

Development of a Mobile Application for the C2NET Supply Chain Cloud-based Platform

Enbo Chen, Wael M. Mohammed, Borja Ramis Ferrer, Jose L. Martinez Lastra
Tampere University of Technology, Tampere, Finland
enbo.chen@student.tut.fi, {wael.mohammed, borja.ramisferrer, jose.lastra}@tut.fi

Abstract—Cloud-based ERP (Enterprise Resource Planning) Platform is a tendency to facilitate collaboration across Small and Medium-sized Enterprises (SMEs) in the industrial supply chain. However, fast and smooth accessibility to the cloud resource is a vital challenge due to efficiency, reliability, security and other critical issues. This paper presents the development of a mobile application by using web technologies for supply chain cloud-based platform to support user to keep informed and make fast decisions based on real-time notification and information. Furthermore, it illustrates the architecture, integration and implementation of a hybrid mobile application by showing real use cases in different scenarios.

Keywords—mobile application; cloud-based platform; supply chain; web-based technologies.

I. INTRODUCTION

Contemporary Small and Medium-sized Enterprises (SMEs) seek for affordable solutions that permit such organizations to be visible and accessible to any stakeholder of their provided solutions. Moreover, within the implementation and enhancement of Information Technology (IT) solutions, both new and legacy systems can be interconnected through networks and cloud-based platforms. Then, SMEs tend to employ IT-based tools and solutions that permit, among other features, i) speeding up the response to customer orders, ii) efficiency analysis of their products, iii) management of enterprise's business and field level operations and iv) the increment of productivity.

In the industrial domain, one way of providing SMEs of aforementioned enhancements is to involve companies in the supply chain of products within collaborative networks [1]. These are of a medium type that allows exchanging and sharing information rapidly and securely. In other words, having all parties, involved in the design, manufacturing and delivery of products, all in the same network preclude SMEs to be isolated of the supply chain. Thus, SMEs can provide their solutions easily and their business can grow without connectivity restrictions.

The main objective of the Cloud Collaborative Manufacturing Networks (C2NET) project¹ is to develop a cloud-based platform that will enable the collaboration and optimization of processes which are carried out in industrial supply chains [2]. This will provide the enhancement of SMEs business efficiency and visibility.

One challenge in the C2NET project is the design and development of a mobile application that permits different types of user accessing to the cloud-based platform. Basically, all users of the platform should have access to it with any device capable of i) hosting the application and ii) connecting to the Internet. In particular, access must allow users not only to use the different available services but also be notified about events happening in the supply chain. It should be noted that this platform differentiates between different types of users, including not only managers and operators of companies, but also clients. Conceptually, any person that requires accessing to the platform will need to interact with the mobile application interface. Furthermore, the interface, actions and notifications in the platform will be adaptable to the different user role.

Therefore, the objective of this manuscript is to present the insights, development and different interfaces of a hybrid mobile application for the C2NET supply chain cloud-based platform. The design of the application has been done taking into account other C2NET components in order to allow data compatibility and interoperability. In fact, features are needed due to the heterogeneity of data which is produced by different data sources [3]. Besides that, the application should address the nature of the architecture of the C2NET platform.

The rest of the paper is structured as follows: Section II presents a short literature review and current enterprise practices of collaboration networks and platforms as a service for industrial supply chains. Then, Section III describes the approach for developing the mobile application for the C2NET platform within an UML component diagram. Afterwards, the implementation of such solution is presented in Section IV, which shows two different interfaces i.e., the Home, Login interface, and the notification user interface. Finally, this research work is concluded in Section V.

II. THEORETICAL BACKGROUND

A. User Collaboration Network

With the huge leap ICT (Information and Communications Technology) had in this century, the concept of having web-based application, cloud solutions and collaborative networks became more achievable [4]. The term of collaborative networks addresses the interaction between persons as users and companies or organizations as networks [5][6]. The interaction could cover different fields. As an example, Facebook² social

¹ <http://c2net-project.eu/>

² <https://www.facebook.com/>

media, Research Gate³ and LinkedIn⁴, share the same fact about having users and companies. Nonetheless, the fields for these examples are very different.

The C2NET, on the other hand, provides a solution for Enterprise Resource Planning (ERP) supply chain activities where users who are connected to certain companies are interacting, collaborating, and exchanging information through the network. Addition to that, the C2NET provides a smart solution for the users to provide an aid for planning and monitoring production plans, logistic plans and delivery plans.

As any other collaborative networks, the C2NET provides a web-based interface for the user. This interface visualizes the features of the C2NET via a web browser. Besides, the interface notifies the user for related events that are triggered in the platform [1][3].

B. Enterprise Resource Planning and Supply Chain

ISA-95 is the international standard for the integration of enterprise and interfaces between the enterprise sectors. It has different levels of hierarchical model from top to bottom: level 4 – business planning & logistics, level 3 – manufacturing operations management, level 2,1,0 – batch control, continuous control and discrete control [7].

Enterprise resource planning (ERP) is a business process management software or system for managing, integrating and automating the important business progress, such as product planning, purchasing, production planning, inventory, sales, marketing and service delivery. Some familiar software facilitates ERP management, such as SAP, FIS Global, Oracle, etc. There is a tendency of cloud-based ERP system become more demanding than on-premise ERP because the former is scalable on-demand and allow users to concentrate on their core business [8].

Supply chain is a value network and process of organization, activities, information and resources involved in distributing a specific product from supplier to customer. Supply Chain Management (SCM) have a large amount of practices in industrial engineering, systems engineering, operations management, logistics, procurement, information technology, and marketing. [9].

Further, the ERP plays a significant role in SCM. It increases speed, productivity, efficiency working flow across multiple departments and organization. Some other benefits for integration of SCM and ERP is that automation of workflow reduces labor cost, improve corporate performance and make supply chain planning more flexible and easier to adapt to rapid changes in industrial supply chain[10].

C. Platform as a Service and OpenPaaS

Cloud computing has been evolving as a mature computing technology delivering on-demand and scalable provision of computer infrastructure, platforms and software as services. The elastic resources allow users to scale horizontally and metered service enables user to pay as you go.

Cloud computing has three stacks (from bottom to top): Infrastructure as a Service (IaaS) provides servers, networking, storage, and data center space and other hardware infrastructure [11]. Platform-as-a-Service (PaaS) offers the cloud-based environment with everything required to support building and delivering web application [12]. Software as a Service (SaaS) enabling software applications are to be used by others and that connect to users' computers via the internet and, usually, a web browser [13].

PaaS allows developing applications and deploying a new web application in minutes, because it offers the same advantages as IaaS but also additional features—middleware, development tools, and other business tools [13].

OpenPaaS⁵ is an open source collaboration platform developed by LINAGORA⁶, an open source software company, which is implemented with PaaS technology for enterprises and organizations. This platform encourages collaboration and teamwork by integrating all the useful tools (e.g. instant message, video-conference, calendar, contacts and email inbox, etc.). The platform is built on NodeJS Express web framework interacting with heterogeneous relational and NoSQL data stores. It allows developers to build their own application independently based on exposed RESTful API and OPEN-PaaS-DataBase API (ODBAPI) [14].

III. APPROACH

As any web-based solution, the C2NET platform tends to allow the user to connect to the platform using remote devices. Thanks to the architecture of the C2NET, a smart phone with internet connection could provide such a feature. This mobile application aims to allow users, e.g. managers and planners, remotely to monitor, visualize, control, share and collaborate with other partners from any digital mobile devices (e.g., tablets, smartphones) across the supply network [2]. This supports decision-maker to make agile and collaborate decision with intuitive user interface with real-time data, finally improving the user experience of usage of C2NET. Besides, the application is intended to notify the user about the occurrence of certain event. It should be noted that this approach tries to exploit the hybrid mobile application design. Which means, the application will include partially, a web view for visualizing the platform and other modules for handling the notifications.

Since the C2NET provides the HTTPS connection via a web browser, the vision of having a mobile application that uses a web view is strongly recommended. In this context, the challenge of having a mobile application that able to visualize and interact with the platform could be considered on the server side. This means the server side should serve both web browsers and mobile web browser. Addition to that, the notifications feature requires a stream of data between the C2NET platform and the mobile application. Therefore, this section presents the architecture of main modules in both the mobile application and the C2NET UCP (User Collaboration Portal). After that, a detailed illustration about the interactions is presented.

³ <https://www.researchgate.net/>

⁴ <https://www.linkedin.com/>

⁵ <http://open-paas.org/>

⁶ <https://linagora.com/>

A. Architecture

The information gathered in C2NET can be available through mobile devices in order to be easily accessible by decision-making personnel or third parties. Consequently, the accessibility to the information will improve in the efficiency of workers and processes, as described in [1].

C2NET mobile app contains two main components: the backend, which is deployed in C2NET User Collaborative Portal (UCP) and the frontend, which is deployed on the mobile device. The mobile front-end application is built on Android

platform while the back-end application is developed within Node.js that resides as a module on OpenPaaS.

The main components of the C2NET mobile application architecture are presented in Figure 1 within a UML component diagram. The diagram shows two main parts: front-end mobile application and back-end server in UCP. The users exchange data into Mobile Application connected to the C2NET websocket. The data are then passed from the UCP to the Data Collection Framework (DCF) component through an API that is to be implemented [2].

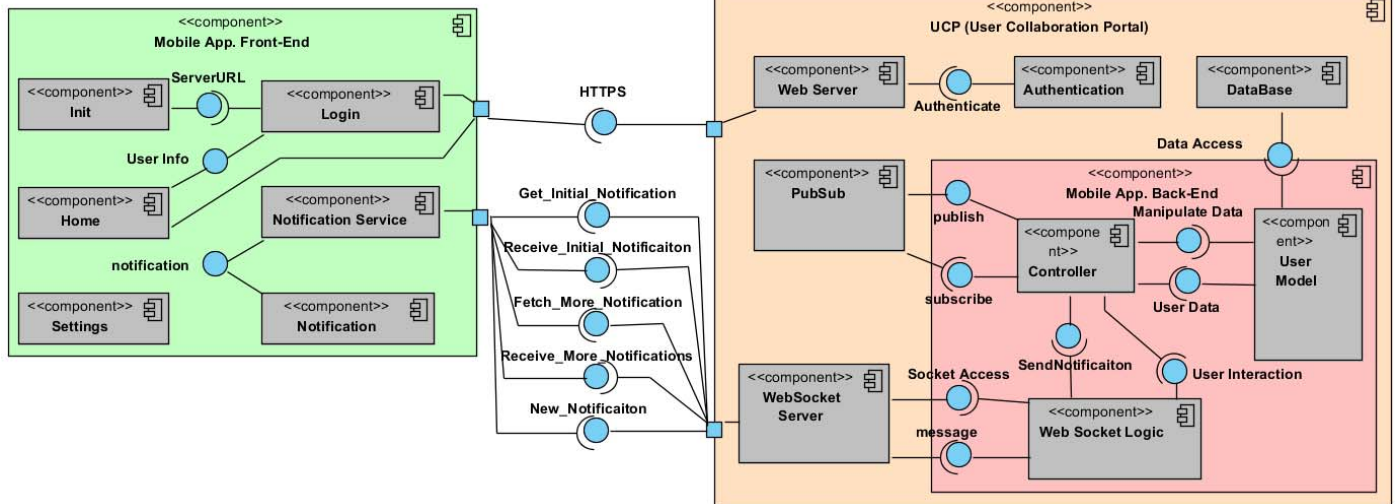


Figure 1: Component diagram of C2NET mobile application

There are six main subcomponents of the front-end mobile application that are identified as follows:

1) *Init*: The Init (i.e. initialization) component is responsible for initialization, testing the connection. Before user's login, it offers a portal that permits developers to configure the required connection parameters such as cookies, server IP address.

2) *Login*: The Login provides a user interface within a username and password entry form in order to validate identity. The mechanism behind this component allows mobile client side to send confidentiality with secure cookies to server by HTTPs protocol.

3) *Home*: A home page is realized after login by the home component by using a web view, widget and toolbar. In this way, the user can easily visit all the resource modules on the cloud platform.

4) *Notification*: The Notification component renders the notifications from cloud platform to notify users immediately. This is important to remote monitor and to be informed on real-time.

5) *Notification Service*: The Notification Service supports notification user interface in the background even if the user switches to another application. This component is bounded to interact with notification and even perform inter-process communication between server by using web socket.

6) *Setting*: The Setting component allows user to change personal settings of notification, preference and confidentiality easily and flexibly.

On the other hand, the back-end mobile application includes three main components which are:

1) *User Model*, which structures the user's data in a reliable schema and provide methods to create, read, update and delete the user's data in database. Each user is represented as an instance of *User Model*.

2) *Web socket Connector* that allows the front-end in the mobile device to communicate with the platform (back-end). This connection is needed for passing the notifications to the users using their mobiles.

3) *Controller*, which manages the web socket connection, user's events and handling the notifications. The controller occurs as a singleton class, which maps the users to the web sockets. In this context, each user could have multi web sockets (mobile devices).

B. Integration

It is vital to implement an efficient and complete integration between the mobile application and the C2NET cloud-based platform. This section mainly focuses on the integration between frontend and backend and the problems about authentication and communication.

The communication between backend and frontend is established and secured with authentication by the login component in frontend and web server component in backend.

The protocol – HTTPS secures the transportation of the user account and password and prevents man-in-the-middle attack since the information on the cloud is private and confidential. Secure cookie, session and token are implemented to improve security as well as provide a frictionless login. Moreover, Access Control and Security (AC&S) provides domain-oriented vision of the Authentication, Authorization and Accounting (AAA) services offered by the PaaS for Service Level Agreement (SLA).

After authentication, a Secure Web Socket (WSS) is established to open an interactive communication session between server and mobile application. This event-driven behavior can ensure the efficient communication from each side.

IV. IMPLEMENTATION

A. Login and notifications use case (different scenarios)

Agile and stable mobile apps shall assist users to make decisions and solve problems in different working situations[16]. Therefore, the one of the goals of C2NET is to develop a simple, clear, efficient and user-friendly mobile application. This section analyzes two most usual scenarios that present distinct behavior of users when interact with the application.

In the first scenario, the login process, if a user tries to log in the Android application, the confidential information (i.e. email and password) are sent to back-end server within the HTTPs protocol. The back-end server will request authorization component in UCP to validate authenticity of users’ credentials. Authentication module is a significant component in UCP, which uses the industry-standard authorization framework and authorization middleware passport.js.

If the login is successful, the mobile application will navigate user to home page, where user can access all the authorized resource on the cloud platform via REST API with JSON Web Token. If someone try to log in with wrong email account or password, application will throw the error message and push notification to a real user who is using the application.

In the second scenario (notification), user is also able to subscribe to the notification in PubSub (Publish and Subscribe) module via mobile application setting, which allows user to customize the preference of notification for pushing a service i.e., which kind of notification a user wants to receive, which devices should be pushed, etc.

B. User interface and client side

Figure 2 (a) shows the interface of login that allows user to input email account and password. Figure 2 (b) presents the home page after login, wherein a main part of it is the web view from OpenPaaS. Figure 2 (c) shows a pop-up window containing all the modules (e.g. mail, calendar, contacts, accounts, etc.) that reside on the cloud.

As it can be seen in Figure 3 (a), for each incoming notification, the notification service creates a notification appearing as an icon in the status bar of the mobile application. This occurs even if the user does not use the C2NET application. Then, the user can check more details by opening

the notification drawer and clicking to navigate into the notification interface, as depicted in Figure 3 (c). If the user is using the application when an event to be notified occurs, the user can also check the notification by clicking the notification icon to enter into notification center.

Moreover, the notification center is an interface that shows all the notifications for a specific user. The notification service will fetch a certain number of least notification messages from the server when the user enters into the notification center. Besides, the user is able to get older notifications by scrolling down the page to the end, wherein the notification service will request the server via a web socket.

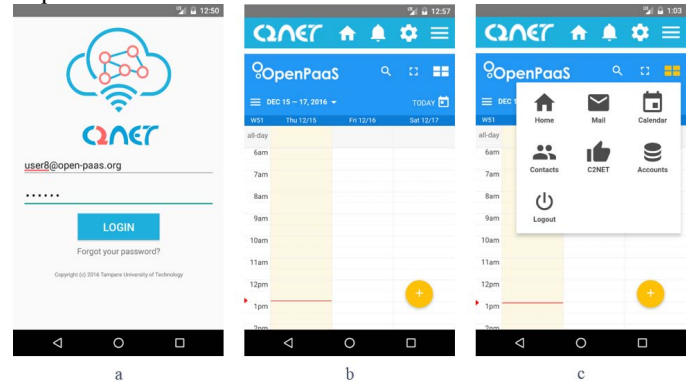


Figure 2: Login and Home user interface

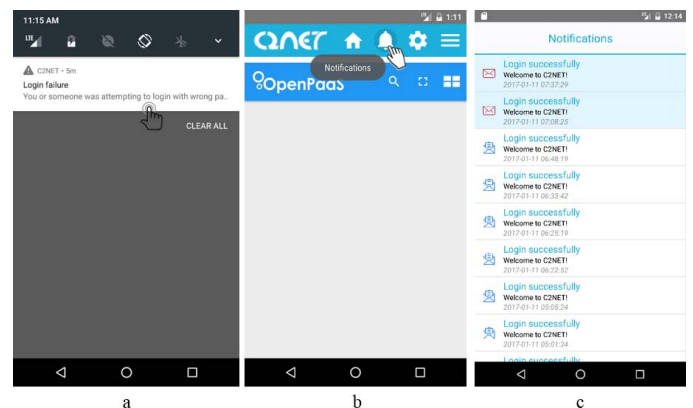


Figure 3: Notification user interface

C. Multi-user management server side

According to the presented architecture, the back-end of the mobile application manages the connected devices. The management process exploits the namespace and room features of the Socket.IO package. During the startup process on the server side, a MobileApp namespace is injected in the socket.io module by the controller. Then, within the connection request from a mobile application, a room will be created for each user since the used Identifier (ID) is listed in the request parameters of the connection. This means, each user could have more than one device connected to the C2NET platform. These devices are sharing the same room of the user that logged in. and notifications are published to all devices of the same user.

To achieve such goals, the mobile application back-end requires certain data model for this purpose. As Figure 4 shows, the user data, which are available in the OpenPaaS, it contains

ID, name and token for web socket connection. Then for each user log in using the mobile, two schemas will be created: *Notifications* and *Devices*. The devices schema contains ID of the device, name of the device (mobile model and serial number) and status that represents the connection status of the device, true for connected and false for disconnected. Lastly, the notifications' schema contains the ID of the notification, the title of the notification, which will be shown as a title in the mobile application, message of the notification, which includes the body of the notification, the status of the notification that represents read or unread and the timeStamp of the notification to show the time and the date of the notification.

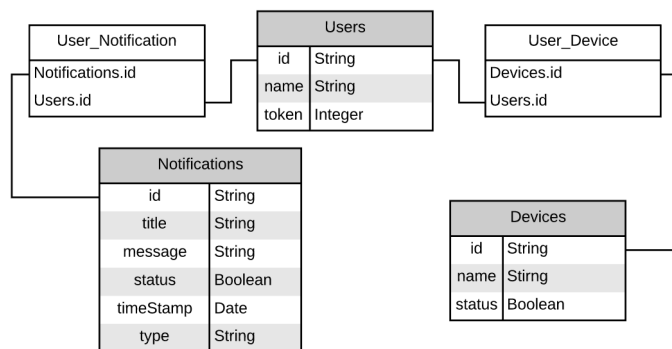


Figure 4: Data model

An internal PUBSUB module in the OpenPaaS generates the notifications. Each notification contains the used id(s) that is (are) related to the notification according to the notification type. As an example, community notifications could include more than one user. Meanwhile, messages include only one user. The type of the notification is exploited in the mobile application by showing different icons for the notification, as it can be seen in Figure 3.

V. CONCLUSION AND FURTHER WORK

This paper presents the development of a mobile application for the C2NET supply chain cloud-based platform. Mobile application facilitates the usability and accessibility between cloud-based platform and end-user, which supports user to keep informed and make quick decisions based on the real-time information and notification.

This research work presented architecture, integration and implementation of hybrid mobile application with real use cases in different scenarios. Some specific scenarios like notification pushing are analyzed to show how mobile application solve the problem of fast and secure communication between cloud-based platform and mobile frontend. Moreover, this paper also illustrates how to achieve multi-user management on the server side as well as user interface design and implement on the client side.

As further work, it is planned to enhance security, reliability and user experience in a long term. Because all these issues are essential and critical when the application ships to the production stage. Furthermore, new technology like Progressive Web Apps (PWA) will be compared with ready-

made mobile application then adopted to make it faster and offline usage possible by modern web technology. As well, the future work could include penetration test to validate the vulnerability of the system.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 636909, correspondent to the project shortly entitled C2NET, Cloud Collaborative Manufacturing Networks.

REFERENCES

- [1] M. J. Islam, B. R. Ferrer, X. Xu, A. Nieto, and J. L. M. Lastra, "Implementation of an industrial visualization model for collaborative networks," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 720–725.
- [2] R. F. Borja, N. Angelica, and L. Sergii, "C2Net | D6.6 - White Paper of C2NET platform / openness and portability - Deliverables," Aug-2016. [Online]. Available: <http://c2net-project.eu/deliverables/-/blogs/d6-6-white-paper-of-c2net-platform-openness-and-portability>. [Accessed: 13-Feb-2017].
- [3] N. Govindarajan, B. R. Ferrer, X. Xu, A. Nieto, and J. L. M. Lastra, "An approach for integrating legacy systems in the manufacturing industry," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 683–688.
- [4] L. M. Camarinha-Matos and H. Afsarmanesh, *Collaborative Networks: Reference Modeling*. Springer Science & Business Media, 2008.
- [5] J. Bastos, A. Azevedo, and R. Almeida, "Collaborative planning in customer-oriented supplier networks - The CoReNet approach," in *Technology and Innovation 2012 18th International ICE Conference on Engineering*, 2012, pp. 1–10.
- [6] G. Liotta, T. Kaihara, and G. Stecca, "Optimization and Simulation of Collaborative Networks for Sustainable Production and Transportation," *IEEE Trans. Ind. Inform.*, vol. 12, no. 1, pp. 417–424, Feb. 2016.
- [7] B. Scholten, *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. ISA, 2007.
- [8] C. Moller and S. Chaudhry, *Advances in Enterprise Information Systems II*. CRC Press, 2012.
- [9] I. V. Kozlenkova, G. T. M. Hult, D. J. Lund, J. A. Mena, and P. Kekec, "The Role of Marketing Channels in Supply Chain Management," *J. Retail.*, vol. 91, no. 4, pp. 586–609, Dec. 2015.
- [10] K. B. Hendricks, V. R. Singhal, and J. K. Stratman, "The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations," *J. Oper. Manag.*, vol. 25, no. 1, pp. 65–82, Jan. 2007.
- [11] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures," *Computer*, vol. 45, no. 12, pp. 65–72, Dec. 2012.
- [12] C. Pahl, "Containerization and the PaaS Cloud," *IEEE Cloud Comput.*, vol. 2, no. 3, pp. 24–31, May 2015.
- [13] H. J. La and S. D. Kim, "A Systematic Process for Developing High Quality SaaS Cloud Services," in *Cloud Computing*, 2009, pp. 278–289.
- [14] R. Sellami, S. Bhiri, and B. Defude, "Supporting Multi Data Stores Applications in Cloud Environments," *IEEE Trans. Serv. Comput.*, vol. 9, no. 1, pp. 59–71, Jan. 2016.
- [15] L. Jean-Pierre, "C2Net | D3.5 - 1st Modular architecture and tools for interoperability and virtualisation - Deliverables," Jan-2016. [Online]. Available: <http://c2net-project.eu/deliverables/-/blogs/d3-5-1st-modular-architecture-and-tools-for-interoperability-and-virtualisation>. [Accessed: 17-Feb-2017].
- [16] P.-O. Raúl, "C2Net | D2.9- State of Play of related projects including IPR - Deliverables," Mar-2015. [Online]. Available: <http://c2net-project.eu/deliverables/-/blogs/d2-9-state-of-play-of-related-projects-including-ipr>. [Accessed: 17-Feb-2017].