9-2017

Mobile Medical Applications: From Cloud-Oriented To Cloud Ready

Ioannis Patias
University of Sofia "St. Kliment Ohridski", BG, patias@fmi.uni-sofia.bg

Vasil Georgiev
University of Sofia "St. Kliment Ohridski", BG, v_georgiev@fmi.unisofia.bg

MOBILE MEDICAL APPLICATIONS: FROM CLOUD-ORIENTED TO CLOUD READY

Research paper

Patias, Ioannis, University of Sofia "St. Kliment Ohridski", BG, patias@fmi.uni-sofia.bg

Georgiev, Vasil, University of Sofia "St. Kliment Ohridski", BG, v_georgiev@fmi.uni-sofia.bg

Abstract

The paper's main purpose is to show the importance, and some improvements of mobile telemedicine information systems in patients' compliance. First the basic definitions of cloud-oriented architecture are provided, and based on that the prerequisites for cloud-ready applications development. This is the basis on which our solution is build. Described is a solution, developed to be in service of patients with different diseases, and their physicians. The web application gives physicians the ability of developing template of care plan for some diagnoses and then to personalize it for specific patients and their needs. The mobile application is disease management tool and it is developed to be in help of patients. Its purpose is to notify them for the next steps of the care plan, which of course are prescribed, and will be followed up by their physicians.

Keywords: telemedicine, cloud-oriented, cloud-ready, mobile application, HL7, FHIR.

1 Cloud-oriented and cloud-ready

Cloud computing is believed to have been invented by Joseph Carl Robnett Licklider in the 1960s with his work on ARPANET to connect people and data from anywhere at any time. Cloud-Oriented Architecture (COA) is a term coined by Jeff Barr at Amazon Web Services Jeff Barr (2010). It describes the architecture, where applications act as services in the cloud and serve other applications in the cloud environment. The aim is an architecture for IT infrastructure and software applications that is optimized for use in cloud computing environments.

In other words the cloud was initially used to indicate the Internet. But, the term's meaning of initially "the Internet" shifted, and now refers to the resources we usually perceive as being on the Internet, together with the means to access them. Respectively the term cloud computing may be undertaken of another name for an existing concept. Thus, the term refers to something new indeed. The cloud is a coherent, large-scale, and in most of the cases publicly accessible collection of computational, storage, and networking resources. All these can be reached through web services, and are accessible on payment for short or long-term use, based on the actual resources consummation. Typical technologies used in cloud computing are TCP-IP, and REST architecture web services.

From the technical point of view the advantages are quite perspective. First of all any need of scaling and enlarging of the system is quick and cheap. Any new hardware becomes accessible directly online, and within minutes. And exactly the same way resources, no longer needed, can be released the same easy way. This practically provides with unlimited scalability. Second there is no need for becoming expert on the supporting infrastructure. This role is now on the provider to keep track on the technical specifications of the hardware used, in order to change and improve over time. For the end user the resources become abstract and thus undifferentiated. And last but not least and experiments are quite cheap, because any risk remains on the provider. We may use IT resources as separate blocks, and pay individually for the blocks used only.

In order to develop cloud-ready applications, the application architecture should take into consideration (Kyle, Brown and Mike, Capern (2014)). An application is cloud-ready if you can effectively deploy it into a public or private cloud. That is, you must design the application so that it can leverage the platform-as-a-service (PaaS) layer on which it runs, and won't break because of design limitations that collide with assumptions that are made in the PaaS layer. If you follow these simple rules in your application design, you can usually make your existing applications cloud-ready without going through an entire reimplementation.

So, the application should be described as collection of services. The application as concept should be service-based or so-called service-oriented architecture (SOA). It is important also to separate the processing of the data, and use them as two separate components. This will help, since clouds normally are complex distributed systems. That means that our system will work better with application architectures like this. But, again separating and keeping close communication should not be confused. Applications components, which constantly communicate with each other, normally will lower the overall performance. This is why we should optimize the communications between the applications components. We should also try to design our system so as it can take advantage of scaling. That means we should well consider how the application could scale when the load increases. And last but still very important to make the security a systemic component, and include it within the application.

2 Case study

In this section we will first give the background of a mobile application, which further will be used as a case study. With this case we aim the above mentioned regarding transition from cloud-oriented architecture to cloud-ready applications to be described.

2.1 Patients compliance

As technology constantly changes, so and the medical world is changing, since technology is of crucial importance for the medical world. Telemedicine can be used for consultation, forward diagnostic images, and the medical record of the patient directly to the doctor for review. Thus, the telemedicine doctor having enough information is able to make diagnosis and even provide treatment plan. Regarding remote patient monitoring systems the picture is the same. We have remote systems, which constantly collect and forward data to the doctor for further interpretation.

There are obvious advantages of telemedicine. First of all is that telemedicine gives access to specialists and information to people, which otherwise would not have. Telemedicine also may save money. And this applies in both directions, first the patient, as receiver of the medical service, the treatment, and of course for the provider too.

The main disadvantage of telemedicine is related to the availability and the cost. In some case it may be expensive for the provider to set up and maintain telemedicine infrastructure. Usually it is too expensive for smaller health services providers to implement such costly systems. And although telemedicine wider the provided treatment methods, still it cannot be the same as a more personal or face-to-face relationship. For sure there are many types of illnesses and medical problems, which require face-to-face relationship, and of course real physical assessment. Such cases cannot be treated through telemedicine.

Having said that many people need instruments to follow their treatment, and share information on the next steps of their care plan with their physician. People with same health problems more or less need to follow the same care plan. But, it is the physician who decides for any deviations of this core care plan. That means a lot of information needs to be shared between the physician and the patient.

But, it is often considered patients' compliance as the sending reminder messages, by voice or SMS, to patients. This is trying to achieve treatment compliance, and overcoming of challenges such as drug resistance, but with limited one-way communication.

According to WHO, World Health Organization (2011), approximately 35% of Member States across WHO regions reported conducting treatment compliance initiatives, with the South-East Asia Region highest (50%) and the Western Pacific Region lowest (23%). In the European Region, reported treatment compliance initiatives targeted chronic illnesses such as diabetes, asthma, obesity, chronic obstructive pulmonary disease (COPD), and chronic heart disease.

Close to 60% of participating countries in the high-income group reported treatment compliance initiatives, compared to approximately 30% for the other income groups. These results are consistent with the literature review, which found a concentration of studies from high-income countries such as Canada, the United Kingdom, and United States with treatment compliance programmes using SMS, mobile phone applications, web browsing and e-mail for chronic diseases such as diabetes, asthma, and obesity. Both the high- and the low-income groups had a large proportion of Member States reporting informal initiatives (around 20%).

2.2 Care plan standardized representation

As we see technology constantly changes, so and the medical world is changing, since technology is of crucial importance for the medical world. Telemedicine can be used for consultation, forward diagnostic images, and the medical record of the patient directly to the doctor for review. Thus, the telemedicine doctor having enough information is able to make diagnosis and even provide treatment plan. Regarding remote patient monitoring systems the picture is the same. We have remote systems, which constantly collect and forward data to the doctor for further interpretation.

Care plan (wiki.hl7.org) is a document that identifies nursing orders for a patient and serves as a guide to nursing care. It can either be written for an individual patient, be retrieved from a computer and individualized, or be pre-printed for a specific disease, condition, or nursing diagnosis and individualized to the specific patient. Standardized care plans are available for a number of patient conditions.

In general all Information Systems, and especially Healthcare Information Systems need formal representation of logic covering from low-level technical specification, going through intermediate logical architecture, and getting to the high-level conceptual description. Being able to formally represent knowledge using expression terms and standardized information models allows us to implement better systems and improves communication between systems.

A basic problem in this direction is the way to express basic logic conditions. Such the ones that will help define requirements constraints, which may be further used in decision support, and get to whole protocol descriptions. This need for expressions gave the definition of languages for representation. Those languages are limited based on the data structures, and the information models for which they are designed to operate. Independence of representation and technology infrastructure is important to assure the usability of the information model. Standard languages often meet these requirements, like Java, C#, and others. But, those languages depend on the platform, and they are general-purpose development languages. This makes them insufficient.

Health Level Seven International (hl7.org) is a not-for-profit, ANSI-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services.

FHIR® – Fast Healthcare Interoperability Resources (hl7.org/fhir) – is a next generation standards framework created by HL7. FHIR is designed as a simple, lightweight, platform- and structure-independent graph traversal language. As such, FHIR meets the requirements, and since can be used in various environments, provides a simple and effective formal representation of expressions.

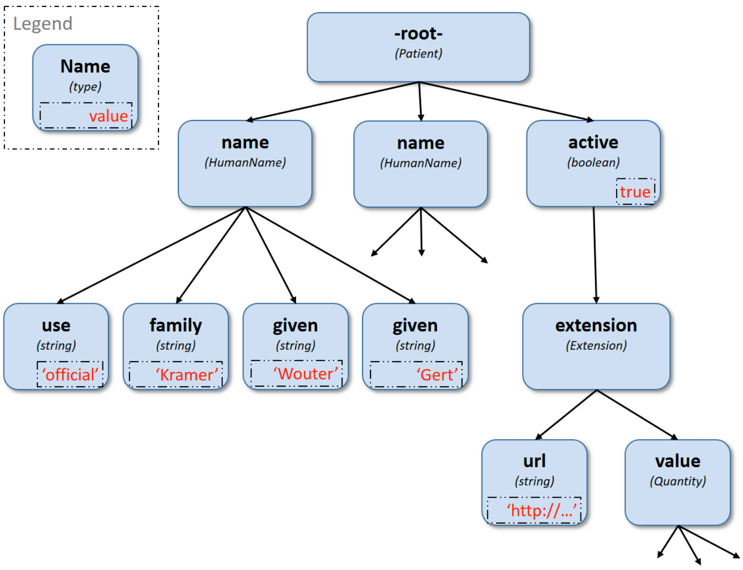

Figure 1: Partial representation of a FHIR Patient resource.

In FHIR, FHIRPath is used within the specification to provide formal definitions for conditions such as validation invariants, search parameter paths, etc. Within Clinical Quality Language (CQL), FHIRPath is used to simplify graph-traversal for hierarchical information models.

In both FHIR and CQL, the model independence of FHIRPath means that expressions can be written that deal with the contents of the resources and data types as described in the Logical views, or the UML diagrams, rather than against the physical representation of those resources. JSON and XML specific features are not visible to the FHIRPath language.

Data are represented as a tree of labelled nodes, where each node may optionally carry a primitive value and have child nodes. Nodes need not have a unique label, and leaf nodes must carry a primitive value. For example, a (partial) representation of a FHIR Patient resource in this model looks like figure 1.

FHIR solutions are built from a set of modular components called "Resources". These resources can easily be assembled into working systems that solve real world clinical and administrative problems at a fraction of the price of existing alternatives. FHIR is suitable for use in a wide variety of contexts – mobile phone apps, cloud communications, EHR-based data sharing, server communication in large institutional healthcare providers, and much more.

Resource CarePlan - Content respectively describes the intention of how one or more practitioners intend to deliver care for a particular patient, group or community for a period of time, possibly limited to care for a specific condition or set of conditions. This is how CarePlan is one of the request resources in the FHIR workflow specification. Care Plans are used in many areas of healthcare with a variety of scopes. They can be as simple as a general practitioner keeping track of when their patient is next due for a tetanus immunization through to a detailed plan for an oncology patient covering diet, chemotherapy, radiation, lab work and counselling with detailed timing relationships, pre-conditions and goals. They may be used in veterinary care or clinical research to describe the care of a herd or other collection of animals. In public health, they may describe education or immunization campaigns. This resource takes an intermediate approach to complexity. It captures basic details about who is involved and what actions are intended without dealing in discrete data about dependencies and timing relationships. These can be supported where necessary using the extension mechanism.

This resource can be used to represent both proposed plans (for example, recommendations from a decision support engine or returned as part of a consult report) as well as active plans. The nature of the plan is communicated by the status. Some systems may need to filter CarePlans to ensure that only appropriate plans are exposed via a given user interface.

For simplicity sake, CarePlan allows the in-line definition of activities as part of a plan using the activity.detail element. However, activities can also be defined using references to the various "request" resources. These references could be to resources with a status of "planned" or to an active order. It is possible for planned activities to exist (e.g. appointments) without needing a CarePlan at all. CarePlans are used when there's a need to group activities, goals and/or participants together to provide some degree of context.

Workflow description, can be given under the scope of workflow as an essential part of healthcare - orders, care protocols, referrals are the drivers of most activity within in-patient settings and a great deal of activity in community care as well. FHIR is concerned with workflow when there's a need to share information about workflow state or relationships, when there's a need to coordinate or drive the execution of workflow across systems and when there's a need to define allowed actions, dependencies and conditions on behaviour.

FHIR does not need to be used for the execution of workflow. Orders, care plans, lab results, hospital admissions, claim payments and other records can all be shared using FHIR resources without the process to solicit fulfilment of those orders or requesting payment of those claims being driven by a FHIR transaction. Interoperable support for workflow execution is a more advanced FHIR activity because it requires a higher degree of standardization. Rather than merely standardizing the data to exchange, interoperable workflow execution requires standardization of the processes, roles and activities across the different systems.

FHIR defines three categories of resources that are involved in activities - requests, events and definitions. Each of these categories has a "pattern" associated with it. Resources that fall into that category are encouraged to adhere to their respective pattern. These patterns provide standard elements that are typical for most resources of each category. Strict adherence is not required as work groups are expected to align with typical domain behaviour and requirements as more authoritative than "desired" architectural patterns. In some cases, capabilities might be supported with extensions rather than core elements where a pattern capability is deemed to be "not common, but still relevant" for a given resource.

In addition to defining patterns for resources used in workflow processes, FHIR supports the execution of those processes as well. However, FHIR does not define a "one size fits all" solution for workflow architecture. FHIR supports a variety of interoperability paradigms and most of them (REST, Messaging and Services) provide support for driving workflow execution.

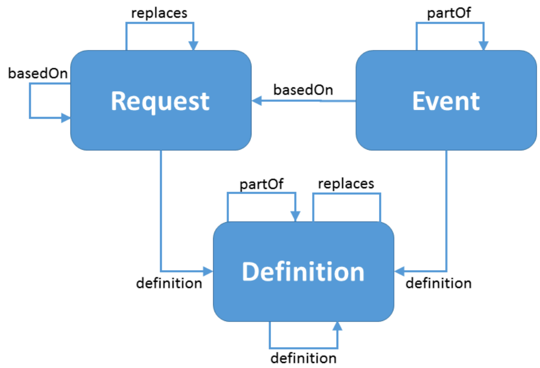

Figure 2: Workflow Resource Relationships.

Not all resources in FHIR are related to workflow - many are used to describe entities and roles (patients, medications, etc.) or infrastructure (structure definitions, value sets, etc.). However, a large proportion of the FHIR resources are devoted to the description of activities in one fashion or another and almost all of these fall into the realm of workflow - they describe things that can be done (definitions), are desired to be done (requests) or that have been done (events). Figure 3 illustrates the workflow resource relationships.

The PlanDefinition resource is used to describe series, sequences, or groups of actions to be taken, while the ActivityDefinition resource is used to define each specific step or activity to be performed.

As the name implies, the PlanDefinition resource is strictly definitional. It does not represent the intention to take any action, nor does it represent that any actions have been taken. Rather, the resource provides a definition that can be applied in the appropriate circumstances. When the plan definition is applied, the result will in general be a set of actions that should be (or potentially even have been) performed.

Note that the PlanDefinition still has action-level information, as well as a reference to an ActivityDefinition. The action-level information defined in the PlanDefinition itself is used to describe how the actions are related to each other within the plan, where the ActivityDefinition contains only information about the activity itself.

3 Improved business model

The improved business model was developed based on research results with existing telemedicine systems characteristics (The Best Telemedicine Apps of 2016). The main points of improvement are three. First point is the ability of individualized care plan. That means that apart of the templates provided as typical care plans for different cases we also provide the ability the physician to decide and adapt the provided to the needs of the concrete patient. This aims to provide a personalized extended treatment plan. The plan should be prepared by a treating physician and be valid only for a patient. To make it easier for the doctor, we add common templates to complement and change with little effort for their different patients, when necessary.

The second improvement is related to physician – patient communication. In some apps, it is provided the ability to make a video or voice chat with a health care professional; others can schedule a personal appointment with a doctor, or to consult with text messages. In none of the applications for treatment of chronic diseases requiring the preparation of a personal treatment plan there is real-time feedback component. Meaning a common module to give feedback to the physician, or the ability both the physician and the patient to see the same measurement results.

Finally the third improvement is all the above mentioned to comply with HL7 standards, including the FHIR Resource CarePlan – Content standard. This will provide with the ability to easily update and share further.

Having all the above the aim is to build an information system consisting of a mobile application to serve the patients and web application, to be used by treating physicians, and both to manage patients treatment plans.

The web application is for physicians only. They can manage the different diagnoses that are subject to planned treatment. Doctors can generate templates with different treatment steps. Doctors can manage their patients, add, delete, and manage their treatment plans and schedule activities.

The mobile application is to be used by patients. They can follow the schedule, as given to them by the treating physician, and can complete their own reminders and tasks. Through the mobile application, patients can upload measurement results and monitor changes in the values of different indicators over time.

4 Applications' specifications and technologies used

4.1 Applications' description

The system, as shown on the figure below, consists of:

- Web application, which is used by doctors: doctors can manage their patients, diagnoses and create care plan for diagnosed or personalized care plan for specific patients.
- Mobile application, which is used by the patients: patients can be notified for the next appointments of the care plan and record their measurements of medical or lab tests. This mobile application is for android systems, as further developments will cover other platforms also.
- Database server: server through which communication between a doctor and a patient is made, based on the appropriate database designed.

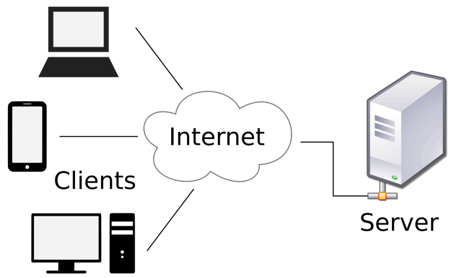

Figure 3: Typical architecture.

In this section briefly are presented the modules and components of the system. For the sake of clarity, they are presented as list of the main features of the modules without going into detail. The modules and components of the system are:

4.1.1 Web application

- Designed for use by physicians,
- Functionality to:

    - add templates for each diagnosis,
    - edit the predefined template for the needs of each specific patient,
    - add a new assignment to the calendar of a patient,
    - add a new task / reminder to all patients with a diagnosis,
    - add a new patient,
- Search by:
    - patient name,
    - diagnosis,
- patient-to-physician feedback, meaning, the physician to be able to monitor what the patient adds to his schedule.

4.1.2 Mobile application

- Designed for use by patients,
- Provides detailed calendar/ schedule with next tasks,
- Functionality to:
    - add alarms and reminder from the calendar,
    - preserve critical measurements, and results,
    - add additional measurements necessary for specific patient, configurable to track user-defined measurements.
- All measurements, and results visualization in tabular form.

4.1.3 Database server

- The server connects the mobile and the web applications. The communication is implemented through REST architecture,
- Security connection is implemented by the OAuth 2.0 protocol,
- The database as designed to be completely separated from the mobile and wed applications, thus access to information is realised through REST architecture again.
- The services provided are implemented as database manipulation by methods that add, edit, and delete information.

4.2 Technologies used

4.2.1 REST

Representational State Transfer (REST) architectural style is developed by W3C Technical Architecture Group (TAG) together with HTTP/1.1. It is based on existing HTTP/1.0. design of "Word Wide Web". The basic idea is that we have client-server architecture. Client sends resource/data requests. Both the request and the reply are focused on transferring the current status of the resource at the moment of the request. Each resource is related to URI (Uniform Resource Identifier). The interaction with given resource is based on: URI, and the concrete actions/methods applicable to the resource (GET, POST, PUT, DELETE, etc). REST services are easy to implement can be used with HTTP/HTTPS protocol, support data formats like XML, JSON, and HTML, and communicate through clients using Java script. All these make them a very flexible tool.

4.2.2 Joda

In order to avoid limitations of Java 7 time libraries, and to synchronize multiple requests, since the system is developed for a large number of users, and multi-thread performance is a priority, the Joda time library is used.

4.2.3 Caldroid

Caldroid is used to allow all schedules to be displayed on a calendar.

4.2.4 MySQL Connector

MySQL database link library, as part of the products of MySQL is used, since the database is used.

4.2.5 Bootstrap

Bootstrap is used, as one of the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile first projects on the web.

4.2.6 jQuery

jQuery is used as a widespread alternative to JavaScript, published in early 2006 by John Rezig. Basically, jQuery simplifies access to any element on a given website, thus allowing easy build of dynamic page functionality. JQuery is free and open source software licensed under MIT license.

4.2.7 Json

JSON (JavaScript Object Notation) as a simplified data exchange format that works for both people and computers is used. It is based on a subset of the JavaScript programming language, Standard ECMA-262 3rd Edition, since December 1999. JSON has a text format completely independent of language implementation but uses conventions familiar to C-like programmers, Including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON the perfect language for data exchange.

4.2.8 OAuth

OAuth 2.0 dropwizard is the next generation OAuth protocol that was created in 2006. It is used as easy to apply, while providing specific authorization scenarios for Internet, desktop, and mobile applications, and various devices. The specification was developed by the IETF OAuth Working Group (oauth.net/2/).

4.2.9 fullCalendar

fullCalendar as JavaScript open source calendar, easy to personalize and use in the code is used, because it is suitable for adding and editing events.

Conclusions

Mobile telemedicine information systems are a strong instrument in patients' compliance. Many systems have proved that. Apart the patient it is also the health-care provider affected by compliance. The role of mobile medical applications as instrument for both structuring the communication and improving the physician-patient relationship has bee proved quite important. The proposed system provides solution for three telemedicine problems. First improves patients' awareness, and doctor – patient communication, second facilitates doctors' information management in regards to the diseases of their patients, and third provides all this on a standardized, and based on HL7 approach. Having in mind the technology developments, and the wide provision of cloud-oriented infrastructure, it is important that the architecture design of the system takes from the beginning into consideration what is necessary in order to be cloud-ready.

Acknowledgements:

This paper is supported by the project ДН 02/9-17.12.2016.

References

Jeff Barr (2010). Host Your Web Site In The Cloud: Amazon Web Services Made Easy: Amazon EC2 Made Easy, Amazon Web Services, ISBN 978-0-9805768-3-2.

Kyle, Brown and Mike, Capern (2014). Top 9 rules for cloud applications, The dos and don'ts of making your application cloud-ready, https://www.ibm.com/developerworks/.

National Center for Biotechnology Information (2016). NCBI, Carme Hernandez, Jennifer Mallow, and Georgia L. Narsavage, Delivering telemedicine interventions in chronic respiratory disease, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4734754/.

World Health Organization (2003). Library Cataloguing-in-Publication Data, Adherence to long-term therapies: evidence for action. Geneva: World Health Organisation. ISBN 92-4-154599-2.

World Health Organization (2011). mHealth New horizons for health through mobile technologies. WHO Library Cataloguing-in-Publication Data. ISBN 978 92 4 156425 0.

Health Level Seven International (hl7.org). Clinical and Administrative Domains, and Education & Awareness, http://www.hl7.org/implement/standards/product_brief.cfm?product_id=435.

FHIR® – Fast Healthcare Interoperability Resources (hl7.org/fhir). Resource StructureDefinition, Workflow Module, Resource PlanDefinition.

The Best Telemedicine Apps of 2016. http://www.healthline.com/health/best-telemedicine-iphone-android-apps.