>Besides rendering the forms, we support plugin-like widgets that could render their own content and provide the interface to various devices linked to the phone and thus provide an open system for data collection. iOS Form Renderer acts as an adapter for iOS devices, with iPhone being the primary device, to be able to visualize the form, return the collected data, which may or may not be validated and populate the form. It also offers a way to introduce messages from the outside, bind them to specific fields or display them as general annotations.</p>
5.3.4	<p>Is an easy to use (e.g. drag-and-drop?) query editor available to create AQL queries based on Archetypes and Templates?</p> <p>Think!EHR Platform provides a generic component (EHRviewer) enabling viewing of any EHR data held in a clinical data repository with a normal web browser. All openEHR EHR information can be viewed in a generic fashion, requiring no programming. Think!EHR Platform empowers users to build queries using simple drag-and-drop features. Query builder feature allows users to create and run complex, multi-dimensional queries—without any need for assistance from IT resources or any knowledge of database design or how to construct query statements.</p>



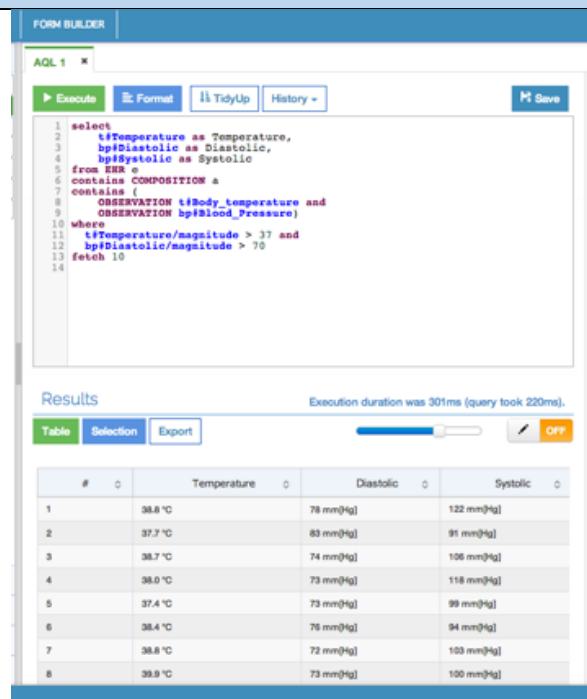
Presentation specific to content can be implemented via javascript renderer, allowing developers to customise display according to clinical requirements.

A view of all Compositions, including previous versions, in a given clinical subject's EHR.

- Filtering of Compositions in an EHR by date, user, topic, archetype, etc.
- Viewing information in different contexts such as information about a diagnosis in a summary list or showing all related information.
- Ability to display the results of queries.

5.3 Tooling & configuration

Answer



EhrExplorer greatly simplifies the display of information from a subject's EHR and allows control of presentation due to a strong grip on the meaning of the information in the EHR. This is done by linking display logic to particular archetypes and particular user categories and means that information can be presented in a manner most appropriate to the current user. It is possible to record the viewing of EHR data for audit purposes and to ensure accountability to patients and maintenance of their privacy.

5.3.5

Are functions like domains or namespaces available to achieve a logical separation of data between different care organisations using a physically shared server instance?

Think!EHR Platform provides ability to run multiple domains on single installation. A domain can be used for a specific application, specialities, care organisations. Rationale for providing multi-tenant deployment options are:

- Cost savings since multitenancy allows for cost savings over and above the basic economies of scale achievable from consolidating IT resources into a single operation.

Release management since multitenancy simplifies the release management process. With the multitenant model, the package typically only needs to be installed on a single server. This greatly simplifies the release management process, and

5.3 Tooling & configuration	Answer
	the scale is no longer dependent on the number of customers.

6 Non-functional requirements

6.1 Infrastructure		Answer:
6.1.1	List supported OS	<p>Think!EHR Server is a software-only solution that runs on most modern operating systems which are supported by the Java JDK/JRE. We test and develop Think!EHR on the following:</p> <ul style="list-style-type: none"> ○ Windows (64-bit is recommended) ○ 64-bit Red Hat Enterprise Linux 6.3 and higher ○ 64-bit Oracle Linux 6 and higher ○ 64-bit Ubuntu Server 12.04 and higher <p>The above operating systems are supported natively and also when running in a virtual environment such as VMWare.</p>
6.1.3	List supported DBMS	<p>Think!EHR Server supports the following databases servers:</p> <ul style="list-style-type: none"> • Oracle Database 11g or 12c • PostgreSQL 9.0 or later • SQL Server 2008R2 or later • MySQL 5.7 or above (or compatible MariaDB, PerconaDB release) with InnoDB Barracuda file format • Azure SQL Server
6.1.4	Support of management packs for Microsoft System Center	<p>For Think!EHR Platform monitoring It is very common to use tools such as Nagios or Zabbix or others that can monitor the server health. Think!EHR supports such tools by allowing access to heartbeat URLs. Each of these URLs can be called (no authentication is required) and will respond with a status of 200 if everything is OK. When there is no response or response is not 200 then there is a problem and a manual intervention is required. Cluster monitoring is a bit special as it will respond with an error once the number of active cluster nodes drops to the configured number of index replicas or less. So if the index is configured to 2 replicas and the number of cluster</p>

6.1 Infrastructure		Answer:
		nodes drops to only 2 nodes then cluster state will report an error.
6.1.5	Describe minimum hardware requirements for a test installation	The minimum requirements for a test installation would be: <ul style="list-style-type: none"> • Linux RAM 2GB, 1 VCPU, 20GB free disk space • Windows RAM 4GB, 1 VCPU, 20GB free disk space
6.1.6	Limitations on using virtualization (hardware/laaS)?	Think!EHR Platform is supported to run in a virtual environment such as VMWare for which it was tested as well as other major virtualization solutions. Also we tested Think!EHR Platform to run in docker containers (for which it requires local storage).

6.2 Security		Answer:
6.2.1	Support of role based authorization? Describe (default/typical) roles	<p>Patient privacy settings can be represented within the openEHR EHR itself or kept in external Policy Repository. A 'pluggable' model allows rules based on access models, including those of CEN and ISO to be used and even changed over time.</p> <p>When new users are created roles are typically added only to a single domain - which allows them access to only a single domain.</p> <p>User roles are as follows:</p> <ul style="list-style-type: none"> • ROLE_ADMIN: allows access to all actions in user's domain • ROLE_READ: allows getting compositions, but no modifications are allowed • ROLE_WRITE: allows creating, updating and deleting compositions • ROLE_WRITE_EHR: allows creating and updating of EHRs, demographic parties and EHR meta-data (EHR_STATUS) • ROLE_WRITE_GENERATED: allows creating 'generated' compositions without any validation. Typically used by CDS engines • ROLE_QUERY: allows executing AQL queries, but no modifications are allowed • ROLE_VIEW: allows retrieving data with views

6.2 Security	Answer:
	<ul style="list-style-type: none"> • ROLE_PRESENTATION: allows access to presentation API endpoint • ROLE_TEMPLATE_ADMIN: allows management of templates • ROLE_USER_ADMIN: allows management of users • ROLE_EVENT_ADMIN: allows management of events • ROLE_VIEW_ADMIN: allows management of views • ROLE_FORM_ADMIN: allows management of forms <p>In addition to roles, users can also be marked as 'Super user' which gives them full access to all domains.</p>
6.2.2. Support of authentication tickets issued by an Identity Provider (e.g, SAML)?	<p>Information in Think!EHR Platform can leave the system only via service API which is protected by authentication approach, where every API call requires a secure session key. The direct access to database is protected by a variety of mechanisms, including data encryption and highly secure login control.</p> <p>The security components allow authentication of users and management of secure Think!EHR Platform sessions. Existing authentication security services can be used via an interchangeable Authentication Adapter configured to work with Think!EhrPlatform. The Session Manager is responsible for initialising an Think!EHR Platform session and managing EHR access. The comprehensive use of versioning, coupled with optional hashing and digital signing of every committed change to the EHR ensures the integrity and authorship of all health information in clinical data repository.</p> <p>Out of the box Think!EHR Platform supports Basic http authentication as well as OpenID Connect (OAuth2). Authentication mechanism is modular and adding support for more types of authentication is relatively simple. Supporting several token issuers is also possible – we have a modular token validation policy, which allows configuration of several token origins.</p>

6.2 Security		Answer:
6.2.3	Support of logging; access and change?	<p>Each call made to Think!EHR Platform can be logged into an audit log. Think!EHR Platform supports 3 types of auditors. Auditing is always performed on a separate thread in order to not slow down the server.</p> <ul style="list-style-type: none"> • XML file audit where each call is logged into a compressed XML file. Files are closed at midnight every night and can be moved to a backup location in order not to use up all the active server disk-space. Part of the Think!EHR distribution is an audit file reader which can be used to process audit files. • Each call can be logged to an ATNA server. • For performance monitoring it's possible to log calls to Think!HUB • Logstash audit where it is possible to log all calls to the ELK stack • Several auditors supports to log to two different services, for example XML log files as well as HUB.

6.3 Training		Answer:
6.3.1	Availability of course or on-line training for administrators? Describe	<p>As part of Professional services, We also provides training courses for end users to use, operate and manage the Think!EHR Platform. Trainings are carried out as on site trainings or can be done remotely as on-line webinars. For administrators we provide Administration Course where experts provide end user administration users all major tasks of administering a Think!EHR Platform, such as deployment planning, Think!EHR Platform installation, Think!EHR Server management tasks and Think!EHR Server troubleshooting. On completion of this course the participants will be able to:</p> <ul style="list-style-type: none"> • Prepare a deployment plan for Think!EHR Platform • Install a Think!EHR Server • Administration of a running Think!EHR Server

6.3 Training		Answer:
		<ul style="list-style-type: none"> • Troubleshooting of an existing Think!EHR Server installation Manage archetypes and templates.
6.3.2	Availability of course or on-line training for technicians? Describe	<p>Our assumption is that by technicians you refer to developers working on top of Think!EHR Platform. As part of Professional services, we also provides training courses for end users to use the Think!EHR Platform. Trainings are carried out as on site trainings or can be done remotely as on-line webinars. For technicians we provide Development course where experts provide end technicians how to develop applications running on Think!EHR Platform. On completion of this course the participants will be able to:</p> <ul style="list-style-type: none"> • Develop an application to store data into Think!EHR • Query data stored in a Think!EHR Server • Use of demographics server • Use terminology services.
6.3.3	Availability of course or on-line training for users? Describe	<p>As part of Professional services, we also provides training courses for end users to use the Think!EHR Platform. Trainings are carried out as on site trainings or can be done remotely as on-line webinars. For users we Think!EHR Platform overview course where experts provide overview of key capabilities of the platform. On completion of this course the participants will be able to:</p> <ul style="list-style-type: none"> • Explain the various openEHR concepts • Basic use of openEHR tools such as ADL Designer • Explain how Think!EHR Platform can fit into different health related scenarios • Recognize different deployment options • List major Think!EHR features

6.4 Usage		Answer:
6.4.1	Is the number of registered users limited, if so what is the limit?	Usage of Think!EHR Platform is not limited to number of registered users. License is limited to the number of unique EHR IDs which is the number of unique patients for which data is stored.
6.4.2	Is the number of simultaneous users limited, if so what is the limit?	Usage of Think!EHR Platform is not limited to number of registered users. License is limited to

6.4 Usage		Answer:
		the number of unique EHR IDs which is the number of unique patients for which data is stored.
6.4.3	Is the number of managed assets limited, if so what is the limit?	<p>It depends on the licence that customer would purchase:</p> <ul style="list-style-type: none"> • If Customer is granted with full use licence than he is not limited with number of managed assets (applications) using capabilities of the Think!EHR Platform • If Customer is granted with Application specific licence then he is allowed to run single 3rd party or other application using Think!EHR Platform as underlying infrastructure. Furthermore, one ASL license applies for one implementation instance of Think!EHR Platform, only.
6.4.4	Does the license model allow usage for research as well as caregiving?	Licence for Think!EHR Platform does not differentiate the use for research or caregiving, and the solution can be used for any of such purposes.
6.4.5	Does the software product provide client libraries to support the development of software against the system, if so in what program languages?	Think!EHR Platform provides Think!EHR-framework client library designed to simplify standard developer tasks, such as presentation and submitting of the openEHR data.

6.5 Management		Answer:
6.5.1.	Is it possible to export system configuration between different instances of the installation? If so how?	System configuration can be used (exported and imported) using simple .conf files that carry the required configuration information.
6.5.2	Is it possible to run multiple instances of the installation on the same network without conflicts? If so how?	<p>Think!EHR server is easily scalable at server level, and can be adjusted over time to handle increasing demand without any service interruption. The Think!EHR Server cluster scales by adding Think!EHR Server servers and declaring them in the IP network. However High Availability properties rely on implied Load Balancing solution.</p> <p>Therefore you can run multiple instances on the same network. Network that cluster nodes communicate on must allow multicast traffic where each logical cluster needs own settings for cluster name. Adding a new node to an existing cluster requires creating a proper configuration. Most</p>

6.5 Management		Answer:
		setting should be the same as on existing nodes (watch out for index and cache cluster names, these must be the same as on existing cluster nodes). New nodes must also use the same clustering strategy (i.e. multicast or unicast) and match parameters for clustering configuration of existing nodes.
6.5.3	Is it possible to run different versions of the same system simultaneously within the same instance?	upgrades always contain only bug-fixes and possibly new functionality which is 100% compatible with the existing version.
6.5.4	Does the software allow soft launches of new versions?	Think!EHR Platform upgrades can be also minor upgrades - in which case only the third portion of the version number is incremented - i.e. an upgrade from 2.1.1 to 2.1.4. These upgrades always contain only bug-fixes and possibly new functionality which is 100% compatible with the existing version. Upgrades don't need to be consecutive - it is possible to upgrade from 2.0.x to 2.2.x directly, without having to upgrade to 2.1.x first.

6.6 Integrations		Answer:
6.6.1.	Does the software product have an interface to support import/export of HL7v2 messages?	Think!EHR Platform provides Adapter (content-based-transform) for message mapping and transformation to openEHR composition. Mapping from HL7 v2 messages are done using TDS. Template Data Schema is a XML Schema (XSD) that can be generated for each openEHR template. The resulting schema is hard-wired to the template's contents, and suited for XML data transformation and communication, for example as a message. This allows data sources such as pathology laboratories to generate their content according to schemas directly describing their result types, without having to understand openEHR. The same logic applies for any kind of content. The TDS approach holds great promise for integration because any data that conforms to TDS XML Schema (XSD) in the standard XML fashion is guaranteed to be convertible to standard openEHR content format. The capability of producing TDSs from templates is effectively a facility for machine-

6.6 Integrations		Answer:
		<p>generation of message definitions, replacing previous manual message definition approaches.</p> <ul style="list-style-type: none"> • Generate TDS from the Template (required to capture the integrated HL7 message content (using Template Designer). • Convert HL7 V2.x ASCII message to canonical XML • Create XSLT to map HL7 2.x XML to TDS (Using XML Authoring Tools) or provide specialized transformation class (Java) • Invoke HL7 v2.x to TDD transform and submit using Think!EHR Platform API
6.6.2	Does the software product have an interface to support import/export of HL7 FHIR messages?	At the moment Think!EHR Platform provides availability of basic observations data (weight, height, blood pressure, body temperature) exposed as FHIR resources. By the end of Q3 2018 we are planning to offer comprehensive interface for import/export of HL7 FHIR messages, in the context of supporting FHIR-based storage components for applications focusing on IoT data and personal device data.
6.6.3	Does the system support automated extraction of required IHE XDS.b data from openEHR compositions?	Think!EHR Platform is validated for IHE Integration profiles, enabling platform to act as an IHE validated registry (XDS.Reg), document repository (XDS.Repo), 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). At the same time, the platform stores data in openEHR structured data format, serving as an advanced clinical data repository with future-proof data analytics, information-flow opportunities and support innovation on top of common data architecture.
6.6.4	Does the system support extraction, mapping and storage of required DICOM metadata from KOS Objects to openEHR compositions	Think!EHR Platform does not provide out of the box extraction and mapping of required DICOM metadata from KOS objects to openEHR compositions and we see this as a custom development per request. As part of R&D activities (R&D Department) we have analyzed extracted KOS objects and mapped it to openEHR composition, but have not developed an out of the box plug-in.

6.6 Integrations		Answer:
6.6.5	Describe other integration support features of the platform.	<p>Think!EHR Platform also provides enablers (library) for directly perform data acquisition from connected health devices to mobile and web applications. Enabler hides the complexity of communications and the device specifics. It feeds health applications with measurements from medical devices without the gateways in between. Enabler automatically discovers, recognizes and communicates with health devices regardless of manufacturer and/or device type. It takes care of all the heavy lifting jobs, such as communicating with the device and decoding binary data and passing it to the application in readable and computable form. With this enabler, connected health device becomes a programmable data source. There is no need of gateway or other intermediaries such as cloud storage or API interfaces, just pump data from device directly into your applications. Enabler is based on Bluetooth Smart technology, devices supporting the implementation of specific profiles and characteristics related to healthcare devices. Enabler is provided as a framework to be included in your iOS application or as a JavaScript library to become part of your web based service just by adding couple of lines.</p>