BULGARIAN ACADEMY OF SCIENCES

PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2023 • Vol. **79**, pp. 3-36
p-ISSN: 2738-7356; e-ISSN: ISSN: 2738-7364
https://doi.org/10.7546/PECR.79.23.01

Sharing Local Resources Within a Community by Enhancing the Potential of Eduroam and EduVPN with Mobile Application for Remote and Local Resources and Through Secure User Identification Over the Network (MARLIN)

Radoslav Yoshinov[1] *and Oleg Iliev*[1]

[1] *Laboratory of telematics at Bulgarian Academy of Sciences, Sofia, Bulgaria*
E-mails: yoshinov@cc.bas.bg, ilievo@cc.bas.bg

Abstract: Universities and scientific organizations share a community with enormous potential. In the last decade, a lot has been done to support the collaboration in this community – eduGAIN, eduroam, eduVPN were created. At the same time, these technologies have their limitations, which do not allow the users to benefit from the local resources of the organizations (printing resources, access to labs, etc.) without requiring the intervention of administrators, support, etc. Solving this challenge not necessarily should be complex to implement and to require a substantial investment. The smartphones we hold in our pockets could be used as biometric readers (fingerprints, face recognition, etc.), so we may have full user identification. Moreover, these smart devices can read QR codes and RFID tags. They could be integrated into a model of a promising solution suggested by this research aiming to provide the community an ability to enhance the ways for interfederation collaboration.

Keywords: *User Identification, Sharing, Remote and Local Resources, Eduroam and EduVPN.*

1. Introduction

The scientific organizations collaborate by working on projects, scientific researches, training programs, etc. This way, the organizations form a powerful community with enormous potential. There are various solutions that aim to support the scientific institutions in the process of providing local resources to the members of the community, such as eduGAIN, eduroam, eduVPN, but at the same time these solutions cannot provide the users with an access to local resources managed by an organization, which is not his or her own organization [1, 21, 22]. In other words, if the user is temporary on the territory of another organization, he or she cannot use any of the local resources of this organization. This limits the capabilities of the community.

This study is focused on the ability of scientists, members of one scientific organization, to access local resources of another scientific organization while they are located on its territory, but at the same time, the authentication of users is performed by their own scientific organization.

According to the stats of published by Statista Research Department in August 2022, around 80% of the population of the world has a smartphone or another smart device [14]. In this regard, some authors deal with developing questionnaires to conduct the survey to influence of smart technologies over human body [8]. The current research aims to offer a novel way to access local resources by simply reading QR codes and using secure authentication and authorization mechanism, together with the ability to acknowledge the user's identity, using biometric data though smart devices. In other words, we are talking about optimizing the processes necessary to provide access to local resources, but not at the cost of lowering security. At the same time, the organizations do not need to make huge investments in infrastructure and equipment such as GPS navigation or IoT [12, 13] even development of complex software systems [9, 11].

The research presents a prototype of a native application called MARLIN (Mobile Application for Remote and Local resources and through secure user Identification over the Network) that may be easily integrated with every scientific institution over the world. Moreover, the native app is integrated with a prototype of an API and Administrative Web System that aims to represent an actual system that could be easily integrated with the organizations' infrastructure and thus provide access control to particular locations as well as printing services. Thanks to the prototype, which was further implemented into Laboratory of Telematics – BAS, we were able to collect the necessary data and complete the research.

The prototype as well as the models created as part of the research are described in details with sequence diagrams and other schemes.

2. Communities created by scientific organizations and universities

Nowadays, the scientific organizations work together on various projects, scientific researches, training programs, etc. It is a typical practice for scientists and teachers from one scientific organization to visit another organization during their joint work or during an organized seminar. This way, the organizations form a powerful community with enormous potential.

Scientific institutions are expected to work on researches as well as to teach students, but while this researches and the learning process is carried out by the scientists and teachers who are members of an organization, the administration of the organizations has the responsibility to provide the necessary resources to enable the scientists and the teachers to carry out their work successfully. Nowadays, for every professional, regardless of the field in which he or she works, it is of utmost necessity to have Internet connectivity, as well as access to a printing resource. Scientists are no exception to this, and in addition they need special laboratories, conference rooms, etc. [23]

2.1. Cooperation between the participants within a community

There are various solutions that aim to support the scientific institutions in the process of providing local resources to the members of the community. Thus, for example, scientific organizations part of the international interfederation service "EduGAIN" have the opportunity to securely exchange information related to identity, authentication and authorization between participating federations [21]. In other words, scientific organizations can easily identify the members of other scientific institutions as well as the institutions themselves. On the other hand, thanks to "eduVPN", members of the scientific institutions can access the local network resources of their organization remotely, being on the territory of another institution, and in a secure way provided by a VPN tunnel [1]. The necessary Internet connectivity is provided by another solution called "eduroam", which allows each member of a scientific institution to use the Internet through a WiFi network provided by the organization in whose territory he or she is currently located, and the user's authentication itself is carried out by his or her own scientific organization [22]. We could summarize that these solutions aim to support the work of the members of the scientific institutions, but at the same time the provided resources that cannot be recognized as local resources to any remote organization in which the users are temporary located, for a seminar for example:

- **eduroam** – provides an Internet connectivity to users, but these users do not get access to the local network of the institution where they are located. In other words, we are only talking about Internet connectivity, and local resources that one would normally be able to access if connected to the scientific organization's local network, such as an Ethernet printer, are not available to users. In the event that a scientist wishes to print his paper to be presented at an active conference held in a remote institution, for example, he or she should ask another user, a member of the same institution, to print the paper on his or her behalf or to seek technical assistance from the administration of the organization.
- **eduVPN** – provides a secure access to the local institution for academics so they can access their local computer, file server, or similar, remotely

through a VPN tunnel, but this solution is again limited to providing remote access to local resources and we cannot talk about providing access to local resources for the remote scientific institution.

This study is focused specifically on the ability of scientists, members of one scientific organization, to access local resources of another scientific organization while they are located on its territory, but at the same time, the authentication of users is performed by their own scientific organization. In other words, we follow the idea of eduroam, in which the scientific organizations authenticate their users and thus provide them with a way to get Internet connectivity in a network maintained by another scientific organization. However, the idea is further extended so that user authentication is not limited to Internet connectivity, but is focused on local resources of the remote institution:

1) printing resources;

2) access to halls, class rooms, laboratories, etc.

If the models developed as part of the research are further developed and step on the possibilities offered by eduGAIN, we could talk about even greater possibilities to identify users and automatically provide them with the access to special resources or customized permissions. However, integration with eduGAIN was not addressed as part of the research and remains the subject of future work by the team.

2.2. Roles and responsibilities in one organization

Each scientific organization uses its own organizational structure. Depending on the size of the organization, it could be relatively flat, but it could be extremely complex. At the same time, if we have to analyze the roles and responsibilities of the members of an organization, we can basically summarize that we have 2 main roles:

- **Administration** – these are all the members of the organization who have the responsibility of supporting the learning process, as well as the work of the scientists who are members of the same organization. We would expect the administrators to support the printer resources and have them up and running – loading them with toner, maintaining them, setting up the computers of the other members of the organization so they can use those resources, and so on. Moreover, the administrators are responsible for all other resources that the organization has – halls, class rooms, laboratories, etc. Very often, the access to these resources is limited and controlled by an access control system or, more simply, with a key – the maintenance of the access is again a responsibility of the administrators.
- **Scientists and teachers** – their job is to teach students, work on scientific researches, organize and attend conferences and seminars. At the same time, often these responsibilities require extra support by the

administration. The teachers may print materials to provide later to students or simply may need a room whose access is restricted to authorized personnel only to conduct their trainings. Scientists also need these types of resources.

These 2 main roles must work in symbiosis for a scientific organization to be successful.

2.3. Existing system architecture of solutions used by scientific institutions

The organizations have a built-in network infrastructure, thanks to which they can provide their users (teachers, scientists) with printing resources, Internet connectivity and other services typical of a local network (Fig. 1).

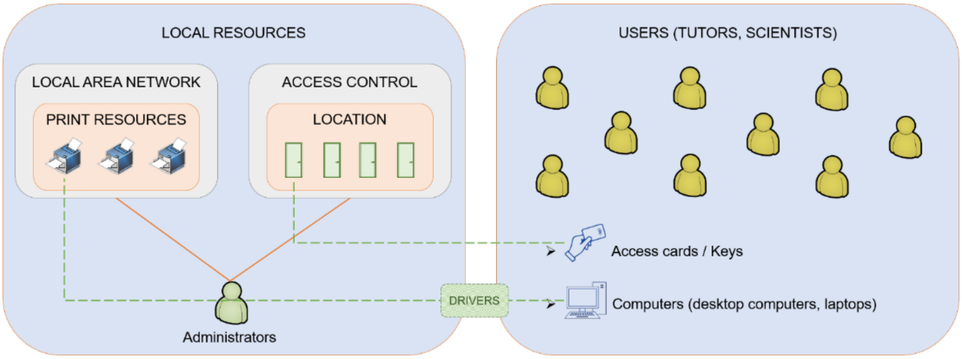


Fig. 1. Existing system architecture of solutions used by scientific institutions

They themselves maintain their local networks, as well as the resources provided through them, with each organization having one or more administrators. These administrators help the users to set up their computers and install the necessary drivers to use available network printers. Usually, all premises such as halls, class rooms, laboratories, computer rooms, etc. are protected by some sort of an access control system, which is again maintained by the organizations themselves. The administrators are responsible for the proper organization of access rights to these premises, usually this is done through access cards or simple keys. These systems are often integrated with the rest of the security system used by the organization.

Users must have access cards with them to use the available resource of locations. The configuration for using network printers is also relatively static and limited to a particular computer used by a member and in case of a failure of this computer or any kind of a replacement with another computer presupposes the need for a new setup by an administrator.

Thanks to eduroam users (teachers, scientists), members of a scientific organization have the opportunity to establish an Internet connectivity on the territory of another scientific organization, and in combination with eduVPN and to access local resources from their own scientific organization remotely (access to their local computer, for example) (Fig. 2).

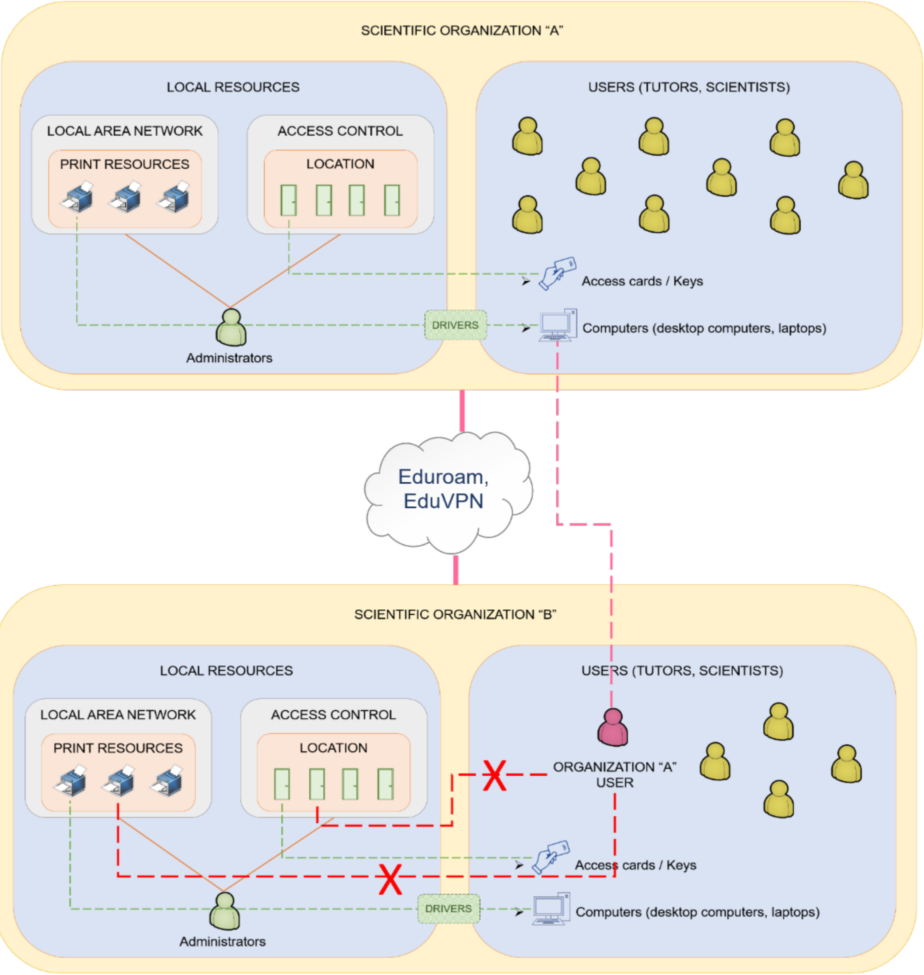


Fig. 2. Existing system architecture of solutions used by scientific institutions to share resources with members of another institutions

At the same time, these users would not be able to get easy access to the local resources of the organization in which they are currently located. This includes access to printers, halls, laboratories, classrooms, etc.

2.4. Taking advantage of smart technologies that are part of our everyday life

According to the stats of published by Statista Research Department in August 2022, around 80% of the population of the world has a smartphone or another smart device [4]. The mobile devices that we carry in our pockets are so powerful and offer such kind of features that it is a shame not to use them in our favour. They could be used as biometric readers (finger prints, face recognition, etc.), so we may have full user identification. Moreover, these smart devices can read QR codes and RFID tags.

The natural question is – How we could use this fact in our benefit? The following research offers a novel way to access local resources by simply reading QR codes and using secure authentication and authorization mechanism, together with the ability to acknowledge the user's identity, using biometric data though smart devices. In other words, we are talking about optimizing the processes necessary to provide access to local resources, but not at the cost of lowering security. At the same time, the organizations do not need to make huge investments in infrastructure, equipment or development of complex software systems.

2.5. Access local and remote resources using nothing but users' own devices

The following research offers a novel way to access local resources by simply reading QR codes and using secure authentication and authorization mechanism, together with the ability to acknowledge the user's identity, using biometric data though smart devices.

The model developed as part of the research is divided into 2 parts:

1) A model oriented to the local solution, which should be implemented by the organizations that are willing to join the idea of expanding the opportunities that the community offers;

2) A model oriented towards the global solution that should be implemented so that local implementations can interact with each other. However, for the overall solution presented in the research to work successfully, both models should be implemented.

When the devices used to access a system and/or authenticate users are not controlled by the organization that administers that system, the security design of the devices themselves and the system must be properly tailored. In other words, users' personal devices differ greatly in their characteristics from the same users' work computers. For example, a Group policy [15] could be applied to office computers to ensure their level of protection. In contrast, this is not applicable to the devices own by the users themselves. Moreover, these devices are used over a network not controlled by the organization, in which case we can accept the

Internet connectivity provided by eduroam for a public and threat it as unsecured network. The same applies to the devices that this research suggests, meant to replace conventional access cards to the rooms and laboratories with a limited level of access. Typically, access cards are issued, maintained, and even owned by the organizations that issued them, which is not the case with users' personal devices. We can summarize that when designing the web system and the mobile application, the following must be taken into account:

1) The models assume the use of the system and access to the organization's local resources through the users' personal smartphones;

2) These mobile devices will be used over networks which the organizations cannot control and may be treated as public and unsecured; 3.) Users should have a secure authentication method that cannot be easily compromised.

The Users' devices can be lost or given to other people without first restoring their factory settings. Through the "Administrative Web System", the devices could easily be disabled, which would cease the possibility of their further use. In other words, in case of device loss, the user simply needs to inform the administrators of his organization or log in himself using his own profile and disable the lost device, as well as one that he no longer owns for other reasons. Moreover, because the mobile application sends the unique device ID as part of each request to the scientific organization's API, in the event that the application is cloned to another device, that unique ID will also be different and the attacker will not be able to use the new device.

A user can have more than one device and this is not a problem at all according to the models developed as part of the research, just that each of the devices must be authorized to access the system by being tied to the given user. This gives users even more flexibility to use more than one way to perform the processes required to access the organization's local resources.

The proposed solution assumes that organizations will continue to manage their printer resources, as well as the access control systems for halls, classrooms, labs, etc., as they have done until now. Moreover, they should continue to manage their users as they have done so far. So, for example, if the scientific organization used "Active Directory" [3] to manage its users, it can continue to use the same user management tool, it just needs to integrate it with the rest of the system framed by the paper. The new elements in the solution that characterize the presented model are the presence of an API of a software system, an administration system, as well as a mobile application (called by us "MARLIN" – Mobile Application for Remote and Local resources and through secure user Identification over the Network) (Fig. 3):

- **MARLIN** – this is the mobile application designed as part of the research. The users should install this mobile application on their smart devices, which in turn must have fingerprint recognition capability and a working camera (this is a prerequisite for the user device to be used). Through this application, the users have the opportunity to access their files and add them to a printing queue, and then print them by scanning a QR code label by the system administrators on the printers. Again, the users may access any room, which meant to have restricted access, controlled by an access control system and used to require an access by an RFID card, by simply scan another QR code label next to the door of the room.

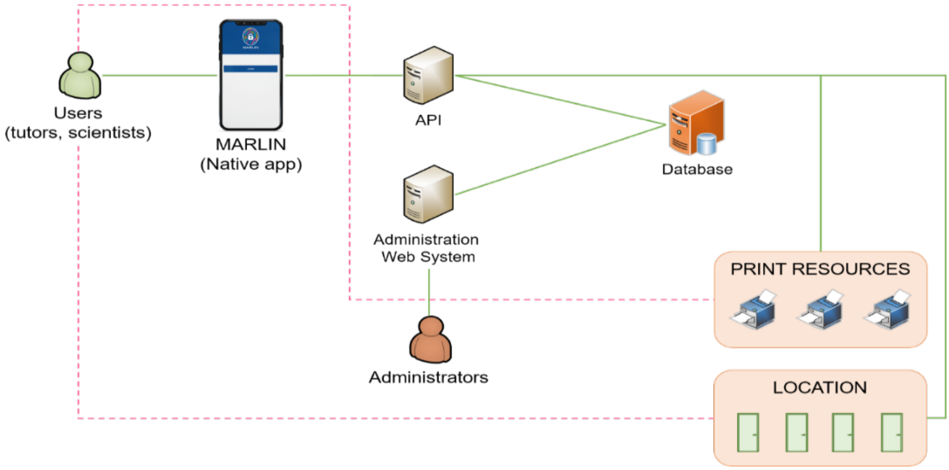


Fig. 3. Local solution using MARLIN

- **API** – the mobile application communicates with the rest of the system through an API, which should be located locally in the scientific organization. Moreover, the machine on which it is located must be able to communicate with the available printers (by means of installed drivers for them and have proper configuration applied), as well as the access control system (most such systems offer this kind of functionality via API)
- **Administration web system** – through this system, the administrators configure the printers, rooms and other local resources and make them "visible" to the API with which the system shares a database. Moreover, through this system, the administrators link the existing user with their trusted devices. In other words, this system offers the ability to perform CRUD (Create Read Update Delete) operations performed on the database containing the configurations used by the API.

The implementation of this solution provides additional advantages to the scientific organization:

1) Safe printing – in a classic implementation the user sends a file to the printer, so it could be printed, but the printer itself may be located far away from the user, so someone else may get the printed file consciously or unconsciously. In contrast, with the solution suggested the user just adds

the file to the printing queue, but the actual printing process is activated by the user once he or she scans the QR code labeled on the printer.

2) Managing the access control using users' personal mobile devices ensures to a greater extent that the means of unlocking rooms, laboratories and others will be available to the user, as practice shows that the users often forget their access cards. However, it is less likely to forget their mobile phones.

As it becomes clear from the presentation of the local solution, which should be implemented by the scientific organization, of fundamental importance for the solution is locating the user, identifying him and performing some subsequent action. For example, the user's files must be printed to a specific printer of his/her choice, and when exactly at the time he/she is in front of that printer, not before he/she goes there. Also, the user should only be able to open a room or lab when they are right in front of it. In other words, it is necessary to find a solution for the localization of the exact place of the user on the territory of the scientific organization. While there are sophisticated solutions for locating users of a Wi-Fi network using vector-triangulation determined by the strength of the signal received by their devices from three different Wireless Access Points, for example The **F**ramework for **I**nternal **N**avigation and **D**iscovery (FIND) [2], which will be considered as opportunity for future development of the research, the current model uses a significantly simpler approach. If some unique marker is placed on the printer or on the door of the hall, which will signal the device only when the user is in front of that marker, we can be sure of the current location of the user, because the marker is static to the printer or the room. In other words, even if the printer is moved to another location, the marker will move with it, and the user device must again indicate to the system that the user is right in front of the printer in question, albeit at a different location. Two potential options were investigated to be used for the so-called "marker" – 1) QR code; 2) NFC tag. Both types of tags can store data that describes their uniqueness, for example a unique ID. What's more, both types of tags can be easily read with a mobile device.

The NFC tags (Fig. 4) are cheap (one tag costs approximately $0.50 as of 2022) and are extremely easy to find on the market. What's more, many of the mobile devices on the market these days offer a reader for such tags.

Fig. 4. Example of an NFC tag

The principle of operation is relatively simple:

1) The system administrator records data on the tag with the so-called tag writer (again, many mobile devices that offer a tag reader also have a tag writer);

2) The tag is a sticker that is placed on the printer or the door;

3) The user brings their mobile device closer to the tag and the mobile device reads the tag indicating that the user is in front of it. While this type of tag is gaining more and more popularity, devices that offer NFC tag reading are still fewer in variety, and also more expensive, than those that simply offer a camera. For this reason, the use of QR codes was chosen for the purposes of the research (Fig. 5).

Fig. 5. Example of an QA code identifying a printer

QR codes are a specific matrix barcode, recognizable by special QR barcode readers or mobile phone cameras. The barcode consists of black modules arranged in a square pattern on a white background. The information in it can be text, URL or other. Generating such codes is relatively easy using a standard algorithm. What's more, the QR code can be printed on a plain piece of paper by a conventional printer and then easily labeled on a printer or door.

The ability to read QR codes is supported by both iOS and Android operating systems and can be easily accessed through the operating system's API in the same way that the ability to read biometrics can be accessed. This functionality is implanted into MARLIN native app.

Using several sequence diagrams, the working model of the local solution is described:

Fig. 6 presents the user device authorization process. The model developed as part of the research assumes that users use their personal devices and thus benefit from their high parameters as well as the advantages they bring us. Moreover, the use of users' devices reduces the investment required to implement the solution to a minimum. At the same time, the model does not assume that users can start using their devices instantly – these devices must first be authorized by users themselves and also the system administrators.

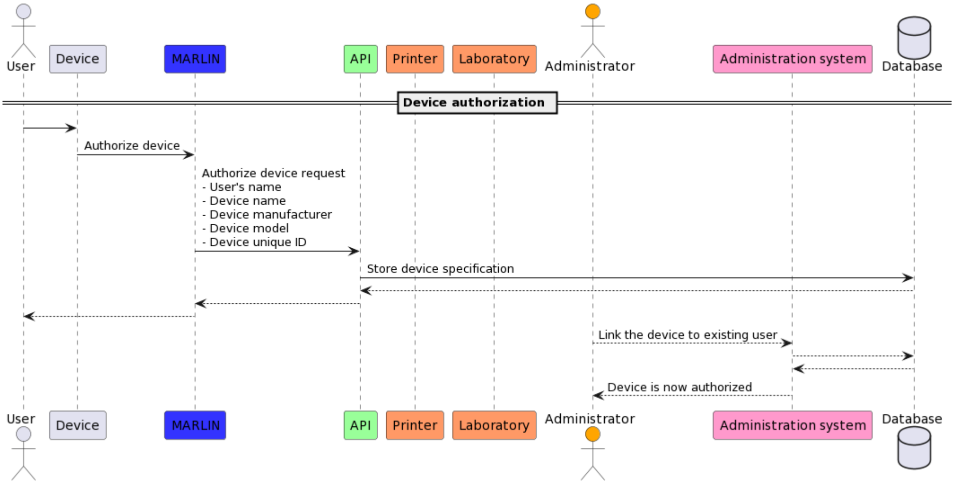


Fig. 6. Process of device authorization

The process begins by installing the MARLIN application on the user's device. This device must have a camera and a fingerprint reader. After installing MARLIN, the user must initiate a device authorization process. He/She enters his/her name to make it easier for system administrators to find the device authorization request. Also, the user should select the scientific organization of which he/she is a member.

Since MARLIN can be integrated with any scientific institution in the world, and at the same time, as specified earlier in this post, the scientific institution has the responsibility to perform user authentication itself, the user must select the correct institution of membership. With this action the user configures the system and from that moment it will work with his/her scientific organization. The list of scientific organizations is fetched from the "shared system" (Inter-organization) with which all the scientific organizations are integrated. This allows new scientific institutions to be easily added without having to update the entire mobile application. The device adds data representing its unique identification to the user name, so the request has: device name; device manufacturer; device model and a unique identifier of the device. The device authorization request is sent to an API integrated in the user's scientific organization, and then recorded in a special database. The authorization process ends in the administrative web system, where the administrator links the device from the request to a specific user from the organization's repository. At this point, the device is formally authorized and the user can start using it to print files and/or access locations.

Once the device is authorized for operation, the user logs into the MARLIN system, as shown in Fig. 7. Before the device is authorized, the login is possible at all.

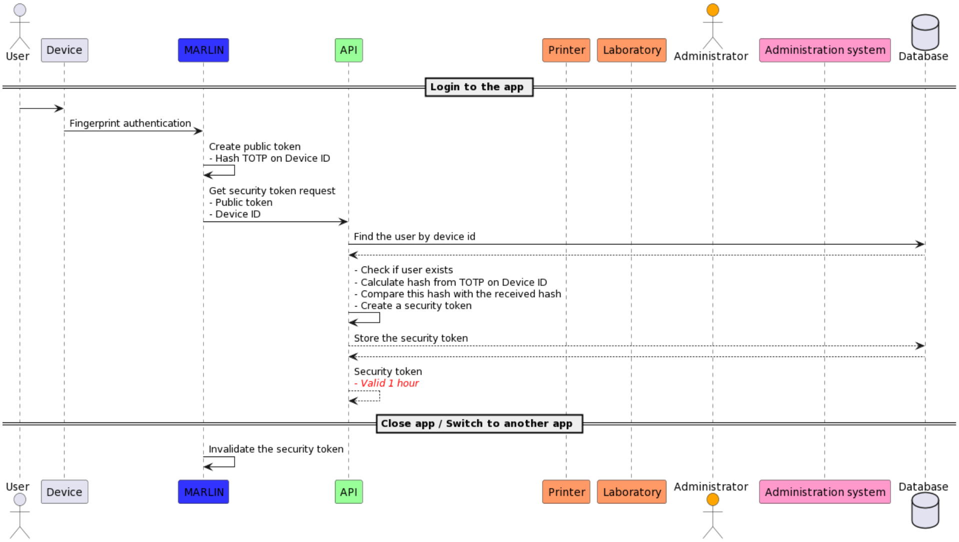

User
Device
MARLIN
API
Printer
Laboratory
Administrator
Administration system
Database
Login to the app
Fingerprint authentication
Create public token
- Hash TOTP on Device ID
Get security token request
- Public token
- Device ID
Find the user by device id
- Check if user exists
- Calculate hash from TOTP on Device ID
- Compare this hash with the received hash
- Create a security token
Store the security token
Security token
- Valid 1 hour
Close app / Switch to another app
Invalidate the security token

Fig. 7. Process of user login to MARLIN and closing MARLIN

The user identifies himself/herself to the MARLIN system using biometric data – a fingerprint. Upon successful authentication, MARLIN generates a hashed (using MD5) version of the time-based one-time password (TOPT), which is recorded as a GUID. This is so-called "public token". It is sent to the API maintained by the scientific organization in combination with the unique device ID, which is sent as part of the header of each request to the system. The system checks if such a device ID is actually recognized by it, and in turn also generates a TOPT, which it compares with the one sent by the user. In case of a match, the system generates a security token valid for 1 hour to be used by MARLIN for future interactions of the user with the system. This token is invalidated every time the user log out, switch to another mobile app, or after 1 hour. In this case, the user must authorize himself again with a fingerprint.

Multifactor authentication is a method of providing the user with an access to a resource that requires two or more phases of the authentication process. It is based on the presence of 3 constituent components of the model:

1) knowledge (something that only the user knows);

2) possession (something that only the user owns);

3) inherence (something only the user is). In order to achieve secure multi-factor authentication, at least two of the three components must be used in the authentication solution [10].

For example, having a bank card issued in the name of a specific user by a given bank for the purpose of carrying out financial operations is something that only the user can possess, and the card's PIN code for the purpose of authorizing payments at a POS terminal is the thing which only the user can know. Payments are usually made with these two components in place – the user presents their card to the merchant, then authorizes the payment with their PIN. Even if the card is lost, another user would not be able to use it without the PIN code.

In the presented solution, the MARLIN system implements a multi-factor authentication model. The user installs the mobile application on his/her personal devices, which is subsequently authorized and then attached to the user's account used by the user to access the system. In other words, the mobile device and the MARLIN application installed on that device cannot be owned by another user. At the same time, the use of a fingerprint ensures secure multi-factor authentication. A fingerprint is the so-called "inherence" component of multi-factor authentication, it is something that uniquely identifies a user.

By itself, this solution provides an opportunity for secure authentication of the user, but the presented model further develops the possibility of multi-factor authentication and provides additional protection of the process. An algorithm for one-time password (OTP) and in particular time-based one-time password (TOPT) is implemented in the model.

TOPT is a relatively simple method to provide multifactor authentication. Most often the user first uses a username and password to authenticate himself/herself and gain access to the system, and then enters an OTP – usually some sort of six-digit number. This six-digit number is generated by some authentication application, it can be a physical token device or a software application installed on the user's mobile device, and according to the algorithm specification, it is generated every 30 seconds (it is possible to be used with a different time interval, but in this case we will not use the original version of the algorithm). The user enters this six-digit number (this is usually the second step of authentication) and then the number is validated with the authentication server, which in turn also validates a six-digit number value using the same algorithm. In the 30 second interval this number must be identical if generated based on the algorithm on both the server and the authentication device. TOTP is defined in RFC 6238 [16]. The main advantage of this authentication method is that the number used for authentication is random, determined only by time and some key, and cracking it is complicated. Moreover, given the fact that the number changes every 30 seconds, even if it is broken, there will be another number replacing the last one after this period. The use of TOPT in this research is embedded in the MARLIN application (Fig. 8).

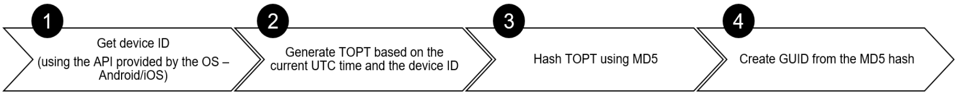

1
Get device ID
(using the API provided by the OS – Android/iOS)
2
Generate TOPT based on the current UTC time and the device ID
3
Hash TOPT using MD5
4
Create GUID from the MD5 hash

Fig. 8. Process of creating hashed TOPT by using a device ID

Thanks to TOPT the application adds a header to each HTTP request sent to the server (to send a print request, for example), which represents a hashed value of TOPT, i.e. it is a hashed value of the six-digit number obtained by the TOPT algorithm. MD5 hashing algorithm was used for hashing, through which a globally unique identifier (GUID) is created as shown in the figure below. Since some secret key must be used to create a TOPT, the ID of the user device is chosen as such key. This ID cannot be forged by the user, as it is described by the operating system and is retrieved directly from there on every request to the server. The device ID is also unique value. In other words, it is not possible to have more than one device with the same ID.

Since the header is sent as part of each MARLIN request to the server, the solution can ensure that the device is not swapped between individual requests. In such a case, the request will be invalidated by the server because the GUID value that came in the header was generated for a different device.

The use of TOPT in combination with an authorized device that only the user can own, as well as a fingerprint guarantee an extremely and secure 3-factor authentication of the user.

However, the solution developed in this research suggests an even higher level of protection. The hashed password is used during the user's initial login to MARLIN to issue a security token from the server. In other words, the hashed value simply enables the user to establish a session with the server, but then the session itself is further validated. The principle of operation is described by the diagram below (Fig. 9).

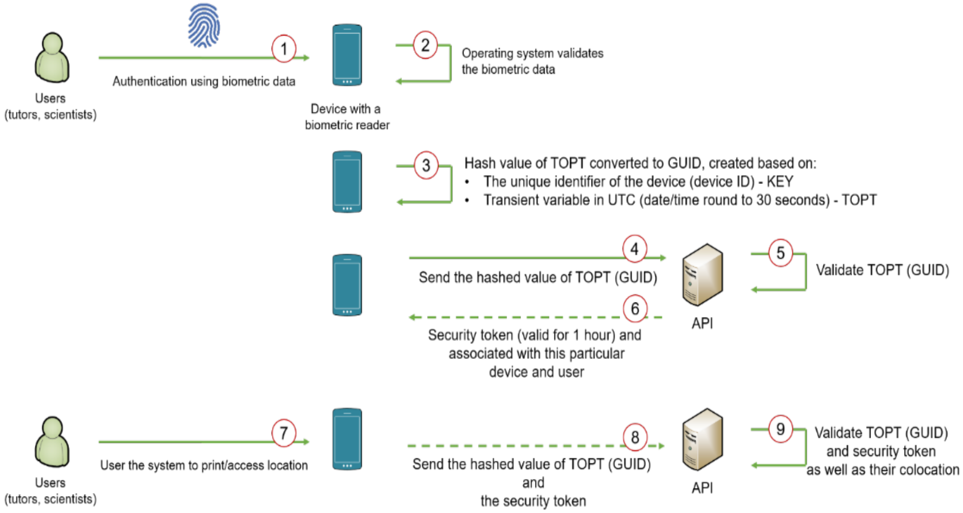

Users
(tutors, scientists)
1
Authentication using biometric data
Device with a
biometric reader
2
Operating system validates
the biometric data
3
Hash value of TOPT converted to GUID, created based on:
• The unique identifier of the device (device ID) - KEY
• Transient variable in UTC (date/time round to 30 seconds) - TOPT
4
Send the hashed value of TOPT (GUID)
API
5
Validate TOPT (GUID)
6
Security token (valid for 1 hour) and
associated with this particular
device and user
Users
(tutors, scientists)
7
User the system to print/access location
8
Send the hashed value of TOPT (GUID)
and
the security token
API
9
Validate TOPT (GUID)
and security token
as well as their colocation

Fig. 9. Process of using TOPT in combination with an authorized device and subsequent validation of requests to the system

The security token ensures the control over the duration of the user's session – it is a maximum of 1 hour. After that time the user is going to need to log in again using a fingerprint. In other words, even if the user has forgotten their device unlocked and MARLIN opened, and another user accesses it, they will only be able to use the system during the time slot in which the session is valid. This will minimize damage caused by user negligence. At the same time, any MARLIN minimization will also invalidate the security token and the user will be forced to log in again using a fingerprint. And since the token is associated with a specific user and device, another user/malicious person would not be able to hijack this session as it would not be valid for him and his device (GUID by TOPT). In this way, a multi-factor authentication with 4 phases is provided and a maximum protection for the user and the organization is guaranteed. In order to generate a TOPT the submission of a unique key is required. The combination of the key and the current time, rounded up to 30 seconds, is the basis of the TOPT algorithm. In this research the unique ID of the device is chosen to be the key. MARLIN accesses (read only) this device ID through an API provided by the operating system. This ID cannot be changed by the user and therefore the process cannot be compromised.

On Android 8.0 (API level 26) and higher versions of the platform, a 64-bit number (expressed as a hexadecimal string), unique to each combination of app-signing key, user, and device. Values of ANDROID_ID are scoped by signing key and user. The value may change if a factory reset is performed on the device or if an APK signing key changes. For more information about how the platform handles ANDROID_ID in Android 8.0 (API level 26) and higher, see Android 8.0 Behavior Changes. On devices that have multiple users, each user appears as a completely separate device, so the ANDROID_ID value is unique to each user [6].

In iOS the device ID comes from so called "identifier for vendor". This is an alphanumeric string that uniquely identifies a device to the app's vendor.

The value of this property is the same for apps that come from the same vendor running on the same device. A different value is returned for apps on the same device that come from different vendors, and for apps on different devices regardless of vendor. Normally, the vendor is determined by data provided by the App Store. If the app wasn't installed from the app store (such as enterprise apps and apps still in development), then a vendor identifier is calculated based on the app's bundle ID. The bundle ID is assumed to be in reverse-DNS format.

In iOS 6, the first two components of the bundle ID are used to generate the vendor ID. If the bundle ID only has a single component, then the entire bundle ID is used. In IOS 7, all components of the bundle except for the last component are used to generate the vendor ID. If the bundle ID only has a single component, then the entire bundle ID is used. If the value is nil, wait and get the value again later. This happens, for example, after the device has been restarted but before the user has unlocked the device.

The value in this property remains the same while the app (or another app from the same vendor) is installed on the iOS device. The value changes when the user deletes all of that vendor's apps from the device and subsequently reinstalls one or more of them. The value can also change when installing test builds using Xcode or when installing an app on a device using ad-hoc distribution. Therefore, if your app stores the value of this property anywhere, you should gracefully handle situations where the identifier changes [7].

A biometric recognition system makes it possible to determine the user's identity with high accuracy and provides high security [20]. Biometrics are body measurements and calculations related to human characteristics. This type of data most commonly used to identify users in a software system are (but not limited to): fingerprints, facial recognition, voice recognition, and iris recognition. Biometric authentication ties some physical characteristic of the user, unique in itself, to its software representation. Specially defined standards are used to save this representation in a database. The aim is to achieve the following working model:

1) Initial capture of biometric data, for example fingerprints
2) Storing biometric data in a database management system following a specific standard, for example "fingerprint templates" (a standard-defined structuring of the data that allows the fingerprint to be stored in digital form) [20]
3) Reading biometric data and turning it into a temporary digital array (without saving the digital array in the database)
4) Comparison of the digital representation just read with one available in the database
5) User identification on match

In other words, devices equipped with some kind of biometric reader should read the user's biometric data, then process and store it digitally in some storage (usually local database), and then associate it with the user. Once the data is saved to the device, the device can compare it to data just captured by the user to determine if there is a match and authorize the user.

Digital representations of biometric data can be stored in two ways, depending on the needs of the system:

1) Locally on the device
2) On a centralized location where it is processed not by the device but by a server

Storing the digital representation of the biometric data locally has many advantages over storage in a centralized database management system, such as:

1) The logic of comparing the digital representation of the newly captured biometric data with the one stored in the database is performed locally on the user's device and thus does not burden the entire system
2) A possible compromise of the database storing a digital representation of biometric data would compromise only the records stored on the device, but not the entire database.

The most used operating systems, such as Windows, Linux, as well as those for mobile devices – Android, iOS, offer the ability to capture biometric data and support a local database on the device to store the data. These operating systems could compare biometric data with the available database and as such to confirms the user's identity upon matching. In other words, through the functionalities of the operating system and a reader located on the device itself, we could easily provide user identification. The nature of these operating systems and the devices that support them is that they deliberately do not allow the storage of biometric data/templates in remote storage, unlike other authentication data such as PIN, password, TOPT, and others, which are validated remotely on some server. This is done in order to ensure their maximum protection, because any data leaving and sending it over the network is a prerequisite for its compromise.

In the Android OS, fingerprint biometrics are required to be stored in the Trusted Execution Environment (TEE), where the information is encrypted and kept in a separate part of the smartphone, completely inaccessible to the regular OS. It can't even be exported. Android can ask the TEE to verify a user's identity using biometrics, but it can't extract the biometric information. In other words, when the user stores their biometric information, such as a fingerprint, they're not sharing that information outside of their own smartphone or tablet; they're just establishing a way to identify themselves to their device [5].

TEE (Fig. 10) is a secure area of the smartphone's main processor. It guarantees confidentiality and integrity of the code and data loaded inside. This separation enables security and protection from hacks, malware and root access.

- All fingerprint data manipulation is performed within TEE
- All fingerprint data must be secured within sensor hardware or trusted memory so that images of your fingerprint are inaccessible
- Fingerprint data can be stored on the file system only in encrypted form, regardless of whether the file system itself is encrypted or not
- Removal of the user must result in removal of the user's existing fingerprint data
- Root access must not compromise fingerprint data

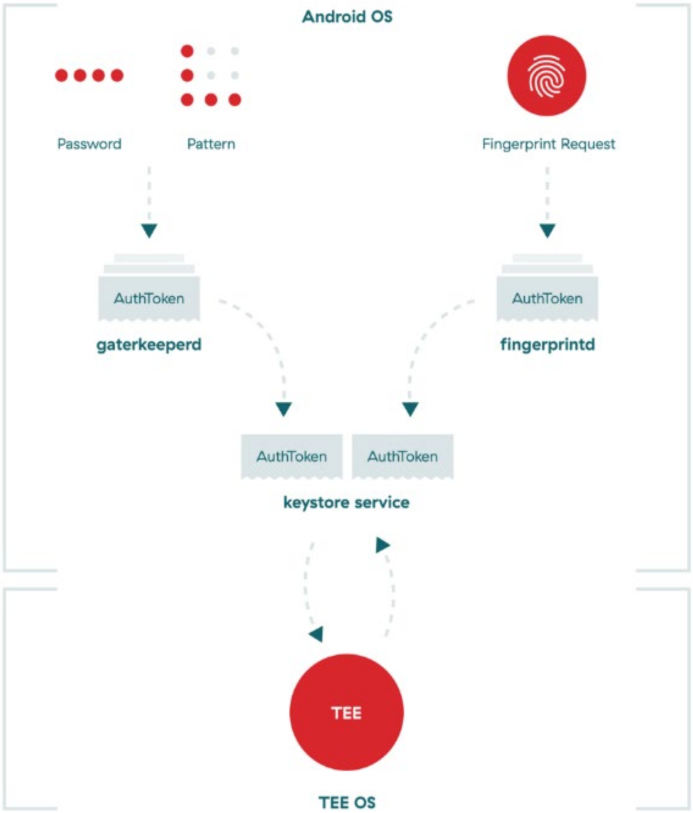


Fig. 10. Android – Trusted Execution Environment (TEE) [19]

iOS offers multiple types of biometric authentication, such as fingerprint (Touch ID), face recognition (Face ID). This biometric data, including mathematical representations of your it, is encrypted and only available to the Secure Enclave. This data never leaves the device. It is not sent to Apple, nor is it included in device backups.

The Secure Enclave is a dedicated secure subsystem integrated into Apple systems on chip (SoCs). The Secure Enclave is isolated from the main processor to provide an extra layer of security and is designed to keep sensitive user data secure even when the Application Processor kernel becomes compromised. It follows the same design principles as the SoC — a Boot ROM to establish a hardware root of trust, an AES engine for efficient and secure cryptographic operations, and protected memory. Although the Secure Enclave doesn't include storage, it has a mechanism to store information securely on attached storage separate from the NAND flash storage that's used by the Application Processor and operating system [18].

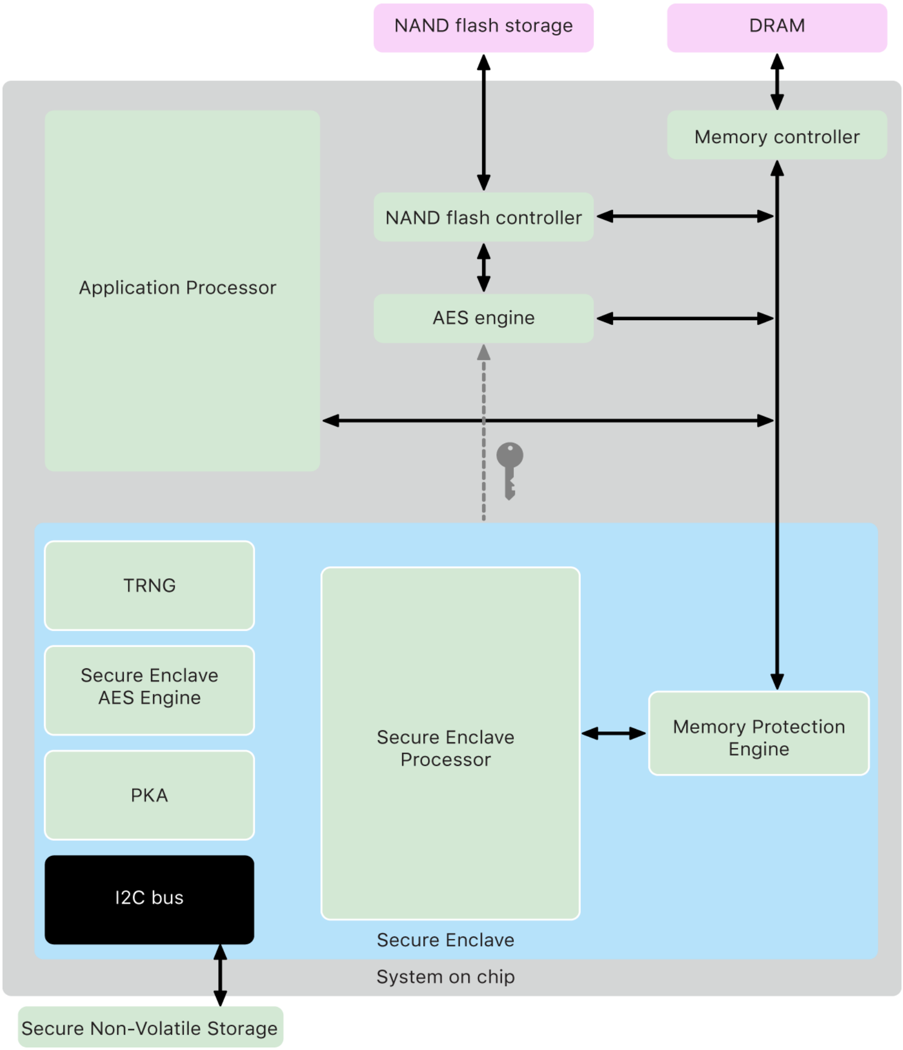


Fig. 11 iOS – Secure Enclave [18]

The presence of powerful readers for biometric data on users' mobile devices, as well as the maximum degree of protection of these data provided by the devices themselves, provides an ability for their easy and "free" use in some system requiring authentication of users. This is embedded in the model of current research. MARLIN implements an API provided by the Android and iOS operating systems that aims to provide user authentication with biometric data.

For the purposes of the scientific study, fingerprint authentication was chosen, since almost all devices on the market today have such a reader, even the cheap ones. Other types of biometric authentication, such as facial and voice recognition, require more complex processes to be performed by devices and are typically supported primarily by higher-end devices. In other words, in order to provide biometric user authentication for the most possible devices and users respectively, fingerprint authentication is chosen.

The Android and iOS operating systems take care of reading and storing fingerprints, as well as their subsequent association with the user of the device and his eventual authorization. Through the API provided by the operating systems, MARLIN receives a response from the operating system whether the fingerprint is recognized or not.

According to the General Data Protection Regulation (GDPR) [17], binding to more than one user authentication method ensures that consumer identity is fully identified. In other words, the combination of a username and a phone number is a sufficient way of verifying a user's identity [4]. Following this idea, various services are now available over the Internet to enable third party verification of the user's identity instead of using well-known electronic signatures. For example, the DocUSign system provides the ability of having a fully legitimate and legally enforceable contracts signed remotely by the users.

The processes of printing files by the user and accessing some locations are illustrated in Fig. 12.

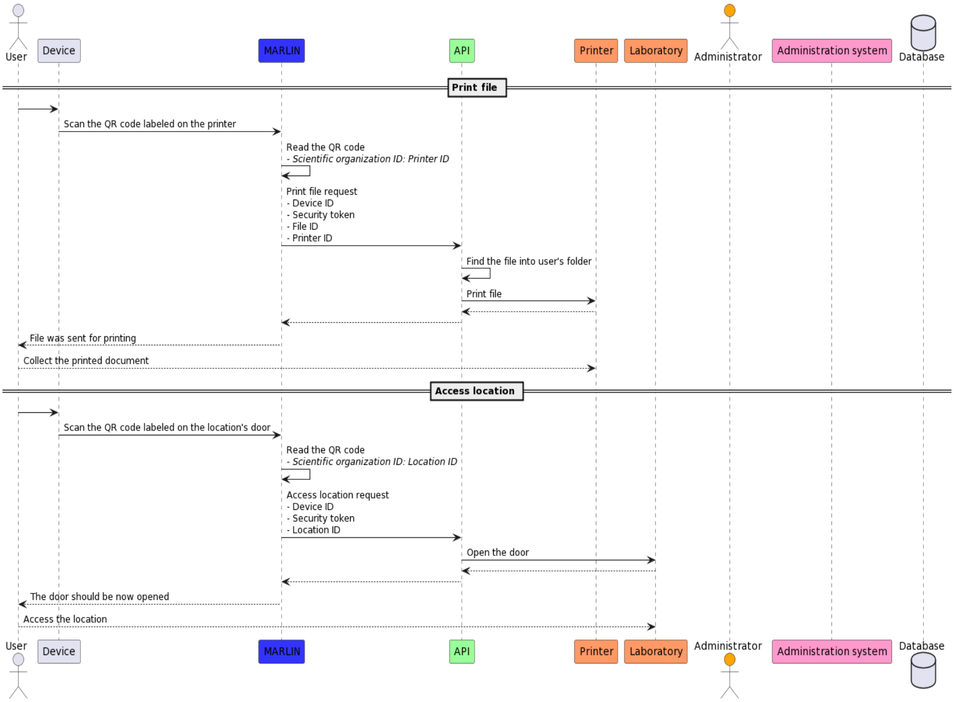


Fig. 12. Process of user login to printing a file or accessing a location

Printing the files starts with the user selecting a file to print. This file should be uploaded to the system, either through the web-based solution or the mobile application (only PDF files are supported in the developed prototype). As the next stage of the development of research, the design and development of a virtual printer is set, which allows creating a "queue" for printing from a wide variety of files and does not require uploading a file to the system, which is then printed.

After the file is uploaded to the system, the user should go to the printer on which he/she wishes to print it. The system administrator must have pre-marked all printers with a unique QR code that can be generated from the system administration where all printers must be pre-entered. What's more, all printers must be installed and ready to work with the organization's API. Once the user is in front of the printer, he/she click on the file they wish to print and MARLIN will automatically turn on the mobile device's camera for the user to scan its QR code.

With these actions, the user has clearly stated his wish to the system, for which file to be printed, as well as on which printer this should happen. MARLIN reads the QR code, behind which information about the institution is locked (a unique ID known both to the local organization and to the shared communication system between different scientific organizations), as well as the unique ID of the printer. The data from the QR code is sent to the API of the user's organization, which in turn checks whether the ID of the organization administering the printer matches the ID of the current organization – in case it does not, the API appeals to the shared system, which should provide a URL for the API of the organization that administers the printer. The local API sends the file and print request to the API of remote organization. The process ends by sending a print command and the print file to the user's desired printer.

The process of accessing locations is similar to that of printing. However, this time the user simply needs to scan the QR code of the location he/she wish to access. The system administrator must have previously marked all locations with a unique QR code that can be generated from the administrative part of the system, where all locations must be entered in advance. Moreover, the organization's API should be integrated with the access control system's API. Once the user is in front of the location, the user scans its QR code through the MARLIN mobile application. MARLIN reads the QR code, behind which information about the institution is locked (a unique ID known both to the local organization and to the shared communication system between different scientific organizations), as well as the unique ID of the location. The data from the QR code is sent to the API of the user's organization, which in turn checks whether the ID of the organization administering the location matches the ID of the current organization – in case it does not, the API appeals to the shared system, which should provide a URL for the API of the organization that administers the location. The API sends the location access request from this user to API of the remote organization. The process ends with sending an instruction to unlock the location to the API of the access control system.

2.6. Inter-organization solution

The inter-organization solution assumes that each autonomous scientific organization has already implemented the local solution model. The individual organizations remain responsible for the administration of their users, printers, access control to halls, laboratories, etc. At the same time, the concept of “Shared service” is introduced into the model (Fig. 13).

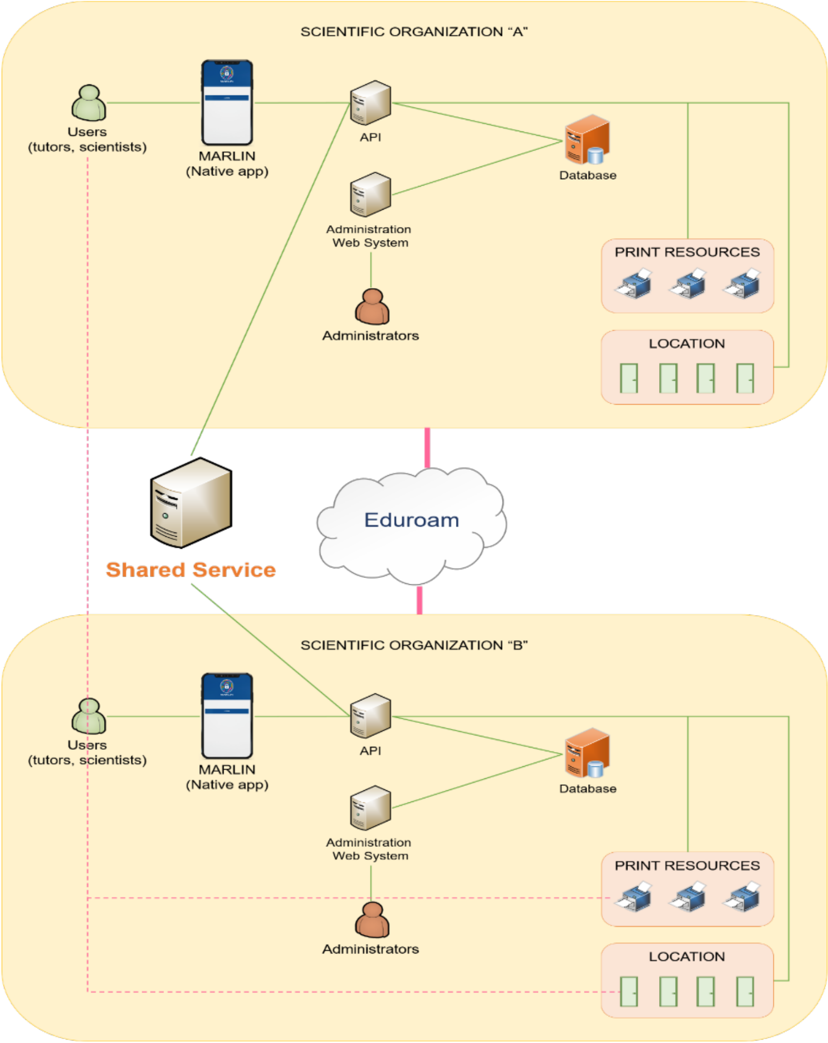


Fig. 13. Inter-organization solution using MARLIN

This system contains the data for all organizations that have implemented the solution described by this research. This data is not confidential and should not conflict with the internal information security rules. The system contains a list of organizations, their unique IDs, and their base URL to access their local API.

The mobile application is the same for all scientific institutions, but through the configuration set in the Shared service it can be used by users of all possible institutions. In other words, there is no need to create a new version of the mobile application when adding a new institution, as the configuration is dynamic and users of the new institution will be able to use the application as soon as the configuration is updated in the Shared service.

Users will be able to access the local resources of the institutions in whose territory they are located. They will simply scan the QR codes when they need to print or access a room, lab, etc. Since the QR codes contain data about the unique ID of the organization for which they were created, MARLIN will check whether this ID matches the user's organization ID, and if that is not the case, it will check against the Shared service, what is the correct base URL (of the API) to send the request. There is not much of a difference between opening a location with a restricted access control from a local or a remote organization. However, the main difference between printing a file to a printer located in the local intuition of the user and printing a file in a remote institution is the print queue with the file is going to be copied from the server of one organization to the server of the other organization. The rest of the printing process remain the same.

3. Prototype

In order to verify the efficiency of the research, the models designed, and the system created, we were able to create a prototype – an actual proof of concept. This prototype was further implemented into Laboratory of Telematics – BAS. Thanks to the actual implementation we were able to collect all the necessary data and to document the results of the research.

Hardware representation and system architecture of the prototype

Fig. 14 presents a diagram of the prototype. For the prototype we have created an actual native app for iOS and Android – MARLIN. The native app is then connected to an API, again, created as part of the prototype implementation. The API is deployed to a Windows Server 2019 virtual box, hosted on a Raspberry Pi 4 8 GB. The virtual box is configured to establish a connection with a network printer in order to prove the printing feature of the system. On the same Raspberry Pi board is installed a 4 relay board that is controlled by general-purpose input/output (GPIO) extension header of the Raspberry Pi board. The idea of the relay board is to simulate an actual access control system, which operates on 12 V DC. The power supply is connected to an electric magnet, typically used to lock doors. The relay operates in normally closed mode, meaning that until the relay is switched on, the electric magnet is going to be charged with 12 V DC, i.e. the door is going to be closed.

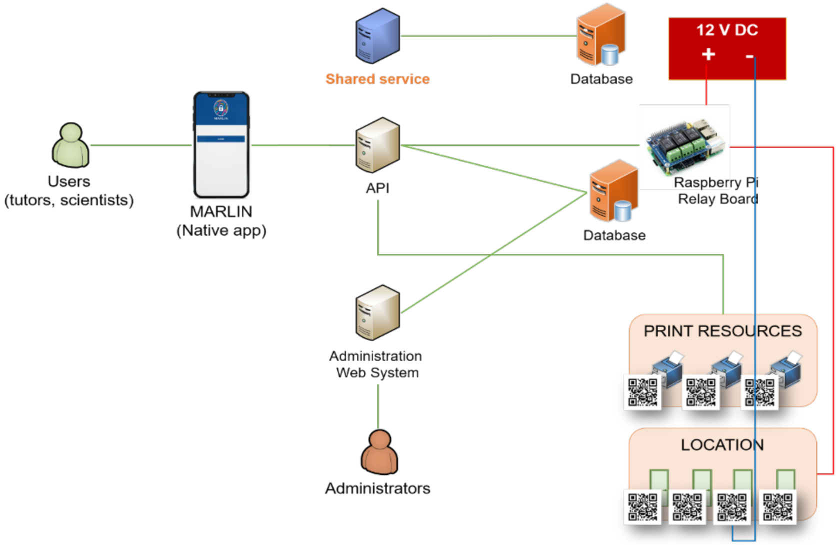


Fig. 14. Schema of the prototype

To control the relay remotely and thus simulate the API, which is usually proved by the access control systems, there was delivered a simple application, which was then hosted on the Raspberry Pi. The web application is actually exposing the relay control behind an endpoint that is being called over HTTP by the API of the MARLIN system.

The API is using a PostgreSQL database, designed as part of the prototype, and hosted into a Docker container. The same database is used by the Administration Web System created. The system is providing a set of CRUD (Create Read Update Delete) operations of the entities describing the users, the devices, the printers, the locations, and the files for printing. The system is hosted into another Docker container. Nevertheless, a prototype of a Shared service is created. This service should be used by all the individual API's of the scientific organizations, part of the community. The Shared service is using its own database and both the service and the database for it are hosted into Docker containers, running on the same Raspberry Pi, where everything else is hosted.

The actual prototype (Fig. 15) is validated and verified using the following test script:

1. A network printer (6) is connected to the same network, which is used by the Raspberry Pi board (2) that controls the relays and hosts all the web applications

2. The system is configured using the Administration Web System hosted on (2) and the configuration is stored on a database, again hosted on (2)

3. The Shared service hosted on (2) is configured to support the newly created “scientific organization”

4. The 12 V DC power supply (1) is connected to the relay board in normally closed mode and to the electric magnet (3)

5. The user installs a MARLIN app (on an Android device) (5) and then sends an authorization request for his/her device

6. The device is authorized through the Administration Web System

7. The user logs into the system and uploads a file for printing though the MARLIN app

8. The user clicks the files name and once the camera of the device is activated, he/she scans the QR code of the printer

8.1. The file should be printed

9. The user then chooses to access a location into MARLIN app and once the camera of the device is activated, he/she scans the QR code meant to be labeled on the door of the location

9.1. The electric magnet should be switched off, so the door should be opened

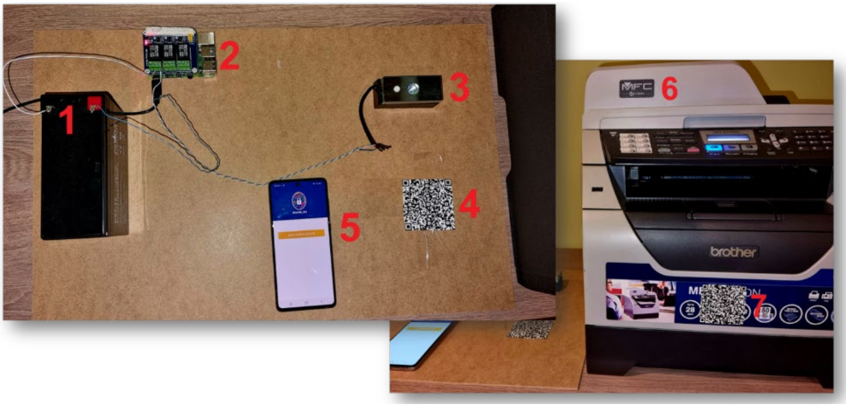


*Fig. 15. MARLIN prototype: **1** – Main power supply (12 V battery); **2** – Raspberry Pi board with a relay board; **3** – Electric magnet (door lock); **4** – QR Code for the location (electric magnet unlock); **5** – Android device with MARLIN installed on it; **6** – Network printer; 7 – QR Code for the printer*

3.1. Software representation of the prototype

The following screenshots taken from MARLIN native app (Android in this case) and the Administration Web System represents the software side of the prototype. The Administrative Web System created as part of the prototype uses a simple authentication method – user and password, as shown on Fig. 16.

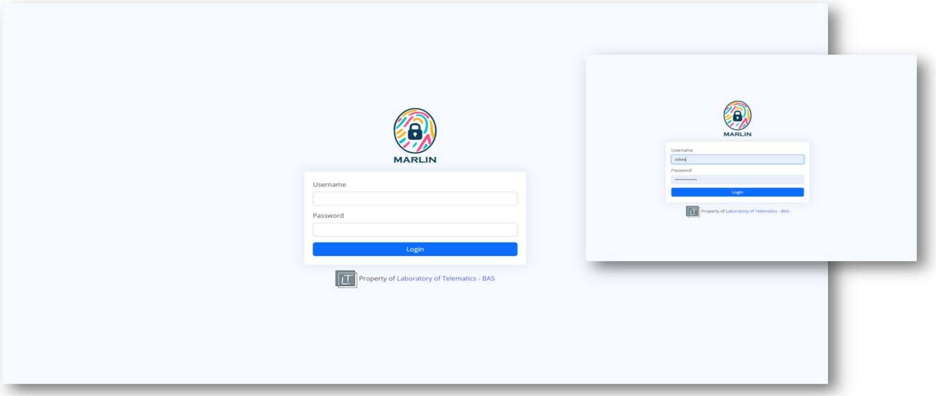

MARLIN
Username
Password
Login
Property of Laboratory of Telematics - BAS
MARLIN
Username
Password
Login
Property of Laboratory of Telematics - BAS

Fig. 16. Administrative Web System – Login

This is how the users are managed. The prototype is not integrated with Active Directory, OAuth2, or any other user authentication method.

The system offers 2 level hierarchy for the user administration. Two user roles are supported – administrators and non-administrators. Each group has access to different set of features offered by MARLIN as shown on Fig. 17.

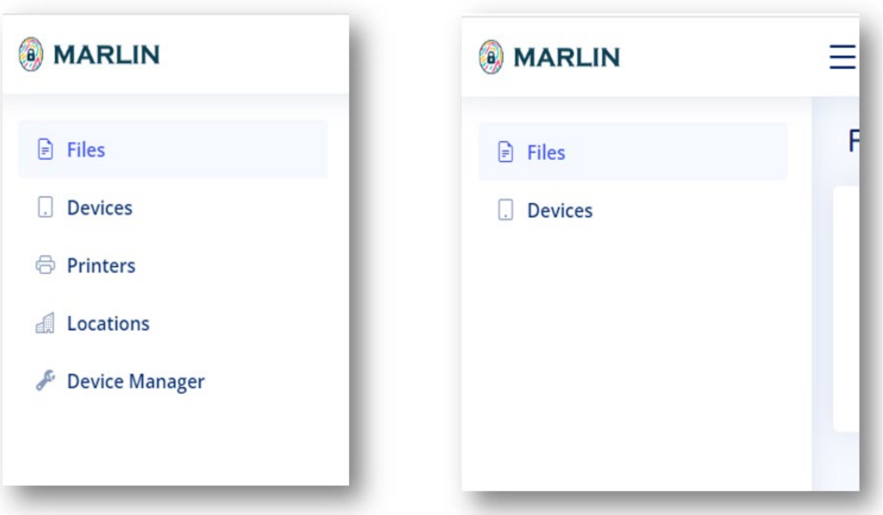
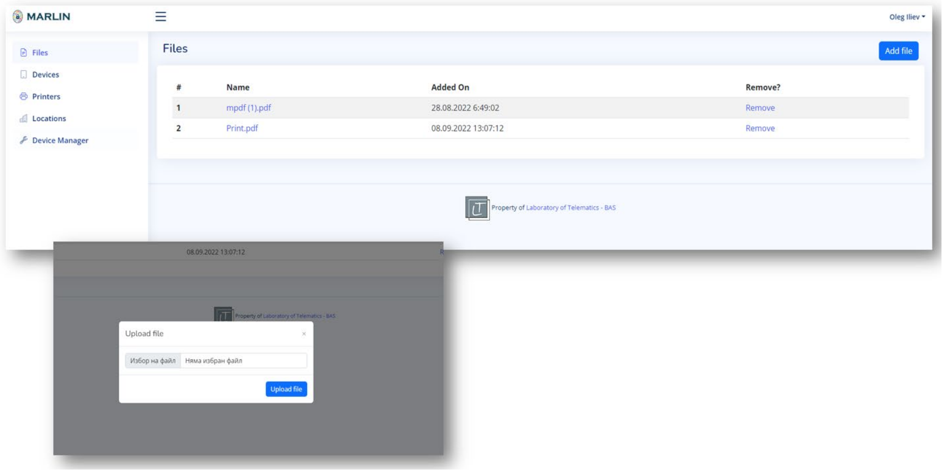


Fig. 17. Administrative Web System – Menu – Administrator vs. Regular user

The administrators of the system are supposed to manage the devices of all the users, the printers, and the locations, whiles the regular users (non-administrators) should see only their own devices (associated with their accounts) and their own printing queue (files to be printed).

In order to print a file, the user needs to upload it first to the system. This could be done both by the MARLIN native app and the web system as shown on Fig. 18. Moreover, the files uploaded to the system for printing could be further managed.



Files

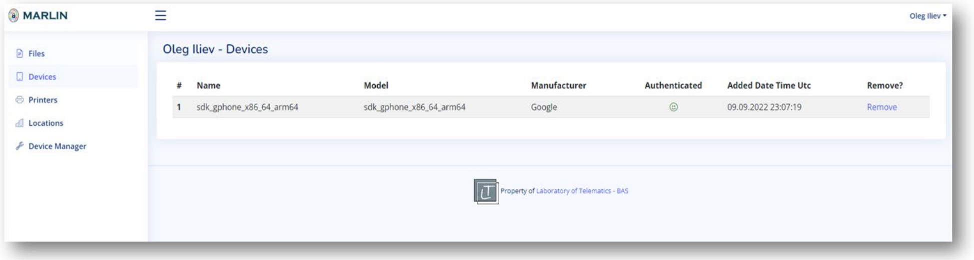
#	Name	Added On	Remove?
1	mpdf (1).pdf	28.08.2022 6:49:02	Remove
2	Print.pdf	08.09.2022 13:07:12	Remove

Upload file

Избор на файл: Няма избран файл

Fig. 18. Administrative Web System – Files to print (user's printing queue)

The user is able to manage his/her own devices – to remove them from the authorized list, as shown on Fig, 19. This is a convenient option for the user who may remove a device that is stolen, but not wait the administrator of the system to do that.

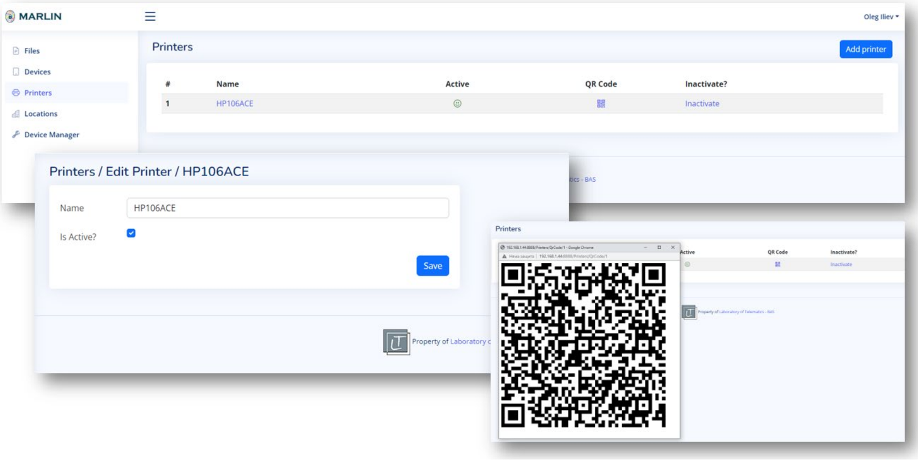


Oleg Iliev - Devices

#	Name	Model	Manufacturer	Authenticated	Added Date Time Utc	Remove?
1	sdk_gphone_x86_64_arm64	sdk_gphone_x86_64_arm64	Google	☺	09.09.2022 23:07:19	Remove

Fig. 19. Administrative Web System – User's devices

Only the administrators of the system could manage the printers as shown on Fig. 20.



MARLIN

Oleg Iliev

- Files
- Devices
- Printers
- Locations
- Device Manager

Printers

Add printer

#	Name	Active	QR Code	Inactivate?
1	HP106ACE	☺	▦	Inactivate

Printers / Edit Printer / HP106ACE

Name: HP106ACE

Is Active? ☑

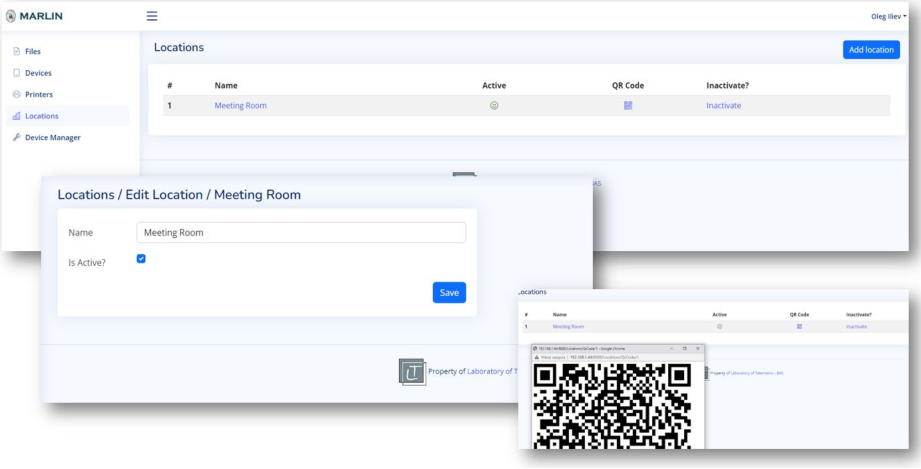
Save

Property of Laboratory of Telematics - BAS

Fig. 20. Administrative Web System – Printers management

They just need to enter the name of the printer as it is recognized by the operating system of the API. Moreover, through the system the administrator could set a printer to be inactive or to generate a QR code that should be then printed and labeled on the device.

The administration of the location is quite similar to the one for the printers as shown on Fig. 21.



MARLIN

Oleg Iliev

- Files
- Devices
- Printers
- Locations
- Device Manager

Locations

Add location

#	Name	Active	QR Code	Inactivate?
1	Meeting Room	☺	▦	Inactivate

Locations / Edit Location / Meeting Room

Name: Meeting Room

Is Active? ☑

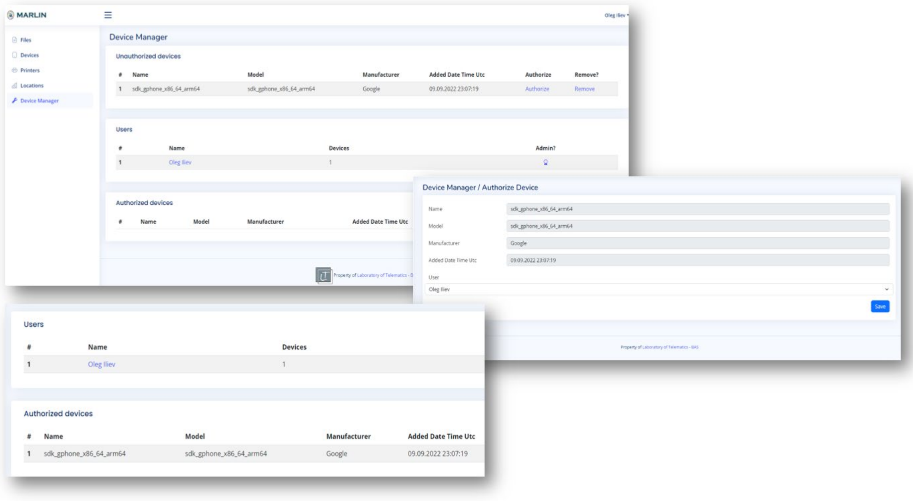
Save

Property of Laboratory of Telematics - BAS

Fig. 21. Administrative Web System – Locations management

Again, the administrator needs to setup a location giving it “name”, which is then recognized by the access control system. Just like the printers the location could be flagged as inactive and the system could generate a QR code for them.

The administration of the users' devices – their authorization, mapping to particular user, deactivation, etc. is done by the Administrative Web System as shown on Fig. 22.



Device Manager

Unauthorized devices

#	Name	Model	Manufacturer	Added Date Time Utc	Authorize	Remove?
1	sdk_gphone_x86_64_arm64	sdk_gphone_x86_64_arm64	Google	09.09.2022 23:07:19	Authorize	Remove

Users

#	Name	Devices	Admin?
1	Oleg Iliev	1	

Authorized devices

#	Name	Model	Manufacturer	Added Date Time Utc

Property of Laboratory of Telematics - B

Device Manager / Authorize Device

Name: sdk_gphone_x86_64_arm64
Model: sdk_gphone_x86_64_arm64
Manufacturer: Google
Added Date Time Utc: 09.09.2022 23:07:19
User: Oleg Iliev

Property of Laboratory of Telematics - BAS

Users

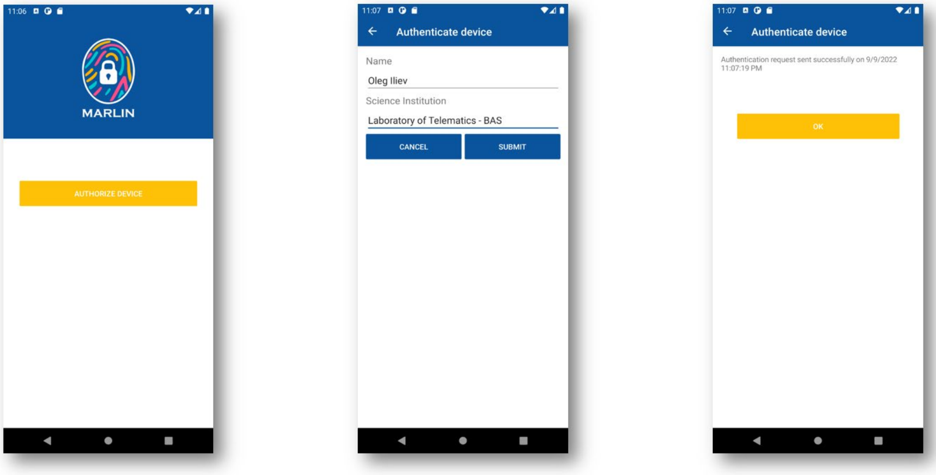
#	Name	Devices
1	Oleg Iliev	1

Authorized devices

#	Name	Model	Manufacturer	Added Date Time Utc
1	sdk_gphone_x86_64_arm64	sdk_gphone_x86_64_arm64	Google	09.09.2022 23:07:19

Fig. 22. Administrative Web System – Users' device management

After installing MARLIN native app on his/her own device the user needs to have the device authorized. The process is shown on Fig. 23.



Name: Oleg Iliev
Science Institution: Laboratory of Telematics - BAS

Authentication request sent successfully on 9/9/2022 11:07:19 PM

Fig. 23. MARLIN native app – Device authorization process

Once the device is authorized, the user may do the actual login process using biometric data as shown on Fig. 24.

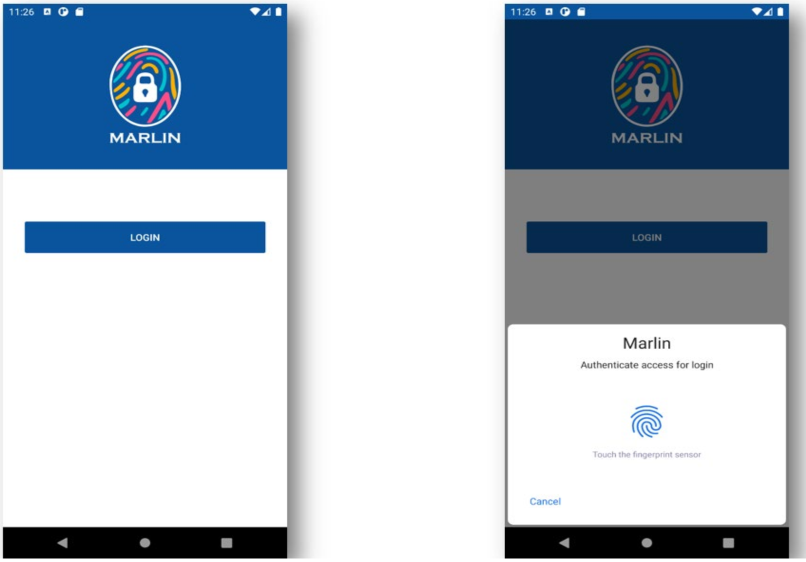


Fig. 24a. MARLIN native app – Login process using biometric data

In order to print a file, the user needs to first have it uploaded to the system. This could be done through the Administrative Web System or through the native app (Fig. 25).

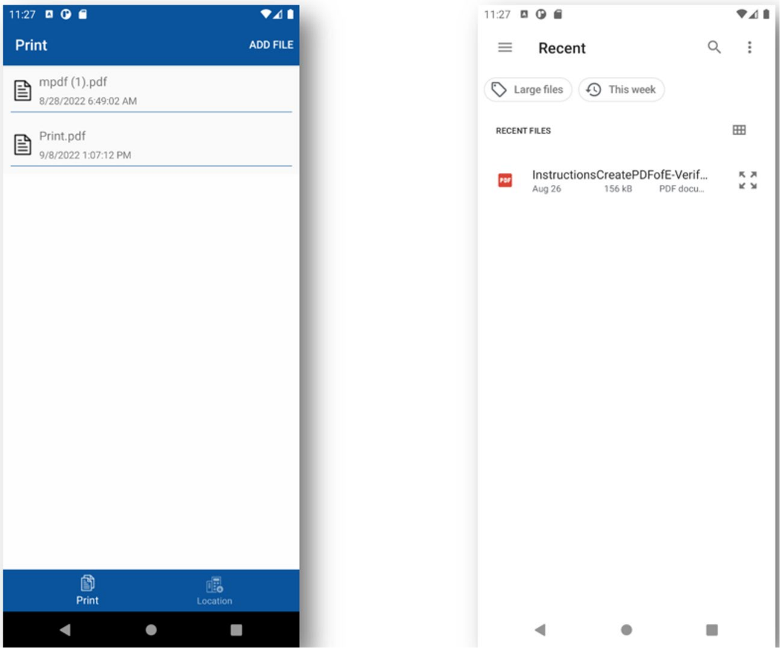


Fig. 25. MARLIN native app – File management and printing

Once the file is in the queue the user simple need to click on it and this is going to active the camera of the device. The user then need to scan the QR code labeled on the printer.

The access to the location and printing a file features are triggered by scanning a QR code by a module integrated into MARLIN native app as shown on Fig. 26.

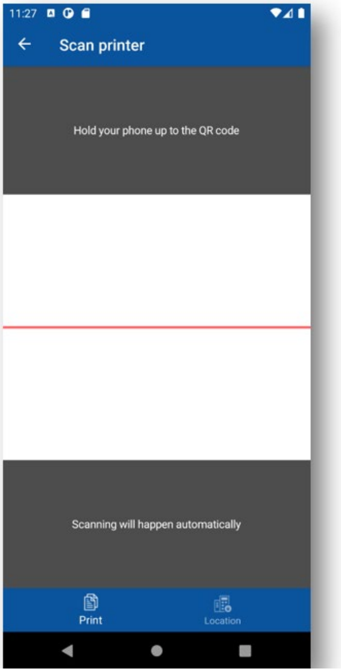

11:27
Scan printer
Hold your phone up to the QR code
Scanning will happen automatically
Print
Location

Fig. 26. MARLIN native app – QR code scanning

More about the prototype could be seen on https://youtu.be/Fp1QIKxawB8 – a short video that we recorded in order to demonstrate the MARLIN system.

4. Conclusions

The opportunities to support the scientific organizations in their collaboration processes are of utmost importance to increase the capabilities of the community they form. Although solutions such as eduroam, eduVPN, etc. already exist, these solutions are limited to providing Internet connectivity to users and remote access to their own local networks. The presented solution aims to provide secure access to the local resources of any organization by all members of the community using a specially created native application – MARLIN, biometric authentication and the mobile devices of the users themselves. In other words, apart from being innovative, the solution does not require large investments from the scientific organizations willing to implement it. At the same time, the possible applications of the solution are not limited to the focus of this research – providing users with access to printing resources and locations with restricted access, but can be further developed.

During the research we were able to outline several improvements for the next stages of the work, such as 1) replacing the QR codes with NFC tags; 2) design and development of a virtual printer, which is going to allow creating a printing “queue” from a wide variety of file types and does not require uploading a file to the system, which is then printed.

Acknowledgment

This research was sponsored by Vietsch Foundation in 2022. Thanks to the funding we were able to deliver the research as well as a prototype of both hardware and software system. The prototype was further implemented into Laboratory of Telematics – BAS. Without the assistance of the Laboratory we would not have been able to collect the necessary data and complete the research.

References

1. About eduVPN. (2022, October). Retrieved from eduVPN: https://www.eduvpn.org/
2. About FIND? (2022). Retrieved from The Framework for International Navigation and Discovery: https://www.internalpositioning.com/
3. Active Directory Domain Services Overview. (2022, October). Retrieved from Microsoft: https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview
4. Amo, D., Alier, M., Garcia-Penalvo, F. J., Fonseca, D., Casany, M. J.: GDPR Security and Confidentiality compliance in LMS' a problem analysis and engineering solution proposal. In Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'19). Association for Computing Machinery, pp. 253–259, 2019, https://doi.org/10.1145/3362789.3362823.
5. Android Fingerprint Security. (2016). Retrieved from Infinium: https://infinum.com/blog/android-fingerprint-security/
6. Androind developers documentation. (2022). Retrieved from Android ID: https://developer.android.com/reference/android/provider/Settings.Secure#ANDROID_ID
7. Apple developers documentation. (2022). Retrieved from An alphanumeric string that uniquely identifies a device to the app’s vendor: https://developer.apple.com/documentation/uikit/uidevice/1620059-identifierforvendor
8. Borissova, D., Dimitrova, Z., Garvanova, M., Garvanov, I., Cvetkova, P., Dimitrov, V., Pandulis, A.: Two-stage Decision-Making Approach to Survey the Excessive Usage of Smart Technologies. Problems of Engineering Cybernetics and Robotics, vol. 73, pp. 3-16, 2020, https://doi.org/10.7546/PECR.73.20.01
9. Borissova, D.: An overview of multi-criteria decision making models and software systems. In: Atanassov K.T. (eds) Research in Computer Science in the Bulgarian

Academy of Sciences. Studies in Computational Intelligence, vol. 934, pp. 305–323, 2021, https://doi.org/10.1007/978-3-030-72284-5_15.

10. Dasgupta, D., Roy, A., Nag, A.: (). Multi-Factor Authentication. In: Advances in User Authentication. Infosys Science Foundation Series, pp. 185–233, 2017, https://doi.org/10.1007/978-3-319-58808-7_5.
11. Dimitrova, Z., Borissova, D., Dimitrov, V.: Design of Web Application with Dynamic Generation of Forms for Group Decision-Making. In: Saeed K., Dvorsky J. (eds) Computer Information Systems and Industrial Management. CISIM 2021. Lecture Notes in Computer Science, vol. 12883, pp. 112–123, 2021, https://doi.org/10.1007/978-3-030-84340-3_9.
12. Garvanov, I., Garvanova, M., Borissova, D., Vasovic, B., Kanev, D.: Towards IoT-based transport development in smart cities: Safety and security aspects. Business Modeling and Software Design. BMSD 2021. Lecture Notes in Business Information Processing, vol. 422, pp. 392–398, 2021, https://doi.org/10.1007/978-3-030-79976-2_27.
13. Garvanov, I., Kabakchiev, H., Behar, V., Garvanova, M., Iyinbor, R.: On the modeling of innovative navigation systems. Business Modeling and Software Design. BMSD 2019. Lecture Notes in Business Information Processing, vol. 356, pp. 299-306, 2019, https://doi.org/10.1007/978-3-030-24854-3_23.
14. Global smartphone penetration rate as share of population from 2016 to 2020. (2022, October). Retrieved from Statista: https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/
15. Group Policy Overview. (2016). Retrieved from Microsoft Learn: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831791(v=ws.11)
16. M'Raihi, D., Machani, S., Pei, M., Rydell, J.: TOTP: Time-Based One-Time Password Algorithm. RFC 6238, DOI 10.17487/RFC6238, May 2011, https://www.rfc-editor.org/info/rfc6238
17. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), https://eurlex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A32016R0679, Accessed 2019-06-27
18. Secure Enclave. (2021). Retrieved from Apple Platform Security: https://support.apple.com/en-gb/guide/security/sec59b0b31ff/web
19. Using biometrics for authentication in Android. (2021). Retrieved from Samsung Insights: https://insights.samsung.com/2021/04/21/using-biometrics-for-authentication-in-android-2/
20. Weaver, A. C.: Biometric authentication. Computer, vol. 39(2), pp. 96-97, 2006, https://doi.org/10.1109/MC.2006.47.
21. What is eduGAIN? (2022, October). Retrieved from eduGAIN: https://edugain.org/about-edugain/what-is-edugain/
22. What is eduroam? (2022, October). Retrieved from eduroam: https://eduroam.org/what-is-eduroam/
23. Yoshinov, R., Iliev, O.: Secured with biometric protection local resource sharing within a community. In: 1st International Conference on New Approaches in Engineering, October 6-7, 2022, (ICNAE'22), Konya, Turkey.