

ALBIGen - A Framework to Create Managed and Distributed Digital Signage Systems Based on Laravel and iBeacons

Christian Roschke
Hochschule Mittweida
Technikumplatz 17
09648 Mittweida
Germany
roschke@hs-mittweida.de

Rico Thomanek
Hochschule Mittweida
Technikumplatz 17
09648 Mittweida
Germany
rthomane@hs-mittweida.de

Robert Manthey
Chemnitz University of Technology
Straße der Nationen 62
09111 Chemnitz
Germany
robert.manthey@informatik.tu-chemnitz.de

Maik Benndorf
Hochschule Mittweida
Technikumplatz 17
09648 Mittweida
Germany
benndorf@hs-mittweida.de

Marc Ritter
Hochschule Mittweida
Technikumplatz 17
09648 Mittweida
Germany
ritter@hs-mittweida.de

Digital Signage Systems represent an important interface between human and machine in public institutions. By distributing services in IoT environments, dynamically generated and personalized content can be displayed. However, currently existing Digital Signage Systems are usually proprietary and difficult to maintain and administer. Furthermore, such systems require the installation and configuration of an application on the client side, which can lead to high costs for deployment and administration. This article describes a framework architecture for creating such systems, based on free and standardized technologies. The focus is on distributed and web-based communication of the components. By using the MVC-Framework Laravel to create a basic system and the extension by plugins, a platform independence, maintainability and extensibility is made possible. In the proposed architecture, external interaction mechanisms such as voice control, location-based services via iBeacons and bidirectional server-client connections are added to the basic system. The framework enables the creation of personalized Digital Signage Systems and connects them with Internet of Things components, such as sensors and actuators.

Digital Signage System, Internet of Things, location-based services, iBeacon, framework Laravel

1. INTRODUCTION

In this paper, a novel approach of a framework for creating and managing digital information and control systems based on current web technologies using Laravel and iBeacons is presented. The "audio and location based interface generator" (ALBIGen) framework enables the visualisation of personalized information via an intelligent user interface using iBeacon technology and a smartphone. In addition, connections to Internet of Things technologies are possible, especially in the Smart Home sector. The following framework can therefore also display information automatically by triggering sensors and actuators. Current information and control systems often exist as proprietary solutions based on predefined hardware and software. Because of missing and non-standardized interfaces, extensions and user-

specific adaptations are only partially insufficiently realizable. The development of web-based systems for displaying specific information based on an MVC framework has already been discussed in several articles. Current work in the papers Anif et al. (2017), Ardiyani et al. (2016), Chin et al. (2017) or Park et al. (2017) describe specific architectures for displaying information using distributed systems. None of these developments focuses on connecting such systems with voice control and location-based services using iBeacons to display personalized content. Likewise, neither the administration nor the development effort was considered in the context of minimization. The architecture of the framework described in this article makes it possible to design an information system, to maintain it easily and to automatically generate and display dynamic content in a user-oriented manner using standardized technologies.

2. SYSTEM ARCHITECTURE

We chose a web-based architecture as the basis because it offers several advantages over native applications. For example, nothing needs to be installed on the client side, since the application runs directly in the browser. Another advantage of the web-based software is the automatic backup of data in a central database. Each transaction of a user is time-exactly related to the current data stock and is stored time-exactly to these.

In the area of web development there are various frameworks to support the development process. We chose the MVC Framework Laravel for ALBIGen, because the logic can be distributed arbitrarily to perform calculations on the server, client and database side. Furthermore, the clear separation of logics enables easy support and enhancement. (Chen et al. 2017)

To show personalized content on a display, we use iBeacons for person recognition. The user's smartphone detects the presence of an iBeacon and sends control commands to the associated display. The display is connected to the server via a unidirectional connection and can receive and evaluate events and visualize personalized content based on them.

In addition to this method, we use speech recognition to customize the content. Voice commands can be used to trigger actions such as highlighting a defined element.

In addition, we link each information display with sensors and actuators from the Smart Home area using a management system. This allows commands to be sent to a defined display after an actuator has been triggered or a value read from a sensor exceeds or falls below a predefined limit.

2.1 Laravel as a basic framework

The open source framework Laravel follows the development pattern Model View Controller (MVC) and is licensed under MIT. Laravel offers multiple components in the standard installation, which can be used by developers for larger and more complex developments. In addition to the integration of Laravel modules, Symfony components can also be integrated. (Anif et. al 2017)

Laravel has a routing module that allows to forward HTTP requests to a controller. It is feasible to differentiate which request method is used and which of the functions are to be called up. This makes it possible to deliver different views on the

basis of the URL and to transfer parameters for requests. The controller processes the transferred parameters and initiates the use of a model to perform operations on the database. The controller is used to pass content from the database to a view. For this purpose, data entered in a form can be checked, database entries queried, views loaded and files uploaded into the system. (Yu 2014)

2.1.1 Events

We chose Server-Sent-Events for the communication between client and server, after the view was transferred. Based on this technology it is possible to establish and maintain a persistent unidirectional connection between the client and the server. The used EventSource API is standardized in HTML5. To open a connection to the server to receive events, an EventSource object with the reference to a server script must be created. When the connection is established, the client starts listening to message events. As soon as a message has been received, it can be integrated into the current view.

2.1.2 The holistic web-based architecture

Figure 2 shows the Laravel-based architecture we developed. The classic Laravel components were supplemented by a plugin manager, which enables the administration of plugins. This enabled us to develop components for communication with iBeacons and extend the classic views to include personalized content.

Furthermore, we have integrated a client-side system that simplifies communication with the server. The Communication Handler manages the exchange of information and forwards it to the controller for processing. A generic exchange format is generated there and made available to the UI or an API for adequate representation or client-side manipulation.

The resulting architecture is based on the client-server principle. The Client Controller forwards requests to the Communication Handler as the interface to the server application. The handler handles sending and receiving messages via AJAX, direct connections, server-sent events and web sockets. The server main system contains the complete management logic, such as user administration, protection of application areas and the management of views and plug-ins. The use of plugins makes it easy to extend the functional core and avoids the risk of negatively affecting the core logic.

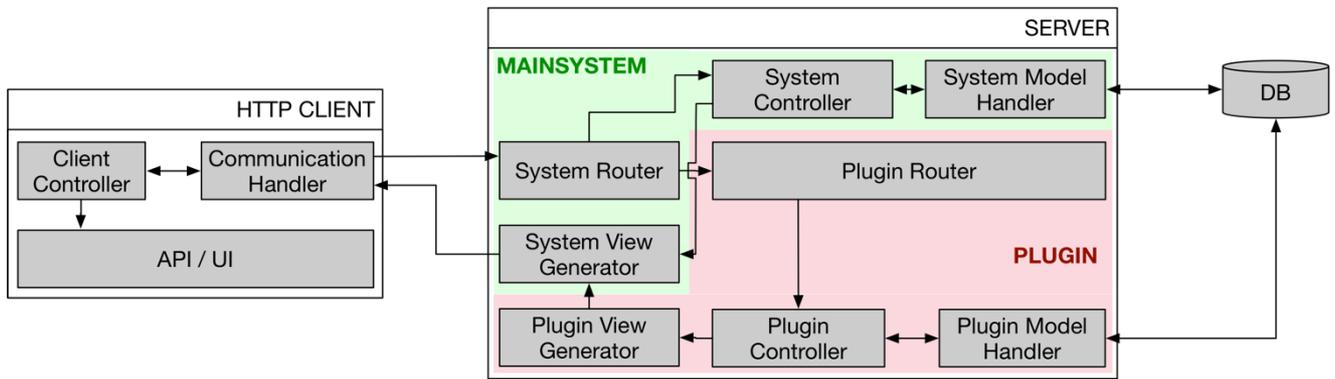


Figure 2: the holistic web-based architecture

2.2 iBeacon

iBeacon is a technology introduced by Apple in 2013 that can be used for indoor navigation. The iBeacon devices emit a low energy Bluetooth network for this purpose. Using the Proximity Beacon Advertising Package (PBA package) sent as a broadcast, mobile applications are able to detect their position on a microlocal level and provide the user with contextual content depending on the location. The following three parameters of the PBA package must be adjusted: Proximity UUID, Major and Minor. (Apple, 2015)

2.2.1 Usage in the Framework

ALBIGen enables the display of personalized information on the interfaces using iBeacon technology. For this purpose, an iBeacon is placed near the interface. The UUID is a 16 byte flag for service-specific identification. With this ID a mobile application is able to detect the iBeacon. The major and minor values are unsigned integers between 0 and 65535 and are used to identify iBeacons with greater accuracy than with UUID alone. ALBIGen uses the major value to identify the location (e.g. building number). The minor value is still used to distinguish the interface.

2.2.2 Extended Finite State Machine

For the detection of the iBeacon on a smartphone, we have developed a prototype iOS app. iOS enables the detection of iBeacons using two mechanisms, monitoring and ranging. Monitoring enables the detection of iBeacons even when the application is finished. However, only the detection of entering and leaving the iBeacon area is supported. If the application is not active during a detection, iOS starts it for a few seconds in the background to handle the event. Please note that time-consuming actions cannot be performed.

Ranging can only be used if the application is active or has only recently been in the background, making time-consuming actions possible. The ranging mechanism can also be used to determine the

distance between the smartphone and iBeacon. (Köhne et al. , 2014)

We use both mechanisms to display personalized information, monitoring when the app is closed and ranging when it is in the background or active. When using Ranging, the PBA packet byte "Measured Power" is also used to determine the distance to the interface. This enables us to only display personalized information if the user is no more than 3m away from the interface, for example.

2.3 Speech control

In addition to control using iBeacon technology, we also use speech control. The Web Speech API used is a specification of the Speech API Community Group and enables the use of functions for speech recognition via JavaScript at the client.

How the functions of the API are made available is not defined. We use Google Chrome and the Google Speech API. Spoken words are sent to external servers, analyzed by neural network models and the results are returned as text. The API recognizes over 110 languages and enables their transcription.

2.4 Remote control

Because of the possibility to transmit control commands via HTTP to ALBIGen, any REST clients or systems with HTTP client support can be used for remote control and so can be used completely autonomously by ALBIGen. Due to this architectural concept, existing management systems from the IoT area can also perform automated control processes.

For remote control of ALBIGen we use the Smart Home System "ZWay" as an example. ZWay is a software solution available for various operating systems and can be used as a Z-Wave controller. Z-Wave is an international ITU-T standard for home automation. (G.9959, 2015), (SDS11847, 2017) ZWay also offers user management and a convenient user interface that can be used to create rules for automation and virtual devices (e.g. switch) for sending HTTP requests. Especially the

connection of sensors and actuators is an interesting application.

3. RESULT AND DISCUSSIONS

Based on the developed framework we were able to design a prototype and integrate it in the University of Mittweida. The developed prototype showed that the underlying architecture offers several benefits compared to conventional proprietary solutions. The system is modular, expandable, platform-independent and allows control by other software using a standardized REST API. Furthermore, development is open source and can therefore be integrated into heterogeneous infrastructures without further costs. Furthermore, Laravel's modular design offers a stable and well documented basis. By distributing the logic, it is possible to regenerate content dynamically and make adjustments at runtime. We were able to minimize the administration and maintenance effort by using generic routes and the instantiated, controllable interface instances. The bidirectional communication between mobile devices and the individual clients also enables the display of personalized content and thus a high target group accuracy.

3.1 Deficits and room for improvement

We currently see deficits and potential for improvement in the detection time for detecting the iBeacon area. To not shorten the battery life of the smartphone unnecessarily, the operating system automatically decides how short the polling times for finding iBeacons should be. Therefore it cannot be guaranteed that the personalized information will be displayed immediately when entering the iBeacon area. Rapid detection of an iBeacon can only be guaranteed if the app is in the ranging mechanism. However, the ranging mechanism is only activated after entering an iBeacon area.

Placing several additional iBeacons in the environment of the interface with a different UUID could therefore lead to the app being placed in the ranging mechanism before entering the interface area. This would also allow the app to quickly find the iBeacon used to display the personalized information. However, this procedure only works if the app is active or running in the background.

4. CONCLUSIONS

ALBIGen enabled us to implement a digital signage system at the Hochschule Mittweida (University of Applied Sciences). The system has already been stable in use for several months and was able to display specific information distributed across several clients. The developed architecture proved

to be extremely stable, low-maintenance and easily expandable. The system currently displayed specific information for approx. 5200 users for 16 weeks as expected and we could not detect any failures. From this it can be assumed that the framework can be used to develop and individually adapt reliable Digital Signage Systems for public institutions. The identified deficiencies are currently being processed successively in further iteration processes and the framework is being continuously improved.

In the future, the framework could be extended in the form of plug-ins with additional functionalities such as a manageable warning system, employee information systems and indoor navigation using iBeacon location determination. Furthermore, we plan an evaluation in the context of scalability and reliability.

5. REFERENCES

- Anif, M., Dentha, A. & Sindung, H. (2017) Designing internship monitoring system web based with Laravel framework. IEEE International Conference on Communication, Networks and Satellite (Comnetsat).
- Apple (2015) Proximity Beacon Specification Release R1. <https://developer.apple.com/ibeacon> (6th April 2018).
- Ardiyani, R., Arham, Z. & Rustamaji, E. (2016) The Development of a Web-based Spatial Information System Utilization of Forest Area (Case Study : Sulawesi Island). International Conference on Cyber and IT Service Management.
- Chen, X., Ji, Z., Fan, Y. & Zhan, Y. (2017) Restful API Architecture Based on Laravel Framework. Journal of Physics: Conference Series, Volume 910, conference 1.
- Chin, T., Chuang, Y, Fan, Y, Jiang, Y, Kang, Y, Kuo, W, To, T & Nishino, H. (2017) Prototyping digital signage systems with high-low tech interfaces. In SIGGRAPH Asia 2017 Posters (SA '17). ACM, New York, NY, USA, Article 18.
- G.9959 (2015) Short range narrow-band digital radiocommunication transceivers – PHY, MAC, SAR and LLC layer specifications. ITU-T.
- Hickson, I. (2015) Server-Sent Events W3C Recommendation. <https://www.w3.org/TR/events> ource (6th April 2018).
- Köhne, M., & Sieck, J. (2014). Location-based Services with iBeacon Technology. Second International Conference on Artificial Intelligence, Modelling and Simulation, 315-321.
- Park, Y., Yang, H., Dinh, T., Kim, Y. (2017) Design and implementation of a container-based virtual client architecture for interactive digital signage

systems. International Journal of Distributed Sensor Networks, Volume 13.

SDS11847 (2017) Z-Wave Plus Device Type Specification. Sigma Designs.

Yu, H. R. (2014) Designing internship monitoring system web based with Laravel framework. International Conference on Computer Science and Electronic Technology (ICCSET 2014).