This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

An enhanced device-transparent real-time teleconsultation environment for radiologists

Christos Andrikos, Georgios Rassias, Panayiotis Tsanakas, Ilias Maglogiannis, *Senior member IEEE*

***Abstract*—This paper describes a novel web-based platform promoting real-time advanced teleconsultation services on medical imaging. Principles of heterogeneous Workflow Management Systems (WFMSs) and state-of-the-art technologies such as the microservices architectural pattern, peer-to-peer networking and the Single-Page Application (SPA) concept are combined to build a scalable and extensible platform to aid collaboration among geographically distributed healthcare professionals. The real time communication capabilities are based on the webRTC protocol to enable direct communication among clients. The paper discusses the conceptual and technical details of the system, emphasizing on its innovative elements.**

***Index Terms*—telemedicine, teleradiology, multidisciplinary teams, MDT, medical collaboration, real-time, telepresence, video-conferencing, video-teleconferencing, telementoring and surgery, trauma, follow-up, education, WebRTC**

I. Introduction

ONE of the most promising practices in healthcare is the deployment of Multi-Disciplinary Teams (MDTs) aiming to improve healthcare services, to augment preventive care and to utilize efficiently the available medical resources. Todays physicians acknowledge that the more they involve other disciplines in their practice, the more likely their patients will receive advanced health services. This holistic approach is gaining ground in the wide healthcare field, as it is currently being implemented not only in large clinics and hospitals but in private practices as well. Any required interaction among MDT members is achieved in the context of multidisciplinary team meetings (MDTMs) placing them in the center of our interest.

Although it is well established that the MDT management increases the quality of healthcare delivery, there are certain **technical barriers**. Poor infrastructure, geographic distribution of medical personnel or data as well as schedule conflicts prevent the complete exploitation of its potential. Small or regional hospitals usually are short of equipment or specialized healthcare personnel discouraging the MDT approach, while crowded metropolitan hospitals suffer from heavy scheduled personnel failing to promote productive MDTs. MDT efficiency is based on the asset of collaboration to enhance healthcare delivery through constant communication and interaction of the involved healthcare specialists [1]. In this context, the paper's contribution lies on the introduction of a novel and scalable Web 2.0 collaboration platform which features real-time tele-consultation services to enhance communication among caregivers. The novelty of the proposed platform is based on two main pillars: (a) technical: its extensible and scalable design opposing to alternative proposed solutions promises efficiency without sacrificing performance and (b) conceptual: its extensible paradigm paves the way for the development of collaborative workflows dedicated to healthcare professionals.

A concrete survey conducted on a total of 368 radiology professionals exhibited that 65% of them were using some kind of teleradiology [2]. The main usages were in-house image distribution (71%) and on-call readings from home (44%). The main benefits are improved collaboration with other radiologists (46%) and efficient distribution of workload (38%). Outsourcing is performed by 35% of the participants, while among them 68% noted that they use commercial services. The major advantages of outsourcing are availability of second opinions (82%) and additional capacity for on-call services (71%). The major disadvantages are insufficient integration of patient history and priors (69%), and limited communication with clinicians (68%). The majority expressed a positive opinion regarding the future of teleradiology (80%) predicting a growing importance (46%). Opportunities ought to be found in emergency reading services, flexible support of small practices and in collaborative platforms.

Lately, collaboration in the electronic health (eHealth) environment is focused on the cooperation of healthcare professionals belonging to the same organization via the Computer-Supported Cooperative Work (CSCW) paradigm as presented in [3] and Share4Health [4]. The majority of the existing infrastructures have been deployed as in-house, centralized systems featuring point-to-point communication between medical servers and clients. However, in practice, collaboration is proved to be an inherently challenging task imposing a variety of additional requirements, such as latency, flexibility, scalability, tolerance and robustness which are usually not met by the common centralized architectures. Typical screen-sharing or accessing a machine remotely means that only one person is in control while the other is only observing. Instant messaging, email, or other tools are great for primitive collaboration but collaborating on real medical data often requires more than an image or a single file to share the necessary context. Additionally, the validity of analysis can be affected by hardware capabilities, while trust is also of high importance, as privacy and security issues are essential in medical data handling. Thus solid data protection policies are required to cover strict rules like EU's impending GDPR

Chistos Andrikos, Georgios Rassias and Panayiotis Tsanakas are with the Greek Research & Technology Network (GRNET) and the School of Electrical and Computer Engineering, National Technical University of Athens, Greece, e-mail: candrikos@cslab.ece.ntua.gr, grassias@cslab.ece.ntua.gr and panag@cs.ntua.gr.

Ilias Maglogiannis is with University of Piraeus, e-mail: imaglo@unipi.gr.

regulation (https://www.eugdpr.org).

The success of any collaboration technology supporting any distributed medical team relies on the understanding of the human computer interaction entailed in the technology and a socio-technical approach in the design process [5] [6]. Any MDT requires an appropriate electronic collaboration infrastructure tailored to standardized work environments, so as to improve efficiency and productivity by tackling any barriers. In this context, we present a platform that provides remote situation awareness and shared artifacts exploring a variety of settings. By utilizing video teleconferencing techniques alongside a rich graphical user interface (GUI), the communication among different specialists is being enhanced to eliminate any MDTM limitations: **the meetings are easily integrated into the work schedule of specialists, while small or district hospitals need no special facilities for teleconferencing to support remote MDTMs**. The proposed platform may also be utilized in **teaching** medical students and junior doctors [7] or in supporting research groups to jointly solve related issues such as image-based co-registration [8] or co-segmentation [9] [10] [11].

The rest of this paper is organized as follows: Section II presents a survey of the literature and delves further into research motivations. Section III exhibits the system architecture and focuses on technical implementation details. Section IV evaluates the overall operation, user experience, performance, utility and usefulness, while Section V concludes exhibiting directions for further research.

II. Related Work and Background information

Collaboration services in medical imaging have a long history of systems and tools. Biomedical imaging researchers have produced various research papers along with a plethora of real world products and standards. In the beginning of 1980s, Digital Imaging and Communication in Medicine (DICOM) was developed by American College of Radiology (ACR), a standard for handling, storing, printing and transmitting information in medical imaging [12]. Since its introduction, DICOM has been extended and expanded to support new applications [13], and studied its potential use in novel use cases [14][15][16].

Due to the proliferation of the web, DICOM committees recognized the necessity of a web medical standard. They specified WADO (Web Access to DICOM Objects) service [17], so that system interaction could also take place over the web in a standardized way, allowing interoperability and proper information management within PACS (Picture Archiving and Communication Systems).

Apparently, such a coherent standard was able to facilitate the development of various biomedical imaging applications and to inspire the early "Internet of Things" (IoT) for medical devices. On one hand, in hospitals and clinics equipped with PACSs, images and reports are transmitted digitally via them, eliminating the need for someone to manually file, retrieve or transport film jackets. On the other hand, healthcare professionals started to use specialized software packages (DICOM viewers) such as HTML5 Zero Footprint DICOM viewer (the latest medical imaging product by LEADTOOLS) [18], Synapse 3D [19] by Fujifilm, OSIRIX [20] by the University of Geneva, Radiant or ImageJ [21] etc. A brief inspection of the long list of DICOM viewers [22][23] shows that rich functionality and suitable interfaces can be found in the commercial as well as in the open source domain of DICOM viewers software. However, each tool has its strengths and weaknesses; an all-rounder solution does not exist.

In the introductory Section I, the value of collaboration among healthcare professionals has been demonstrated. Furthermore, a survey of literature reveals two outstanding initiatives adopting early Internet technologies to support this claim, INTERMED and TeleMed. INTERMED was an early project designed to enhance the communication among patients and multidisciplinary health providers. INTERMED proved to be a reliable method for classifying patients' care needs, especially when used by experienced raters scoring by patient interview. It can be considered a useful tool in assessing patients' care needs as well as the level of needed adjustment between general and mental health service delivery [24][25]. TeleMed was a promising visualized, action-oriented decision-support tool, providing collaborative sessions and access to basic patient data to physicians [26]. However, the insufficient hardware, the poor connectivity infrastructure combined with the lack of mature software solutions, made any of those ventures almost inviable. The work in [27] presents the development of a web-based collaborative system for medical image analysis and diagnosis aiming to equalize healthcare delivery services among areas of higher and lower population density in Australia. The proposed platform was implemented in Java as a set of applets and Common Gateway Interface (CGI) scripts. In [28], a collaborative working environment for physicians is introduced, enabling peer-to-peer exchange of electronic health records based on the HL7 standard [29]. The paper treats technological issues such as video, audio and message communication, workspace management, distributed medical data management and exchange, while it emphasizes on the security issues arisen, due to the sensitive and private nature of the medical information. Authors of [30] depict a universal method to integrate computer-aided detection and diagnosis (CAD) results with PACSs to assist physicians in decision making process.

A medical image atlas is described in [31]. The researchers developed a platform allowing registered users to upload, visualize, process and comment medical images in a collaborative manner by using Web 2.0 Personalized Learning Environment tools for social learning and knowledge construction and sharing. In a similar manner, the work in [32] exhibits the way that a social network can be provided to physicians and patients, so as to foster cooperation and to overcome the problem of unavailability of doctors on site any time; patients can submit their medical images to be diagnosed and commented by several experts instantly. Another collaboration network intended to gather patients' medical images and physicians' annotations expressing their medical reviews and advices, is proposed in [33]. Authors in [34] present a potential solution on integrating grid computing and DICOM to feature advance medical imaging capabilities and security while researchers in

[35] introduce a virtual laboratory for medical image analysis; the platform is based on Dutch grid and adopts a service oriented architecture to decoupling the user-friendly clients running on the user's workstation from the complexity of the grid application. Finally, P2Care, motivated by real-world social groups, organizes healthcare professionals, providers and patients in groups according to common characteristics and interests [36] [37].

Last but not least, three related patents have been filed so far to further demonstrate the value of medical collaborative environments. The first one refers to a system that aggregates medical images via an uploading application; the system can further transmit the acquired images to the plurality of client applications to be displayed [38]. The second one presents a pure browser-based medical imaging system that requires no installation of application software or any browser plug-in and functions in the same way as traditional full blown medical imaging PACS viewer fat clients [39] [9], while the third patent depicts the idea of a web-based medical collaboration platform [40] between two browser-based clients.

Despite the long list, the surveyed applications found in literature do not meet the requirements of a holistic approach enhancing user experience. Medical image data exchange is a bandwidth intensive task, due to the size of imaging data as well as the demand for high QoS (Quality of Service) and minimum response times. Thus server-based architectures that assumes subsequent server operations to produce intermediate image processing tasks to be rendered by a client app, may introduce serious bottlenecks [28] that lead to poor user experience. Moreover, the majority of the existing systems are device dependent or highly coupled with the DICOM standard while they do not support collaborative creation of workflows over shared medical imaging data. Consequently, our contribution relies on the complete integration of both concepts —collaboration services and biomedical imaging— into a **web-based real-time platform** tackling all those issues and limitations, listed in Section I, that incommode collaboration among healthcare professionals. In previous works, we proposed an intelligent collaboration system fusing the principles of heterogeneous WFMSs, AI, existing electronic medical infrastructure (PACS, EMR) and standards (HL7, WADO) [41] [42]. In this one we focus on the conceptual and technical details of design and implementation process, delving deeper into the overall system architecture and the way it facilitates the provision of easy-to-use, transparent and interoperable online MDTM capabilities..

The presented platform leverages state-of-the-art web technologies (WebRTC, HTML5, CSS3) that provide rich and fluent communication (audio/video streams, etc.) and real-time interaction mechanisms combined with medical image processing functionalities. The motivation of the specific work comes from (a) the benefits for public health that stem from interconnecting MDT members in an efficient and transparent way, (b) the advantages of peer-to-peer networking and its suitability for developing real-time collaboration services as it features robustness, scalability, dynamic adapted system architectures and low-cost, (c) the maturity of the web technologies offering advanced programming capabilities to facilitate the deployment of image processing algorithms and (d) the lack of academic research as well as commercial solutions in the specific domain.

The main innovative features of the proposed platform are the exploitation of client-side computing (fat client) paradigm as well as the efficient cloud utilization; a Single Page Application (SPA) based on the newly introduced WebRTC standard, adopting the most recent web technologies, enables users to communicate, share and work on medical images simultaneously [43] while everything is orchestrated by a highly scalable ecosystem of fine-grained backend microservices. In this way clinicians are able to collaborate in real-time following intelligent pathways using a medical image processing toolbox on any device without any installation concerns. In addition, the proposed system is easily extensible to offer additional functionality and it does not suffer any serious burden of maintenance.

III. Proposed System architecture and Utilized Key Technologies

The overall architecture of the proposed systems is illustrated in Figure 1. More specifically, the proposed system involves three main modules to provide the real time teleconsultation services to radiologists: User Access Control (UAC), User Management (UM) and Content Management (CM). UAC involves user authentication and authorization, either by interacting with some kind of private data structures or by negotiating with some third party Authentication and Authorization Infrastructure (AAI). UM enumerates securely all registered user profiles as well as manages any social activity of theirs (collaborative circle, networking, friend requests, etc.), while CM defines the means and the policies behind content delivery. The whole system features strict security requirements due to highly sensitive medical data manipulation and exchange, thus it leverages encryption protocols in all communication stages. The following paragraphs describe the main components of the Micro-architecture.

A. Micro-architecture

1) Microservices: To start explaining the microservice style and the reason why it is adopted in this work a comparison to the monolith paradigm is useful. A monolithic application built as a single unit, which is a single logical executable. Enterprise Applications are often built in a monolithic way. In that case, horizontal scalability involves scaling the monolith by running many instances behind a load-balancer, a fact that may affect seriously the efficiency of any cloud infrastructure and increase the cost of maintenance. On the opposite direction, *microservices* represents a specialization of service-oriented architectures (SOA) used to build flexible, independently deployable software systems. Services in a microservice architecture (MSA) are processes that communicate with each other over a network in order to fulfill a goal. On top of decentralizing decisions about conceptual models, microservices also **decentralize data storage**. At the most abstract level, it means that the conceptual model of the world

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

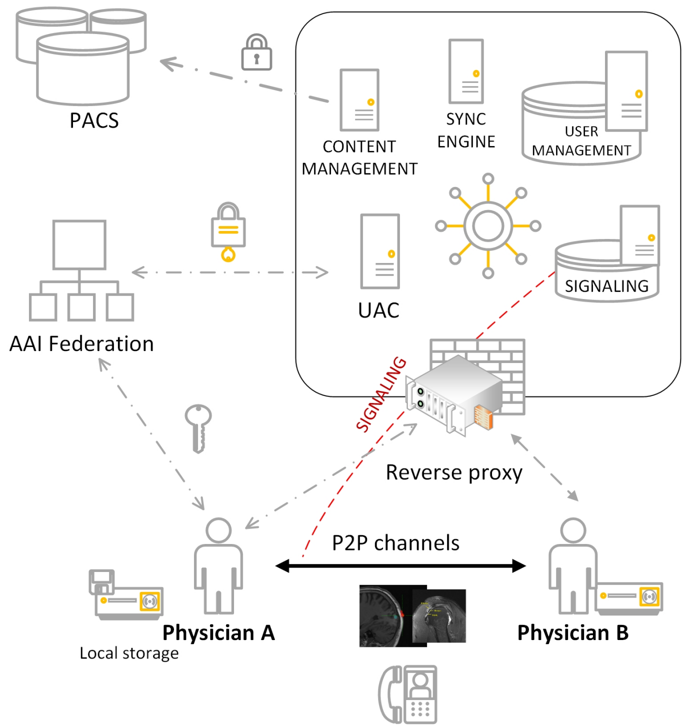


Fig. 1: Overall system architecture

will differ between systems or even subsystems of the same system. Microservices prefer letting **each service manage its own database** —either different instances of the same database technology or entirely different database systems, an approach called *polyglot persistence* (see Figure 3a). In addition, **just-in-time scalability** is facilitated [44], since there is no more need for scaling the entire application but only the services that require more resources (Figure 3b).

2) Fine-grained backend services: Splitting the monolith's components into services eliminates any programming language, software engineering paradigm or database system obligation. The developer can tackle any sub-challenge — MSA is the adoption of divide and conquer principle in the field of software design— utilizing the tools and technologies that fit best. Such heterogeneity can contribute further to the overall system performance but not always at zero cost. Remote calls are more expensive than in-process calls, and thus remote APIs need to be more coarse-grained, which is often more awkward to use. If someone needs to change the allocation of responsibilities between components, such shifts of behavior are harder to do when crossing process boundaries.

TABLE I: Architecture components and their short description

Component	Short description
EDGE SVC	Reverse proxying, dynamic routing, resilience and basic security
CRED SVC	User registration, authorization handling, token issuance
FRONTEND SVC	UI views and SPA serving
SOCIAL API	User and contact management
WEBSOCKETS API	Message passing, WebRTC signaling, real-time features support
Metadata Harvester	GRNET's AAI metadata aggregator
MONGODB-Store	Various instances as backing stores of the rest microservices
Orthanc	Open-source DICOM server with REST APIs [45]

Figure 2 depicts the plain system architecture following the MSA paradigm. The backend features a **network of cooperating autonomous services** (see also Table I). The whole network resides into a Demilitarized Zone (DMZ) — all services communicate through a private network and they are not directly accessible over public Internet— in a two-tier fashion architecture. The first layer consists of all the services that constitute the provided API, while the second layer comprises "rich" database management systems, which autonomous services providing database and storing functionalities through REST APIs.

EDGE SVC is the front gate for all the requests to the backend of the introduced system. It is built to provide dynamic routing, resilience and security. Featuring a rich and divergent API, it interacts with the corresponding services to fulfill any incoming request. Moreover, it assigns a unique ID to every autonomous incoming request or to any group of requests that constitute a more complex operation. This unique identification of requests is vital for monitoring due to the distributed fashion of MSA. Last but not least, the core EDGE SVC manages cookies, sessions as well as interacts with third-party AAI (Authentication and Authorization Infrastructure) to offer a flexible and robust AAA (Authentication, Authorization and Accounting) scheme.

FRONT-END SVC delivers content to the user such as the client application as well as any supplementary static assets (i.e. HTML pages, CSS files, etc.) while **CRED SVC** is responsible for the management of the registered users's credentials. Registration, notification e-mails or any further update of registration data is been provided via that service.

SOCIAL API features the social networking infrastructure based on GraphQL. GraphQL is a novel query language introduced by Facebook for managing data efficiently in a transparent fashion. The service implements a runtime for executing queries and mutations by using a well-defined type system for data. GraphQL isn't tied to any specific database or storage engine, queries and mutations accompanied by use regulations are defined and can be applied to any database system.

WEBSOCKETS API is the core of every real-time feature of the provided platform. The service hosts a WebSockets connection coming from each online user. In this manner, the backend can asynchronously interact with any client, in order to facilitate status discovery, notifications, instant messaging and signaling, they are vital for driving remote conferencing sessions. The service depends on a Redis key value store to store any temporal data before archiving them to the corresponding persistent data stores through SOCIAL API.

Metadata Harvester is dedicated to aggregate all the necessary SAML (Security Assertion Markup Language) metadata [46] coming from the third-party AAI federation. In this way the system is aware of the members of the federation and thus users are able to be authenticated via an XML-based SSO (Single Sign-On) scheme.

MONGODB-Store hosts both registration and social

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

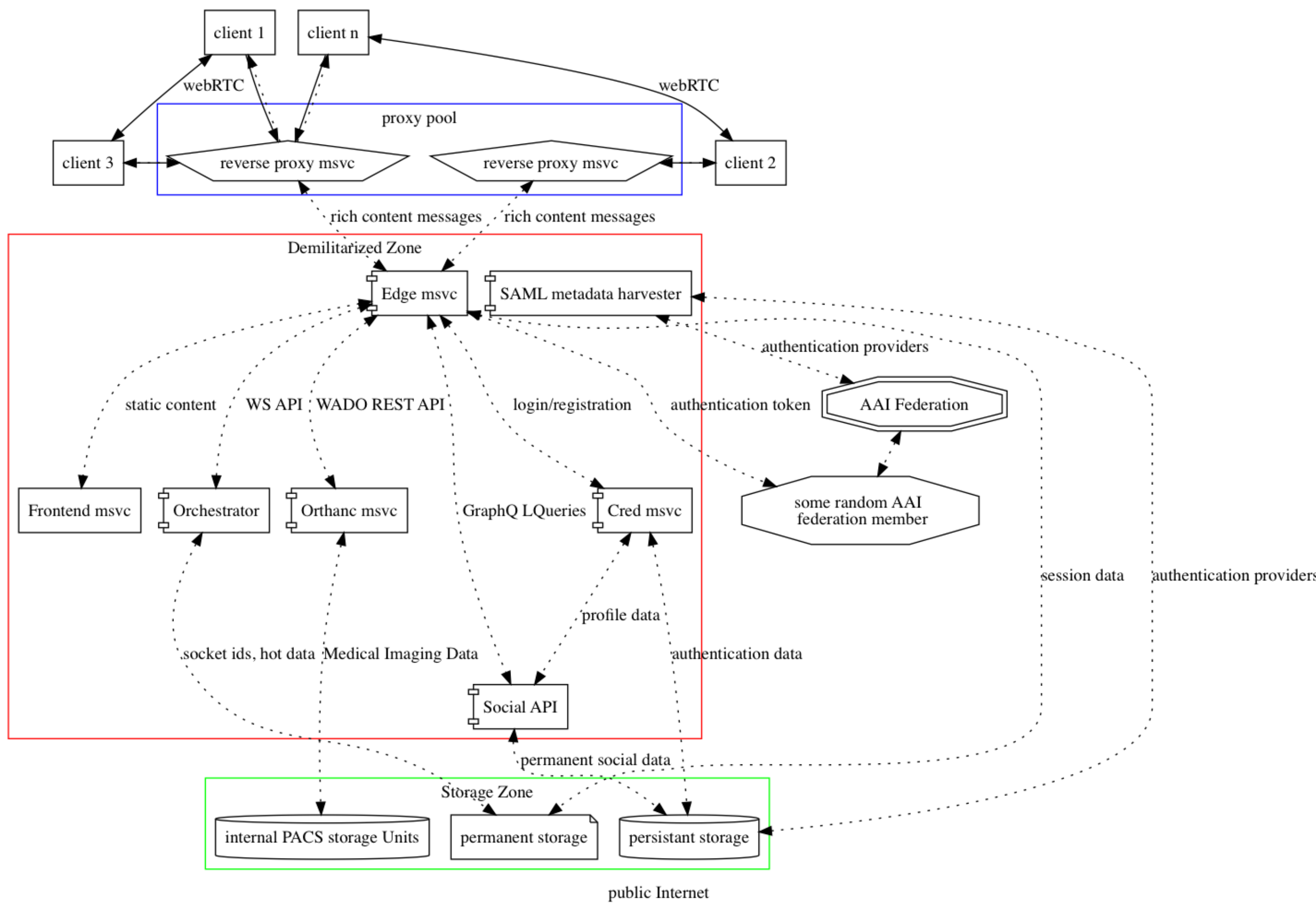


Fig. 2: System micro architecture

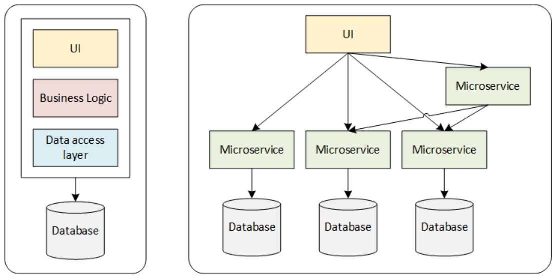
databases while a **PostgreSQL DB** powering an Orthanc instance [45], a free and open-source DICOM server whose source code is available to the benefit of hospitals and researchers, and they are combined to mimic a fully functional PACS system. Decentralizing responsibility for data across microservices leads to more flexible and performant system but it introduces implications for managing updates.

3) Design for failure, deployment automation: Each one of the presented services is "containerized" —an OS-level lightweight visualization technique for deploying and running distributed applications without launching an entire VM for each one. Thus, multiple isolated systems, called containers (i.e. **Docker Containers** (https://www.docker.com/what-docker)), can be run on a single control host relieved from the burden of hardware virtualization, and access a single kernel that is shared with the host itself. Consequently, hosts treat the provided services as isolated internal processes resulting to a more efficient scheme regarding to memory, CPU and storage. Any service call could fail due to unavailability of the supplier; the client has to respond to this as gracefully as possible to reduce failure impact to the user experience.

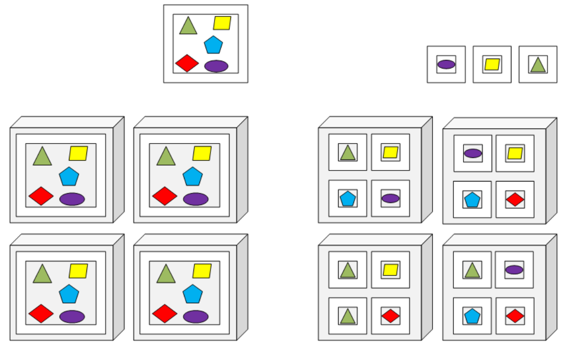
The proposed system deals with that issue through replication of services and smart endpoint deployment over SDN (Software Defined networks). Each container is replicated and every replica participates in a private virtual network that spreads out on the entire cluster of hosts. Replicas of the same service reside behind smart endpoints facilitating ad-hoc discovery of replicated services in order for the latter to be reachable in a transparent way. Thus the proposed architecture ensures high availability and network partitioning tolerance. The lifecycle and replication factor of each microservice is managed by a newly introduced master-slave orchestration system by Google, Kubernetes (K8s). In kubernetes deployment plans and networking rules are defined in a plain declarative way while a description of the state of the entire cluster is backed to an ETCD key-value store to provide failure recovery. As containers can be created and terminated much faster than hypervisor-based instances, our deployment approach is much more agile than the typical hypervisor-based ones.

4) Inter-service communication: Concerning the building of communication structures between different processes, the microservice community favors the approach of **smart endpoints and dumb pipes**. Applications built from microservices aim to be as decoupled and as cohesive as possible —they own their own domain logic and act more as filters in the classical Unix sense— receiving a request, applying logic as appropriate and producing a response. These are usually choreographed using simple REST protocols rather than complex protocols, such as WS-Choreography or BPEL or orchestration by some central tool. The two protocols used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics



(a) Polyglot persistence



(b) Coarse vs fine-grained scaling, monoliths vs microservices scaling

Fig. 3: Visual representation of differences between Monolith and Microservices paradigm

most commonly are HTTP request-response with resource APIs and lightweight messaging [47].

The second approach in common use is messaging over a lightweight message bus. The infrastructure chosen is typically dumb (dumb as in acts as a message router only) —simple implementations such as RabbitMQ or ZeroMQ do not do much more than provide a reliable asynchronous fabric– the smarts still live in the endpoints that are producing and consuming messages, i.e. the services. In a monolith, the components are executing in-process and communication between them is via either method invocation or function call. The biggest issue in changing a monolith into microservices lies in changing the communication pattern. A naive conversion from in-memory method calls to RPC leads to chatty communications which do not perform well. Instead you need to replace the fine-grained communication with a more coarse-grained approach.

In the proposed system, inter-service communication is based on the REST approach which offers many benefits, such as development simplicity, rapid prototyping, flexibility in scaling and independence from implementation language and underline platform. Nevertheless, as someone might expect, REST/HTTP is not a silver bullet; there are scenarios that it simply does not fit in. Such a case is the intrinsic asynchronicity regarding instances of the same service: two users are connected to two different instances of the WebSockets service and one of them is trying to send a message to the other one. In that case, every instance is not aware of the users connected to the others, so there is not an obvious way to relay the messages through some REST API call. The solution to this problem is supplied by the use of the aforementioned message brokers/buses that implement well-known and useful patterns for these purposes, like Pub-Sub. According to Publish/Subscribe messaging paradigm, senders (publishers) are not programmed to send their messages to specific receivers (subscribers). Rather, published messages are characterized into channels, without knowledge of what (if any) subscribers there may be. Subscribers express interest in one or more channels, and only receive messages that are of interest, without knowledge of what (if any) publishers there are. This decoupling of publishers and subscribers can allow for greater scalability and a more dynamic network topology. In our case, Redis provides such kind of functionality.

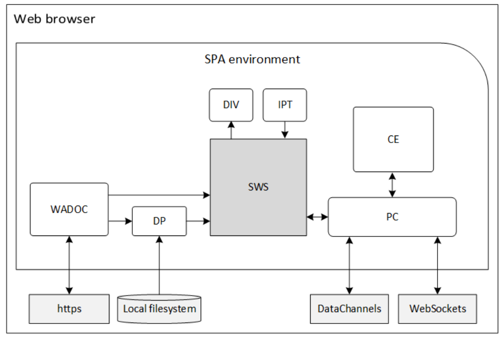
5) Logging and Monitoring: Software solutions based on microservices architectural pattern involve increased inter-service communication, which should not be taken always for granted. As a result, algorithms and well-known workflows are altered to groups of asynchronous processes. Therefore block diagrams should be substituted by sequential diagrams embracing any potential communication failure.

In order to efficiently monitor the operation of the platform we ended up with the Elasticsearch, Logstash and Kibana (ELK) stack [48][49]. Elasticsearch is a search server based on Apache Lucene; it provides a distributed, multitenant capable full-text search engine with a RESTful web interface and schema-free JSON documents. Logstash is an open source, server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it and then sends it to some "stash" (in our case Elasticsearch), while Kibana is a rich visualization tool consuming the Elasticsearch API. In this fashion, every node of the constituted Kubernetes cluster has been configured to monitor any deployed containers as well as a bunch of individual process that are considered vital for the operation of the cluster. All data logs are aggregated in a dedicated private server running Elasticsearch and Kibana to be further analyzed.

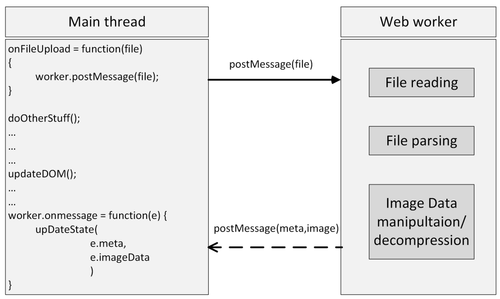
B. Single Page Application - SPA

The SPA provides a sound DICOM viewer. It supports reading and displaying a wide-range of single-frame and multi-frame DICOM files, encoded according to most common transfer syntaxes. Interaction with PACS via WADO (Web Access DICOM Objects) as well as local storage devices in conjunction with smart series integration (synthesis from distinct DICOM files) and browsing is also supported. Basic image processing (e.g. zooming and panning, rotation and flips, brightness and contrast adjustment, preset window level setting for Computed Tomography and Magnetic Resonance modalities, etc.) filtering and ROI (Region Of Interest) selection are supported to enhance image viewing.

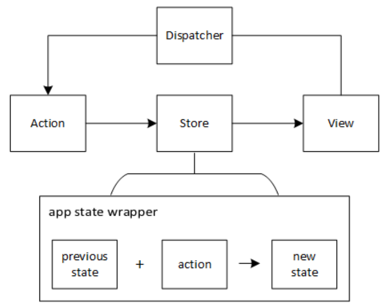
However, the **basic innovation** of the SPA itself is the real-time collaboration feature based on multi-channel video conferencing, workspace and file sharing, chat rooms and



(a) The internal architecture of SPA.



(b) Parsing DICOM files concurrently.



(c) SPA state updates.

Fig. 4: SPA internal organization.

shared operations and annotations. While typical implementations of similar collaboration environments involve centralized architectures, we introduce a pure peer-to-peer interconnection scheme to eliminate any intermediate party overheads and handle scalability issues. The communication mechanism relies absolutely on the WebRTC protocol, which is a free, open source project that provides browsers with secure (encryption embedded) RTC (Real-Time Communication) capability. Implementation details are provided in the following paragraphs.

1) Client Application & Implementation Details: The introduced software consists of seven autonomous entities: WADO Connector (WADOC), DICOM Parser (DP), DICOM Image Viewer (DIV), Image Processing Toolkit (IPT), File Sharing Engine (FSE), Conferencing Engine (CE) and PeerConnector, as depicted in Figure 4a. The whole SPA is implemented in HTML5, CSS3 and JavaScript. The User Interface (UI) has been built in a reactive manner [50] exploiting the declarative approach and one-way data flow of **ReactJS**, an efficient, and flexible JavaScript framework for building user interfaces. By enforcing a purely functional programming style, the classic MVC (Model-View-Controller) paradigm is upgraded to support explicit modeling of complex time-dependent relationships in a high-level declarative way.

Views are the declarative React components which are responsible for the UI layout that is being presented to end users through composition, while **actions** are plain JS objects with a type property that indicates the type of action being performed. **Dispatcher** is a special component/function that produces the actions as instructed by the registered event handlers of views. Finally, the **store** is essentially a wrapper around the whole application state, implemented by a single JS object; it ensures the latter's **immutability** by deterministically calculating and saving the new state based on the previous one and a single dispatched action. The SPA consists of views that are organized in a hierarchical way. High-order views get notified whenever a new state is produced; they consequently propagate parts of that state to their child views which update themselves accordingly in order to reflect the changes.

This structure allows us to **reason easily about our SPA** in a way that is reminiscent of functional reactive programming, or more specifically data-flow programming or flow-based programming, where data flows through the application in a single direction —there are no two-way bindings. The different parts of the application remain **highly decoupled**. The two-way data bindings originating from the heavy use of **Backbone.js** and its core dependency, jQuery, often resulted in cascading and subtle updates that were difficult to reason about. Now, updates due to user interaction can only change data within a single round and thus the system as a whole becomes more predictable.

Shared Workspace State (SWS) is the most important component of the SPA. It comprises only a part of the whole application state but it deserves special reference as it needs to be kept synchronized across client instances for all the duration of a remote collaboration session. In other words, as long as a call is being carried out, SWS essentially becomes shared state with eventual consistency semantics. SWS holds all the information from the loaded DICOM files —images and metadata, as well as all the user applied filters and annotations, in order for the connected clients to be able to render the same output every moment. The specifics of the synchronicity mechanisms utilized by SWS are examined below (subsection III-C).

WADO Connector (WADOC) is the logical part of the

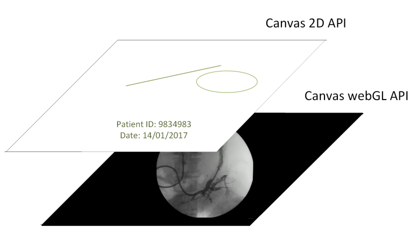
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

application which is responsible for requesting and receiving DICOM persistent objects residing in remote PACS through HTTPS, based on DICOM UIDs. It relies on the WADO standard, which is part of the DICOM specification. Data can be retrieved either in a presentation-ready form as specified by the requestor (i.e. JPEG or GIF) or in a native DICOM format.

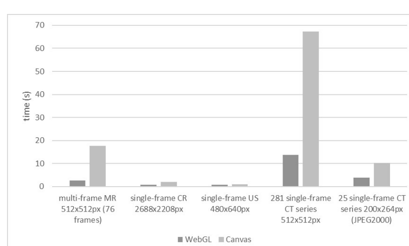
DP (DICOM Parser) is the JavaScript module that handles the parsing of DICOM files and the extraction of the image data and its associated metadata. It is designed to utilize modern multi-core architectures efficiently and reduce memory footprint. Typically, in common server-based architectures DICOM parsing would have taken place server-side. Nevertheless, we hand off that parsing process to the client to reduce network traffic and improve application responsiveness. Therefore, the client-side component (DP), written in JavaScript, is becoming larger and more compute-intensive, increasing the demand for high performance JavaScript execution. Client-side computation helped to improve the performance of JavaScript engines in the web browsers. Furthermore, considering the widespread deployment of multicores in modern computing systems; exploiting parallelism in these applications is a promising approach to meet their performance requirement [51]. However, JavaScript has traditionally been treated as a sequential language with no support for multithreading, limiting its potential to make use of the extra computing power in multicore systems. Therefore, the SPA utilizes Web Workers, a Web API that provides a simple means for web content to run scripts in background threads. The worker thread can perform tasks without affecting the user interface. Once created, a worker can send messages to the JavaScript code that created it by posting messages to an event handler specified by that code (and vice versa). In this manner, DP resides in a web worker, main thread posts files to be parsed and DP responds with a similar post message containing all the extracted data (Figure 4b); obviously, the performance of the main thread is not affected and hence SPA remains always responsive.

DICOM Image Viewer (DIV) is responsible for displaying medical images, while Image Processing Toolbox (IPT) provides key functionality for image manipulation and processing; the combination of these two modules produces a fully-fledged web-based DICOM viewer which is the core unit of the introduced SPA. User experience has been highly prioritized while designing and implementing the specific modules. On one hand HTML5 comes with the canvas element which supports drawing graphics via scripting in JavaScript and while utilizing GPU for rasterization. However, in our case, canvas paradigm suffers some serious performance degradation; any image processing operation involves (1) transferring all pixel data of an image from GPU memory to RAM, (2) executing some JavaScript routine to transform a proportion of that pixel data, and finally, (3) summoning pixel data back to GPU memory. In addition, rendering the content of DICOM files utilizes the canvas-specific APIs, a process that requires special handling of the underline pixel data. That, in many cases, should lead to user experience degradation. For instance, loading DICOM series of multiple instances involve excessive CPU computations resulting in unresponsiveness. To tackle these serious performance issues, a hybrid architecture has been adopted (see Figure 5a). By stacking up two canvas elements, we achieved the performance of pure WebGL without sacrificing the utilization of HTML5 canvas element for tasks that WebGL lacks sufficient programming capabilities, such as text rendering, etc. The bottom layer supports medical image rendering and post-processing by delegating image processing computations directly to the GPU using GLSL routines, while the upper one facilitates text features and annotation capabilities. Figure 5b depicts the performance increment as the decrease of a complete loading cycle (UI latency). The latter is defined (in the scope of the paper) as the overall time required from the selection of a DICOM file by the user until the completion of loading all of its content into the SPA. DICOM content loading requires excessive computation compared with any other visual update. Consequently, it can be considered the bottleneck of each UI update operation and this is the reason why it is defined as the maximum UI latency. Note that time has been asserted in millisecond level accuracy by specialized low-level monitoring routines. Obviously UI latency decreases drastically by the adoption of the aforementioned hybrid architecture.

PeerConnector (PC) is a library developed to harness the power of the latest Web APIs offered by most major web browsers, like WebSockets and WebRTC-related (i.e. RTCPeerConnection, RTCDataChannel). All these relative new standards offer alternatives for the traditional client-server communication or add new functionalities formerly met only in custom standalone solutions, like peer-to-peer



(a) Architecture layout in practice.



(b) Hybrid vs Canvas-based architecture performance. Diagram presents the overall load time of a DICOM file.

Fig. 5: Hybrid architecture of DIV.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics


```
{
  "peerID": "ho234jf34jfsdfl_43kf",
  "seriesInstanceUID": "1.3.12.2.1107.5.99.2.5562.4.0.575080411334689",
  "left": 458,
  "top": 317,
  "angle": 270,
  "scale": 1,
  "flipX": false,
  "flipY": false,
  "windowCenter": 40,
  "windowWidth": 400,
  "type": "SeriesRendererModel",
  "negativeFilter": false
}
```


(a) A JSON example of what was transmitted in our previous work.


```
{
  "peerID": "ho234jf34jfsdfl_43kf",
  "seriesInstanceUID": "1.3.12.2.1107.5.99.2.5562.4.0.575080411334689",
  "type": "SERIES_MOVE",
  "left": 458
}
```


(b) A JSON example of what is transmitted in our current implementation.

Fig. 6: The decrease of transmitted bytes increases performance and reduces network bandwidth utilization.

communication. More specifically, PC utilizes WebSockets for client availability detection, chat messages relaying and call signaling –since WebRTC as a standard does not cover this area. The media negotiations and direct session establishment among clients are achieved through the use of the RTCPeerConnection API, whereas the peer-to-peer data transfers (for DICOM files and JSON messages representing user actions) are offered by the RTCDataChannel API. PC manages to abstract away all the quirks of this variety of technologies that are implemented differently among browsers and exposes a simple and intuitive API to the higher-level component providing the social functionality of the SPA, i.e. the Conference Engine (CE). The latter manages all the views enabling user communication in real time, including the contact panel, chat panel, video elements, user and call notifications. The developed SPA may run on a variety of different devices, following the Bring Your Own Device (BYOD) trend.

C. Synchronization and concurrency control

The real-time medical collaboration environment deals with user interactions, network I/O, as well as synchronization and data consistency among clients. This kind of functionality is referred to as *groupware*. One of the most studied issues, in this type of applications is the concurrency control problem. Traditional concurrency control methods cannot be applied directly to groupware because system interactions include people as well as computers [52]. For that reason, the well known straight-forward solutions, i.e. locks to parts of the common viewport, would result to degraded user experience.

Interaction between distributed sites can be thought of as the exchange of messages or execution events. An event goes through a number of stages in its existence, usually something like: creation, local execution, transmission, reception and remote execution [53]. The problem arises from the fact that this is a multi-way process, so without proper concurrency control, events could be executed in different order at different sites, which could lead to inconsistencies. In [42] we had mentioned as a possible solution the use of a coordinator-based protocol [54], with the coordinator being a dedicated site (i.e. Firebase) or a site elected arbitrarily from the collaborating clients. Under that protocol, when multiple concurrent and overlapping operations are generated, the operation first arriving the coordinator will win —in other words, the coordinator will re-arrange the order of operations resolving in that way any conflicts.

In this work, we present a slightly different approach regarding the coordinator entity. This time the role of the coordinator is assigned to the WebSockets service, as all actions regarding user interactions are now relayed through WebSockets channels. The change from the previous RTCDataChannel-based implementation was made explicitly because we did not want to overburden the PeerConnector and, in addition, we would like to examine the possibility of storing user actions and associating them with the corresponding DICOM objects (see Section V for details). Finally, a rollback functionality has been implemented allowing clients to *atomically* undo any arbitrary series of actions, in order to eliminate any inconsistencies.

IV. The System in Practice

A. Distributed client state

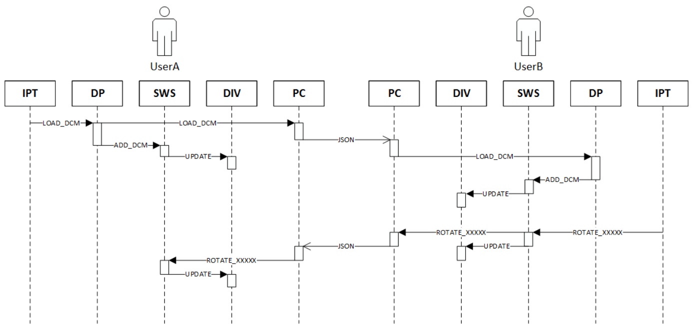
The developed platform is called HERMES and it is publicly available at `https://hermes.grnet.gr/`. In a previous approach [42], we described a state representation and implementation based on dynamic data structures conforming to DICOM hierarchy that resided in client memory.

Figure 6a depicts a JSON containing instructions of how to render a particular series. This is an indicative example of what is used to transfer back and forth between clients in an active session. In the described implementation, the size of the transferred data is reduced drastically (Figure 6b). This is achieved through the radical redesign of the client local state representation based on immutability rules. In this context, every possible user action over the application data, is assigned with a unique name (string of characters) and then it is describe (in code) how that action, when fired, produces a new state. The action representation itself, being a plain JavaScript object, contains the least amount of information required to capture the special meaning of that action and, due to the collaborative nature of the application, it is the only thing that has to be transferred (see Figure 7a). At this point, it should be mentioned that the advantages in addition to the reduced bandwidth utilization extend to reduced CPU utilization, since less data to be serialized/deserialized.

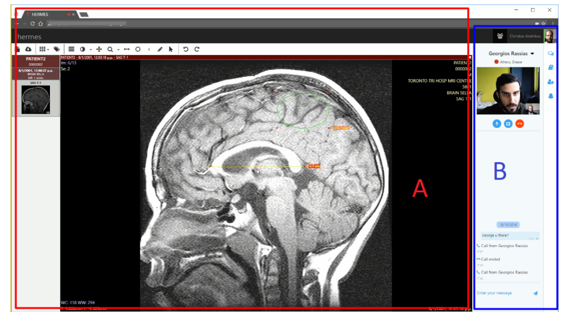
B. User interface & visualization

A significant objective of this work is the provision of visualization tools that are intuitive and can be easily used by the physicians [55]. Browsing across multiple DICOM

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics



(a) Interaction of client modules.



(b) SPA screenshot with highlighted subpanels.

Fig. 7: Client experience.

sources, aggregating and displaying the evidence required in making informed decisions and reusing the retrieved results to address challenges in different contexts, are aspects of the utmost importance.

To facilitate the intuitive exploration of the information available in any DICOM source, while enhancing the collaboration process among MDTM members, we present a User Interface (UI) comprising various components enabling real-time image processing. The resulting workspace conforms to the basic design guidelines dictated by major medical imaging software solutions (i.e. RadiAnt, OsiriX).

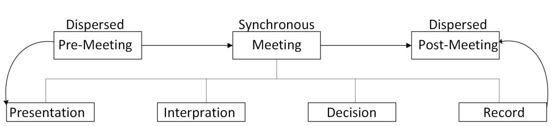
Figure 7b exhibits an indicative screenshot of the introduced SPA demonstrating the aforementioned assets. Figure 7b-A depicts the **medical imaging panel**, the core of our SPA, since it hosts all the key functionality required for medical image viewing. A toolbar facilitates the access of DICOM files or directories and enables user interaction with the loaded images. At the left-side column, there are previews of all the opened series organized according to patient and study unique identifiers. A user can navigate to the desired series and display the corresponding images, by clicking the appropriate thumbnail view, while the actual rendering takes place in the adjacent interactive viewport.

Figure 7b-B presents the **"social" panel** where all social widgets reside to support remote MDTM and enhance collaboration among healthcare professionals. It is divided into two subpanels. The first one dynamically renders information depending on the user selection in the adjacent second one. So, it might contain all user contacts highlighted according to their availability status, the grouped history of user past interactions with others, screen for searching and adding new people, or a window for direct interaction with a specific remote user. The latter combines the video elements that output the local and remote video streams, when available, and the chat history.

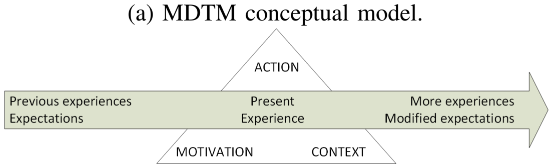
C. Use case evaluation and User experience

In this subsection we present the establishment of a collaboration session. Consider two potential users: physicianA and physicianB as illustrated in Figure 1. Upon login, physicianA acquires an authentication token; by triggering UAC to request a digital signature from a third trusted party. User management service returns to physicianA the manifest of all online participants matching to his contact list. After physicianB logins, the same service updates asynchronously the contact list of physicianA, while the latter may access the supported PACS service of his choice to retrieve and process any medical data. A conference session between the two physicians takes place after the initiator creates a new room. Initial negotiations and further signaling are propagated to participants through the WebSockets API, to result to symmetric WebRTC data channels set up.

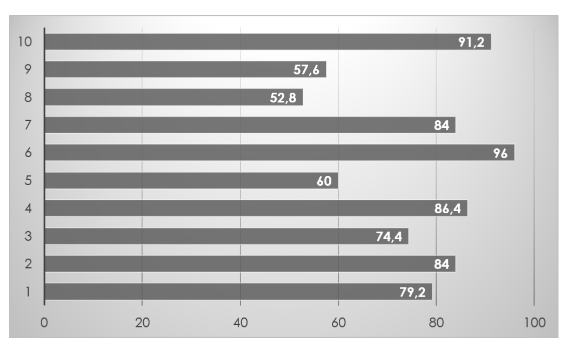
Figure 8 depicts the graphical representation of user experience conceptual model. User actions are based on previous experiences and expectations to form novel experiences while re-adjusting future expectations. As a consequence, enhancing user experience equates reinforcing the amplifying character of



(a) MDTM conceptual model.



(b) User experience conceptual model.



(c) SUS evaluation.

Fig. 8: User experience evaluation by ten random physicians.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

TABLE II: Related work, brief feature comparison. **Ref** column holds the number of the corresponding citation.

Ref	audio-visual communication	real time workspace sharing	end-to-end encryption	P2P connections	image processing toolkit	Integration with PACS	3D reconstruction	Session Recording	BYOD - installation free	Social Networking aspects
[20]					✓		✓			
[21]					✓		✓			
[18]					✓	✓	✓		✓	
[19]					✓	✓	✓		✓	
[27]		✓			✓	✓				
[28]	✓	✓			✓					
[31]	✓				✓			✓		✓
[38]	✓	✓	✓	✓	✓	✓		✓	✓	
[56]					✓	✓	✓	✓	✓	
HERMES	✓	✓	✓	✓	✓	✓	✓			✓

that model. Guided by this concept we applied a novel design resulting to an agile and efficient architecture; any further API extension, forming a distinct microservice, will not affect the provided functionality neither the performance of the platform and thus user experience will not decline. Furthermore, fine-graining the backend infrastructure while maintaining low overall complexity (i.e. microservices interact each other as less as possible) eliminates any design bottlenecks.

The provided architecture in conjunction with the SPA ergonomic design ensures that (i) increased workloads can be easily handled by dynamically spawned service instances, (ii) further functionality extension could not affect the performance and responsiveness of the platform.

Our claim about user satisfaction is not untenable; rather it is backed by the results of a SUS questionnaire [57] that has been filled by ten random physicians, after they engaged in a collaboration session with one of the members of the development team. They were asked to evaluate certain aspects of the platform related with the user-friendliness, the ease of learning, the provided feature integration and observed inconsistencies, if any. The results, depicted in Figure 8c, are highly positive and motivational, especially if someone takes into account the alpha release that physicians were given access to —well above 68 that is considered as the baseline and pretty close to 80, above which people are expected to recommend it to their friends or colleagues.

V. Conclusion and Future Work

In this work, the design, implementation and deployment of a coherent and agile CSCW environment for teleconsultations over medical images are presented. The proposed system enhances communication among different specialists and overcomes MDTM barriers in a robust way. Table II provides a qualitative assertion of the system's functionality compared with the related high-valued works found in literature. The meetings are now easily integrated into the work schedule of specialists, while small or regional hospitals need no special facilities for teleconferencing to support remote MDTMs. Its rich functionality may also be utilized as a valuable tool in teaching medical students and junior doctors [7] or it can further support research groups to collaborate efficiently to resolve issues such as image-based co-registration [8] or co-segmentation [9] [10] [11].

One extension plan is to enable the participation of more than two physicians in a collaboration session. Special research is also performed on discrete ROI analysis and classification through a machine learning cloud portal, aiming to invest in the area of Computer Assisted Diagnosis (CAD). Linking selected ROIs to semantically organized taxonomies should enhance the discovery of novel knowledge through generation of new hypotheses [58]. In this way, the available resources would enable clinical personnel and researchers across different disciplines to discover associations among their areas of interest and those that have already been studied and classified.

Offline collaboration could be featured as well. Practitioners and researchers would be able to setup "workspaces", consisting of medical data accompanied by diagnosis workflows and invite some colleagues to assist them at a later time. Ex post assistance would probably reduce distractions and increase concentration levels when it is required.

However, the biggest challenge would be the transformation of the platform to a more generic one, in a way that would enable researchers to add functionality to be consumed directly by specialists; consider dedicated biomedical image analysis operations or machine learning modules to form an open ecosystem of software components generated by members of some community, evaluated by healthcare specialists and then applied on real medical data to extend and enrich existing medical services, provide novel ones or even assist education. Many sub-areas of further development that would reduce vulnerability and enhance dependability can be identified. They range from low-tech solutions such as microphone placement to the use of latest technologies such as mobile, ubiquitous computing and Internet of Things (IoT). Pointing and annotation devices are highlighted, as well as meeting records and decision support tools, as having potential to improve MDTM services.

We can argue that our collaboration technology ecosystem has the potential to decisively contribute in many novel ways that remain to be explored.

Acknowledgment

This work has been supported by the Greek General Secreteriat of Research and Technology under GSRT grant No 93557/2015.

References

[1] J. Pethybridge, "How team working influences discharge planning from hospital: a study of four multi-disciplinary teams in an acute hospital in england," *journal of interprofessional care*, vol. 18, no. 1, pp. 29–41, 2004.

[2] E. R. Ranschaert and F. H. B. Binkhuysen, "European teleradiology now and in the future: results of an online survey," *Insights into Imaging*, vol. 4, no. 1, pp. 93–102, Feb 2013. [Online]. Available: https://doi.org/10.1007/s13244-012-0210-z

[3] P. Duysburgh and A. Jacobs, *Collaboration through ICT between Healthcare Professionals: The Social Requirements of Health 2.0 Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 165–172. [Online]. Available: https://doi.org/10.1007/978-3-642-11745-9_26

[4] "Healthcare professional's collaboration space (share4health)," https://distrinet.cs.kuleuven.be/research/projects/Share4Health, accessed: 2017-11-30.

[5] T. Ruhstaller, H. Roe, B. Thrlimann, and J. J. Nicoll, "The multidisciplinary meeting: An indispensable aid to communication between different specialities," *European journal of Cancer*, vol. 42, no. 15, pp. 2459 – 2462, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959804906005557

[6] M. S. Ackerman, "The intellectual challenge of cscw: The gap between social requirements and technical feasibility," *Hum.-Comput. Interact.*, vol. 15, no. 2, pp. 179–203, sep 2000. [Online]. Available: http://dx.doi.org/10.1207/S15327051HCI1523_5

[7] F. Gao, "A case study of using a social annotation tool to support collaboratively learning," *The Internet and Higher Education*, vol. 17, no. Supplement C, pp. 76–83, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1096751612000802

[8] P. Wang, O. Ecabert, T. Chen, M. Wels, J. Rieber, M. Ostermeier, and D. Comaniciu, "Image-based co-registration of angiography and intravascular ultrasound images," *IEEE transactions on medical imaging*, vol. 32, no. 12, pp. 2238–2249, 2013.

[9] R. Huennekens, S. Fry, B. Walker, J. Klingensmith, N. Pool, V. Burgess, and W. Kanz, "Vascular image co-registration," https://www.google.com/patents/US7930014, 2011.

[10] J. Schmid, N. Nijdam, S. Han, J. Kim, and N. Magnenat-Thalmann, "Interactive segmentation of volumetric medical images for collaborative telemedicine," in *3D Physiological Human Workshop*. Springer, 2009, Conference Proceedings, pp. 13–24.

[11] P. Mseide, "The deep play of medicine: Discursive and collaborative processing of evidence in medical problem solving," *Communication & medicine*, vol. 3, no. 1, pp. 43–54, 2006.

[12] P. Mildenberger, M. Eichelberg, and E. Martin, "Introduction to the dicom standard," *European radiology*, vol. 12, no. 4, pp. 920–927, 2002.

[13] W. D. Bidgood, S. C. Horii, F. W. Prior, and D. E. Van Syckle, "Understanding and using dicom, the data interchange standard for biomedical imaging," *journal of the American Medical Informatics Association*, vol. 4, no. 3, pp. 199–212, 1997.

[14] M. O. Gueld, M. Kohnen, D. Keysers, H. Schubert, B. B. Wein, J. Bredno, and T. M. Lehmann, "Quality of dicom header information for image categorization," in *Medical Imaging*. International Society for Optics and Photonics, 2002, Conference Proceedings, pp. 280–287.

[15] R. Graham, R. Perriss, and A. Scarsbrook, "Dicom demystified: a review of digital file formats and their use in radiological practice," *Clinical radiology*, vol. 60, no. 11, pp. 1133–1140, 2005.

[16] C. Costa, C. Ferreira, L. Bastio, L. Ribeiro, A. Silva, and J. L. Oliveira, "Dicoogle-an open source peer-to-peer pacs," *journal of digital imaging*, vol. 24, no. 5, pp. 848–856, 2011.

[17] G. V. Koutelakis and D. K. Lymperopoulos, "Pacs through web compatible with dicom standard and wado service: advantages and implementation," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006, Conference Proceedings, pp. 2601–2605.

[18] "Leadtools html5 zero-footprint medical viewer," https://www.leadtools.com/sdk/medical/html5.

[19] "Synapse 3d, radiology visualization software," http://3dimaging.fujimed.com.

[20] A. Rosset, L. Spadola, and O. Ratib, "Osirix: an open-source software for navigating in multidimensional dicom images," *journal of digital imaging*, vol. 17, no. 3, pp. 205–216, 2004.

[21] D. P. Barboriak, A. O. Padua, G. E. York, and J. R. MacFall, "Creation of dicomaware applications using imagej," *journal of Digital Imaging*, vol. 18, no. 2, pp. 91–99, 2005. [Online]. Available: http://dx.doi.org/10.1007/s10278-004-1879-4

[22] D. Haak, C.-E. Page, and T. M. Deserno, "A survey of dicom viewer software to integrate clinical research and medical imaging," *Journal of digital imaging*, vol. 29, no. 2, p. 206215, April 2016. [Online]. Available: http://europepmc.org/articles/PMC4788610

[23] D. Haak, C. E. Page, K. Kabino, and T. M. Deserno, "Evaluation of dicom viewer software for workflow integration in clinical trials," in *SPIE Medical Imaging*. International Society for Optics and Photonics, 2015, Conference Proceedings, pp. 94 180O–94 180O–9.

[24] F. J. Huyse, J. S. Lyons, F. C. Stiefel, J. P. Slaets, P. de Jonge, P. Fink, R. O. Gans, P. Guex, T. Herzog, and A. Lobo, "intermed: a method to assess health service needs: I. development and reliability," *General hospital psychiatry*, vol. 21, no. 1, pp. 39–48, 1999.

[25] F. C. Stiefel, P. de Jonge, F. J. Huyse, P. Guex, J. P. Slaets, J. S. Lyons, J. Spagnoli, and M. Vannotti, "intermed: a method to assess health service needs: Ii. results on its validity and clinical use," *General hospital psychiatry*, vol. 21, no. 1, pp. 49–56, 1999.

[26] D. G. Kilman and D. W. Forslund, "An international collaboratory based on virtual patient records," *Communications of the ACM*, vol. 40, no. 8, pp. 110–117, 1997.

[27] Y. S. Lim, D. D. Feng, and T. W. Cai, "A web-based collaborative system for medical image analysis and diagnosis," in *Selected papers from the Pan-Sydney workshop on Visualisation*, vol. 2. Australian Computer Society, Inc., 2000, Conference Proceedings, pp. 93–95.

[28] I. Maglogiannis, C. Delakouridis, and L. Kazatzopoulos, "Enabling collaborative medical diagnosis over the internet via peer-to-peer distribution of electronic health records," *journal of Medical Systems*, vol. 30, no. 2, pp. 107–116, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10916-005-7984-1

[29] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. E. Mattison, "The hl7 clinical document architecture," *journal of the American Medical Informatics Association*, vol. 8, no. 6, pp. 552–569, 2001.

[30] A. H. T. Le, B. Liu, and H. K. Huang, "Integration of computer-aided diagnosis/detection (cad) results in a pacs environment using cadpacs toolkit and dicom sr," *International journal of Computer Assisted Radiology and Surgery*, vol. 4, no. 4, pp. 317–329, 2009. [Online]. Available: http://dx.doi.org/10.1007/s11548-009-0297-y

[31] J. Millan and L. Yunda, "An open-access web-based medical image atlas for collaborative medical image sharing, processing, web semantic searching and analysis with uses in medical training, research and second opinion of cases," *Nova*, vol. 12, no. 22, pp. 143–150, 2014.

[32] A. Messaoudi, R. Bouslimi, and J. Akaichi, "Indexing medical images based on collaborative experts reports," *International Journal of Computer Applications*, p. 9, May 2013.

[33] R. Bouslimi, J. Akaichi, M. G. Ayadi, and H. Hedhli, "A medical collaboration network for medical image analysis," *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 5, no. 1, p. 10, 2016. [Online]. Available: http://dx.doi.org/10.1007/s13721-016-0117-6

[34] D. K. Michal Vossberg, Thomas Tolxdorff, "Dicom image communication in globus-based medical grids," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 145–153, 2008.

[35] P. T. d. B. Slvia D. Olabarriaga, Tristan Glatard, "A virtual laboratory for medical image analysis," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, pp. 979–985, 2010.

[36] F. G. Andriopoulou, K. Birkos, and D. Lymberopoulos, "P2care: A dynamic peer-to-peer network for collaboration in personalized healthcare service delivery," *Computers in Industry*, vol. 69, pp. 45–60, 2015.

[37] F. G. Andriopoulou, K. D. Birkos, and D. K. Lymberopoulos, "P2care: A dynamic self-organized healthcare network for ubiquitous healthcare environment," in *e-Health Networking, Applications & Services (Healthcom), IEEE 15th International Conference on*. IEEE, 2013, Conference Proceedings, pp. 294–298.

[38] M. Valdiserri and W. Goble, "Medical collaboration system and method," https://www.google.com/patents/US20110238618, 2011.

[39] L. Xie, "Adaptive distributed medical image viewing and manipulating systems," https://www.google.com/patents/US20120084350, 2012.

[40] R. Ahmed, "Systems and methods for medical diagnostic collaboration," https://www.google.com/patents/US20170195377, July 2017, uS Patent App. 15/462,336.

[41] C. Andrikos, I. Maglogiannis, E. Bilalis, G. Spyroglou, and P. Tsanakas, "An intelligent platform for hosting medical collaborative services," in *Hellenic Conference on Artificial Intelligence*. Springer, 2014, Conference Proceedings, pp. 354–359.

[42] C. Andrikos, G. Rassias, P. Tsanakas, and I. Maglogiannis, "Real-time medical collaboration services over the web," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2824312, IEEE Journal of Biomedical and Health Informatics

Biology Society (EMBC). IEEE, 2015, Conference Proceedings, pp. 1393–1396.

[43] S.-y. Katayama, T. Goda, S. Shiramatsu, T. Ozono, and T. Shintani, "A fast synchronization mechanism for collaborative web applications based on html5," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 14th ACIS International Conference on*. IEEE, 2013, Conference Proceedings, pp. 663–668.

[44] J. Yang, J. Qiu, and Y. Li, "A profile-based approach to just-in-time scalability for cloud applications," in *IEEE International conference on Cloud Computing*. IEEE, 2009, Conference Proceedings, pp. 9–16.

[45] S. Jodogne, C. Bernard, M. Devillers, E. Lenaerts, and P. Coucke, "Orthanc-a lightweight, restful dicom server for healthcare and medical research," in *2013 IEEE 10th International Symposium on Biomedical Imaging*. IEEE, Conference Proceedings, pp. 190–193.

[46] K. D. Lewis, "Web single sign-on authentication using saml," *aInternational Journal of Computer Science Issues (IJCSI)*, vol. 1, pp. 41–48, August 2009.

[47] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, "Web services composition: A decades overview," *Information Sciences*, vol. 280, no. Supplement C, pp. 218 – 238, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514005428

[48] S. Bagnasco, D. Berzano, A. Guarise, S. Lusso, M. Masera, and S. Vallero, "Monitoring of iaas and scientific applications on the cloud using the elasticsearch ecosystem," *journal of Physics: Conference Series*, vol. 608, no. 1, p. 012016, 2015. [Online]. Available: http://stacks.iop.org/1742-6596/608/i=1/a=012016

[49] A. Amoroso, F. Astorino, S. Bagnasco, N. A. Balashov, F. Bianchi, M. Destefanis, S. Lusso, M. Maggiora, J. Pellegrino, L. Yan, T. Yan, X. Zhang, and X. Zhao, "A modular (almost) automatic set-up for elastic multi-tenants cloud (micro)infrastructures," *journal of Physics: Conference Series*, vol. 898, no. 8, p. 082031, 2017. [Online]. Available: http://stacks.iop.org/1742-6596/898/i=8/a=082031

[50] E. Czaplicki and S. Chong, "Asynchronous functional reactive programming for guis," in *ACM SIGPLAN Notices*, vol. 48. ACM, Conference Proceedings, pp. 411–422.

[51] M. Mehrara, P.-C. Hsu, M. Samadi, and S. Mahlke, "Dynamic parallelization of javascript applications using an ultra-lightweight speculation mechanism," in *IEEE 17th International Symposium on High Performance Computer Architecture*. IEEE, 2011, Conference Proceedings, pp. 87–98.

[52] S. Greenberg and D. Marwood, "Real time groupware as a distributed system: Concurrency control and its effect on the interface," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, ser. CSCW '94. New York, NY, USA: ACM, 1994, pp. 207–217. [Online]. Available: http://doi.acm.org/10.1145/192844.193011

[53] C. A. Ellis and S. J. Gibbs, "Concurrency control in groupware systems," *SIGMOD Rec.*, vol. 18, no. 2, pp. 399–407, Jun. 1989. [Online]. Available: http://doi.acm.org/10.1145/66926.66963

[54] C. Sun, "Optional and responsive fine-grain locking in internet-based collaborative systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 994–1008, Sep 2002.

[55] M. R. Kamdar, D. Zeginis, A. Hasnain, S. Decker, and H. F. Deus, "Reveald: A user-driven domain-specific interactive search platform for biomedical research," *journal of biomedical informatics*, vol. 47, pp. 112–130, 2014.

[56] L. Xie, "Adaptive distributed medical image viewing and manipulating systems," https://www.google.com/patents/US20120084350, 2012.

[57] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.

[58] M. Saleem, M. R. Kamdar, A. Iqbal, S. Sampath, H. F. Deus, and A.-C. N. Ngomo, "Big linked cancer data: Integrating linked tcga and pubmed," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 27, pp. 34–41, 2014.

Christos Andrikos received a Diploma in Electrical & Computer Engineering from the National Technical University of Athens (NTUA), Greece. Currently, he is a PhD candidate in the School of ECE, NTUA, a member of Computing Systems Laboratory (CSLab), ECE, NTUA and a research associate with GRNET. His research focuses on AI algorithms and applications in medicine.

Georgios Rassias received a Diploma in Electrical & Computer Engineering from the National Technical University of Athens (NTUA), Greece. Currently, he is a PhD candidate in the School of ECE, NTUA, a member of Computing Systems Laboratory (CSLab), ECE, NTUA and a research associate with GRNET. His research interests include cloud computing, machine learning and AI applications in medicine and bioinformatics.

Prof. Panayiotis Tsanakas received his BS degree in electrical engineering from the University of Thessaloniki, his MSc in Computer Engineering from Ohio University and his PhD in Computer Engineering from the National Technical University of Athens. He is currently serving as professor of computer engineering in the School of Electrical & Computer Engineering of the National Technical University of Athens. Has participated in several national and EU-sponsored projects, in subjects covering e-Infrastructures, distributed systems, parallel computer architectures, embedded systems, grid computing and medical informatics. He has published four textbooks and more than 100 research articles in scientific journals and international conferences. He is currently the Chairman of GRNET, responsible for the deployment and operation of the national R&E network, along with large-scale national HPC and IaaS facilities. He has also served as a member of the national telecom regulator (EETT). His research interests include high-performance computer systems architectures, cloud computing and distributed applications in medicine.

Ilias Maglogiannis (M00, SM11) received a Diploma in Electrical & Computer Engineering and a Ph.D. in Biomedical Engineering and Medical Informatics from the National Technical University of Athens (NTUA) Greece. He has served as faculty member in the University of Aegean and the University of Thessaly. He is now an Associate Professor in the Dept. of Digital Systems in the University of Piraeus and director of the Computational Biomedicine Laboratory (www.cbml.ds.unipi.gr). His scientific interests include Biomedical Informatics, Computer Vision and Pervasive Healthcare Systems. His published scientific work includes five (5) books, 92 journal papers and more than 180 international conference papers, while has received more than 4000 citations on his published work. He is a senior member of the IEEE, ACM, SPIE and Vice President member of the Hellenic Association of Biomedical Engineering.