BMM Platform: a booking, management and monitoring service for the IST Shuttle service

Duarte Barbosa
Instituto Superior Técnico / INESC-ID / ULisboa
Av. Cavaco Silva, Taguspark, Oeiras, Portugal
duarte.barbosa@tecnico.ulisboa.pt

Abstract—This paper proposes a dynamic and real-time booking, management and monitoring system for the IST shuttle service. Members of the community can access, in real-time, the central platform to book a seat or monitor the fleet of bus. An embedded system is deployed in each bus that grants access to authorized users, giving priority to the ones that have booked seats. This systems periodically communicates with the central platform, to inform about the location, speed and seat occupancy. BMM platform was validated in a field trial experiments and the results have shown that the system is suitable for use and improves the user experience of the shuttle service.

Keywords—Mobility management, Real-time Booking system, Shuttle service

I. INTRODUCTION

In the past few decades, there was an enormous evolution in the ICT [1] [2]: cloud computing, low cost embedded systems equipped with sensing devices, GPS and advanced wireless communications are now the basis of the Internet. To face the sustainability challenges that we face nowadays, a lot of new technologies and products are emerging that will define the Internet of the future.

One of the most critical problems in modern societies is mobility and the technological revolution is creating, once more, the means to provide new approaches to the existing problems [3]. Different research and standardization initiatives are emerging in this area and a significant effort to develop new architectures are creating new market opportunities and products [4] [5].

With a growing mobility problem, the shuttle service is one of the most important solutions for current difficulties in moving between Alameda and Taguspark IST campi. However, the current service still has several limitations, since IST can not adjust it to the demand and users do not have any guarantee of being transported, specially at rush hours. This thesis aims at providing a contribution to this problem, by providing a Booking, Management and Monitoring (BMM) service for the IST Shuttle service.

The paper is organized as follows: section II describes the back-ground and related work; in section III the architecture of our system is presented and, in section IV, the implementation; the validation of the platform is described in section V and, the conclusions, in section VI.

II. BACKGROUND AND RELATED WORK

A. Electronic identities

Electronic identities are pretty common these days; what once started as a simple way of people communicating with each other using names, e-mail addresses or alias has evolved into useful tools for asserting the real identity of a given person. This goal poses multiple security issues and across all Europe e-ID cards are being developed and deployed [6]. Some e-ID cards offer three forms of information security: Identification (I), Authentication (A) and Signature (S) [7]. We are particularly interested in Identification and Authentication, as they will provide the basis of our system. However, this can only be accomplished if the system is capable of verifying that both attributed identity and biographical identity for a given user are the same [8]. The ability to link a set of information to its owner and the effective and secure handling of entity-specific data are essential to numerous different interactions [9].

B. Access control

Access control can be defined as protection of system resources against unauthorized access [10]. To perform adequate access control, one must be able to verify a claimed identity and confirm the access rights to that resource. Several methods exist; we will focus on electronic tickets and booking systems.

1) Electronic ticket: An electronic or digital ticket is a way to claim some goods or services by means of digitalization of rights. It is commonly used for access control to public transportation. It is a voucher or certificate that guarantees that the owner has the right to claim the services written on the electronic ticket [11]. An issue transaction is an action in which the issuer gives ownership of an electronic ticket to the user. A transaction is when an electronic ticket is verified by the service provider which fulfills the service represented by the electronic ticket; the ticket is then consumed by the checking machine. Both security and service quality are equally important for an effective electronic ticket system.

Some of the technologies used for implementing an Electronic Ticket system include barcode, QR code, smart card, RFID and NFC [12]. Regarding railroad transports, passengers are usually required to pay a booked seat upfront with their train tickets.
2) Booking systems: Many booking systems have been proposed over the years. Classical solutions exists in the majority of transportation systems. Flight companies [13] generally employ authentication of the ticket by barcode or QR code and an identifying document of the passenger, like a passport. The bus services on the other hand do not have any special constraint. Passengers buy a ticket, find an empty place and enjoy the trip.

A similar area where significant technological advances occur is the area of parking lots systems. Smart applications using RFID have been researched [14]. VANET technology has also been used in this context [15]. Finding a vacant parking space is hard and taking advantage of the facilities provided by VANET may help you in that demand. Intelligent parking services application may help to save fuel and drivers parking time. In some ways, our problem is quite similar to finding a vacant parking space; we need to be able to distribute in real-time and in an efficient the remaining seats of a given shuttle. That author of [16] propose a new Reservation-based Smart Parking System. Multimodal systems are used for storing and traveling within various transportations means. One of those systems is the Lisboa Viva card; a contact and contactless smart card used to store tickets and passes that provide access to public transports in the Lisbon Metropolitan Area. It is based on the Calypso platform [17], a set of technical specifications developed since 1990 by a consortium of European transport operators.

III. BMM ARCHITECTURE

A. Main components

The BMM platform, as depicted in Figure 1, comprises three main modules: Shuver, BMM Clients and YAEC.

Fig. 1: System architecture

The central node of our architecture is the Shuver. It collects and processes all the information flow from YAEC and it is responsible for the heavy lifting behind the BMM logic. The Shuver is responsible for managing the booking process, maintaining updated information for each route, providing reports of the status of the service and managing options to users and administrators. This component acts as a sink node; it collects and provides information in real-time to whom requests it. Therefore, all the data must be updated and be available as soon as possible. There are no real-time constraints for the availability of the information; yet, if its severely outdated, it may lose any relevance. The services and hosting provided by Shuver are exposed by an API. End-users cannot access it direct.

BMM clients present and expose the available services and data. This is accomplished by interacting with Shuver, receiving information and sending requests. Those messages are then translated and presented to the end-user. There are no strict rules regarding implementation or interface; BMM clients may be web and mobile applications or even command line interfaces. This is the public facade of the BMM service and may be implemented in multiple platforms requiring only Internet access.

Finally, the embedded system available in each vehicle is called YAEC and is responsible for the access control upon entering the shuttle and provides vehicle and journey data. The YAEC allows efficient authentication and authorization every time someone hops in the shuttle and provides journey and vehicle data. This information can be the occupation data and the location of the shuttle or its route; access to this information may extend the scope of applications and possibilities our underlying technology can provide. YAEC synchronizes periodically with Shuver, receiving booking information and sending the collected data.

BMM clients and YAEC interact with Shuver in a client-server architecture. The deployed solution will be composed by several YAECs in the field and one Shuver in a 1-to-Many Architecture. Similarly, several BMM clients may connect to the same Shuver.

B. Interactions

1) BMM service clients and Shuver: Prior to being able to access Shuver’s services, BMM service clients need to authenticate the user. This is done requesting the (possibly external) authentication service to identify the user.

Once the users are authenticated, the BMM service clients are able to connect to Shuver and fetch the status of the service. From this moment on, the services are available to the end-user.

Booking, management and monitoring all follow the same pattern: BMM clients issue requests to Shuver and its responses are shown to the user. There are no middlemen in this case; the user is already authenticated and the client is known to Shuver.

The messages interchanged between the BMM Service clients and Shuver are depicted in Figure 2.

2) YAEC and Shuver: Once YAEC is powered on, it connects to Shuver and retrieves an updated list of all the booked seats for that node and day. This request takes an arbitrary large time depending on the number of booked seats and current YAEC state. As soon as the shuttle closes its doors and starts the journey, all the passengers have already been authenticated. YAEC is now able to notify Shuver who were the users that hop into the shuttle and the ones that occupied
Fig. 2: Messages between BMM service clients and the Shuver

booked seats; YAEC will upload this list to Shuver and once it is acknowledged, it will be considered in a sane state. YAEC also sends monitoring updates periodically. These have a fixed size and are sent with the same $\Delta t$ time interval.

It is important to note that delivery is not guaranteed. YAEC tries to synchronize itself to a sane state and once it successfully accomplishes this, it just broadcasts the monitoring messages.

Figure 3 summarizes the messages and their order between the BMM Service clients and Shuver.

IV. BMM IMPLEMENTATION

A. Shuver

Shuver is the central node of our architecture and offers an API that exposes access to the BMM functionalities.

Shuver was implemented in Python 2.7 and runs as a standalone process which spawns a thread for each of the BMM components. The database is external though; a MongoDB instance is launched upon system boot and we access it directly using a custom driver and helper wrapper.

Shuver exposes its services as a publicly available RESTful API. Clients that want to consume this API, need only to connect to Shuver and send GET and POST requests to the desired endpoints. Shuver runs as a headless server; there are no GUI’s to describe. Shuver only accepts requests to its endpoints from known Clients. As such, any Client must be registered in Fenix and its keys need to be manually introduced in Shuver. Any user that wants to access Shuver services needs also to authorize the client to access his own personal information inside Fenix. Once everything is set up, the client will be able to send requests to Shuver.

B. BMM Client

The BMM client was developed as a Single Page Application. The goal was to build a fluid experience akin to the ones most users are already used to. AngularJS allows one to declarative program the DOM. From a myriad of different web frameworks, the MEAN stack was used because it was conceived exactly for this kind of application. As of February 2015, this is a popular stack for responsive web development. Since we are not building native applications but rather a responsive web application, this stacks abstract lots of required work. Although we do not actually need MongoDB running in the prototype, it was originally used as the DB of the system and remains as a useful leftover (it allows us to verify requests prior to being sent to Shuver).

The BMM clients run completely in client-side and thus, do not store any user data. An user is forced to authenticate in Fenix and all consequent data exchanges are performed just to identify the user (between the BMM platform and Fenix Framework).

Figure 4 depicts an actual screenshot of the booking functionality.

C. YAEC

YAEC ensures an efficient access control every time someone hops in the shuttle and provides journey data. YAEC
Fig. 4: Booking area

was developed as an embedded system to be deployed in the entrance of the shuttle. Upon entering, users authenticate themselves using either the official IST card or a specially crafted QR code. YAEC ensures that booked seats are available and manages the data stream to Shuver, requesting the list of booked seats and notifying Shuver of the current passengers as well.

YAEC was implement in Python 2.7 and includes code in C and bash scripting. Makefiles are available for building and deploying the system.

YAEC initial process spawns a different thread for each module. All the synchronization is made using a message queue that each of the modules listens too.

Figure 5 depicts the actual physical configuration of YAEC.

V. BMM VALIDATION

In order to start deploying the system in the shuttles, we performed a full test where the different aspects of the platform were assessed in real operational conditions. The next section details the tests that were realized for this purpose.

A. Goals of the test

The aim of these tests is to perform a full evaluation of the BMM service so that one can identify whether the system is ready to be deployed or not. As the QR-Code reader had not arrived at the date of the test, the only missing part was the authentication using QR codes.

The tests were divided in three distinct groups that correspond to the way the service is implemented. They comprise: the booking in the BMM service client; the authentication and access control in the shuttle; and the monitoring of service.

B. Test preparation and schedule

In order to run these tests, an YAEC device must be deployed in a bus of the IST shuttle service. The access was granted by fleet operator that owns and provides the shuttle service to IST. Unfortunately, the vehicle did not had an OBD-II port. The vehicle used during the test is the one that is permanently allocated to IST and the driver was informed in advance why, when and how the test was going to be realized and what type of cooperation was expected.

A special attention was dedicated to the communication with the IST community and an institutional email was sent the day before, with detailed information regarding the service, booking policies and access conditions to the vehicle, as well as the scheduled date and journeys of the test. It was clearly mentioned that:

- There would be only one booking per journey and per user;
- The booking was not transmissible;
- The bookings were only available to the IST community using Fenix identity;
- For evaluation purposes, only 10 seats per journey would be available for booking;
- Every passenger that hopped into the vehicle had to authenticate himself, on a card reader available at the entrance of the vehicle;
- For this purpose the passenger had to have a valid IST card, otherwise, exceptionally, he could use the CC.

To restrict the impact of the test on the shuttle service we decided to realized the test in a less busy day. Hence, we selected the last day before holidays to run the tests: the 31/03/2015, which is the last day before Easter holidays. We chose three distinct journeys on the same day for evaluation purposes: from Taguspark to Alameda at 15:10 and 17:10 and from Alameda to Taguspark at 16:10. Those journeys were chosen based on the occupation information we gathered from the IST staff and in order to test the system in both rush-hour and in schedules with less passengers.

C. Scenario description

The preparation of a full operational test required the entire system working properly. To achieve this, both Shuver and
BMM service clients were installed and tested in advance, in order to being used without interruptions by the IST community during the test.

The Shuver and BMM service client systems were deploying in a VM of the institutional domain. The BMM service client went live some days before.

In the day of the test, before passengers hoping into the shuttle, YAEC was deployed in the vehicle by connecting the power source to the cigarette lighter socket. In order to guarantee that every passenger is authenticated, the card-reader is placed in a visible place, near the entrance. Figure 6 illustrates the YAEC system in the vehicle, at the beginning of the test.

![Fig. 6: YAEC in operation inside the shuttle](image)

Since there were some problems with the power consumption and current stability, the YAEC used an Wi-Fi network adapter instead of the 3G modem. A 4G enabled smartphone (LG-Google Nexus 5) provided a tethering connection to YAEC which was used for all the communication purposes. Given that the QR code reader had not arrived by the evaluation date, all the authentication was made using smart-cards instead. The smart-card reader was initially coupled to the YAEC prototype box but since the power cables were getting in between the driver and the steering wheel, we decoupled the device and unfolded the cable.

At the same time passengers hopped in, they also authenticated themselves. The first two journeys (round-trip) also evaluated the entrance of passengers in intermediate stops. Unfortunately, due to power issues, the initial authentication of the 16:10 shuttle was not performed. Since it began to work properly shortly after, it did manage to authenticate passengers hopping in Sete-Rios and a full authentication was repeated in the end of the journey. The test ended with the departure of the 17:10 shuttle from Taguspark.

D. Results and analysis

1) BMM service client: The first step of our analysis of the full operational test is focused on the first interaction of the users with the system: they need to access the BMM service client to book seats.

The evaluation of the BMM service client comprises the two stages that are needed to fulfill this task: the access to the BMM service client and the booking process. The next sections details each one of them.

Access to BMM service client – To assess how users interact with our platform some raw information was retrieved. Google analytics was used to gather the number of pageviews, sessions and users. The logs of the BMM platform were used to retrieve the number of BMM authorized users and bookings. This information was gathered at four different instants of time;

- t0 - immediately before the email was sent.
- t+6h - before the first journey starts.
- t+24h - the day after the tests
- t+48h - two days after the tests.

Based on that, several different metrics have been defined that measures different level of interactions with the systems, as follows:

- **User Conversation Rate (UCR)** - defines how many users that access the website authenticate themselves through Fenix credentials and is given by:
  \[
  UCR = \frac{Users}{Pageviews}.
  \]

- **BMM platform conversion Rate (BMMCR)** - defines how many Fenix authenticated users are converted into BMM platform users and is given by:
  \[
  BMMCR = \frac{BMMUsers}{Users}.
  \]

- **Booking Conversion Rate (BCR)** - defines how many users performs books requests and is given by
  \[
  BCR = \frac{Booking}{BMMUsers}.
  \]

Using the information of this table, the UCR, BMMCR and BCR were computed. Figure 7 depicts the results that have been achieved.

![Fig. 7: Assessment of user interaction with the BMM client platform](image)
own accesses to the service. Similarly, the authorized users and booking requests were made by ourselves to test various parameters of the system.

Things got interesting shortly after; as the service went live, all the metrics skyrocketed, managing to achieve 78 authorized users with 23 booking requests in just 6 hours.

As stated in the figure, the UCR was fairly high before the field test and most of those users were actually converting into BMM service users. As the time passed by, the rate of new users dropped but the BMMCR is constantly high; this indicates that there is interest in the service. Still, this interest did not manifest itself in the number of booking requests which remained fairly low. Surprisingly, even after the booking period, the rate of booking requests remained constant.

**Booking process** – The next stage of the evaluation of the BMM service client is related with the booking process. Using the logs of the BMM platform, three different type of data have been retrieved:

- **Valid booking request** - the number of valid booking requests that have been performed by BMM authorized users. This number accounts for one booking per BMM authenticated user per journey.
- **Invalid booking request** - the number of invalid booking requests that have been performed by BMM authorized users. This number accounts for all the remaining cases.
- **Granted booking** - the number of booking requests that have been granted. This accounts for the number of valid requests of BMM users that have credits to book seats, under the maximum allowed number of bookings that was defined.

The booking process is one of the most important aspects for the success of our service. In order to assess it, two different metrics have been defined:

- **Booking request Success Rate (BreqSR)** - defines the percentage of valid booking request and is given by: \[ BreqSR = \frac{\text{Validbookingrequests}}{\text{Totalbookingrequests}}. \]
- **Granted booking Success Rate (GbSR)** - defines the percentage of valid bookings that converts into granted bookings and is given by: \[ GbSR = \frac{\text{Grantedbooking}}{\text{Validbooking}}. \]

Figure 8 depicts the success rates of the booking request (BreqSR) and granted booking (GbSR) processes.

The analysis of figure showed that more than 50% of invalid booking requests have been registered. This means that, either the users failed to set the date, or just repeated the same request. Considering just the valid requests, almost all the users had their request accepted, as the value GbSR is close to 100%. This happened because for most of them, there is no past history to consider and all the users that booked seats had enough ranking to do it. Therefore, none of them lost ranking because they failed to authenticated in a previous journey where he booked a seat; neither gained more credits due to his very good behavior. The only two exception were ourselves, who previously seeded the database in order to get a failed booking request.

2) **Access control in the vehicle**: After deploying our service, users need to authenticate themselves in order to access the IST shuttle, as access will be granted only to authorized users. Hence, the next step of our evaluation is focused on the shuttle access control and authorization processes. The next section details both tests.

a) **Authentication process**: In order to perform an effective access control, every user that hops in the vehicle must use the IST Card for authentication purposes. However, given the results of the previous tests, we were not quite convinced about the possibility of having the great majority of users with this card. Hence, we decided to repeat the authentication tests to be sure of the basic assumptions that need to be guaranteed before the deployment of the service.

Figure 9 illustrates the authentication process of a user hoping into the shuttle.

**Access Control** – The authentication module of YAEC ensures shuttle passengers are who they claim to be. This means that, when using our system, access to the bus should only be granted to authenticated users. However, to ensure that they do not have reasons to complain about the service, the process...
must be fast enough to avoid the creation of long queues of users that are waiting for their time to do it and hop in the vehicle. Hence, the authentication must be correctly performed and fast.

The authentication at the entrance of the shuttle introduces an additional overhead and might delay the users entrance in the bus. Therefore, the goal was to validate that the new procedure would neither deteriorate passengers experience nor cause any unsustainable overhead. As part of the evaluation process, each time a smart-card was used for authentication and some time measurements were taken. Specifically:

• When a card was detected;
• When a card was identified and authenticated;
• When any error occurred.

This information was used to compute the Authentication Response Time (ART). This is the time since the card is detected until the result of the detection is generated. This result may represent a card that was identified and authenticated or an error.

Based on the information that was gathered, we computed the ART for each one of values for the two cards and for the errors situations. The results are shown in figure 10 and figure 11.

According to the figures, the time taken to authenticate when using the IST Card or the CC card are quite similar, although the former is slightly lower. This can be explained by the extra number of queries made for CC. Nevertheless, the variation of the ART for the IST Card is higher than the one observed at the CC. This is caused by the usage of the contactless interface instead of the contact one.

Concerning the detection of error situations, the value increases significantly, reaching near 2 s. Again, the variation is higher because of the same interface issue. Moreover, the IST card also offers a contact interface, which users sometimes tried to use instead of the contactless one. Given that approaching the slot is enough to power up the card and we were constantly helping users, these values although high may be considered normal.

Authorization – Once the booking period ends, a set of users will be automatically authorized upon authentication. On the other hand, not only those users are authorized to enter the vehicle but a handful of others. This experiment took in account the already authenticated users and the ones that lacking the booking were also authorized to enter the vehicle. Figure 12 depicts the overall results of this process.

Every user got access to the vehicle. Although there were few granted booking requests for each journey, the number of users with bookings that hopped in the shuttle do not match. In fact, this number was much lower maybe due to the fact that the majority tried to authenticate with the CC instead of the IST card. Moreover, a number of people had no identification at all or hopped in without using the system.

3) Monitoring of the service: A lot of information can be retrieved from the shuttle during a journey and used for monitoring purposes. Hence, the last step of our evaluation is focused on different types of information that might be retrieved from the YAEC. The next section details these results.
Journey duration: Given that passengers hopping in the shuttle need to authenticate themselves, it is possible to extrapolate a lot of information just by looking at the time those authentications were made. Every-time YAEC was powered on, a “begin” event was recorded with a timestamp. The shuttle departs only after all passengers get on board (obviously, with a schedule constraint); since an authentication is made for each one, we can safely say that the departure time is later than the moment the last passenger authenticated himself. On the other hand, passengers may hop in the vehicle on intermediate stops; this allows us to extrapolate the moment the shuttle stops in those points just by following the authentication timestamps and finding a larger gap than usual (typically around minutes). YAEC records an “end” event once the authentication module turns off the devices. This allows us to box all the values in a given time range.

Figure 13 depicts the YAEC times for beginning, departure, intermediate stops and end of each journey.

Fig. 13: Journey’s times

VI. CONCLUSIONS

Is it possible to devise a real-time booking system for the IST Shuttle service that is able to support a controlled access to the shuttle for the IST community and guests?

To address this question, we proposed the Booking, Management and Monitoring service. The BMM service extends the existing Shuttle service by allowing the IST community and other authorized individuals to book seats, manage different options regarding the configuration of the service and monitor the stats of the shuttle and the vehicle.

With the introduction of this new system we intend to make the shuttle service more efficient and to provide the tools to manage it in a more optimal way. This is achieved by a set of novel components:

- An embedded system available in each vehicle (YAEC), responsible for the access control upon entering the shuttle and providing vehicle and journey data.
- A central node (Shuver), which collects and processes all the information flow from YAEC and it is responsible for the heavy lifting behind the BMM logic.
- BMM clients which present and expose the available services and data.

The field tests gave us a very positive feedback, with high conversation rates and proving this service is non-disruptive and an effort in the right path. We can conclude that the answer to the original question is indeed possible. As always, compromises were made; however, we feel this solution has more advantages than disadvantages and brings new power to the management entities. Furthermore, this thesis has set the foundations and enabled future work to take place, specifically:

- YAEC suffers a lot from power issues. Support for other power sources like solar or battery and improved energy consumption shall be researched. The notifications from YAEC shall be improved in order to overcome user experience issues. YAEC may also provide Wi-Fi onboard.
- The BMM client shall be extended with analytics of the service and more monitoring and management options. It may even be integrated with Fenix and Fenix Mobile.
- The service may use VANETS as a routing system for network messages.
- There is no reason why this work can not be extensible to other vehicle providers and institutions.

REFERENCES


