Horizon 2020
European Union Funding
for Research & Innovation

C3-Cloud

"A Federated Collaborative Care Cure Cloud Architecture for Addressing the Needs of Multi-morbidity and Managing Poly-pharmacy"

PRIORITY Objective H2020-PHC-25-2015 -Advanced ICT systems and services for integrated care

D8.2 – Design of the implementation of the pilot application scenarios

Work Package:	WP8-C3-Cloud Pilot Application Development & Deployment
Due Date:	31 August 2017
Actual Submission Date:	31 August 2017
Project Dates:	Project Start Date: 01 May 2016
	Project End Date: 30 April 2020
	Project Duration: 48 months
Deliverable Leader:	Kronikgune

Project funded by the European Commission within the Horizon 2020 Programme (2014-2020)		
Dissemination Level		
PU	Public	X
CO	Confidential, only for members of the consortium (including the Commission Services)	
EU-RES	Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)	
EU-CON	Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)	
EU-SEC	Classified Information: SECRET UE (Commission Decision 2005/444/EC)	

Document History:

Version	Date	Changes	From	Review
V0.1	13/07/2017	Deliverable template	Kronikgune	Kronikgune/Osakidetza
V0.2	18/07/2017	Inputs in sections 1, 2, 3 and 4	Kronikgune	Kronikgune/Osakidetza
V0.3	21/07/2017	Extended version of the Deliverable, inputs in sections 1 to 6	Kronikgune	Kronikgune/Osakidetza
V0.4	28/07/2017	Inputs from SLCK and updating the fact that Windows is an option to deploy Docker containers	Kronikgune/ WARWICK (SLCK)	Kronikgune/Osakidetza/ Warwick
V0.5	01/08/2017	Responses to SLCK comments	Kronikgune	Kronikgune/Osakidetza/ Warwick/SLCK
V0.6	02/08/2017	Inclusion of latest suggestions and changes	Kronikgune	Kronikgune/Osakidetza/
V0.7	03/08/2017	Inputs from SRDC, SWFT and RJH	Kronikgune	Kronikgune/Osakidetza/ SRDC/SWFT/RJH
V0.8	10/08/2017	Inputs from all partners	Kronikgune	Kronikgune/Osakidetza/
V0.9	22/08/2017	Inputs from Internal Reviewers	Kronikgune	Phil Johns/Lei Zhao/Kronikgune
V1.0	30/08/2017	Final review	Warwick	

Contributors (Benef.)	Adolfo Ranea, Esteban de Manuel, Lola Verdoy, Ane Fullaondo (KG) Nicolás González, Antonio de Blas (OSAKIDETZA) Matias Wurschmidt-Wang, Mikael Lilja (RJH) Phil Johns, Danny Roberts, Marie Beach (SWFT) Lei Zhao, Sarah N. Lim Choi Keung, George Despotou, Theodoros N. Arvanitis (WARWICK) Gokce B. Laleci Erturkmen, Mustafa Yuksel (SRDC) Pontus Lindman (MEDIXINE) Damien Leprovost, Marie-Christine Jaulent (INSERM) Rong Chen (CAMBIO)			
Responsible Author	Esteban de Manuel		**Email**	edemanuel@kronikgune.org
	Beneficiary	Kronikgune	**Phone**	+34944007794

EXECUTIVE SUMMARY

C3-Cloud is planned to be deployed in three pilot sites South Warwickshire (United Kingdom), Basque Country (Spain) and Region Jämtland-Härjedalen (Sweden). This deliverable analyses and describes how the solutions provided by the C3-Cloud components can be deployed in the regional healthcare systems. It presents a general method to deploy C3-Cloud and specific aspects related to each component.

This deliverable describes a High-level Deployment Design (HDD) that defines the deployment requirements of the C3-Cloud High Level Components (HLC) to be implemented at the pilot sites. It will be used as a reference manual to understand C3-Cloud components interaction and the strategy on how the whole system can be deployed and used in their real production scenarios.

The HDD maps the components identified in the logical architecture to physical servers and other network devices to create deployment architecture.

The High-level Deployment Design of the system includes:

- The Application Architecture is a high-level view of the layers.
- The Technology Architecture represents the frameworks and technology used in each layer. It specifies for each of the HLC: the Development platform, Web Server, Application Server, Database Server and Messaging Server.
- The Deployment Architecture depicts how the software is packaged and installed and describes the resource requirements needed to deploy it. It involves resource sizing in a physical environment, which includes computing nodes in an intranet or Internet environment, CPUs, memory or storage devices. Strategies differ from Windows (Patient Empowerment Platform) to Docker (rest of the components).
- The Physical and Network Architecture defines and describes the hardware resources and communication infrastructure. Each module is deployed only in one machine at a time. They include a reverse proxy and load balancer on top of the modules as a single entry point.
- The Non-functional Requirements (system qualities) are persistent qualities and constraints of the system that ensure the usability and efficacy of the entire system to satisfy internal business, user, market needs, or regulatory or standards agencies. They provide guidance on hardware configurations for performance, availability, scalability, and other related quality of service (QoS) specifications.

The two main options in the Project for software distribution (organization of architecture layers and hosting responsibilities to make software available to end-users) are “On-premises” and “Infrastructure as a Service” (IaaS).

The High-level Deployment Design described in this deliverable can be adjusted. It is based on the current estimation of the High Level Components requirements. The details provided in the HDD will be adapted to the final High Level Components and Pilot Sites requirements based on security and privacy protection policy and distribution model at each site. The design will be cross validated and refined in related work tasks in work packages 5, 6, and 7, and be reviewed in the “Deployment and Operation of C3-Cloud Pilot Application” (Task 8.3), before the actual deployment per pilot site.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS AND ACRONYMS

Abbreviation/Acronym	Definition
CAMBIO	Cambio Healthcare Systems AB
C3DP	Coordinated Care and Cure Delivery Platform
CDA	Consolidated Clinical Document Architecture
CDSM	Clinical Decision Support Modules
CPU	Central Processing Unit
DNS	Domain Name System
EHR	Electronic Health Record
EMIS	Egton Medical Information Systems
EPR	Electronic Patient Record
EuroREC	European Institute for Health Records
empirica	empirica Gesellschaft für Kommunikations und Technologieforschung Mbh
FHIR	Fast Healthcare Interoperability Resource
HDD	High Deployment Design
HISA	Health Information Service Architecture
HL7	Health Level 7
HLC	High Level Components
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
ICD	International Classification of Diseases
ICT	Information Communication Technology
IdP	Identity Providers
INSERM	Institut National de la Sante Et de la Recherche Medicale
ITK	Interoperability Toolkit
JMS/MQ	WebSphere MQ classes for JMS
KG	Kronikgune
LCS	Local Care Systems
MDT	Multidisciplinary team
MEDIXINE	MEDIXINE OY
NFR	Non-functional requirements
NHS	UK National Health Service
O/S	Operating System
OSAKIDETZA	Osakidetza, Basque Public Healthcare Service
ORU	Örebro University
PCPDP	Personalized Care Plan Development Platform
PEP	Patient Empowerment Platform
PHG	Personal Health Folder
PHR	Personal Health Record
QoS	Quality of Service
RESTful	Representational state transfer
RJH	Region Jämtland-Härjedalen
SCR	Social Care Records
SIS	Semantic Interoperability Suite
SOA	Service oriented architecture
SPS	Security and Privacy Suite
SRDC	Software Research and Development and Consultancy Ltd.
SSL/TLS	Secure Sockets Layer/Transport Layer Security
SWFT	South Warwickshire NHS Foundation Trust
TCP	Transmission Control Protocol
TIS	Technical Interoperability Suite
UDDI	Universal Description, Discovery and Integration
WARWICK	The University Of Warwick
XML	eXtensible Markup Language

1. INTRODUCTION

C3-Cloud will establish an ICT infrastructure enabling a collaborative care and cure environment to enable continuous coordination of patient-centred care activities by a multidisciplinary care team and patients/informal caregivers. It includes several high level components. A Personalised Care Plan Development Platform will allow collaborative creation and execution of personalised care plans for multimorbid patients through systematic and semi-automatic reconciliation of clinical guidelines. Decision Support Modules will help with risk prediction and stratification, recommendation reconciliation, poly-pharmacy management and goal setting.

C3-Cloud Interoperability Middleware will be composed of Technical Interoperability Platform, Semantic Interoperability Platform and Privacy Protection and Security Mechanisms. They will altogether achieve continuous access to and fusion of patient and care provider data by seamlessly integrating with the Electronic Health Record (EHR), Social Care Records (SCR), Personal Health Record (PHR) and home/community care information systems. Active patient involvement and treatment adherence will be achieved through a Patient Empowerment Platform (PEP) ensuring patient needs are respected in decision making and taking into account preferences and psychosocial aspects[i].

The project original high-level components can be seen in the figure below. In the architectural design phase, Personalised Care Plan Development Platform (PCPDP) was merged with C3-Cloud Coordinated Care and Cure Delivery Platform (C3DP).

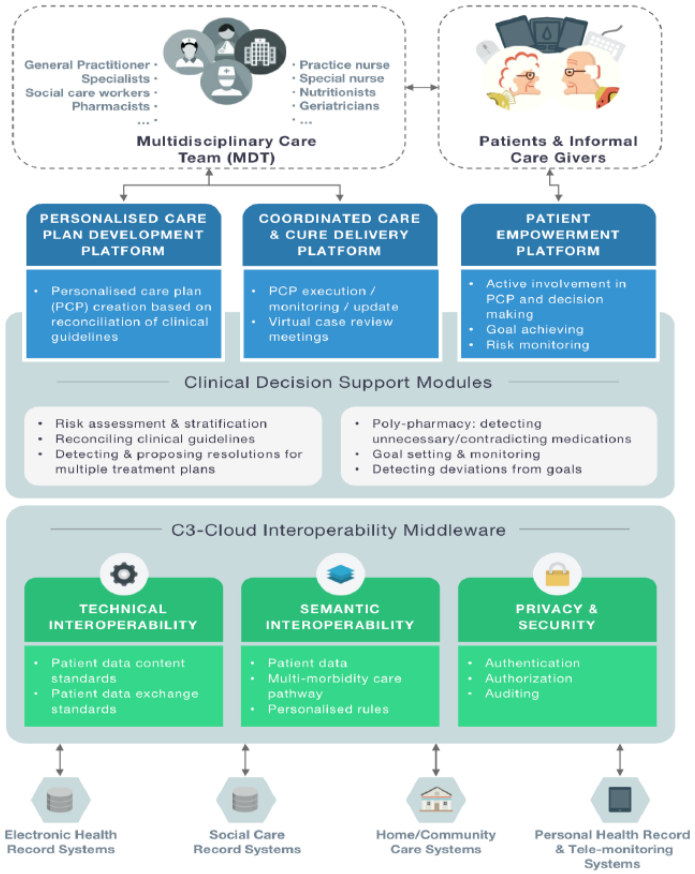

Figure 1: C3-CLOUD Overall Architecture

C3-Cloud is planned to be deployed in three pilot sites: South Warwickshire (United Kingdom), Basque Country (Spain) and Region Jämtland-Härjedalen (Sweden). This deliverable analyses and describes how the solutions provided by C3-Cloud components can be deployed in the Regional Healthcare

Systems. It presents a general method to deploy C3-Cloud and takes into account specific aspects related to each component.

The High-level Deployment Design (HDD) shown in this deliverable is based on the previous work already developed in work packages WP8 and WP3 of the Project. They comprise the requirements of the C3-Cloud pilot application and the design of C3-Cloud architecture. The former is covered in Deliverables 8.1 "Use cases and requirement specifications of the pilot application" and Deliverable 3.2 "Requirements specification of the C3-Cloud architecture". The design of C3-Cloud architecture at a conceptual level is in Deliverable 3.3. It provides a technical reference to continue implementation work for each high level component to the technical partners (D3.3 "Conceptual Design of the C3-Cloud architecture").

HDD also describes the resource requirements of the High Level Components that are being developed in WP5 (Patient Empowerment Platform), WP6 (Technical and Semantic Interoperability, and Security and Privacy Suite) and WP7 (Clinical Decision Support Modules and both the Personalized care Plan Development and the Coordinated Care and Cure Delivery Platforms). These WPs are still in progress so some changes to what is described in this document may be introduced in future. The High Level Components will be completed by month 18 (SIS, TIS and SPS), month 22 (PEP), month 20 (CDSM), and month 26 (PCPDP-C3DP) of the project. The corresponding deliverables will describe each HLC and refine the deployment requirements when necessary.

In Section 1 the Purpose, Context and Approach and Scope are presented, Section 2 discusses the Methodology, while Section 3 presents the High Deployment Design and finally Section 4 outlines the Distribution Model.

Apart from the "Design of the Implementation of the Pilot Application Scenarios ", Task 8.2 includes "The detailed mock-ups of the user interfaces to be presented to the actual users of C3-Cloud pilots, i.e. formal and informal care givers and patients, will be prepared and updated based on feedback (layer 1 evaluation). Storyboards of the planned scenarios depicting example control and information flow will be documented". This is a work in progress as High Level Components (HLCs) of the C3-Cloud project are currently being developed. The reporting of these subtasks will be done in the deliverables of the corresponding WPs (D9.3 Test and Evaluation Report for C3-Cloud Components for mock ups and D7.4 C3-Cloud Coordinated Care and Cure Delivery Platform for storyboards).

1.1. Purpose

The HDD explores the resource requirements for the C3-Cloud components to be deployed at the pilot sites[ii]. The healthcare information systems of the three regions have different characteristics. This deliverable will be used as a reference manual to understand C3-Cloud components interaction and the strategy on how the whole system can be deployed and used in the real production scenarios.

1.2. Context

The project aims to demonstrate the applicability of C3-Cloud integrated care approach and supporting set of innovative ICT components in varying clinical, technological and organisational settings. It will be piloted in three European regions (South Warwickshire, Basque Country and Region Jämtland-Härjedalen) with quite different health and social care systems and Information and Communication Technology (ICT) landscapes. This requires defining how the solutions provided by C3-Cloud components can be deployed in the pilot sites. Feasibility and integration with current information systems and ICT infrastructures are a "must" to achieve user acceptance and scale-up.

The ICT context is different in each Pilot Site. South Warwickshire (SWFT) adopted the 'Lorenzo' enterprise Electronic Patient Record (EPR). Initially an administration system, it was expanded in 2015/16 to include order communications, clinical documentation and electronic prescribing. The Global Assessment Platform system is developed on open source components and provides electronic scheduling and assessments capability in an on and off line mode. GP practices across South Warwickshire have moved to a single Information Technology provider, Egton Medical Information

Systems (EMIS). There is currently no integrated county-wide patient record accessible to all relevant agencies and no widespread use of telehealth or telemedicine technologies.

Basque Country Healthcare Service (Osakidetza) Electronic Health Record (EHR) service and inter-consultation functionality are provided through Osabide Global in primary and secondary care. The e-Prescription service Presbide is provided by a unique system in both care sectors. This system has been integrated as a module within the EHR systems (Osabide Global). The Personal Health Record service provided through the Personal Health Folder (PHF) system is integrated. Osakidetza uses a service-oriented architecture (SOA) using the Oracle SOA Suite platform. Osakidetza e-health applications are interoperable with UDDI v3 compliant services and JMS/MQ based middleware. HL7 CDA is used as the content standard in the e-prescription, EHR and Personal Health Record (PHR) systems.

In Region Jämtland-Härjedalen (RJH) fully integrated electronic health records are used in all health care centres and at the secondary care level for over 20 years. The Region uses electronic health care system Cosmic® by Cambio. Cosmic® is based on pre-European standard Health Information Service Architecture (HISA), supports open-EHR archetypes as core information models, and uses HL7 v2 and v3 messages for inter-system communication. Electronic prescription is achieved through Cosmic® and the national prescription database, NOD, the dose prescription support system, referred to as Pascalor or databases with medicine lists and codes, SIL, FASS etc. RJH planned to allow all patients' access to their records through EHR in 2016-2017. There is a national and local website (www.1177.se) that is intended to be the patient's first contact point with the care system. The website also includes education and information. ICD-10 is used for hospitalisation and in primary/secondary care; Diagnostic Related Groups (DRG) in secondary care. Within home care, wireless tele-monitoring is used (e.g., to register access by care givers to the patients' home).

A HDD to implement C3-Cloud pilot is defined in order to describe a physical environment for production. HDD depicts a system and helps to identify high level components deployment needs. Deployment architecture has to meet functional specification defined by the technical work team and the pilot sites conditions.

1.3. Approach and Scope

A High-level Deployment Design will be described, depicting how High-Level Components (HLC) will be deployed and used in the Pilot Sites. To do so, this task:

- Gathers information from previous WP tasks to define HDD and understand application components requirements, such as the database architecture, application architecture (layers) and technology architecture[iii].
- C3-Cloud logical architecture is analysed to design the deployment architecture: hardware and network. Non-functional requirements like security, reliability, maintainability are taken into account.
- Assesses within each Pilot Site the requirements and feasibility of the solution proposed.

As part of the pilot application design, it explores how the solutions provided by C3-Cloud components can be deployed at each Pilot Site. HDD covers the deployment scenario created by the logical design and technical requirements but also the Pilot Site needs and conditions.

The HDD includes the following:

- The Application Architecture depicts the high-level view of the components.
- The Technology Architecture represents the frameworks and technology used in each layer: It specifies for each of the HLC the Development platform, Web Server, Application Server, Database Server and Messaging Server.
- The Deployment Architecture shows how the software is packaged and installed; and describes the resource requirements needed to deploy it.
- The Physical and Network Architecture defines and describes the hardware resources and communication infrastructure.

- The Non-functional Requirements (system qualities) are persistent qualities and constraints of the system that ensure the usability and efficacy of the entire system.
- The Distribution Model is the organization of architecture layers and hosting responsibilities to make software available to end-users.

C3-Cloud is a research and innovation project, and prototypes will be delivered at the end. It does not aim to have the same level of completeness of a solid product (e.g., in terms of disaster recovery), but some minimum non-functional requirements must be met to guarantee backup and disaster recovery. The Deployment Architecture and Physical Architecture described below have been designed with that in mind. Therefore, a basic layout without hardware redundancy, clustering or heavy setups is used. The idea is to optimize physical resources that can be shared among many C3-Cloud components and minimize infrastructure costs.

The High-level Deployment Design is based on the current definition of the High Level Components requirements. The details provided in the HDD will be adapted to the final High Level Components and Pilot Sites requirements based on security and privacy protection policy and distribution model at each site. The design will be cross validated and refined in related work tasks in work packages 5, 6, and 7. Any changes will be reviewed before the actual deployment per pilot site.

2. METHODOLOGY

The HDD maps the components identified in the logical architecture to physical servers and other network devices to create deployment architecture. The HDD uses non-technical to mildly technical terms, which should be understandable to the administrators of the system. Key points in the specification include physical computing resources, location, data partitioning, scaling, security, backup, etc.

The methodology used to define the HDD has been:

1. A HDD form (template) has been designed. The HDD documentation presents the structure of the system, such as the database architecture, application architecture (layers) and technology architecture. Each C3-Cloud high level component responsible team have completed the template with detailed information on:
    - Component name and short description
    - Component role: C3DP, PEP, etc. (as documented in WP3)
    - Component responsible beneficiary
    - Development platform
    - Technologies:
        - Virtualization platform
        - Operating System
        - Client platform
        - Web Server
        - Application Server
        - Database Server
        - Database connector
    - Audit and Security:
        - Authentication
        - Scope of use: internal, public
        - Identity management: internal, external
        - Auditing requirements
        - Other security requirements
    - N-tier architecture: Presentation layer, Businesses or Logic layer, Data access or Persistence layer.
    - Input/output devices

- Interoperability:
  - List of related components or systems
  - List of exposed web services and technology used to implement them
- Provisioning:
  - Software packaging
  - Software deployment

2. Kronikgune has summarized and consolidated all HDD forms into one HDD global system form. Common features have been identified and organized. Concepts have been agreed and homogenized across all components.
3. The Design Deployment architecture has been defined, mapping a logical architecture to a physical environment. It includes a high-level view of the Project layers, describing the frameworks and technology used in each layer, how the software is packaged and installed, and describes the requirements needed to deploy it and the mapping of a logical architecture to a physical environment. The hardware resources, communication infrastructure, qualities and constraints of the system that ensure the usability and efficacy of the entire system have been defined.
4. The available Distribution Models have been analyzed (On-premises, Infrastructure as a Service, Platform as a Service and Software as a Service). The final option takes into account the responsibility on different architectural layers management (networking, storage, servers, virtualization, Operating System (O/S), Middleware, Databases, Runtime, Data and applications).
5. A final analysis and review have been made by the Pilot Sites ICT technical teams to check and fine-tune the adequacy, feasibility and resources needed for the deployment solutions proposed with local strategies and capacity.

To do so, the steps followed were:

- The draft of the template was discussed with partners and a final template was agreed and completed by technical partners in charge of software development (HLC).
- A consolidated document was produced with the inputs from High Level Components requirements (PEP from MEDIXINE, TIS from WARWICK, SIS from INSERM, SPS from SRDC, CDSM Technology from CAMBIO and PCPDP-C3DP from SRDC). The completed forms can be found in Section 6 (Appendix).
- A first draft of the HDD was produced after discussion with the partners in June 2017.
- Extended technical details with system sizing requirements were included in a second version.
- Further versions were produced, taking into account pilot sites requirements, clarifying queries and detailing information.
- A final version with the proposed HDD and Distribution Model was agreed with technical partners and Pilot Sites.

The details provided in the HDD will be reviewed before the actual deployment at each Pilot Site. The deployment will be completed by Month 30, and then the pilot will be operated for 15 months. During operation, some minor updates and bug-fixes that could not be caught during component testing will be applied by the developer partners.

3. HIGH-LEVEL DEPLOYMENT DESIGN

The High-Level Deployment Design of a global system includes five sections:

- The Application Architecture is a high-level view of the layers. In the architectural design phase, PCPDP was merged with C3DP. In the rest of this document, whenever C3DP is mentioned, it includes PCPDP as well.
- The Technology Architecture represents the frameworks and technology used in each layer: It specifies them for each of the HLC the Development platform, Web Server, Application Server, Database Server and Messaging Server.
- The Deployment Architecture depicts how the software is packaged and installed and describes the requirements needed to deploy it. It involves sizing the deployment to meet the system requirements, computing nodes in an intranet or Internet environment, CPUs, memory or storage devices. Strategies differ from Windows (PEP) to Docker (rest of the components)
- The Physical and Network Architecture defines and describes the hardware resources and communication infrastructure. Each module is deployed only in one machine at a time. They include a reverse proxy and load balancer on top of the modules as a single entry point.
- The Non-functional Requirements (system qualities) are persistent qualities and constraints of the system that ensure the usability and efficacy of the entire system to satisfy internal business, user, market needs, or regulatory or standards agencies. They provide guidance on hardware configurations for performance, availability, scalability, and other related quality of service (QoS) specifications.

HDD is a general guide that will be used as a reference to define specific physical architectures in each pilot site, according to its own resources and procedures.

3.1. Application Architecture

"Application Architecture" is a high-level view of the Project layers. C3-Cloud is a composite of several modules:

- **Coordinated Care and Cure Delivery Platform (C3DP)**: C3DP facilitates collaborative management of care of patients with multi-morbid conditions. With the help of Clinical Decision Support Modules (CDSM), it provides care team members with the capability to define, update, reconcile and share care plans, and organize online meetings for care plan review. It also allows care team members to navigate a patient's medical history along with his/her care plan history. It should be noted that Personalised Care Plan Development Platform (PCPDP) that has been analyzed as an individual component in the requirements analysis phase is indeed a sub-component of and deeply integrated with the C3DP.
- **Patient Empowerment Platform (PEP)**: PEP System provides RESTful services for PEP Client System components to exchange information related to PEP functionalities (care plan and patient-originated data related to the care plan). The information will be exchanged using FHIR STU3 standard. PEP System will further provide browser-based user interface to its functionalities. The user interface can be used on all modern desktop and mobile browsers. PEP supports data collection from connected self-monitoring devices.
- **Clinical Decision Support Module**: The CDSM is a RESTful service that provides a hook-based pattern for invoking decision support from within a client's (e.g., a clinician's EHR) workflow.
- **Technical Interoperability Suite (TIS)**: TIS is a RESTful service, which enables C3-Cloud subsystems to access patient data, and share care plans with local care information systems through FHIR-based RESTful interface. TIS connects local systems through their native APIs.
- **Semantic Interoperability Suite (SIS)**: SIS provide a RESTful service to Technical Interoperability Suite which enables it to convert patient data from their local representation to FHIR-based representation, and care plans from C3-Cloud to this local representation.

- **Terminology Service (TS)**: Terminology service is a RESTful service based on FHIR Terminology Service Specification for representing, accessing and disseminating terminological content. Terminology service is maintained at a central location outside pilot site deployment and so is not included in the deployment design.
- **Security and Privacy Suite (SPS)**: SPS provides RESTful services for user (i.e. Care Team Member) authentication, user authorization and audit trail management purposes. These services are based on open standards wherever possible, such as OpenID Connect authentication layer that is designed to fit web applications as well as native / mobile apps. SPS will integrate with local Identity Providers (IdP) for enabling single sign-on with existing user accounts.

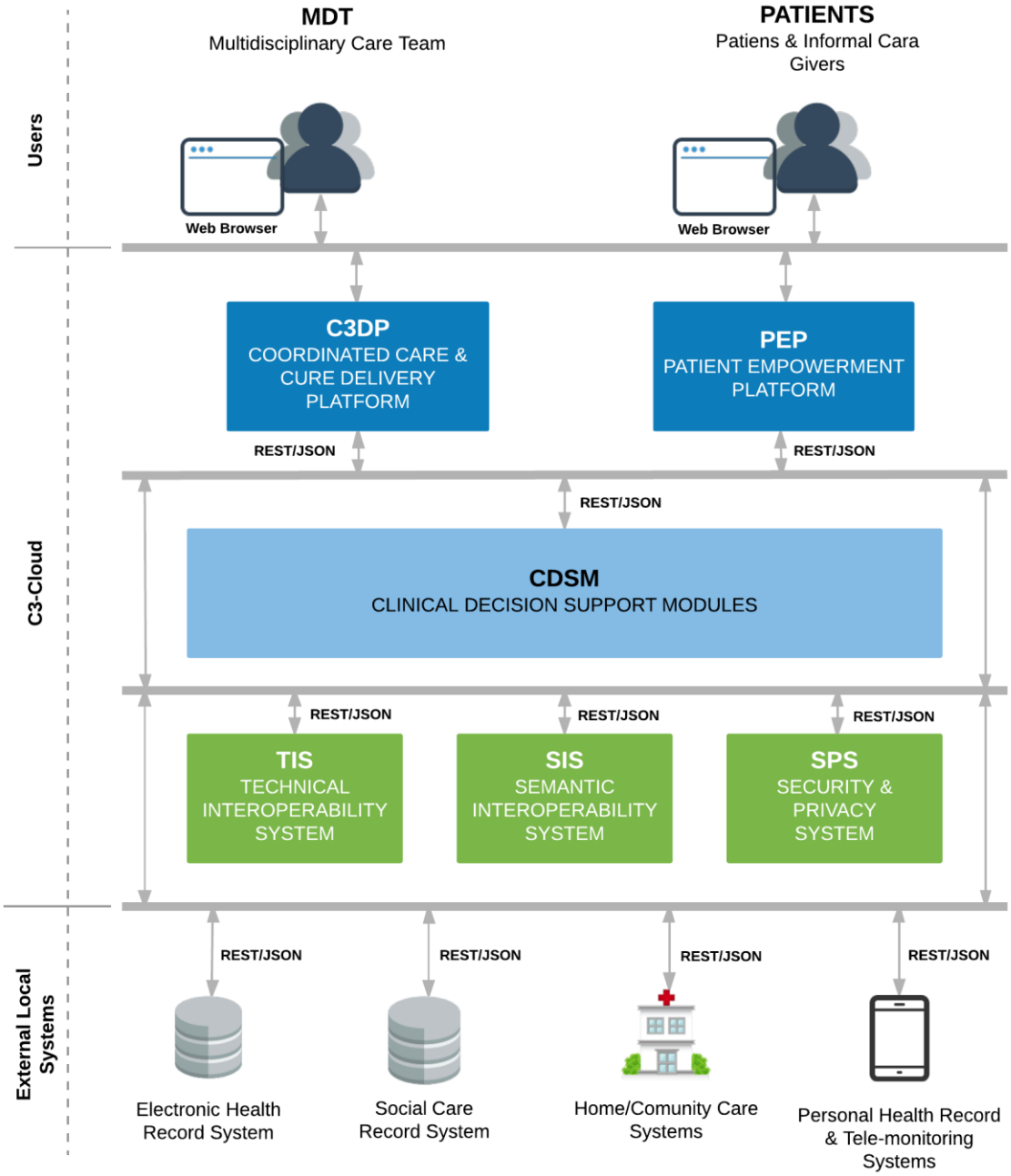

Figure 2: C3-CLOUD Application Architecture

3.2. Technology Architecture

The Technology Architecture represents the frameworks and technology used in each layer. It evolves from the Application Architecture view and takes into consideration the frameworks and technologies each HLC needs:

- **Coordinated Care and Cure Delivery Platform (C3DP)**:
  - Development platform: Scala, Java, Node.js, Angular 4
  - Web Server: Nginx
  - Application Server: Akka HTTP
  - Database Server: MongoDB
- **Patient Empowerment Platform (PEP)**:
  - Development platform: Microsoft .NET Framework 4.5.2
  - Web Server: IIS 7.0/7.5/8.0/8.5
  - Application Server: IIS 7.0/7.5/8.0/8.5
  - Database Server: Microsoft SQL Server 2008 (R2) / 2012
  - Messaging Server: Email, SMS
- **Clinical Decision Support Module (CDSM)**:
  - Development platform: Java 8
  - Web Server: Tomcat 8
  - Application Server: Tomcat 8
  - Database Server: No database is needed
- **Technical Interoperability Suite (TIS)**:
  - Development platform: Java 8, Spring Boot, Kafka
  - Web Server: Tomcat 8.5, Java 8
  - Application Server: Tomcat 8.5 + Spring 4.3
  - Database Server: H2 1.4
  - Messaging Server: Kafka 0.10.2.1, Zookeeper 3.4.10
- **Semantic Interoperability Suite (SIS)**:
  - Development platform: Java 8
  - Web Server: Tomcat 8
  - Application Server: Tomcat 8
  - Database Server: NoSQL (embedded in tomcat environment)
  - Messaging Server: N.A.
- **Security and Privacy Suite (SPS)**:
  - Development platform: Scala, Java, Angular 4
  - Web Server: Nginx
  - Application Server: Akka HTTP, WildFly (for Keycloak)
  - Database Server: MongoDB, mySQL (for Keycloak)
  - Messaging Server: N.A.

The technology architecture is depicted in Figure 3.

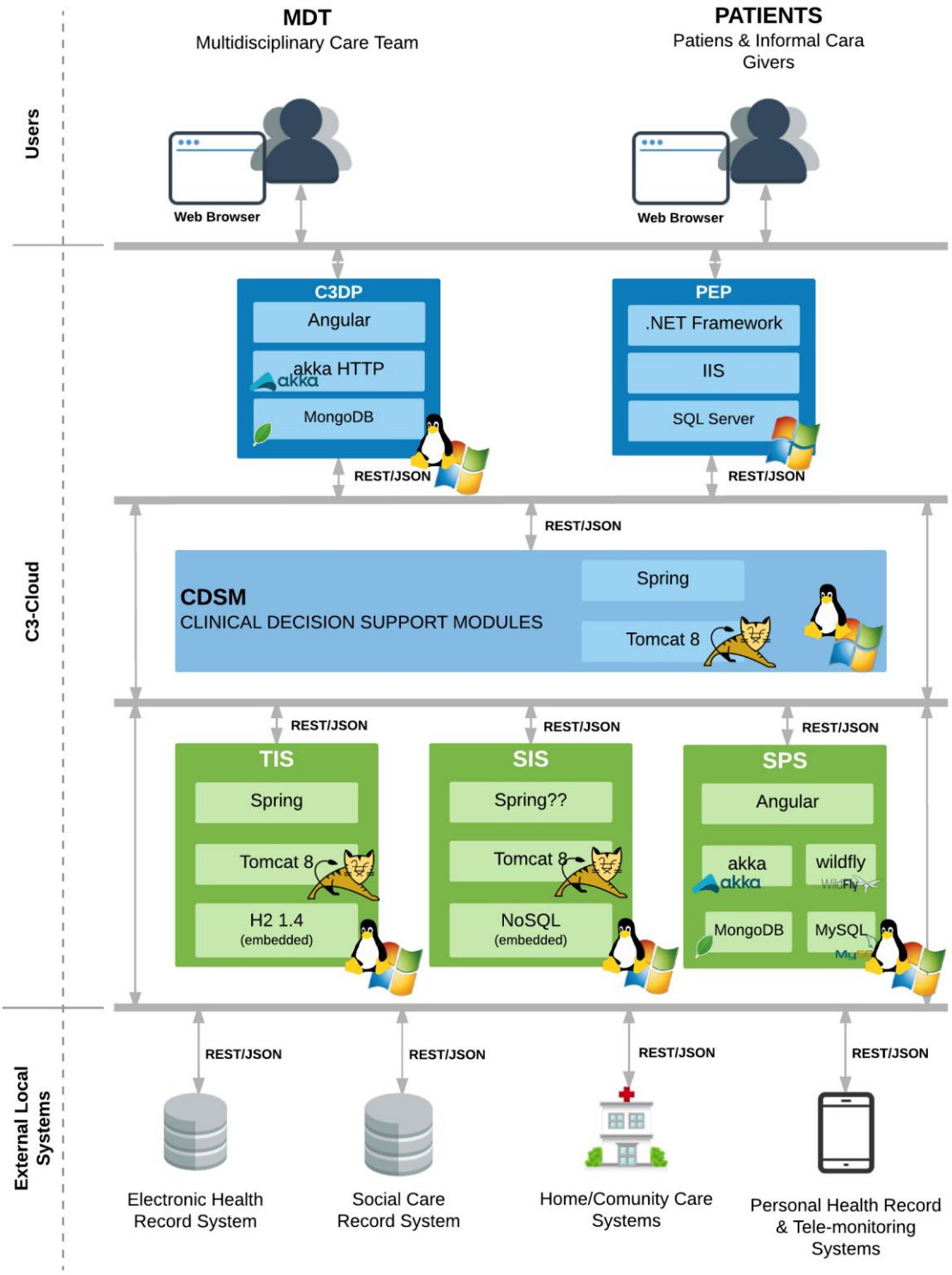

Figure 3: C3-CLOUD Technology Architecture

3.3. Deployment Architecture

Deployment architecture depicts how the software is packaged and installed and describes the requirements needed to deploy it and the mapping of a logical architecture to a physical environment. The physical environment includes the computing nodes in an intranet or Internet environment, CPUs, memory, storage devices, and other hardware and network devices.

Designing the deployment architecture involves sizing the physical resources necessary to meet the system requirements specified during the technical design phase. The deployment architecture is used to optimize resource usage by analysing the results of sizing the deployment to create a design that provides the best use of resources within business constraints. We need a detailed deployment architecture to evaluate the requirements for an organization that wants to install C3-Cloud on its own infrastructure.

There are two main deployment strategies: Windows Server for PEP and Docker[iv] for the other HLCs. Only the PEP has strong dependency to Microsoft Windows operating system. The remaining components do not have any strong operating system dependency, and can run on Linux, Windows and macOS operating systems. In addition, these components will also be packaged as Docker images. Docker simplifies the deployment and helps in the administrative task.

Deployment Architecture (and Physical Architecture described below) are designed keeping in mind the project's nature. So, a basic layout without hardware redundancy, clustering or heavy setups is used. We will define hardware requirements for each separately as shown in Figure 4.

- **Coordinated Care and Cure Delivery Platform (C3DP):**
  - Application Server: 1 Docker Container Akka
  - Database Server: 1 Docker Container MongoDB
- **Patient Empowerment Platform (PEP):**
  - Application Server: Microsoft Windows Server 2008
  - Database Server: Microsoft SQL Server 2008 (R2) / 2012
- **Clinical Decision Support Module (CDSM):**
  - Application Server: 1 Docker Container
- **Technical Interoperability Suite (TIS):**
  - Application Server: 1 Docker Container
  - Database Server: Embedded in application server container
- **Semantic Interoperability Suite (SIS):**
  - Application Server: 1 Docker Container
  - Database Server: Embedded in application server container
- **Security and Privacy Suite (SPS):**
  - Application Server: 1 Docker Container Akka, 1 Docker Container WildFly
  - Database Server: 1 Docker Container MongoDB, 1 Docker Container MySQL

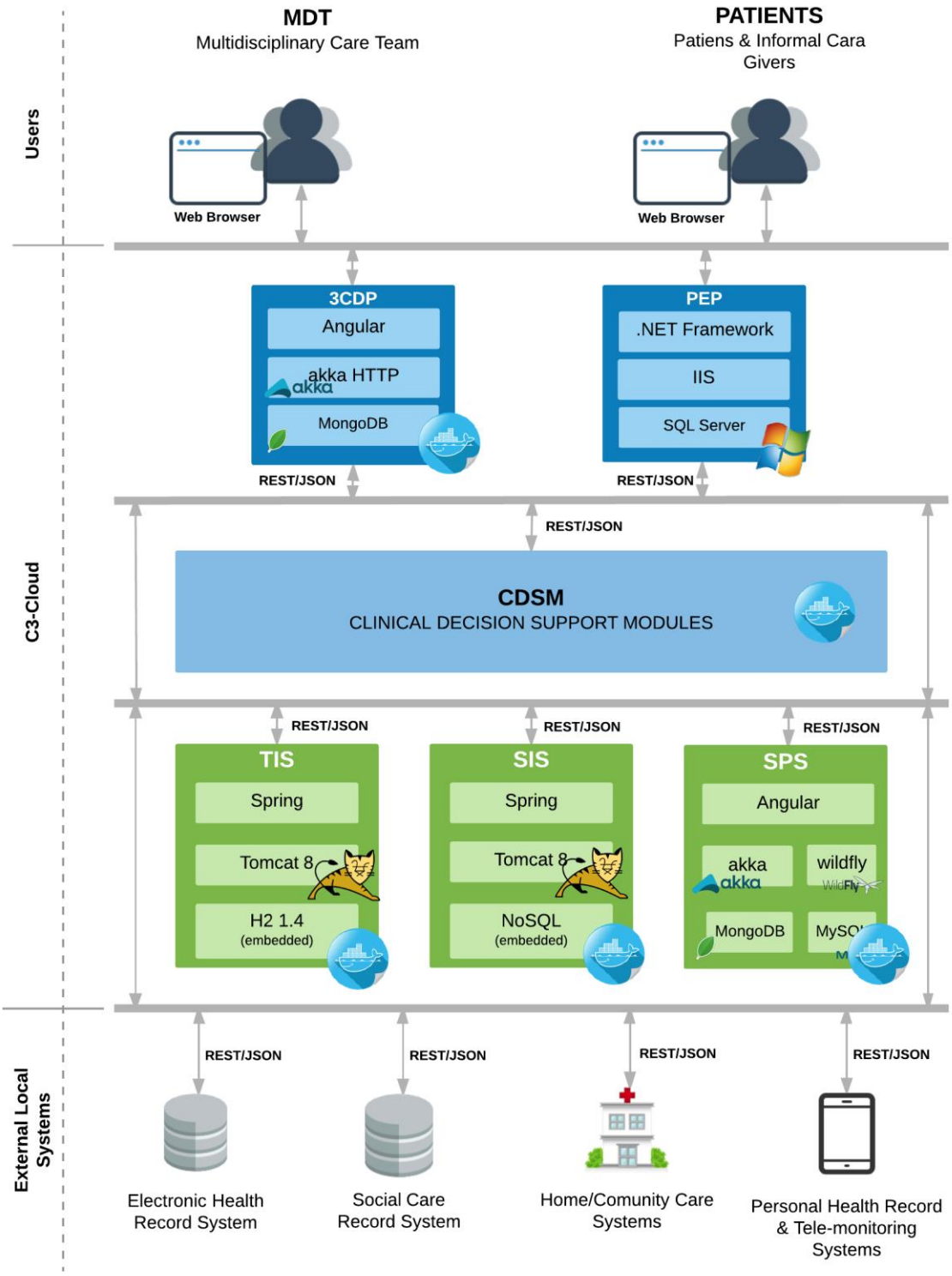

Figure 4: C3-CLOUD Deployment Architecture

3.3.1. Docker Deployment Strategy

Docker is a platform for packaging, running and managing applications in a flexible and efficient manner. Its approach to container-based virtualization is gaining traction among OS and cloud providers

such as Redhat, Microsoft and Amazon. They can be deployed in any of the operating systems supported by Docker. Linux should be the first option to choose in order to save cost, but Docker containers can run on Windows also.

Docker takes advantage of Linux kernels' ability to run applications in containers, which are sometimes described as "chroot on steroids." Containers provide each application an independent runtime environment, while avoiding the overhead of a full-fledged virtual machine. Each container gets its own virtual file system, process listing and network stack; however, containers share the OS kernel with each other and the underlying host. In this respect, the isolation provided by containers is less robust than that of real virtual machines, which have independent kernels and run on top of a hypervisor. Yet, sharing the kernel allows containers to run faster and offers management features that are difficult to accomplish with traditional virtualization.

An application distributed as a Docker image incorporates all the dependencies and configuration necessary for it to run, eliminating the need for end-users to install packages and troubleshoot dependencies. This approach allows developers to be certain that if the application worked in development, it will work in production. Docker provides the tools necessary to build, run and manage applications packaged as Docker images.

Docker images can be distributed in many ways. The simplest and free way is to use a private registry. A private registry can be installed on each pilot site. The other way is to use a public private registry but it has a cost. In both cases, distribution to C3-Cloud machines would be the same. Docker compatible C3-Cloud components will be released as Docker images:

- Creates Docker image and push it to private repository
- Access pilot site machine and pull docker image
- Stops previous container and start a new container based on latest image.

Modules to be deployed with Docker are PCPDP-C3DP, TIS, SIS, SPS and CDSM

Machine 1: Application Server Containers

Module	Mounts		Network Settings	CPUs (cores)	RAM
	Name	Disk Requirement	Ports		
C3DP	/mount/c3dp	10GB	80 or 443	2	8GB
	/mount/fhirrepo	10GB	80 or 8080		
TIS	/mount/tis	20GB	80 or 443	4	8GB
SIS	/mount/sis	10GB	80 or 443	2	8GB
SPS	/mount/sps	5GB	8280	2	4GB
	/mount/keycloak	5GB	8180		
CDSM	N.A.	10GB	N.A.	N.A.	N.A.

Other requirements: No other requirements reported

Machine 2: Database Server Containers

Module	Mounts		Network Settings	CPUs (cores)	RAM
	Name	Disk Requirement	Ports		
C3DP	/mount/fhirdb	50GB	27017	2	4GB
TIS	*N.A. database embedded in application container*				
SIS	*N.A. database embedded in application container*				
SPS	/mount/mysql	5GB	3306	1	2GB
CDSM	*No database needed*				

Other requirements: The same MongoDB instance will be used for SPS and C3DP, it is not necessary to have another instance.

3.3.2. Windows Deployment Strategy

PEP has to be deployed in Windows Server. The software can be packaged for manual deployment or it can be automatically deployed from a build server if the infrastructure provides the required connectivity between the build server and application servers.

The document "Installation and Maintenance Guide - Medixine Suite" has additional details on deployment of the Medixine Suite software.

Microsoft Windows Server, Microsoft IIS and Microsoft SQL Server have additional license costs that have to be faced by Project budget. Each pilot site has to take into account legal and licensing issues, as the hosting has an additional cost.

Machine 3: Application Server

Module	Disk	Network Settings	CPUs	RAM
PEP	100 gb + 100 gb (system & app)	Ports: 443 & 80 TCP (3389 for remote management TCP & UDP)	2 cores	4 GB

Machine 4: Database Server

Module	Disk	Network Settings	CPUs	RAM
PEP	100 gb + 100 gb (system & data)	1433 TCP (3389 for remote management TCP & UDP)	4 cores	8 GB

3.4. Physical and Network Architecture

This Physical and Network Architecture defines and describes the hardware resources and communication infrastructure. Each module is deployed only in one machine at a time. This model is not designed to scale and high availability strategy is not defined.

3.4.1. Load balancer and proxy server

A high availability load balancer and proxy server for TCP and the HTTP-based applications that spreads requests across multiple server will be used on top of the modules to:

- Serve as a single entry point of C3-Cloud system
- Manage SSL/TLS connections: SSL/TLS certificate would be installed and managed

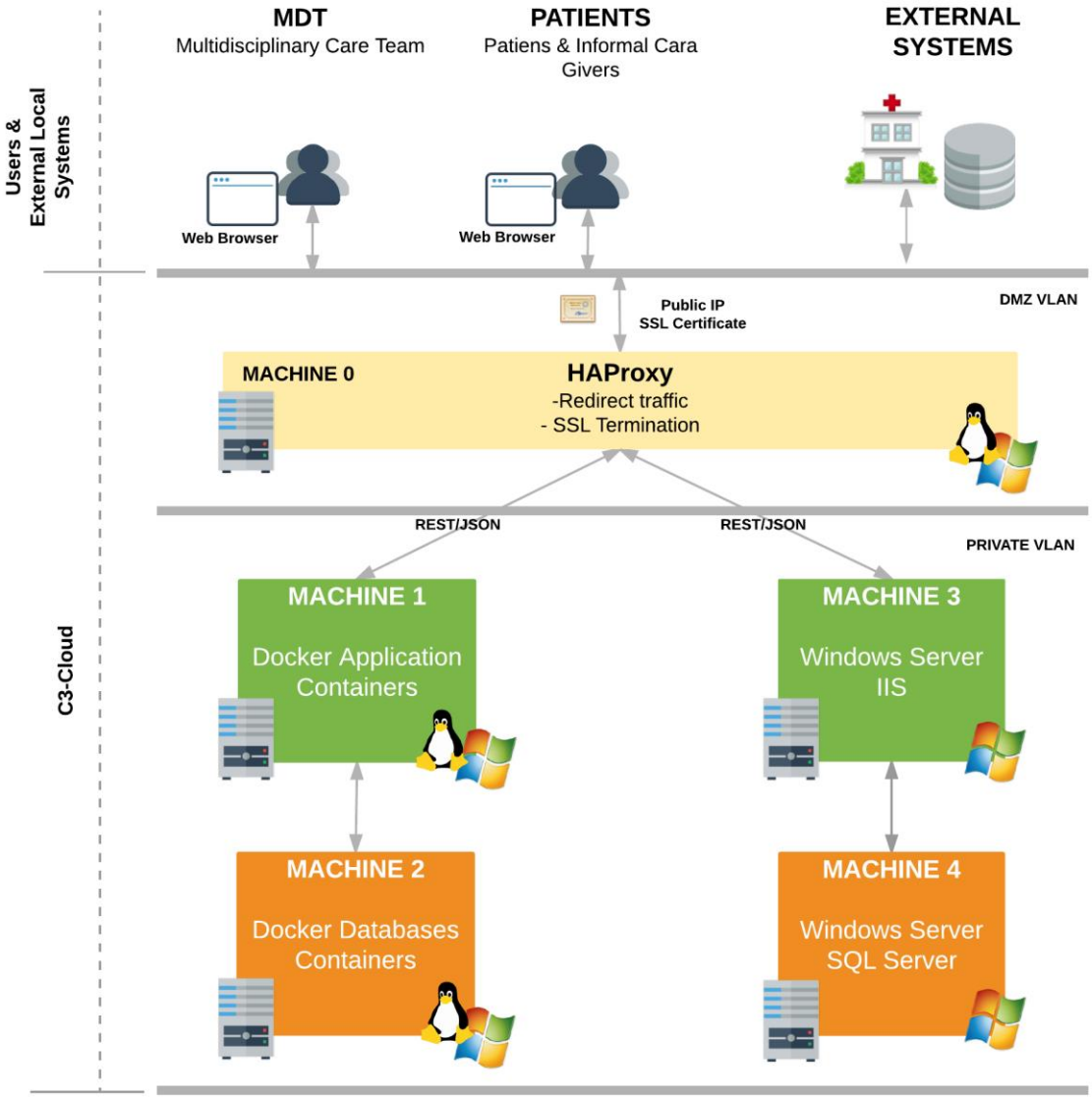

Figure 5: C3-CLOUD Physical and Network Architecture

There are many load balancers and reverse proxies, both software and hardware, in the market. In case that a load balancer is not available in Pilot Sites, HAProxy is recommended. HAProxy[v] is a free, very

fast and reliable solution offering high availability, load balancing and proxy for TCP and HTTP-based applications. It has benefit in:

- Virtualization: The clients do not need to know where the services are deployed. Clients always call to HAProxy and HAProxy redirects the requests to the right machine. It is easy to change redirection configurations in HA Proxy, making it useful during maintenance or migration task.
- DNS: The configuration for DNS is easier because all DNS names points to a single machine, the HAProxy one.
- Security: HAProxy is the only machine exposed in Internet. Machines where modules are deployed are not directly accessible from the Internet, which adds a level of security. Security policies, including SSL, are managed by HAProxy. SSL certificates configuration is done only once in HAProxy. It would be more difficult to configure certificates per module/server.

HAProxy is particularly suited for very high traffic web sites and powers quite a number of the world's most visited ones. HAProxy is available in DMZ VLAN and receives requests from the Internet. HAProxy managed SSL/TLS connection and redirect request in a plain protocol to web servers: Machine 1 and Machine 3. Redirection can be configured based on Name Server. For this requirement, a subdomain could be defined for each C3-Cloud module. This is a DNS proposal:

Module	DNS Name	Redirects to...
C3DP	c3dp.c3cloud.org	Machine 1 IP
	fhir.c3cloud.org	
TIS	tis.c3cloud.org	Machine 1 IP
SIS	sis.c3cloud.org	Machine 1 IP
SPS	sps.c3cloud.org	Machine 1 IP
CDSM	cdsm.c3cloud.org	Machine 1 IP
PEP	pep.c3cloud.org	Machine 3 IP

One domain name for the FHIR repository is needed, as it will be directly accessible by other components like TIS and PEP. With this approach, SSL/TLS certificate must support wildcard domains to protect all C3-Cloud subdomains.

3.4.2. Hardware and Network requirements

The Hardware Requirements are:

Machine 0: HAProxy, SSL/TLS Termination

Resource	Requirement
CPU	Core2 duo
RAM	3Gb~4Gb
Disk	250Gb
Network	2 interfaces (service and managemenrt)
OS	Centos 7
Other Software requirements **libraries or preinstalled software**	

Machine 1: Application Server Containers

Resource	Requirement
CPU	8 cores
RAM	32GB
Disk	200GB
Network	2 interfaces (service and management)
OS	Linux/ Windows or others
Other Software requirements **libraries or preinstalled software**	Docker

Machine 2: Database Server Containers

Resource	Requirement
CPU	4 cores
RAM	8GB
Disk	100GB
Network	2 interfaces (service and management)
OS	Linux/ windows or others
Other Software requirements **libraries or preinstalled software**	Docker

Machine 3: Windows Server Application

Resource	Requirement
CPU	2 cores
RAM	4 GB
Disk	200 GB
Network	2 interfaces (service and management)
OS	Microsoft Windows 2008 (R2) / 2012 (R2) Server
Other Software requirements **libraries or preinstalled software**	IIS 7.0/7.5/8.0/8.5 Microsoft .NET Framework 4.5.2

Machine 4: Windows Server Database

Resource	Requirement
CPU	4 cores
RAM	8 GB
Disk	200GB
Network	2 interfaces (service and management)
OS	Microsoft Windows 2008 (R2) / 2012 (R2) Server
Other Software requirements **libraries or preinstalled software**	Microsoft SQL Server 2008 (R2) / 2012

3.5. Non-functional requirements

The Non-functional Requirements (NFRs or system qualities) are persistent qualities and constraints of the system that ensure the usability and efficacy of the entire system to satisfy internal business, user, market needs, or regulatory or standards agencies. Some of NFR are backup, disaster recovery, fault tolerance, analytics and reporting.

C3-Cloud is a Research and Innovation Project, and prototypes will be delivered at the end. The project is not expected to have the same level of completeness as a solid product, e.g. in terms of disaster recovery, scalability and high availability. The software deployment will comply with the Pilot Sites Procedures and the Non-Functional Requirements:

- Basic:
  - Backup: As a minimum requirement, databases must be backed up.
    - Docker: This can be achieved using Docker volumes
    - SQL Server: Backup full database
  - Disaster recovery: In case of disaster, the strategy is to reinstall Docker containers or Windows Server software and recover database backup.
- Optional: Each pilot site will define it according to the specific procedure in the site.
  - Fault tolerance (e.g., Operational System Monitoring, Measuring, and Management):
  - Analytics and reporting.

4. DISTRIBUTION MODEL

The Distribution Model is the organization of architecture layers and hosting responsibilities to ensure software availability for end-users[vi]. There are several deployment models available in the market. Each of them has its own separation of duties. The hosting infrastructure for the C3-Cloud High Level Components has to suit local needs best at each pilot site.

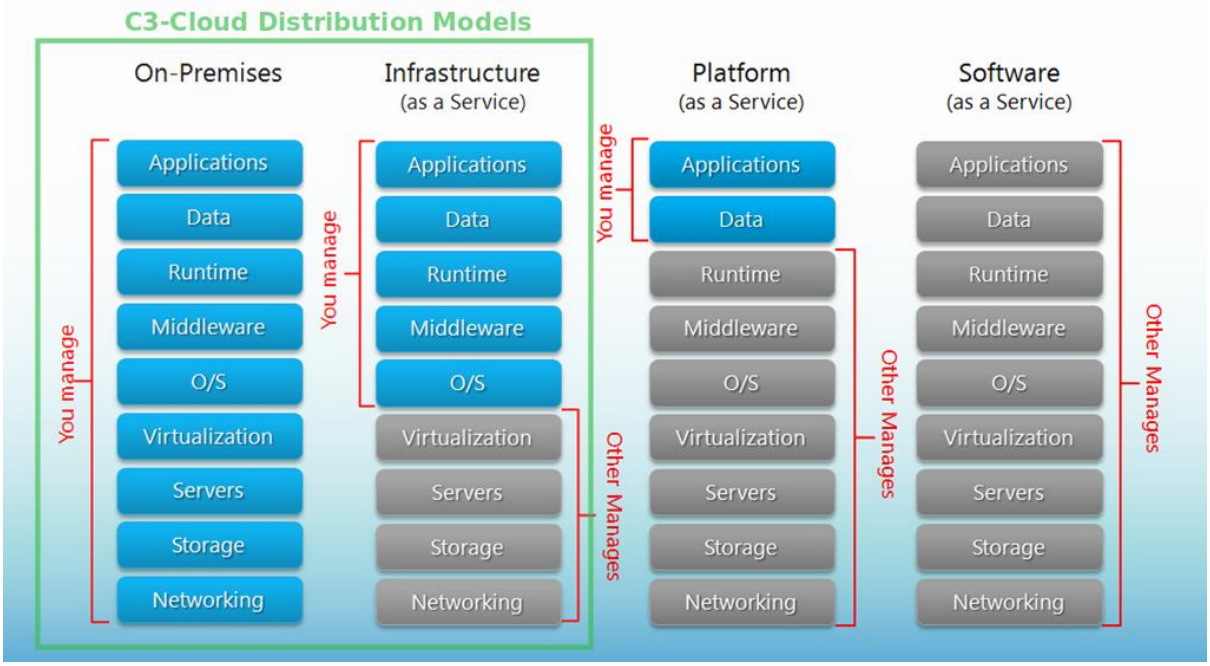

Figure 6: C3-CLOUD Distribution Models[vii]

On-premises is software located within the physical confines of an enterprise as opposed to running remotely on hosted servers or in the cloud. Infrastructure as a Service (IaaS) involves outsourcing the equipment used to support operations, including storage, hardware, servers and networking components. Platform as a Service (PaaS) is a paradigm for delivering operating systems and associated services over the Internet without downloads or installation. Software as a Service (SaaS) is a software distribution model in which applications are hosted by a vendor or service provider. They are made available to customers over a network, typically the Internet.

The two main options in our Project are On-premises or Infrastructure as a Service (IaaS).

1. **On-Premises**: In this model, the Pilot Sites can make available sufficient server resources to support the C3-Cloud Project. These would be locally managed by the Pilot Site and would include public facing web access to allow the services to be accessed by all users. C3-Cloud project resources would manage the application and would be provided with adequate access privileges in order to fulfil this task.
2. **Infrastructure as a Service (IaaS)**: This model refers to online services that abstract the user (technical administrator) from the details of infrastructure. Instead of having to purchase hardware outright, users (technical administrators in each Pilot Site) can purchase IaaS based on consumption (the same way as we contract our electricity supply or other utility billing).

In IaaS distribution model:

- Networking, storage, servers and virtualization are managed by IaaS providers
- Operating System (O/S), Middleware, Databases, Runtime, Data and applications must be managed by C3-Cloud project resources.

Another key issue on deployment decision is cost. Although **On-Premises** model could save some costs in physical infrastructure (hardware), other costs must be taken into account like proprietary software licenses. One aim of the deployment design is to identify hardware and software cost:

- **Hardware**: In case of **IaaS**, the fee of the IaaS provider. If **On-Premises** is used, this cost can be avoided if hardware already purchased can be used for C3-Cloud deployment.
- **Software**: C3-Cloud is based on Open Source software. However, there is a component (PEP) that requires Microsoft webserver and databases, which has a fee that must be taken into account for both **IaaS** and **On-Premises.**

The Pilot Sites will choose the Distribution Model that best meets their ICT strategies and available resources. The decision will be made by each site at a later phase of the project.

5. REFERENCES

[i] C3-Cloud Description of Action (DOA) of the C3-Cloud project

[ii] http://docs.oracle.com/cd/E19396-01/819-0058/dep_architect.html

[iii] C3-CLOUD Deliverables: D3.2, D3.3, D7.1, D8.1.

[iv] https://www.docker.com/

[v] http://www.haproxy.org/

[vi] http://searchcloudcomputing.techtarget.com/definition/SPI-model

[vii] https://yourdailytech.com/cloud-architecture/why-use-platform-as-a-service-paas/

6. APPENDIX: HIGH-LEVEL DEPLOYMENT DESIGN REQUIREMENTS

6.1. PEP

- Component name: Medixine Suite
- Short description: The system that serves as the foundation for the Patient Empowerment Platform functionality.
- Component responsible: Medixine
- Technologies
  - Server platform

    Server technologies: Virtualization platform, Operating System, Web Server, Application Server, Database Server, Database connector

    Medixine Suite Server Platform:

    - Microsoft Windows 2008 (R2) / 2012 (R2) Server
    - IIS 7.0/7.5/8.0/8.5
    - Microsoft .NET Framework 4.5.2
    - Microsoft SQL Server 2008 (R2) / 2012

    Please see document "**Medixine Technology Overview**" for additional details on the server architecture and technologies.

  - Client access

    Medixine Suite clients:

    - Medixine Suite can be used on all major browsers on both desktop and mobile devices.
    - No additional 3rd party components are needed.
    - Additionally some patients will use connected measurement devices to upload measurements to the Patient Empowerment Platform.
    - Scope of use. The client users connect from public Internet-facing devices.
- Security
  - Identity management and authentication

    Identity management is integrated. PEP Client Systems synchronize the details of eligible patients in the pilots as described in the D3.3 deliverable.

    Patient access user authentication is internal. Each pilot system is configured to use one of the available authentication methods in Medixine Suite.

    Medixine Suite supports basic username/password authentication, can be configured to use two-factor authentication (TOTP) or integrated to use external authentication providers. Authentication is based on Microsoft ASP.NET Identity (https://msdn.microsoft.com/en-us/library/mt173608(v=vs.108).aspx) which supports transferring data from other contexts using claims.

    Description of patient access user identification and authentication workflow:

    - PEP Client Systems components (mainly C3DP) synchronizes active care team, professional and patient details to PEP. No manual management of this information foreseen.
    - For each new patient, an invitation to register and access PEP is sent by email. An additional invitation code is shared by other means (most likely by SMS) to ensure that the user registering is the right person. Informal caregivers can

be invited in a similar fashion using available Medixine Suite functionality (available both via UI to users and via APIs to related components).
    - When the user opens the invitation link in a browser, the user can register and start using the PEP solution. The patient access user is authenticated as configured in the Medixine Suite system configuration.
  - Other

    Please see document "**Medixine Products and Services Security Overview**" for additional details on security-related topics.

- Interoperability

  List of related components or systems:

  - C3DP.
  - Email and SMS gateway for authentication and registration purposes and for notification and reminder delivery.

  List of exposed web services and technology used to implement them:

  - APIs for the C3 cloud integrated care plan purposes as decribed in T3.3 deliverable.
  - Medixine Suite product APIs.

  See **Medixine API documents** for additional details.

- Provisioning

  "Software packaging. Software deployment."

  The software can be packaged for manual deploy or it can be automatically deployed from a build server if infrastructure provides the required connectivity between the build server and application servers.

  Please see document "**Installation and Maintenance Guide - Medixine Suite**" for additional details on deployment of the Medixine Suite software.

6.2. TIS

- Component name: Technical interoperability suite (TIS).
- Short description: TIS (T6.1) establishes secure connection between the information systems in local care settings and C3-Cloud and enables data exchange between the two environments. The component, during the whole pilot period, continuously pulls data from the local care systems at each pilot site, transforms data into FHIR structure (with help from SIS), and pushes data into C3DP FHIR repository.
- Component role: C3DP, PEP, LCS, etc (as documented in WP3): TIS
- Component responsible: WARWICK
- Development platform: Java 8, Spring Boot, Kafka
- Technologies:
    - Virtualization platform: Support both Docker and popular Virtual Machines.
    - Operating System: Support both Linux and Windows. Linux (Centos or Ubuntu) is recommended.
    - Client platform: Supports all popular browsers, including Internet Explorer 9 and above.
    - Web Server: Tomcat 8.5 (embedded)
    - Application Server: Tomcat 8.5 + Spring 4.3
    - Database Server: H2 1.4 (embedded)
    - Database connector: H2 JDBC driver
    - Messaging Server: Kafka 0.10.2.1, Zookeeper 3.4.10
- Audit and Security:
    - Authentication: SPS
    - Scope of use: internal, public: Internal
    - Identity management: internal, external: Internal
    - Auditing requirements: SPS
    - Other security requirements: VPN to data publishing interface of local care systems
- N-tier architecture: Presentation layer, Businesses or Logic layer, Data access or Persistence layer.
- Presentation layer: Spring MVC RESTful services
- Business layer: Spring middleware
- Persistence layer: Kafka
- Input/output devices: None
- Interoperability:
    - List of related components or systems : C3DP, SIS, Local Care Systems
    - List of exposed web services and technology used to implement them: On-demand Update RESTful service: for clients (e.g., C3DP) to schedule an immediate patient data extraction. Implemented using Spring MVC. Monitoring and management RESTful services: a number of built-in HTTP endpoints by Spring Boot Actuator for IT administrator to monitor and manage the application. Custom endpoints could be added when requested.
- Provisioning:
    - Software packaging: Support Docker image, VM image, or zipped package (with service installation scripts)
    - Software deployment: Provide installation instructions and technical support.

6.3. SIS

- Component name: Semantic Interoperability Suite (SIS).
- Short description: The SIS provide a platform to both handle structural mappings among different information models and resolve semantic mismatches due to use of different terminology systems and different compositional aggregations to represent the same clinical concept.
- Component role: SIS
- Component responsible: Inserm
- Development platform: Java 8
- Technologies:
    - Virtualization platform : none used
    - Operating System : Linux (Debian is preferred) (recommended conf: 15GB RAM, 20GB HDD, CPUs: minimum of 12 cores)
    - Client platform: Web client for Server side administration
    - Web Server : Tomcat 8
    - Application Server
    - Database Server : NoSQL (embedded in tomcat environment)
    - Database connector
- Audit and Security:
    - Authentication
    - Scope of use: internal, public
    - Identity management: internal, external
    - Auditing requirements
    - Other security requirements
- N-tier architecture: Presentation layer, Businesses or Logic layer, Data access or Persistence layer.
- Input/output devices
- Interoperability:
    - List of related components or systems
    - List of exposed web services and technology used to implement them
- Provisioning:
    - Software packaging
    - Software deployment

6.4. SPS

- Component name: Security and Privacy Suite (SPS).
- Short description: SPS is responsible for guaranteeing authentication and authorisation of Care Team Members while they are managing personalised care plans and accessing sensitive personal data; and ensuring that all data exchange within and across C3-Cloud software components is encrypted and audited properly. C3-Cloud components utilizes its audit logging feature, and it provides a single-sign on mechanism to enable the care team members to use C3-Cloud applications by using a single account, the one that is already being used in local care system (when integration with local care sites' identity provider sites is possible).
- Component role: C3DP, PEP, LCS, etc (as documented in WP3): SPS
- Component responsible: SRDC
- Development platform: Scala, Java, Angular 4
- Technologies:
    - Virtualization platform: Docker
    - Operating System: Linux (Ubuntu is preferred) or Windows
    - Client platform: Web browser with HTML5 support
    - Web Server: nginx
    - Application Server: Akka HTTP, WildFly (for Keycloak)
    - Database Server: MongoDB, mySQL (for Keycloak)
    - Database connector: MongoDBScala driver, mySQL JDBC driver (for Keycloak)
- Audit and Security:
    - Authentication: Ideally through integration with the existing Identity Provider system(s) at the local site. In cases where this is not possible, an OpenID Connect compliant C3-Cloud Identity Provider will be provided.
    - Scope of use: internal, public
    - Identity management: internal, external
    - Auditing requirements: SPS has an Audit Record Repository (ARR) sub-component to keep C3-Cloud audit trails, especially those created by the C3DP.
    - Other security requirements:
    - Data transfer between C3DP and other components will be encrypted
- N-tier architecture: Presentation, Businesses or Logic, Data access or Persistence layers.
    - Complies with 3-tier architecture principles: Presentation, Business, and Data tiers.
- Input/output devices: Regular PCs.
- Interoperability:
    - List of related components or systems:
        - Local Identity Provider system(s)
        - Keycloak Open Source Identity and Access Management solution will be used for enabling integration with the existing user directories such as Active Directory and LDAP
        - C3-Cloud Coordinated Care and Cure Delivery Platform (C3DP)
    - List of exposed web services and technology used to implement them
        - HL7 FHIR STU3 RESTful API for AuditEvent resource: Totally compliant to STU3 3.0.1 specifications. Implemented with Scala and Akka HTTP.
        - OpenID Connect API: Implemented by Keycloak with Java.
        - Smart on FHIR App Authorization: Implemented with Scala and Akka HTTP.
        - Access Control Policy CRUD RESTful API: An API to manage access control policies, i.e. care team member permissions.Implemented with Scala and Akka HTTP.
- Provisioning:
    - Software packaging: SRDC will provide runnables and the necessary Docker images.
    - Software deployment: Deployment will be done at pilot site premises together with IT personnel of the pilot site.

6.5. CDSM

- Component name: Clinical Decision Support Modules
- Short description: Clinical Decision Support Modules combine information that is specific to an individual patient, with knowledge based on medical evidence, to provide guidance or advise on goals setting, activities to perform or on what is the best treatment for the patient.
- Component role: C3DP, PEP, LCS, etc (as documented in WP3): CDSM
- Component responsible: CAMBIO
- Development platform: Java 8, Dockter/Kubernetes ready
- Technologies:
  - Virtualization platform: Docker
  - Operating System: Linux (Ubuntu is preferred) or Windows
  - Client platform: Web browser with HTML5 support
  - Web Server: Spring-boot embedded jetty or tomcat server
  - Application Server: Spring-boot
  - Database Server: No database is needed
  - Database connector: NA
- Provisioning:
  - Software packaging: Docker images
  - Software deployment: Docker

6.6. PCPDP/C3DP

- Component name: Coordinated Care and Cure Delivery Platform (C3DP).
- Short description: C3DP facilitates collaborative management of care of patients with multi-morbid conditions. With the help of Clinical Decision Support Modules (CDSM), it provides care team members with the capability to define, update, reconcile and share care plans, and organize online meetings for care plan review. It also allows care team members to navigate a patient's medical history along with his/her care plan history.It should be noted that Personalised Care Plan Development Platform (PCPDP) that has been analysed as an individual component in the requirements analysis phase is indeed a sub-component of and deeply integrated with the C3DP. Therefore, in the architectural design phase, PCPDP was merged with C3DP.
- Component role: C3DP, PEP, LCS, etc (as documented in WP3): C3DP, PCPDP
- Component responsible: SRDC
- Development platform: Scala, Java, Node.js, Angular 4
- Technologies:
  - Virtualization platform: Docker
  - Operating System: Linux (Ubuntu is preferred) or Windows
  - Client platform: Web browser with HTML5 support
  - Web Server: nginx
  - Application Server: Akka HTTP
  - Database Server: MongoDB
  - Database connector: MongoDB Scala driver
- Audit and Security:
  - Authentication: Ideally through integration with the existing Identity Provider system(s) at the local site. In cases where this is not possible, an OpenID Connect compliant C3-Cloud Identity Provider will be provided.
  - Scope of use: internal
  - Identity management: internal
  - Auditing requirements: All kinds of access to patient data by care team members for care plan management will be audited via the Security and Privacy Suite (SPS).
  - Other security requirements:
    - C3DP will be OpenID Connect and hence OAuth2 compliant
    - Data transfer between C3DP and other components will be encrypted
    - Role-based access control was offered by SRDC but pilot sites declared that it is not necessary; hence all health professionals as care team members will have similar privileges
- N-tier architecture: Presentation layer, Businesses or Logic layer, Data access or Persistence layer.
  - Complies with 3-tier architecture principles: Presentation tier, Business tier, and Data tier.
- Input/output devices: Regular PCs.

- Interoperability:
  - List of related components or systems
    - C3-Cloud Clinical Decision Support Modules (CDSM)
    - C3-Cloud Patient Empowerment Platform (PEP)
    - C3-Cloud Technical Interoperability Suite (TIS)
    - C3-Cloud Semantic Interoperability Suite (TIS)
    - C3-Cloud Security and Privacy Suite (SPS)
    - External Teleconference System that is used at the pilot site
  - List of exposed web services and technology used to implement them:
    - HL7 FHIR STU3 RESTful API: Totally compliant to STU3 3.0.1 specifications. Implemented with Scala and Akka HTTP.
    - C3DP Event API: A RESTful API simple events like “care plan is read by the patient” or “the message is read by the patient” from the PEP. The decision is not finalized yet but it will most probably be implemented with Scala and Akka HTTP again. Alternative is Node.js + Express.
- Provisioning:
  - Software packaging: SRDC will provide runnables and the necessary Docker images.
  - Software deployment: Deployment will be done at pilot site premises together with IT personnel of the pilot site.