jamk.fi

Tactilon Dabat and commercial applications

Juuso Ohra-aho

Bachelor’s thesis
May 2018
School of Technology, Communication and Transport
Degree Programme in Software Engineering

jamk.fi

Description

Author(s) Ohra-aho, Juuso	Type of publication Bachelor's thesis	Date May 2018
		Language of publication: English
	Number of pages 29	Permission for web publi- cation: x

Title of publication
Tactilon Dabat and commercial applications

Degree programme
Software Engineering

Supervisor(s)
Luostarinen, Hannu

Assigned by
Versine Oy

Abstract

The goal of this project was to inspect and document what is needed to port already commercial applications from Google Play Store to be compatible with Tactilon Dabat device. This project was assigned by Versine Oy, a company facing this procedure for their product Secapp. The goal was to make Secapp work flawlessly on Dabat.

Tactilon Dabat is a TETRA radio and 4G mobile phone hybrid device that runs on Android operating system. Therefore, it can run regular Android applications; however, it has some special restrictions and limitations. On the other hand, it offers some new possibilities through TETRA.

Secapp is a secure and centralized communication solution. It supports a great range of messaging channels including mobile applications, email, SMS, TETRA, voice call, automated robot calls and possible social media integrations. Since Secapp and TETRA devices already share similar user bases and use case scenarios, Secapp is a perfect application for Dabat users.

Managing to get an application to run flawlessly on Dabat greatly depends on the application and the features and device capabilities it uses. If the application relies too heavily on any of the features Dabat restricts, it could be completely incompatible with it. Some restrictions can be more or less easily eluded with similar enough solutions and if the application does not use single restricted feature, it will run "Out of box" on Dabat once licensed.

Keywords/tags (subjects)
Tactilon Dabat, TETRA

Miscellaneous (Confidential information)

jamk.fi

Kuvailulehti

Tekijä(t) Ohra-aho, Juuso	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2018
	Sivumäärä 29	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x
Työn nimi **Tactilon Dabat and commercial applications**		
Tutkinto-ohjelma Ohjelmistotekniikka		
Työn ohjaaja(t) Hannu Luostarinen		
Toimeksiantaja(t) Versine Oy		

Tiivistelmä

Opinnäytetyön tavoitteena oli tutkia ja dokumentoida vaatimuksia portata jo kaupallisia applikaatioita Google Play -kaupasta Tactilon Dabat-laitteelle. Projektin toimeksiantaja oli Versine Oy, jolla tämä toimenpide oli edessä tuotteelleen Secappille. Lopullinen päämäärä oli saada Secapp toimimaan saumattomasti Dabatilla.

Tactilon Dabat on TETRA-radion ja 4G gsm-puhelimen välinen hybridilaite, jossa on Android-käyttöjärjestelmä. Näin ollen sillä voi ajaa normaaleja Android-applikaatioita, mutta koska siinä on kuitenkin muutamia erillisrajoituksia perus-Androidiin verrattuna, vaativat applikaatiot sille todennäköisesti jonkin verran töitä, jotta ne toimivat oikein. Toisaalta Dabat myös tarjoaa TETRA-toiminnollisuutta applikaatioille, joka vaatii sekin erillistyötä.

Secapp on turvallinen ja keskitetty viestintäalusta. Se tukee laajaa valikoimaa viestikanavia mukaan lukien mobiiliapplikaation, sähköpostin, SMS-viestit, TETRA-viestit, äänipuhelut, automaattiset robottipuhelut sekä sosiaalisten medioiden integraatiot. Koska Secapp ja TETRA-laitteet jakavat jo valmiiksi samat käyttäjäkunnat ja käyttötapaukset, se on loistava applikaatio Dabat-käyttäjille.

Applikaation saaminen täysin toimimaan Dabatilla riippuu suuresti applikaatiosta ja eritoten laitteen omaisuuksista, joita se käyttää. Jos applikaatio toimii liian vahvasti poissuljettujen laitteistojen varassa, voi sen saaminen toimimaan Dabatilla olla täysin mahdotonta. Joitakin Dabatin rajoituksia pystyy enemmän tai vähemmän kiertämään tarpeeksi samanlaisilla ratkaisuilla, ja mikäli applikaatio ei valmiiksi jo käytä mitään rajoitettuja ominaisuuksia, toimii se Dabatilla suoraan, kunhan se on Airbussin toimesta lisensoitu.

Avainsanat (asiasanat)
Tactilon Dabat, TETRA

Muut tiedot

Contents

Figures

Terminology

Tactilon Dabat

A smartphone with TETRA radio, developed by Airbus.

ETSI, European Telecommunications Standard Institute

Independent non-profit standardization organization in Europe. Founded by CEPT, European Conference of Postal and Telecommunications Administrations, in 1988. ETSI has more than 800 members in 66 countries and five continents. (Etsi)

TETRA, Terrestrial Trunked Radio

European standard for trunked radio system. More detailed explanation is found in Chapter 1.2.

ACELP, Algebraic Code-Excited Linear Prediction

Algorithm used by TETRA to compress voice. Patented by VoiceAge Corporation. (Acelp)

GSM, Global System for Mobile communications

ETSI standard for 2G, second generation, digital cellular system. Successor of 1G analog cellular networks and succeeded by UMTS, 3G. In this paper, term GSM is mainly used to refer to commercial 2-, 3- and 4G networks. (GSM)

UMTS, Universal Mobile Telecommunications System

3G, third generation, digital cellular system. Component of IMT-2000 standard set by ITU, International Telecommunications Union. (UMTS)

LTE, Long Term Evolution

First candidate for 4G, fourth generation, digital cellular systems. (LTE)

Android

Operating system for smartphones and tablets currently developed by Google. Based on Linux kernel. Variants also exist for smart TVs, cars and wearables. Over 80% of sold smartphones run on Android (Global mobile OS market share 2017).

APK, Android application package

Package file format used by Android operating system to distribute and install apps.

API, Application programming interface

API is an abstract definition allowing different programs and processes to communicate with each other.

Bluetooth, IEEE standard IEEE 802.15.1

Bluetooth is a wireless technology standard for short range communications. 100% of new commercial phones, tablets and laptops sold have Bluetooth support. (Bluetooth distribution N. d.)

SIG, Bluetooth Special Interest Group

Global community of over 30 000 companies formed in 1998. Oversees the development of Bluetooth specifications and publishes them. (SIG)

USB, Universal Serial Bus

Standard for connection peripherals to computers. Usually the term USB refers to the USB connector.

Wi-Fi, IEEE standard IEEE 802.11

Technology for wireless local area networking (WLAN). Wi-Fi operates in 2.4GHz and 5GHz frequencies. (WiFi)

VPN, Virtual Private Network

Technology to extend private networks over public internet or "tunnel" one’s internet connection to another location.

NFC, Near Field Communication

Set of very short range, some centimeters, communication protocols. Widely used with contactless payment. (NFC)

SDK, Software development kit

A set of software development tools, APIs and applications for creating applications on some specific platform or system.

GPS, Global Positioning System

Satellite based positioning system owned by the United States government.

OAuth

Standard for access delegation between different systems. Usually in form of letting users (clients) to log into a third-party service through some more mainstream service, like Google or Facebook. In this paper, the term OAuth refers to OAuth version 2.0 that is not compatible with its predecessor, Oath 1.0. (OAuth 2.0)

1 Introduction

1.1 TETRA

TETRA, Terrestrial Trunked Radio (formerly Trans-European Trunked Radio), is a European standard for a trunked radio system standardized by ETSI. Its developed for mission critical environments to meet the needs of organizations such as public safety, transportation, utilities, government, and military with scalable architecture allowing networks ranging from single local area to national coverage. It has been developed since 1980s and was first published with version TETRA 1.0 in 1995. TETRA is comparable to Project 25, which is similar North American version. (Tetra pocket guide 2015)

TETRA devices can be operated either in DMO (Direct-Mode Operation) with other TETRA devices or in TMO (Trunked-Mode Operation) with TETRA base stations. TETRA device can also act as relay for DMO - DMO or DMO - TMO connections for other TETRA devices making it possible for TETRA devices without trunk network coverage to still be able to communicate over trunked network. Basic TETRA operations include voice transmission and status-, SDS- (Short Data Services) and call-out messages. (Tetra pocket guide 2015)

For some technical details, TETRA uses TDMA (Time division multiple access) channel method with four channels per radio carrier. It can transfer data up to 7.2 Kbit/s per timeslot, totaling of 28.8 Kbit/s with all 4 timeslots. It is modulated by $\pi/4$ differential quadrature phase-shift keying and its baud rate is 18 000 symbols/s, resulting in 36 000 bit/s gross. Voice is sampled at 8kHz and compressed using ACELP (Algebraic Code-Excited Linear Prediction) resulting in data stream of 4.567 Kbit/s. It is then encoded with error protection increasing the data rate to 7.2 Kbit/s, entirely filling up 1 timeslot. (TETRA specification 2007)

Even though in state of constant development, TETRA network has seen little development, especially for its data transfer speeds, in its 20-year lifespan. It still has many advantages compared to commercial GSM networks, including (Vizocom 2016)

1. Greater range.
   With much lower frequency than GSM, far longer ranges can be achieved with same transmit power.
2. Direct mode.
   If one, however, finds oneself in place without trunk network, one can still communicate directly to other TETRA users.
3. Operational modes.
   While GSM provides only one-to-one connections, TETRA supports one-to-one, one-to-many and many-to-many connections.
4. Security
   TETRA supports terminal registration, authentication, air-interface encryption and end-to-end encryption.

However, the technology behind TETRA is getting obsolete and data requirements, in particular, have skyrocketed in the past two decades. This can be seen directly in GSM network evolution, where its theoretical speeds have increased from 50 Kbit/s with 2G to 300 Mbit/s with LTE, which is a 600 000% increase.

The disadvantages of TETRA if compared to GSM are listed as follows:

1. Requirement of linear amplifier
   TETRA devices require linear amplifier to meet their RF specifications. This increases the size and power consumption of radio.
2. Data transfer speed
   While TETRA can provide full-duplex voice channel, its data transfers speeds are far from GSM or today’s requirements: 28.8kbit/s with one channel or 691.2 Kbit/s in an expanded 150 kHz channel.

1.2 TACTILON DABAT

Tactilon Dabat is a revolutionary product to the TETRA family of devices as it brings modern data services to TETRA systems. It is the world’s first smartphone with a TETRA radio. Figure 1 shows the Dabat Tactilon device.

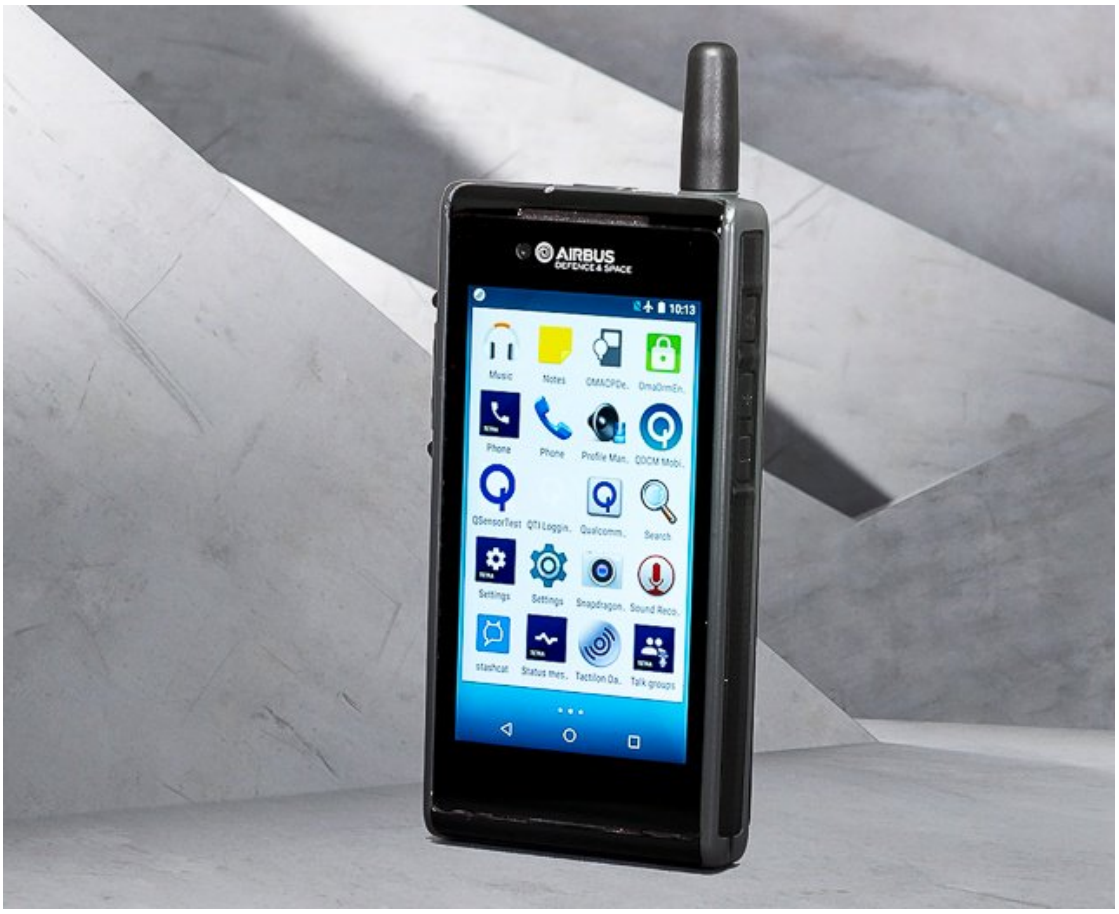

Figure 1 Tactilon Dabat

Much like TETRA networks, TETRA radios have developed in their lifetime; however, not really as much as they should have. They have become lighter, smaller and more durable. They have color displays now and they have gained some new features. But. They are still very much the same devices with the same functions as 15 years ago.

Meanwhile, the consumer devices and data speed requirements for internet usage have completely changed. Mobile phones with 3G internet access have been a norm for 10 years already. This has led to a situation, where many professionals have to carry two devices with them: TETRA radio for communication and a mobile phone for internet access.

Tactilon Dabats lives up to its slogan "Tactilon Dabat - both a smartphone and a TETRA radio" (Dabat). First of all, Tactilon Dabat is a fully-fledged smartphone. For commercial connectivity it supports LTE (up to 150 Mbit/s) and 3G GSM networks. Some of its smartphone features are listed as follows (Dabat technical specification 2017)

- Android 6.0 Operating system
- Octa core ARM CPU.
- 16GB Internal memory expandable with microSD card.
- GPS with A-GPS, Glonass and Beidou.
- Sensors: accelerometer, gyro-meter, proximity sensor and compass
- 16MP back camera with LED-flash
- 5MP front camera
- IP65 and IP67 certifications
- 2.4 and 5 GHz a/b/n/ac Wi-Fi and Bluetooth v4.2
- 4600 mAh Li-Po battery

Tactilon Dabat is also a fully-fledged TETRA radio. It has the emergency- and two functional buttons familiar from other TETRA voice devices. With TETRA voice it supports (Dabat technical specification 2017)

- Talking group selection
- Making and receiving group calls
- Making and receiving individual calls
- Making and receiving express calls
- Making and receiving emergency calls
- Making and receiving direct mode calls
- Priority scanning
- Late entry

With TETRA messaging it supports (Dabat technical specification 2017)

- Sending and receiving status messages
- Sending and receiving SDS messages
- Receiving callout SDS messages
- Receiving Flash messages
- Sending and receiving SDS and status messages also in direct mode

With TETRA networks it supports (Dabat technical specification 2017)

- Direct mode
- Trunk mode
- Sharing location over SDS/LIP
- Call log view
- Priority scanning
- Voice override in group calls

Airbus also offers PMR API for Tactilon Dabat that is a bridge between Android applications and TETRA features for any 3rd party applications on the device. The PMR API is an interface between the device's TETRA functionalities and applications installed on the device. It is still in development; however, the latest version already offers support for the following TETRA features on the device (PMR API specification 2017)

- Call status monitoring
- SDS and status messaging
- Network status monitoring

When implemented, the PMR features act like any other Android APIs. They require permissions belonging to a dangerous category; hence, they must be declared in apps manifest file, and they also require runtime access grant by the user. (PMR API specification 2017)

The communication with the PMR API works with intents, much like any other Android API, in two ways depending on the direction of communication (PMR API specification 2017)

- Binding to service (from application to PMR API)
  When application needs to use PMR API, it uses the Android Contexts method bindService to access functions provided by PMR API.
- A broadcast from service (from PMR API to application)
  Application can receive PMR APIs actions and events by registering an Android receiver for intents sent by PMR API.

1.3 Thesis objective

Dabat solves the problem of professionals' need to carry two different devices with them, however, its licensed only app selection creates new problem: a smartphone without good app selection is useless. Airbus needs to get developers to enroll and port their apps to Dabat for it to be useful and fill its purpose.

The aim of this thesis was to inspect and document the requirements and challenges of porting a commercial Android application from Google Play store to Tactilon Dabat and possibly make it easier for anyone doing so. This process itself is handled as abstract and applicable to any app there is, however, it reflects heavily the porting of Secapp to Dabat.

1.4 Disclaimer

This thesis relies on the current generation of Dabat, PMR API, development guidelines, kits and software versions. These might and most likely are subject to changes when matured and more customer feedback is received. Some of the restrictions or limitations mentioned in this thesis could, for instance, be changed by software and/or firmware updates from Airbus.

2 Limitations and challenges

The operating system of Tactilon Dabat is Android 6.0 Marshmallow, slightly customized by Airbus. Therefore, it is able to use all the native Android APIs, and it also inherits all limitations of Android.

Android is a relatively "free" operating system for developers; especially if compared to its "rival" Apples iOS. However, on the last couple major Android versions the capabilities a developer has at his/her use, especially regarding background execution, have decreased. Of course, these limitations do not affect Dabat since it runs on Android 6, rather than the latest Android 8, Oreo.

Because of its mission critical use cases, some restrictions have been implemented to Dabat's OS for security purposes. Many of the next limitations and restrictions can be disabled or bypassed if Dabat is in development mode, however, since this thesis covers the production circumstances, the development mode is completely ignored.

2.1 Google Mobile Services

Google Mobile Services (GMS) is a collection of Google apps and APIs usable by 3rd party apps. There is a more detailed description later on in this paper about it.

However, the absence of GMS is very unusual in commercial devices, which affects a great portion of already commercial apps. This will most likely be the biggest single limitation of the device, varying greatly with the features used by the app.

2.2 Application installation

On a normal Android system, applications can be installed straight from APK (Android application package) packages. In addition, almost every commercial Android device has at least one store app that allows buying and/or installing other apps.

With Dabat, both of these options are disabled. All apps one installs must be developed through Airbuses' smarTWISP program, and then it is licensed and approved by Airbus. After that it is still unclear if the Dabat has Appstore for apps from smarTWISP or are they installed directly from licensed APK packages.

Either way, Dabat will miss a huge number of apps offered from various app stores (mainly Google Play and Amazon app store) and direct APK packages from different sources.

2.3 Bluetooth

Bluetooth may be operated in numerous profiles; there are currently 27 still active (not yet deprecated) Bluetooth profiles specified by SIG. (Bluetooth profile list)

Dabat supports only Headset profile (HSP), Hands-Free Profile (HFP) and Serial Port Profile (SPP) Bluetooth profiles. In addition, the supported Bluetooth accessories must be whitelisted by product and manufacturer.

Bluetooth Low Energy (BLE) is also supported, however, only with LE Secure Connections.

2.4 USB

In USB protocol, a connection is established between two devices that operate in host and peripheral modes. In standard USB connections, the host and peripheral are defined by the cable, the device with A type connector is host and the one with B type is peripheral. Later on, SIG has introduced a specification for normally peripheral devices with B type connectors to act as hosts, USB OTG or USB On-The-Go. This allows devices to act as either host or peripheral, which ever seems more suitable. (USB On-The-Go)

Even though Dabat's hardware supports USB OTG, the host mode is disabled, leaving only the peripheral mode available to use. (Developing applications for Tactilon Dabat Terminal 2017)

2.5 Wi-Fi

Wi-Fi networks work traditionally with one router as access point and 0-N devices connecting to that network through a router. This is how every home network with Wi-Fi, coffee shop network or any public network operates (Pinola 2018)

Wi-Fi also supports direct communication from one device to another with no access point, which is called Wi-Fi Direct.

Another access pointless Wi-Fi operation mode is ad hoc Wi-Fi. In this mode, one device starts acting as a virtual access point, allowing other devices to join their network. If the device offering this "access point" has an internet connection through some other interface, such as mobile data or Ethernet, it may act as "Wi-Fi Hotspot", allowing internet access to devices in its network. (Toh & Delwar & Allen 2002)

Dabat may only be connected to whitelisted Wi-Fi networks. In addition, Wi-Fi Direct and Wi-Fi Hotspot modes are completely disabled. (Developing applications for Tactilon Dabat Terminal 2017)

2.6 VPN

To access the internet with Dabat, the device must be connected to Virtual Private Network (VPN). If the VPN configuration is missing, all data flow over Wi-Fi or mobile data is garbage. (Developing applications for Tactilon Dabat Terminal 2017)

2.7 NFC

NFC (Near field communication) is a technology used for very short distance data communications. To be operated, NFC requires a dedicated chip that most modern mobile devices have, Dabat included. One great potential of NFC communication is that if the communication is only one-way, the “tag” being read can be completely static, e.g. contactless bankcards. (NFC, what it does)

Even when Dabat has an actual NFC chip, the NFC features are programmatically disabled by the operating system. (Developing applications for Tactilon Dabat Terminal 2017)

2.8 Challenges

Many of the restrictions above simply cannot be eluded, for example, the required usage of VPN. It is unlikely that it affects one’s app in any way (unless the app inspects the device’s LTE or Wi-Fi connections for some reasons, e.g. location); however, this should still be kept in mind.

Nevertheless, most of the restrictions can be eluded in some way, although it could mean just using different techniques and actually ignoring the restricted feature. Unless any of the restricted hardware features is a part of the core functionalities of one’s app, the biggest challenge of porting apps to Dabat will most likely be the absence of Google Mobile Services. Surely, if one is making an app to pay groceries with, unusable NFC will be the biggest deal.

Even if it feels like not so significant a matter, apps that do not use any Google Services are rare. Google Mobile Services include apps such as Google Search, Google Chrome, YouTube and Google Play Store. Of course, there are many alternatives to these and they can be replaced easily, however, a bigger impact is caused by the absence of the other part of the Google Mobile Services, Google Play Services APK.

2.9 Google Play Services

Google Play Services APK is a collection of individual Google Services running as background service in Android. It can be interacted by other apps with its client library containing interfaces for all individual Google Services. The background APK may be silently updated from Google Play Store in the background; hence, apps always have the latest APIs to use. This is descripted in the following Figure 2. (Google Mobile Services, overview)

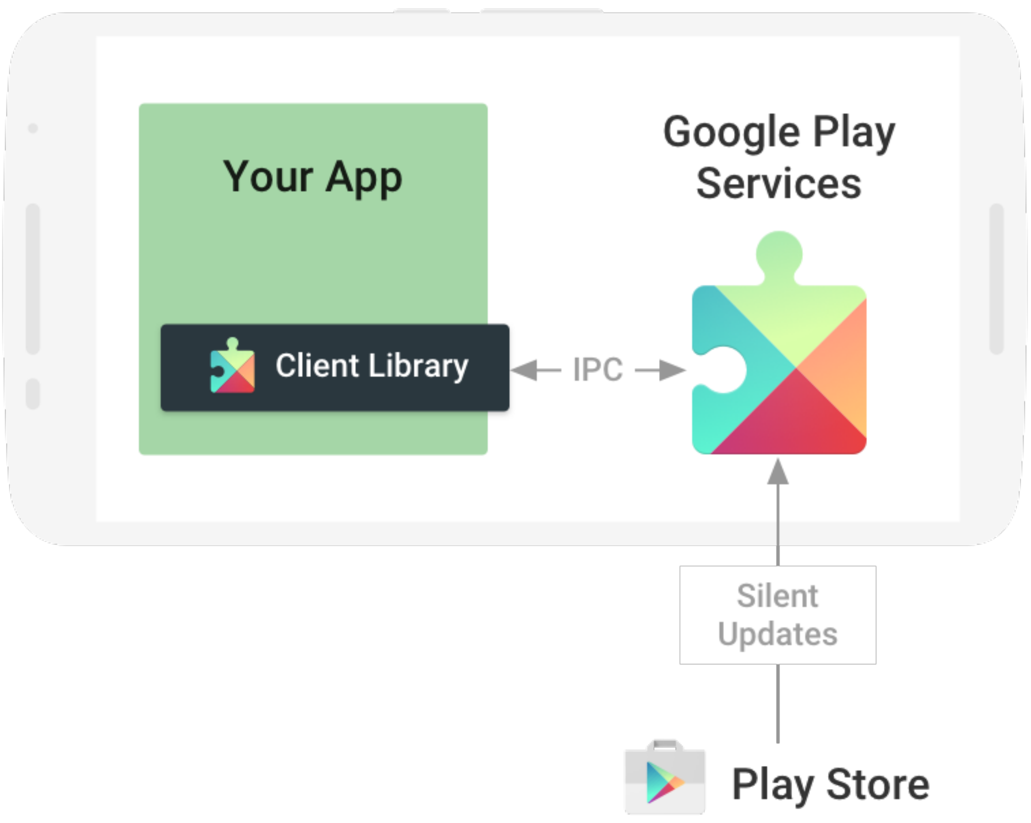

Figure 2. Google Play Services

Google Play Services contain at least 21 different APIs that can be separately imported on any app using Google Play Services. (Google Mobile Services, setup)

Individual services worth mentioning from the writers' perspective are Google+ that grants access to Google's social media features, Google Cloud Messaging that grants ability to receive push messages from Google's Cloud Services and Google Maps, granting access to Google Maps in apps.

3 Solutions

How to avoid all these restrictions? To simplify it, it is not possible. For some restrictions, alternative routes achieving same results can be found; however, there is no silver bullet here.

The restrictions could be categorized in two groups: hardware and software. Hardware group consists of features and capabilities supported by Dabat's hardware; however, they are either completely or partially disabled by OS. Software restrictions are special restrictions on Dabat's OS compared to stock Android.

These groups are:

- Hardware
  Bluetooth, USB, Wi-Fi, NFC
- Software
  Application installation, no Google Mobile Services

VPN "restriction" is hard to categorize, since it is the only one in the format of "must" instead of "cannot" and it does not directly affect any hardware or software features.

3.1 Hardware

Hardware restrictions are in format "this cannot be used" and are quite straightforward. They either affect the whole feature, like in case of NFC, or just parts of it, like with Bluetooth.

The only completely disabled hardware feature is NFC; all other restrictions are just partial, and those features can be used to some extent. For an alternative solution, there are far too many use cases of NFC to simplify a single valid alternative.

Many NFCs use cases could be covered with QR codes + camera where both devices are connected to the internet, e.g. pairing devices or exchanging small amounts of data, such as contact info. In situations where another device is offline, light data exchange is still possible, however, only to one direction without user interaction. For example, one’s contact data could be loaded to a QR image on a screen where the other party can read and save it, whereas if both devices were connected to the internet, the QR code could say "Hey, contact me in this address" and then the devices would have bidirectional communication.

Another approach without the internet could also be a USB connection initiated manually (or also with QR codes, for example). Since USB Host mode is disabled in Dabat, a straight USB connection from Dabat to Dabat is impossible, just like direct Wi-Fi connections with disabled Wi-Fi Direct and Hotspot modes. However, if the other device is not Dabat and can act as USB host, one way of replacing NFC could be a direct USB connection.

Therefore, even if NFC is completely disabled, there are alternative routes of communication. The only cases where communication is impossible are when the other device cannot act as USB host, does not support Bluetooth SPP (or headset profiles if one is really persistent) and is unable to provide data visually for reading with a camera. A debit card with contactless payment is a good example of this.

Other restricted hardware features are Bluetooth, Wi-Fi and USB. None of them is completely disabled; some parts of them are just either disabled, restricted or both.

Dabat supports only three previously mentioned Bluetooth profiles and secure connections. Therefore, if one needs to use Bluetooth with Dabat, the only way is to make the other device compatible with modes supported by Dabat: SPP, HFP or HSP. Dabat to Dabat data communication through Bluetooth is accessible through SPP profile.

For USB Dabat supports only the peripheral mode. Hence, just like with Bluetooth if the user needs to use USB, there is only one valid solution, which is to make another device compliant, which must act as host.

In Dabat, Wi-Fi does not support direct or hot spot modes. If it is crucial for the app that the device can act as router or connect to other devices directly, then alternative options for these operations need to be used.

For direct connect cases where one to one connection is needed between devices, Wi-Fi could be replaced with a Bluetooth connection or even USB if possible. Hot spot case is more complicated, since it requires one to many connections. Of course, also Bluetooth and USB could be used for this, however, if other devices in the hotspot network need to communicate to each other, the device would have to be emulated programmatically to act as router, which is a mode neither Bluetooth or USB is familiar with.

If the aim is to use the app actively between Dabats, then for small amounts of data Dabat’s PMR API offers some great functionalities for communicating with devices in both larger TETRA networks and directly between devices in the direct mode. It should be remembered that spamming a larger TETRA network with large data continuously is very likely illegal. Therefore, the amounts of data should be small, especially in trunked networks. The writer doubts that no one cares that much if one's own devices are suffocated in direct mode.

3.2 Software

Software limitations are the leftovers after hardware specification: App installation and missing Google Mobile Services. Neither of these is an actual limitation or restriction but rather a left out basic Android functionality.

As explained in Chapter 2.3, Dabat has disabled Android's capability to freely install any APK one gets one's hands on, and since the default app selection does not offer any market apps, there is no way of getting any unlicensed apps to run on Dabat. For this, the only viable option is to enroll to smarTWISP program and get the app licensed by Airbus.

The absent Google Mobile Services contains 21 different useful APIs missing from Dabat, and each of them has their own problems and solutions. The next subsections contain descriptions of some of the most used libraries from GMS and examples of how to do without them: Push messages, maps, locations and auth.

3.2.1 Push messages

One feature that is likely missing is the push messages from Google Cloud Messaging. An advantage is that with Android there are some alternative options. If one was to have the same kind of situation with iOS, the absent APNS (Apple Push Notification Service) that would be it. With iOS's extremely limited background functionality, there is no way to receive messages from one's server. Luckily, as said with Android, there are ways to go around this.

With Android, the background execution has not been so limited until very recently, or the current Android version O, Oreo, to be specific (Android background execution limits. N. d.). Whereas Apple limits the app's background processing time by default to max three minutes, there are no such limits with Android. Therefore, theoretically background service can be run forever with socket open for incoming data.

Of course, it is not that simple. The major reason is that Android OS can anytime it needs more memory, gets too low on battery or for any other reason it sees viable kill the app from background without warning. If one has a high-end device the battery charge of which never gets too low and one does not run too many CPU or memory intensive tasks, it is very unlikely this will happen. However, it can happen at any time and with older devices quite often and easily.

This must, of course, be prevented. One way is to run one's background process as foreground process, which effectively prevents the system from spontaneously killing one's app; however, it also bothers the user with an irremovable sticky notification on the status bar. Another type of solution could be using Android's own alarm manager to periodically check if the process is running and starting it if it is not. This solution also has some downsides. With the device on low-power idle mode, there is no guarantee that these alarms will be fired in up to 15 minutes after they should have been fired. Therefore, there is a possibility of up to at least 15-minute downtime anytime the OS decides to kill one's process. In addition, scheduling alarms frequently can significantly affect the device's battery life and blame it on the app.

Luckily, ditching the GCM is not just downhill; there are also some upsides of using "custom push channel" compared to GCM.

First, the pushes can be delivered much faster. GCM pushes are a "best effort" service, meaning that when one sends pushes to GCM, they try to deliver them as soon as possible, yet, without any guarantees. They may be delivered immediately, or after 15 minutes, or 2 hours. Usually they are delivered immediately, however, not always. Another part of this is that the device may also choose to ignore or delay the pushes. If one's battery is running low, the system may decide not to deliver the push to the app until the device is plugged in or to delay it and deliver it in bunch with other apps to save battery, with great latency. With custom channel, the OS cannot interfere with the actual receiving process.

Another missing feature of GCM is the delivery status. With GCM, when push is sent, the status of the GCM service receiving the push to be sent is obtained, not the actual delivery status. If the service returns error, one knows for sure that the push did not go through. But if the service returns OK status, one should not get overexcited because one still does not know if the push did go through to the device, or even that the service already has tried to send it. With custom channel, one will immediately get this info, since on is communicating directly with the recipient device instead of some man in the middle.

Sending pushes directly to devices can be more secure, however, also less secure if done wrong. Using GCM, all the user's data goes through Googles service and if the user does not manually encrypt the content of their pushes before sending and then again manually decrypts them in device when received, it is all visible plain text to Google. An advantage is that sending data to GCM ensures one is using an encrypted connection, preventing anyone between the user and Google reading the data. As mentioned, using custom push channel can be much less secure when implemented carelessly. If one does not use secure connection or encrypt one's data, it is visible to anyone between the user and the recipient device. However, when done right, using either a secure connection between device and server, manually encrypting data content or both, it can be more or equally secure compared to GCM.

GCM pushes are one-way deliveries. When the device receives a push from GCM and possibly needs to respond to the server, it needs to open a new connection. If the custom push channel is just a socket connection, then it could also be bidirectional allowing the device to answer to the backend through same channel it received the push message.

3.2.2 Maps

Another feature quite possible in use from Google Play Services is in app maps. This may not be as big a hindrance as push messages, yet, still noteworthy mainly because the Google maps is by far the easiest map solution to use in Android since it is free, and semi built in.

When Google maps are out of the question, another app needs to be found to do the job. Alternatively, not an actual app, since those cannot be installed on Dabat but rather a library that can be imported in one's own app to provide maps.

One candidate offering such is HERE maps. With free starter SDK, it offers 2D map tiles, satellite maps, car routing, pedestrian routing and many other features. (Here map SDKs)

A feature also likely used with Google Maps, this time with the standalone app, is navigation. Usually navigating in Android is carried out by sending a broadcast intent “I want to navigate to…” that is handled by any app capable of handling this action, however, since Dabat has no built-in handlers for navigation, that cannot be relied on. Therefore, if it should be ensured that one has a working navigation on Dabat, one has to provide it oneself in one’s app.

3.2.3 Locations

Android itself provides an easy way to listen location updates through the system’s own Location Manager. One can register to listen to these updates from different providers: network, GPS, passive and fused. (Android Location manager)

Of these providers, network and GPS are unambiguous: the network offers updates based on cell tower and Wi-Fi lookups through the internet and GPS are, of course, calculated from satellite signals and work offline. Unlike network and GPS, passive and fused modes are not so simple “get locations from this” modes.

The passive mode tries to provide locations without actually initiating any fetching. It listens to updates that may occur if any other app uses locations. This way the user can, for example, get locations in the background with no battery footprint for the app if there is any other app using locations somewhat continuously in the background. Another more ambiguous mode is fused. It provides combined location updates from all other providers for best possible updates. (Android Location manager)

However, a popular and easy way to use geolocations on Android is through Google Services. With GoogleApiClient from Google Services, location fetching can be easily set up with desired priority, interval, provider and other settings. None of these is actually any harder to set on Android’s native Location Manager, however, the biggest profit of using Google’s client might be that it has a shared location cache between apps. (Google locations API, overview)

The native Location Manager does not have any kind of cache. This means that when a location is requested from it, it is always a new one, even when using a low priority (a low accuracy) mode to save battery. This is the desired way to act if really frequent or accurate updates are needed, for example, if a user's running route is tracked. However, if the device's location needs to be tracked continuously in the background, this might be catastrophic for both devices' battery life and apps' battery usage statistics. (Android Location manager)

Using a passive location provider will save the app from the blame; however, if there are no other apps regularly using locations then one also gets no locations. This is where Google's client comes handy. With its shared location cache, fetching can be set up so that the locations already fetched within last X minutes are accepted. For example, if the device's location is tracked every 10 minutes, one could accept locations within five minutes so that the actual interval between locations would range between 5 to 15 minutes. This is completely acceptable for background tracking use case and with some luck, the app does not have to initiate any location fetching meanwhile being guaranteed to receive the location's updates. (Google locations API, client)

Hence, using Google's location client is effectively much like using Android's Location Manager with a provider combining passive and active (GPS, network or fused) modes. In addition, the more apps on the device use locations through Google client, the greater the benefits are for all of them by means of saved battery.

3.2.4 OAuth login

Google provides an easy way for developers to use OAuth login with Google account credentials through GMS. Using OAuth can eliminate the need for users to create dedicated accounts for single apps and rather let them run those apps requiring login with the same Google account.

This can be highly sought after by both app developers and users. For developers since they do not necessarily have to keep user database themselves, or if they do, they do not have to store sensitive login data. In addition, for users it is, of course, convenient if they do not have to create their own account for every single app they use. Developers should still keep in mind that not all users want to share single login between everything, so responsible developers should also provide their own login for an app even while offering Google or other OAuth logins.

Unlike previously handled Google features completely disabled with absent GMS, Google OAuth is an exception, it is still usable; just not through absent Google API that lets developers use it with some lines of code.

The easiest way to use Google OAuth without GMS is probably Google's own OAuth library for Java. It might be designed to be used in backend running Java; however, it should run flawlessly on Android.

In addition, of course, if the app just needs OAuth login and it does not need to be Google, one could use any other OAuth "provider" wanted: Facebook, Twitter, Amazon. All of them provide their own auth libraries.

3.2.5 Monetization and Ads

It is highly unlikely that any app that is wanted to be ported into Dabat actually uses adds or in-app purchases, but since they are so widely used it is covered in this paper anyway. A vast majority of apps on Google Play are free, over 90% and the majority of these apps, especially those targeted to consumers, are either "freemiums", add based or more often combination of both. (Distribution of free and paid Android apps in the Google Play Store from 3rd quarter 2017 to 1st quarter 2018 2018)

Freemium means that the app itself is free, however, its usability and/or functions are limited until a "full app" purchase or some subscription is bought. Many popular free apps in Google Play are freemiums with limited functionalities. Another form of freemiums are games that are free-to-play; however, playing them is agonizingly slow and annoying without in-app purchases. According to the article by Tristan Greene (Greene 2017), four out of five most downloaded apps on Google Play from 2017 were freemiums, with the last one being actually free children's app.

The other half of the majority of free apps is add funded. These apps have no restricted functionality despite being free, however usually they show continuously an add bar at top or bottom of the screen and maybe even a full screen popup add from time to time. If one has a light data plan and tries to conserve mobile data, these ads may silently eat up much of usable data and blame it on the host app. There is one popular model where the app shows continuously adds, however, they can be removed with one time in-app purchase.

Google Play Services offers great tools for both these options. Google In App Billing API makes it easy to buy an app content, and Google Mobile Ads makes it easy to display and monetize adds.

To replace in-app purchases all one must do is to implement another in-app billing provider. Since the app is not going to Google Play, there is no need to care about legal issues with Play Store about using 3rd party in-app purchases. If one can take credit card payments oneself, one can possibly make one's own "store", i.e. just billing the credit card for needed purchases. If that is not possible or far easier (and much likely safer) solution is sought after, one can use an already commercial 3rd party for this, Stripe for example.

The same goes for in-app ads. Google's AdMob is probably by far the most used platform for ad monetizing on Android; however, it is not the only option. This time even for apps on Google Play, since unlike in-app purchases there is no legal issues of using 3rd party provider for ads in Google Play. Some 3rd party ad providers are Airpush, InMobi and LeadBolt, all of them providing equal or even better services than AdMob.

4 Conclusions

If the app to be ported absolutely requires and is unable to operate without some of the restricted hardware, it is simply not portable. Contactless payment app cannot be made work on Dabat.

Otherwise, the amount of work required in porting an existing application from Google Play to Tactilon Dabat varies greatly with the features and hardware the app uses. If the app does not use any of the restricted features, it can be ported just by getting it licensed through smarTWISP.

Unless an existing restricted hardware interface needs to be changed to work in some new way, for example USB connection to Bluetooth SPP, most of the work will just be covering missing Google features. Luckily none of them is a complete show-stopper and with more or less extra work, they can all be converted to other approaches or providers.

Estimating the time required for the porting of an app is not easy, for it being so app specific. Shortest process, only licensing, may take something from a few hours to some work days for paperwork and probably some weeks of waiting for the licensing process, whereas reworking one or more restricted hardware interfaces could alone require several weeks or even months of work.

References

Acelp. N. d. Accessed on 11 May 2018. Retrieved from https://web.archive.org/web/20071014095716/http://www.voiceage.com/relatedstandards.php

Android background execution limits. N. d. Accessed on 26 April 2018. Retrieved from https://developer.android.com/about/versions/oreo/background

Android Location manager. N. d. Accessed on 11 May 2018. Retrieved from https://developer.android.com/reference/android/location/LocationManager

Bluetooth distribution. N. d. Accessed on 11 May 2018. Retrieved from https://www.bluetooth.com/markets/phone-pc

Bluetooth profile list. N. d. Accessed on 26 April 2018. Retrieved from https://www.bluetooth.com/specifications/profiles-overview

Toh, C.-L. & Delwar, M. & Allen, D. 2002. Evaluating the communication performance of an ad hoc wireless network. Accessed on 11 May 2018. Retrieved from. https://ieeexplore.ieee.org/document/1017498/

Dabat. N. d. Accessed 27 April 2018. Retrieved from https://www.dabat.com/

Dabat technical specification. 2017. Accessed on 11 May 2018. Retrieved from https://cdn2.hubspot.net/hubfs/542132/pdf/Tactilon_Dabat_EN_technical_specification.pdf

Developing applications for Tactilon Dabat Terminal. 2017. Develop Applications for Tactilon Dabat.pdf. Retrieved from https://www.tactilon-smartwisp.com/

Distribution of free and paid Android apps in the Google Play Store from 3rd quarter 2017 to 1st quarter 2018. 2018. Accessed on 7 May 2018. Retrieved from https://www.statista.com/statistics/266211/distribution-of-free-and-paid-android-apps/

Etsi N. d. Accessed 11 May 2018. Retrieved from http://www.etsi.org/about

Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017. 2017. Accessed on 20 April 2018. Retrieved from https://www.sta-tista.com/statistics/266136/global-market-share-held-by-smartphone-operating-sys-tems/

Google locations API, client. N. d. Accessed on 11 May 2018. Retrieved from https://developers.google.com/android/reference/com/google/android/gms/loca-tion/FusedLocationProviderClient

Google locations API, overview. N. d. Accessed on 11 May 2018. Retrieved from https://developers.google.com/android/reference/com/google/android/gms/loca-tion/package-summary

Google Mobile Services, overview. N. d. Accessed on 11 May 2018. Retrieved from https://developers.google.com/android/guides/overview

Google Mobile Services, setup. N. d. Accessed on 20 April 2018. Retrieved from https://developers.google.com/android/guides/setup

Greene, T. 2017. Google Play's top downloads in 2017: And the award goes to… Ac-cessed on 11 May 2018. Retrieved from https://then-extweb.com/apps/2017/12/07/google-plays-top-downloads-in-2017-and-the-award-goes-to/

GSM. N. d. Accessed on 11 May 2018. Retrieved from http://www.etsi.org/in-dex.php/technologies-clusters/technologies/mobile/gsm

Here map SDKs. N. d. Accessed on 26 April 2018. Retrieved from https://devel-oper.here.com/develop/mobile-sdks

LTE. N. d. Accessed on 11 May 2018. Retrieved from http://www.etsi.org/technolo-gies-clusters/technologies/mobile/long-term-evolution

NFC. N. d. Accessed on 11 May 2018. Retrieved from https://nfc-forum.org/what-is-nfc/about-the-technology/

NFC, what it does. N. d. Accessed on 11 May 2018. Retrieved from https://nfc-fo-rum.org/what-is-nfc/what-it-does/

OAuth 2.0. N. d. Accessed on 11 May 2018. Retrieved from https://oauth.net/2/

SIG. N. d. Accessed on 11 May 2018. Retrieved from https://www.blue-tooth.com/about-us

Pinola, M. 2018. Understanding Wi-Fi and How it Works. Accessed on 11 May 2018. Retrieved from https://www.lifewire.com/what-is-wi-fi-2377430

PMR API specification. 2017. PMR API function description for Android V1.0.8.pdf. Retrieved from https://www.tactilon-smartwisp.com/

Tetra pocket guide. 2015. Accessed on 11 May 2018. Retrieved from http://pocket-guide.tetra-association.com/english/files/inc/7ec1974c25.pdf

TETRA specification. 2007. Accessed on 11 May 2018. Retrieved from http://www.etsi.org/de-liver/etsi_en/300300_300399/30039202/03.02.01_60/en_30039202v030201p.pdf

UMTS. N. d. Accessed on 11 May 2018. Retrieved from http://www.etsi.org/technol-ogies-clusters/technologies/mobile/umts

USB On-The-Go. N. d. Accessed on 11 May 2018. Retrieved from http://www.usb.org/developers/onthego/

Vizocom. 2016. TETRA Radios vs GSM and Wi-Fi – What is the Correct Choice? Ac-cessed on 11 May 2018. Retrieved from http://www.vizocom.com/blog/tetra-radios-vs-gsm-wi-fi-correct-choice/

WiFi. N. d. Accessed on 11 May 2018. Retrieved from https://www.wi-fi.org/certifica-tion/programs