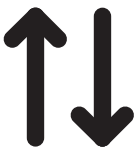
MOBILEKEY

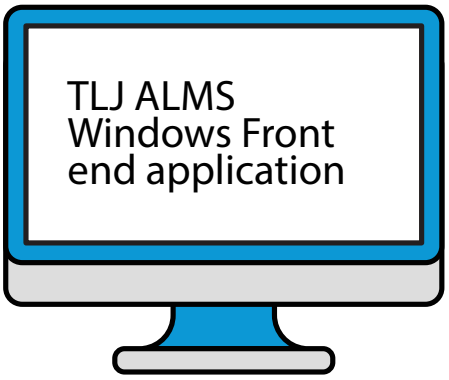
TLJ Mobile Keys

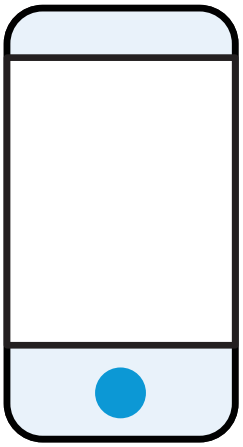
Enterprise

Data flow
infographic

TLJ hosted cloud
containing SQL database







Smartphone App,
iOS or Android. TLJ
native app of
custom App
integration using
TLJ SDK.

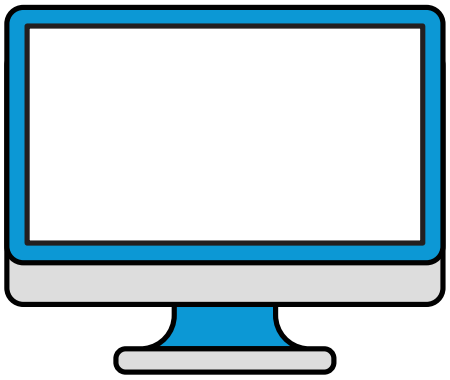
3rd Party Integration
using TLJ protocol,
allows Mobile Keys to
be issued by others

TLJ SECURITY SYSTEMS

The Cloud

The TLJ Cloud is hosted by UKFast, a scalable, 100% uptime, ISO certified & PCI complaint Windows Server running an SQL database. The data centres are exclusively UK based with award winning cyber and physical security, sitting behind a fully managed redundant Cisco ASA shared firewall.

Scalability is very important to TLJ as we expect demand for our Cloud based access control system to grow as well as an increasing mandate for Mobile Keys, TLJ protocol is to utilise a cloud provider that can accommodate such growth, UKFast more that fit the bill.



The Front End

The TLJ front end, officially known as Accommodation Locking Management System (ALMS) is now in its 6 version, and stacked full of features. Ultimate flexibility and control. It also incorporates the Mobile Key functionality, unlike other brands, TLJ did not want to 2 platforms; one for RFID cards and the other for Mobile Keys. Instead the operator can issue Mobile Keys and RFID cards right from the one platform and even at the same time, no hoping between screens and applications. The front end is developed in Delphi language in house by TLJ, and no scripts have ever been outsourced. TLJ have full control over their software, adapting to industry/market changes and fixing bugs is on-going process within the company and is part of the ISO9001 quality management.

The front end communicates to the SQL cloud database by TCP/IP using a dedicated port. This same protocol is used for receiving inbound commands from a 3rd party PMS (property management system). It can receive the following commands:

- Issue RFID token (key card, key fob, wristband etc)
- Clear RFID token
- Issue Mobile Key
- Delete Mobile Key
- Issue a multiple of the above within one request, i.e. Key Card and Mobile Key
- Bulk issuing of both Key Cards and Mobile Keys, i.e. one Student Accommodation site could send the all Mobile Keys in one press of a button from within their own PMS

TLJ offer 3rd party vendors an interface specification document which details all.

Multiple versions of the front end can be run at different locations simultaneously but must be from a Windows OS machine. The front end requires an encryption USB dongle to be present in the machine before it will load, this is a high end security feature implemented by TLJ from the very first mark 1 version of the TLJ front end. An RFID encoder device does not need to be present in order for the software to run, this is with a view that some users may also use the Mobile Key function or perhaps want to view a properties access control activity from a far.

There are different layers of operators within the software, allowing for unlimited flexibility on a users roles, and accessible doors once logged in. So for instance, Lisa on reception may only be able to issue replacement RFID cards, or Mobile Keys, where as David the Property Manager may be able to issue full site staff cards. This is all determined by the roles set with the program, and often managed by a system administrator.

Upon load a login dialogue box appears with a drop down combo box listing all users, and then requesting a password. Each login is logged with the SQL, as well everything done within it, such as issue records.

A multi site client may well opt to have an additional breakdown within the login dialogue, where by firstly the specific site is chosen from a combo box, this then pulls the operator names located within that site. This will ensure the operator list isn't so large.

The Smartphone App

The TLJ Smartphone Applications (iOS & Android) are both made in house using X-Code and Java respectively. They are hosted in the respective marketplaces by TLJ directly and regularly maintained and updated as and when required, i.e. Android 10's modification to the DeviceID library. TLJ constantly monitor changes in iOS and Android coding & functionality and thus apply App updates where applicable. Any reported bugs are also maintained by TLJ, and again fixed in latest releases. The app is relatively small in size, around 4.5Mb currently but with additional features to be added in Q1 2020.

- Minimum Specification iOS 10, with minimum 64bit CPU so that makes the iPhone 5S the current oldest handset capable of using the App.
- Minimum Specification running Android 5
- Minimum Bluetooth chipset within the device is Bluetooth 4.0

Operation – Registration & Terms and Conditions of use

On first run, the App will automatically display the brief user instructions on how the app works, with a view of limiting the number of support tickets. After the instructions the App displays T's & C's of use, these are detailed in GDPR & TLJ Mobile keys.pdf. The user is required to scroll right to the bottom in order to advance to the registration process, in an attempt to ensure they are read by the user. The content ensures that GDPR regulations have been followed.

Afterwards, the App will require a user registration (not applicable with 3rd party SDK releases). The required information is gathered within the app; user email address, chosen password, repeat chosen password. No other information is required from the user directly for registration. The required password must be of a minimum 8 characters in length and include at least one numeral, upper-case letter and lower-case letter.

Upon entering the registration information, the user is required to press a 'Submit' button which in turn will generate an automated email with a randomly generated verification code to be sent to the email address entered within the registration information. The App offers a text box and instructions to the user to collect the verification code from their inbox. The purpose is clear, it ensures the user has access to that email inbox prior to retrieving any mobile keys that have been issued to it. Successful input of the verification code will prompt the user with a successful registration message and will then display the ogin screen. The user would naturally now login using the email and password they have already entered during registration. The email address will be remembered for any future logins as a measure of convenience.

A successful login attempt will perform a background service which will generate a unique deviceID of that particular smartphone used to login. This will be sent to the TLJ Mobile Keys cloud, appending the user record table against that particular email address. Referred to as a device paring by TLJ.

It is used to prevent multiple devices being able to login to the same account, and in effect prevents unauthorised Mobile Key sharing. On any subsequent login's the same service will run, and if the DeviceID field is not blank it will perform a comparison to ensure the login attempt is from the same previous device. If yes, then the app will login and sync keys from the cloud. If no the login attempt will be denied and a prompt will show to the user advising them that this device is not the same as the previous one used, they should contact reception to reset the pairing. This can be performed within the TLJ front end application. In addition, there is an automatically generated email advising the user of an attempted login on another device, and that if it wasn't them they should immediately change their user password and take care not to share it.

With SDK 3rd party Apps the Registration process is not required and the responsibility is with the 3rd party. Mobile Keys in the cloud will not require a registered, verified user to retrieve them. It is paramount that adequate security measures are taken by the 3rd party. If TLJ feel this is not the case they have the ability to limit cloud access to that 3rd party.

The App will stay logged in for 7 days before requiring the user to login again, this has been designed this way in order to increase the speed of the user opening the door. It is advised with the first run instructions that the user uses some inbuilt OS security on their device, such as passcode, pattern, fingerprint or FaceID in to secure their device content.

Operation – Retrieving a Mobile Key

Mobile keys are automatically retrieved from the cloud upon logging in to the app. The email addressed used to login is used as a filter parameter in the Mobile Keys cloud SQL. Any Mobile Keys assigned to the same email address as that logged in will download to the device and will be stored locally on the device itself. This will assist with increased speed of door opening, when compared to an alternative process which involves synchronising from the cloud upon every login and app opening.

In this case it prevents failed door openings when the door is out of data signal range (WiFi or GSM) as the App stores the Mobile Key locally within the device and has no reliance on a cloud connection to open doors.

Preventative measures have been taken when it comes to revoked Mobile Keys. If a Mobile Key is deleted from the cloud, then of course this needs to reflect on the users App, which has the Mobile Key stored locally. Upon Login is not the only time the App will sync to the cloud, it also syncs if the app has been inactive for 2 hours, it can still be open in the background just not in use. The RAM of the device will flush and require a sync to the TLJ Mobile Key cloud, thus then deleting the revoked Mobile Key. A refresh can also be forced from with the App by pulling downwards.

Operation – Using a Mobile Key

Once a Mobile Key is downloaded, there will be a dialogue within the app advising the user that Mobile Keys are available, however they must be within range for them to actually appear and thus be operational by the app. i.e. the phone must be within approximately 5 meters of the door. The distance from which a door can be opened does vary, depending on the Bluetooth chipset within the users device, as well as the material/thickness of door in which the lock is installed.

This also acts as a further preventative measure for unauthorised Mobile Key use, as the App does not display the available keys until close enough to the door, so if the device was in the wrong hands one would not be able to determine the room number until/unless they were right in front of it.

Once within range of the door, the Mobile Key appears as green box displaying the room number or name. The user is then permitted to select it and press the 'padlock' icon to open it.
The users device then communicates to BLE (Bluetooth Low Energy) chip within the door lock using Bluetooth, it sends the encrypted open door command to the door lock who in turn will verify the Mobile Key is valid. The door lock will ensure:

* The Mobile Key matched the locks Authority Code (Serial Num)

* The Mobile Key Building ID matched the locks Building ID

* The Mobile Key Area ID matched the locks Area ID

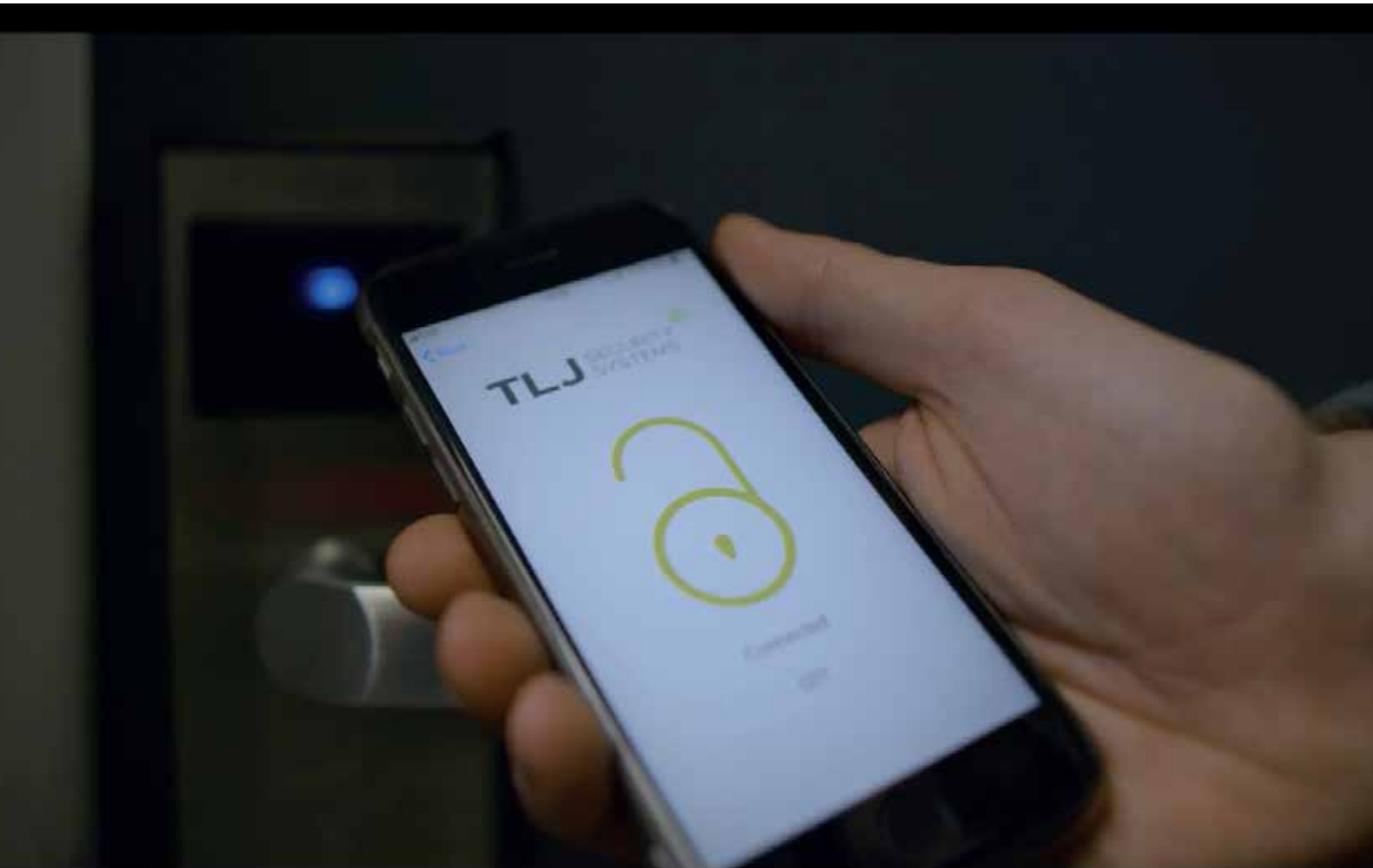
* The Mobile Key Floor ID matched the locks Floor ID

* The Mobile Key Room ID matched the locks Room ID

* The Mobile Keys start and end time/date is with the valid time, i.e. its not expired or too early

* The Mobile Key is not in the Blacklist stored within the locks memory

* The Mobile Key has the ability to over-ride the deadbolt double lock which may have been applied from the inside



Once this information has been verified, and there aren't any non-matching parameters the lock will unlock, sound an optional audible beep and display an LED. In addition the lock then notifies the APP the door has been successfully opened, which in turn then performs 2 tasks; 1) it pops up a prompt to the user that the door is open instructing them retract the lever 2) It sends the audit the log to the Mobile Keys cloud, which then appends the event viewer within the front end. Offering the operator real time records of who opened what door and when.

In the event that the smartphone does not have a data connection at the time of opening the door, the audit record will of course not reach the cloud, however as soon as a connection is back present the audit log will sync.

Contrary to that, if a Mobile Key fails to successfully open a door, the reason for rejection by the lock is prompted to the user within the App, for example – 'Mobile Key is not yet active', or 'The double lock is applied and the Mobile Key doesn't have the privileges to over-ride'. Finally if the lock has low battery power, this will prompt the user within the App, and in a future Q1 2020 release it will also transfer this to the cloud, and again the cloud to the front end, this will assist heavily with Maintenance and help prevent lock out issues.