ETSI GS E4P 007 V1.1.1 (2021-05)

Europe for Privacy-Preserving Pandemic Protection (E4P); Pandemic proximity tracing systems: Interoperability Framework

Disclaimer

The present document has been produced and approved by the Europe for Privacy-Preserving Pandemic Protection ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/E4P-007

Keywords
COVID, eHealth, emergency services, identity, mobility, pandemic, privacy, security, smartphone

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Europe for Privacy-Preserving Pandemic Protection (E4P).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The COVID-19 pandemic has generated significant challenges for many countries and their citizens and showed that digital technologies could play an important role in addressing this and future pandemics. Various applications, services and systems for contact tracing (identification and notification of those who come in contact with a carrier) have been developed in different regions.

Despite the similar goal of automated detection of COVID-19 exposure as a complementary solution to manual tracing (interviews with people diagnosed with COVID-19 to track down their recent contacts), their functionality, technology, scale, required data and limitations are different and may not interoperate.

These systems are currently being deployed in different countries and many more are expected in the near future. In particular, mobile devices with their contact tracing applications can support public health authorities in controlling and containing the pandemic. In that purpose, E4P has been created to provide a technical answer to pandemic crisis not limited to COVID-19 by specifying interoperable digital contact tracing systems.

1 Scope

The present document defines an interoperability framework for pandemic digital contact tracing systems which allows the centralized and decentralized modes of operation to fully interoperate. The present document is part of the ISG E4P specifications describing contract tracing systems and thus aligned with ETSI GS E4P 003 [1]. It is mainly focused on interoperability between ROBERT and DP3T/GAEN, but also contemplates general interoperability mechanisms when more than two protocols can be present in a given geographical area.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS E4P 003 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); High level requirements for pandemic contact tracing systems using mobile devices".

[2] ETSI GS E4P 006 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); Device-Based Mechanisms for pandemic contact tracing systems"..

[3] ETSI GS E4P 008 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); Back-End mechanisms for pandemic contact tracing systems"..

[4] Bluetooth® Core Specification V5.2.

NOTE: Available at https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GR E4P 002 (V1.1.1): "Europe for Privacy-Preserving Pandemic Protection (E4P); Comparison of existing pandemic contact tracing systems".

[i.2] "Decentralized Privacy-Preserving Proximity Tracing", 2020.

NOTE: Available at https://github.com/DP-3T/documents/blob/master/DP3T%20White%20Paper.pdf.

[i.3] Exposure Notifications API.

NOTE: Available at https://developers.google.com/android/exposure-notifications/exposure-notifications-api.

[i.4] "Centralized or Decentralized? The Contact Tracing Dilemma", 2020.

NOTE: Available at https://infoscience.epfl.ch/record/277809.

[i.5] "ROBERT: ROBust and privacy-presERving proximity Tracing", v1.1, May 2020.

NOTE: Available at https://github.com/ROBERT-proximity-tracing/documents/blob/master/ROBERT-specification-EN-v1_1.pdf.

[i.6] "On the interoperability of Decentralized Exposure Notification Systems", June 2020.

NOTE: Available at https://arxiv.org/abs/2006.13087.

[i.7] "Interoperable Digital Proximity Tracing protocol (IDPT)", May 2020.

NOTE: Available at https://upcommons.upc.edu/handle/2117/189356.

[i.8] "Herald International Interoperability draft standard", 2020.

NOTE 1: Available at https://vmware.github.io/herald/specs/payload-interop.

NOTE 2: Herald exposure notification solution, hosted by Linux Foundation Public Health.

[i.9] "DESIRE: A Third Way for a European Exposure Notification System Leveraging the best of centralized and decentralized systems, version 1.0", May 2020.

NOTE: Available at https://hal.inria.fr/hal-02570382/en/.

[i.10] "Interoperability of decentralized proximity tracing systems across regions", v2.2, 2020.

NOTE: Available at https://drive.google.com/file/d/1mGfE7rMKNmc51TG4ceE9PHEggN8rHOXk/.

[i.11] "European Proximity Tracing: An interoperability architecture for contact tracing and warning apps", 2020.

NOTE: Available at https://ec.europa.eu/health/sites/health/files/ehealth/docs/mobileapps_interop_architecture_en.pdf.

[i.12] "European Interoperability Certificate Governance: A Security Architecture for contact tracing and warning apps", 2020.

NOTE: Available at https://ec.europa.eu/health/sites/health/files/ehealth/docs/mobileapps_interop_certificate_governance_en.pdf.

[i.13] "A state-of-the-art Diffie-Hellman function".

NOTE: Available at https://cr.yp.to/ecdh.html.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

dBm	Decibel-milliwatts
KB	Kilo Byte
MB	Mega Byte

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

4G	Fourth generation of broadband cellular network technology
5G	Fifth generation of broadband cellular network technology
API	Application Programming Interface
app	Application
BCGL	Backend Certificate Governance and Lifecycle
BF	Back-end to Federation
CC	Country Code
CH	Confédération Helvetique
COVID-19	COronaVIrus Disease 2019
DCT	Digital Contact Tracing
DCTS	Digital Contact Tracing System
DH	Diffie-Hellman-Merkle
DP3T	Decentralized Privacy-Preserving Proximity Tracing
EBID	Ephemeral Bluetooth® IDentifier
ECC	Encrypted Country Code
EFGS	European Federation Gateway Service
EphID	Ephemeral IDentifier
EU	European Union
FGS	Federation Gateway Service
GAEN	Google Apple Exposure Notification
GATT	Generic ATTribute Profile
GDPR	General Data Protection Regulation
HTTP	Hypertext Transfer Protocol
ID	IDentifier
IDPT	Interoperable Digital Proximity Tracing
IO-C	InterOperability Challenge
JSON	JavaScript Object Notation
LE	Low Energy
LTE	Long Term Evolution
MAC	Medium Access Control Address
MTU	Maximum Transfer Unit
OS	Operating System
PDU	Protocol Data Unit
PKI	Public Key Infrastructure
RI-ECC	RI Encrypted Country Code
ROBERT	ROBust and privacy-presERving proximity Tracing
RSSI	Received Signal Strength Indication
SIG	Bluetooth® Special Interest Group
TLS	Transport Layer Security
TV	Television
TXpower	Transmitted power
UK	United Kingdom
UTF	Unicode Transformation Format
UUID	Universally Unique IDentifier
UWB	Ultra-Wide Band

4 Interoperability challenges

4.1 Overview

This clause discusses the challenges that appear for achieving interoperability between different DCTSs. As discussed in clause 4.1 of ETSI GR E4P 002 [i.1], DCTSs aim at providing an automated, privacy-preserving method of detecting potential contagion and warning people to apply for screening. The aim of interoperability is to make possible this functionality for people using different DCTS apps which, for instance, have been developed by different public authorities.

The main requirements related with interoperability are the following; see clause 5.11 of ETSI GS E4P 003 [1]:

- [HL-IO-01] Epidemiological criteria alignment;
- [HL-IO-02]: Mobile Application interoperability;
- [HL-IO-03]: Infrastructure in a Federation; and
- [HL-IO-04], [HL-IO-05]: Diagnosed roaming user.

Regarding the reference device architecture defined in ETSI GS E4P 006 [2], the main involved reference points are:

- reference point DB (Device - Backend System) - Backend interface; and
- reference point DD (Device - Device) - Contact proximity detection interface.

While for the reference backend architecture defined in ETSI GS E4P 008 [3], the main involved reference point is:

- Reference point BF (Federation Interface).

Clause 4.2 discusses Bluetooth® LE implementation challenges: Standards compliance, functional breadth, and reliability of low-level protocols like Bluetooth® on consumer systems that were designed for accessories like Bluetooth® audio, and not accurate medical risk estimation or contact tracing applications.

Clause 4.3 discusses the interoperability challenges for digital contact tracing protocols. The main contact tracing protocols developed so far are based on decentralized or centralized design approaches, as defined in ETSI GS E4P 003 [1]. The protocols that will be covered in the present document are:

- DP3T/GAEN: the version of the DP3T decentralized protocol specified in [i.2] which is based on the use of the GAEN API [i.3]; and
- ROBERT: the centralized protocol specified in [i.5].

ETSI GS E4P 006 [2] describes a third protocol, DESIRE, that can operate following either centralized or decentralized approach; see [i.9]. However, interoperability aspects for this protocol are not covered in the present document.

Clause 4.4 examines the challenges of keeping the same privacy characteristics of the DCTS when they work in stand-alone in case of interoperability, while clause 4.5 briefly discusses the problem that could arise in terms of amount of exchanged traffic when many users of DCTS interoperate. Clause 4.6 briefly discusses the interoperability challenges that appear when different DCTSs that use different risk scoring algorithms, although a detailed description is left out of the scope of the present document. Other challenges to interoperability not covered in the present document are:

- Access to technology (e.g. handsets, wearables) and network access (Internet) varies greatly according to geography, income, and community. Supporting only the latest handsets denies poorer and more at-risk communities access to this technology.
- Governments may take different decisions on approach based on local needs. These decisions have, by necessity, been taken independently with urgency. This includes different risk appetites and approaches for individual privacy and national security.

Clause 5 is devoted to discuss the interoperability between the device-to-device payload exchange protocols used by different DCTSs. Clause 6 is devoted to the case when the DCTS use the same design approach, while clause 7 discusses the interoperability when the design approach of the involved DCTSs is different. This distinction is necessary as the interoperability challenges are considerably different in the two cases. The main problems of interoperability between DP3T/GAEN and ROBERT appear due to the different privacy properties of these two protocols; see clause 7.2 of ETSI GS E4P 008 [3]. A direct interoperation between them would lead to major changes in privacy properties for some of the users in relation with standalone systems, as discussed in clause 7.1. A solution that implies some modifications in the protocols, but that preserve the privacy properties of the different systems is discussed in clause 7.2.

Finally, clause 8 discuses interoperability between the device-to-device payload exchange protocols when more than two protocols need to be supported.

4.2 Bluetooth® LE Interoperability challenges

4.2.1 Introduction

If two systems are using different Bluetooth® LE advertisement modes, the related applications might not be able to share data and trace contacts when at proximity.

If two systems are using the same Bluetooth® LE advertisement modes, but a different Bluetooth® LE payload, the related applications might not be able to understand each other data and trace contact when at proximity. This is discussed in detail in clause 5 of the present document.

4.2.2 Bluetooth® LE mode to advertise and scan

The applications of two different DCTSs could use different advertisement modes, as presented in clause 5.1.2 of ETSI GS E4P 006 [2].

The applications should be capable of using the different mode of advertisement described in in clause 5.1.2 of ETSI GS E4P 006 [2].

4.2.3 Bluetooth® LE Advertisement Payload

The applications of two different contact tracing systems could use a different payload as described in clauses 5.2.1.1.1 and 5.2.2 of ETSI GS E4P 006 [2].

It is recommended that all applications are using the same payload format and content. If not, the applications should be capable of sharing and understanding the different payload described in clauses 5.2.1.1.1 and 5.2.2 of ETSI GS E4P 006 [2].

If two applications use the same payload content and format, they should use the same UUID as described in [4].

If two applications do not use the same payload content and format, they should use a different UUID.

For instance, the UUID and payloads contents of the application used in France and in Germany are different. The UUIDs are respectively 0xFD64 in France and 0xFD6F in Germany; the payloads contents are also different as different protocols are used.

4.2.4 Bluetooth® standard versions in use

Since its introduction in 2010, Bluetooth® LE has gone through several versions, each introducing additional modes and features. Using the latest Bluetooth® 5 protocol and its security features would mean a large proportion of the population could not access the benefits of DCTS applications. Many wearable and embedded chips used for DCTS applications may also only support older standards such as Bluetooth® LE 4.0 and 4.1.

It is recommended that DCTS applications ensure their protocols can be used back to Bluetooth® LE 4.0.

4.2.5 Bluetooth® hardware support for advertising

Many phones' use of Bluetooth® is limited to accessing external peripherals such as speakers and car hands free kits. They were not designed primarily to act themselves as peripherals. This means many Bluetooth® chipsets, whilst technically capable of being used for advertising, do not have the firmware necessary to provide these services to the host operating system. During 2020, handset manufacturers have improved firmware, but these updates may not be widely applied in existing handsets without automatic updates enabled.

The implication of this is that an advertising only protocol may not allow certain devices, even though they may have been produced in the last 2 years, to be 'seen' and recorded as a contact in a DCTS application.

4.2.6 Operating System Bluetooth® compliance

Many parts of the Bluetooth® standard are optional. As mentioned in the previous clause the use of phones in both central and peripheral modes was not a common use before 2020. As a result, mobile phone OS' support varies for certain parts of the standard.

These challenges start from the modes of advertising (passive, active, etc.) and scanning (continuous with callbacks, or duration based), extend through the handling of connection sessions (e.g. expecting interactions between devices being serial in terms of request/responses, and stalling and timing out if this may not occur), characteristic modes supported (e.g. no write without response) and finally to the handling of data exchanges (e.g. fixed MTU, buggy MTU negotiation on Android).

All of these variations from the standard require specific handling or a reduction in the number of Bluetooth® features that can be relied upon with which to implement a device-to-device protocol and payload exchange over Bluetooth®; see clause 5.

4.2.7 Accessing information necessary for accurate contact detection and risk calculation

Much research has occurred in 2020 in to distance estimation, and thus risk estimation, based on Bluetooth® RSSI data. In order to best correct these estimations, it is not only needed to know the local RSSI for the remote device, but also ideally the phone make and models for each device, and the transmit power of the remote advertising device. Some of this is present in the clear via Bluetooth® standard services but is not always present across all phones and mobile operating systems' Bluetooth® advertisements.

4.3 General challenges related to the functional requirements

4.3.0 Satisfying requirements for interoperability

One of the main interoperability requirements is functional requirement HL-IO-03, which is repeated here for convenience:

- **[HL-IO-03]:** Functionality in a Federation: The Federation shall allow to notify, within the delay mentioned in Timing of notification of users at risk, a User at risk in one of its DCTS that was at risk because of its proximity to a User tested positive in another of its DCTS.

Satisfying requirement HL-IO-03 is challenging even across DCTSs with a common design approach (see clause 6) and even more challenging across those with different design approaches (clause 7). Conceptually, a high-level solution to this requirement differs depending on whether a User that tested positive (Diagnosed User) is located in their home country at the time of proximity event (i.e. the User is a Local) or in their roaming country at the time of the proximity event (i.e. the User is a Traveller). Therefore, the two Users in HL-IO-03 can be two Locals, a Local and a Traveller, and two Travellers. As the case of the proximity encounter of two Locals is not an interoperability challenge as it needs to be solved by any individual DCTS, the focus is put on the remaining cases.

Consider first the case of proximity encounter of a Local and a Traveller. Two different cases should be distinguished:

1) the case where the Diagnosed User is a Traveller and a User to be notified is a Local; and

2) the case where the Diagnosed User is a Local and a User to be notified is a Traveller.

4.3.1 Notification to a Local about a diagnosed Traveller

This case is depicted in Figure 1. Alice lives in country A. Alice gets in proximity of a Traveller, Bob, who could be from any country in the world. Bob returns to his country B two days later and is tested positive. Alice, aware of Bob's symptoms, is concerned she might be infected, too. How does Alice learn about Bob's infection, without being a User of all DCTSs in the world?

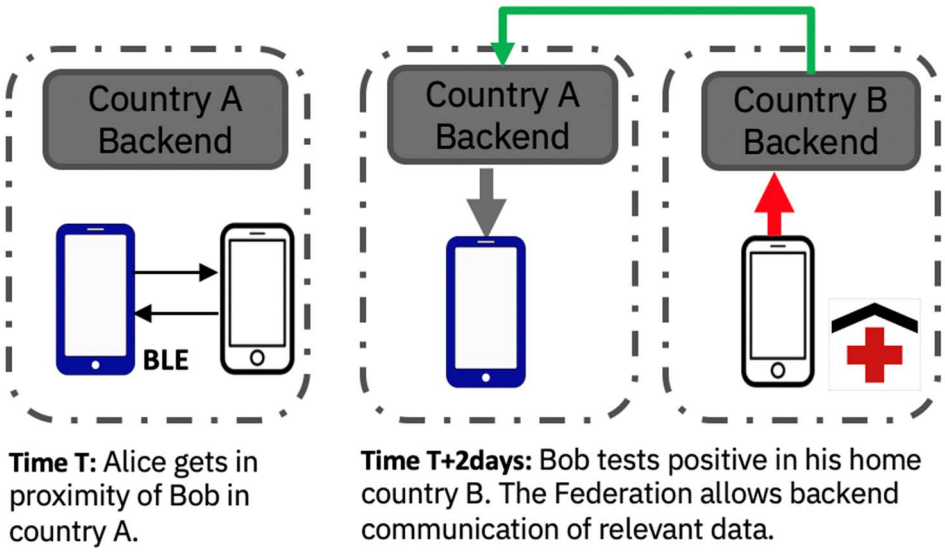


Figure 1: Local Alice infected by traveller Bob

The first case to be considered appears when the system includes geographic information in the exchanged beacons, as it is the case of ROBERT. In this system, the beacons include an Encrypted Country Code (ECC) of the sender of the beacon. In ROBERT, the system exchanges exposed anonymous identifiers, meaning that Bob will include the beacons received from Alice in the list of exposed keys uploaded to the Country B backend when he notifies a positive test to the app. The Country B backend should have thus a method for de-crypting the ECC contained in the beacons of Alice, to known to which server it should relay Alice's exposed keys.

The second case appears when no geographical information is included in the exchanged beacons, as is the case of DP3T/GAEN. Notice that the main issue here is that Alice has no idea from which country Bob comes from. Clearly, it is unfeasible for Alice to install all possible Mobile Applications pertaining to every DCTS. Even if Alice's and Bob's countries use DCTSs with the same mechanism, it might be impossible for Alice to unselectively listen to all backends from all countries due to sheer volumes of data. For instance, assuming an scenario in which 300 000 daily COVID-19 infections are notified in the world using a federated DP3T/GAEN system, every user would need to download more than 70 MB of data daily; see [i.6].

Instead, if Diagnosed Users would upload some coarse-grained travel/roaming information to backends, this information would be very helpful to improve scalability of the Federation. For usability and privacy requirements, the information about visited countries should not be fine grained. There are two options in this case:

- Partial replication, on a need-to-know basis, across a majority of countries. In this case, Bob would inform the Federation (starting from country B backend) about the fact that he visited country A. This would allow Federation to propagate critical information from country B to country A.

- All-to-all replication, across a cluster of affiliated countries (e.g. EU countries). In this case, Bob would not need to upload his travel information to country B backend, but all data would be replicated across all backends belonging to a cluster.

Consequently, interoperable backends of DCTSs, comprising the Federation, shall be able to communicate with each other in a secure and authenticated manner and disseminate critical information among each other.

4.3.2 Notification to a Traveller about a diagnosed Local

If the system includes geographic information in the exchanged beacons, as it is the case of ROBERT, the situation is identical as in the previous clause.

In the case of a solution without geographical information exchanged in the beacons, as it is the case of DP3T/GAEN, and conversely to the case described in the previous clause, if Alice (Local) gets infected, there is no way she could direct the diagnosis information to be propagated from country A backend to country B backend (recall that Alice has no idea where Bob comes from). As global all-to-all replication involves prohibitive volumes of data, the Federation needs to allow Bob to listen to information coming from country A backend; see also illustration in Figure 2. In this case, Bob knows to which country backend to listen, as he knows he travelled to country A.

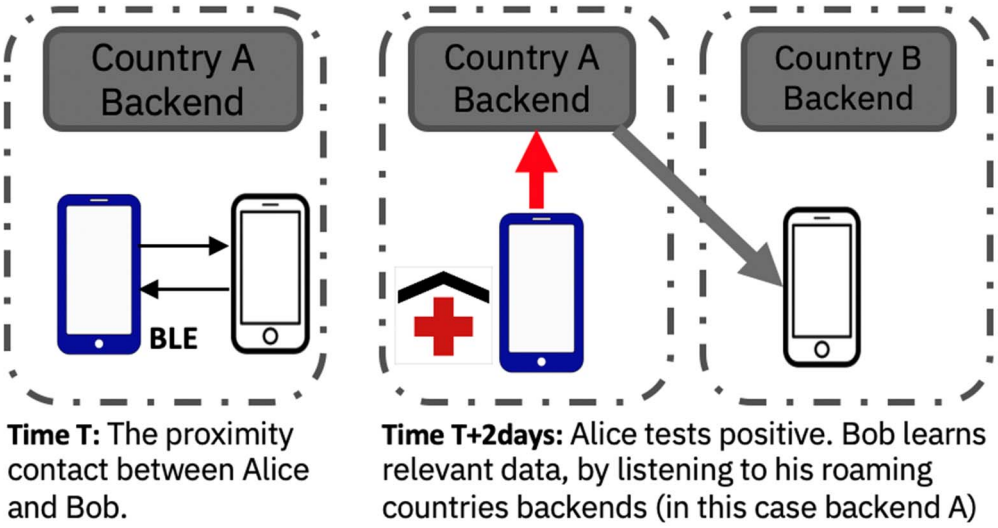


Figure 2: Traveller Bob infected by Local Alice

In a cluster of country backends, which perform all-to-all replication, as discussed in clause 4.3.1, Bob may get relevant data directly from backend B.

4.3.3 Notification to a Traveller about a diagnosed Traveller

If the system includes geographic information in the exchanged beacons, as it is the case of ROBERT, the situation is identical as in the previous clauses.

In the case of solution without geographical information exchanged in the beacons, as it is the case of DP3T/GAEN, it is fairly easy to show that a Federation system which solves challenges described in clauses 4.3.1 and 4.3.2 can also solve this challenge.

4.3.4 Authenticity of positive test information in case of a roaming User

Clause 4.3.1 discussed the case in which Bob gets tested positive in his home country B. The situation changes if Bob is tested positive in country C (different from B). Assuming that Bob cannot reach his home country B (in which case solution outlined in clause 4.3.1 would apply), the Federation needs to allow Health Authorities of country B accept positive test results of country C so Bob is allowed to upload relevant diagnosis data to country B backend.

Assuming Federation requires Bob to upload relevant diagnosis data to country B backend upon Bob tests positive in country C, this requirement can be satisfied leveraging a Verifiable Credentials standard compatible solution, which could be implemented on a decentralized verifiable credentials platform, such as a permissionless or permissioned block-chain. In a nutshell, in such a solution, public certificates of health certificate issuers (Health Authorities) are stored on the decentralized verifiable credentials platform (with no information pertaining to Users being stored on the said platform).

Alternative approach would be to require Bob to upload relevant diagnosis data to country C backend. However, this approach would pose serious operational and implementation problems as mobile applications are normally capable of communicating with the home backend of that application, not with an arbitrary roaming backend.

4.4 Maintaining privacy and security characteristics between different systems

The proposed digital contact tracing protocols can be vulnerable to several potential attacks to privacy and security; see clause 7 of ETSI GS E4P 008 [3]:

- Risk of obtaining the identity of a user from the knowledge of the ephemeral identifiers (i.e. the possibility of tracking people).
- Risk of disclosing the graph of contacts of users.
- Risk of identification of infected people. All digital contact tracing methods are vulnerable to this attack for individual users. Large-scale attacks are a potential vulnerability of some of the methods.
- Risk of injection false at-risk alerts.
- Risk of being pressed to opt-in.

One challenge for interoperability is to ensure that the DCTS interoperability infrastructure shall retain, to the extent possible, the security and privacy provided by individual DCTSs.

This is especially difficult to achieve when the interoperability between systems with different mechanism is considered, as they can have very different properties regarding privacy and security; see clause 7.

4.5 Generated traffic

As discussed in clause 4.3.1, interoperability across different DCTSs imply that some information is exchanged between backend servers supporting different DCTSs. Depending on the adopted architecture (e.g. all-to-all or partial replication) the amount of exchanged traffic can be substantially different. This is an important factor to be taken into account when interoperability across DCTSs with many users (potentially, billions of users) is aimed.

4.6 Interoperability between DCTS applications

The present document mainly deals with technical aspects of DCTS interoperability. There are, however, other aspects that are key to achieve interoperability between DTCS applications, for instance:

- Harmonization between countries of procedures to request tests, to obtain and enter authorizations to release proximity events.
- Criteria and algorithms used to record proximity events, and the way proximity events are structured and handled in the DCTS app.

The DCTS apps should be able to record significantly more events than would be found with manual contact tracing, which requires the '15/1.5' style criteria to be abandoned; furthermore, that two sets of criteria are used for proximity events, with:

- one set for recording events;
- the second set for selection of events for uploads.

This would allow events to be recorded for the benefit of the user (mapping recorded encounters per day, per week, to give the user an idea of possible risks) and for research, while the number of proximity events used to generate warnings could be controlled separately.

Finally, multiple sets of criteria could be loaded, where each set would correspond to the criteria to be used for a certain region or risk level that could be coupled to regional indications by the mobile operators.

This could be realized as follows:

- use agreed, much more sensitive values for the pair time/distance parameters: suggested is 3 minutes/2 meters; or

- provide a mechanism to download the pair of parameters to be used from the backend server; and
- agree upon the way DP3T/GAEN is used, the algorithm used to identify contacts and the semantics and format of the resulting contact records in the Apps, where the English and Swiss implementations provide examples of practices to be followed concerning the reduction of multiple encounters between the same pair of devices within 24 hours.

5 Bluetooth® LE layer interoperability

5.0 General considerations

There is a need to provide a general approach to support the interoperability of Bluetooth®-based DCTS worldwide. This needs to encompass existing protocols but also future or evolving protocols, rather than providing protocol-to-protocol conversions or adapters for each pair of device-to-device DCTS protocol.

5.1 Layers of operation

5.1.0 End-to-end DCTS exposure notification flow

There are five technology layers of a DCTS:

- Bluetooth® OS layer;
- detection of supported device-to-device protocols;
- exchange of Bluetooth® LE payloads, that it is called "device-to-device payloads";
- decoding and storage of payload data; and
- onward transmission of elements from payloads.

This supports an end-to-end DCTS exposure notification flow such as the one below. Ideally, this would allow multiple states to interoperate across multiple device to device protocols, with mutual exposure notification. This can be seen in Figure 3.

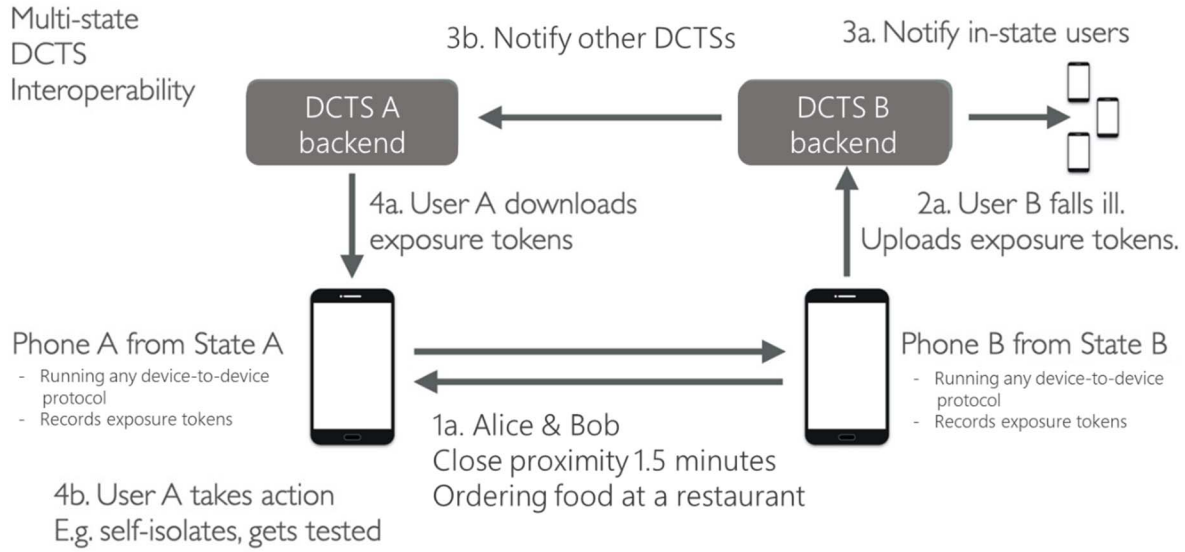


Figure 3: End-to-end DCTS exposure notification flow

This flow involves several steps:

1) Two people (Alice, user A, and Bob, user B) are in close proximity; e.g. in a restaurant. Their apps detect the presence of each other, and perform distance and perhaps risk estimation. They log the contact time date and duration, and the exchanged exposure tokens.

2) User A (Our Traveller) returns to her state. User B falls ill, and uploads his exposure tokens.

3) The DCTS app then determines based on the other devices seen, how to notify those devices:

   a) For contacts with the same state app, these exposure tokens are downloaded and actioned.

   b) For contacts from other states, these exposure tokens are passed on to that state's DCTS app through a pre-agreed mechanism.

4) User A (Our Traveller) downloads the latest exposure tokens, including that from User B's phone. Having reached a particular exposure risk threshold, the user is advised to take action. This may include self-isolation, for example, and getting tested.

Items 1, 2 and 4 are described in clauses 5.1 to 5.3 in the present document. Item 3 is described in clause 5.2.

In the present document, 'exposure tokens' are the parts of the exchanged device-to-device payloads between two phones that are uploaded to a DCTS backend for onward transmission. This clause considers only the DP3T/GAEN and ROBERT systems, leaving wider interoperability discussions until clause 8.

5.1.1 Bluetooth® OS layer

This layer is provided by the mobile phone or wearable operating system. This is the API exposed by the operating system to DCTS application developers.

The standards compliance and runtime behaviour of these OSs provided libraries constrain what a developer can achieve via Bluetooth®. Certain operating system versions also only support certain Bluetooth® LE standard versions and features. This further restricts the protocol functionality choices of developers.

How these layers operate may affect the efficacy of any protocol built on top of them. Android, for example, does not have a 'scan and call-back on discovery' mode, only a 'scan for X time period' mode. This means, potentially, that some devices could be missed if the non-scanning time window is too large. In other OSs, device discovery is impeded by a bug that affects background device discovering another background device. The same restriction does not apply to background devices from other manufacturers.

5.1.2 Detection of supported device-to-device protocols

There is a need to provide a general approach to support one or more ways of detecting a device-to-device payload exchange protocol. This typically involves looking for particular Manufacturer Data Area codes in an advertisement, or particular GATT service and characteristic UUIDs. This provides a means to determine if the device supports device-to-device payload exchange protocols, which protocols and versions they are, and the ability to locate an exposure token or other payload for sharing, thus logging a contact event.

A single device can scan for multiple device-to-device protocols and payloads with relative ease. There are some potential restrictions, however, on advertising multiple device-to-device protocols and payloads for reading. If multiple protocols use advertising, for example, then the sum of their manufacturer data areas should not exceed the Bluetooth® LE PDU size if they are to be in the advertisement data area.

For this reason, it is likely that a DCTS app or wearable-based DCTS will only advertise one or two device-to-device payload exchange protocols themselves, but perhaps be able to locate and interact with a range of other protocols around it. The effect of this is that a single device may only be able to be 'discovered' via two protocols, but may be able to discover devices running many more protocols and protocol versions. This functionality is also useful for forward compatibility.

Moving toward a single protocol for device-to-device protocol discover and exchange that supports multiple device-to-device payloads that shared a common header description format would greatly alleviate interoperability.

This separation of 'interaction protocol' and 'device-to-device payload' will be used in clause 8 of the present document. Many current DCTS protocols specify both the device-to-device protocol and payload. Examples include DP3T/GAEN and ROBERT. Others have separated these out. Examples include Australia's DCTS app that originally used the device-to-device payload exchange protocol created by the Singapore government but their own device payload, and now uses the Herald protocol which specifies a protocol independent on their device payload; see [i.8].

It should also be noted that retrofitting each mobile app so it has basic local interoperability with other individual protocols is a large technical effort and may not be possible for every pair of protocols in each interaction.

Clauses 6 and 7 discuss two-protocol interoperability. These are the DP3T/GAEN and ROBERT protocols which are the protocols currently in use in the EU. A separate approach to multiple payload/protocol interoperability is discussed in clause 8.

5.1.3 Exchange of device payloads

Once the device-to-device protocol(s) supported by a remote Bluetooth® LE device are determined, then those payloads should be retrieved. For advertising-based protocols, this can simply be recognizing and logging the payload's bytes in the advertisement area. For connection-based protocols, this will require a connect, interact, disconnect loop.

Given Bluetooth® chipsets have an upper limit on the number of simultaneous connections they can operate, a connection-based protocol will have to actively manage its connections in order to provide for regular and timely discovery and exchange.

Conversely, advertising-based protocols should ensure their scan and record activities operate regularly enough so that they too log these payloads. A typical Bluetooth® LE device will advertise on three advertising channels 1-5 times per second, whereas some advertising only protocols such as DP3T/GAEN only scan and log these advertisements once every 2+ minutes, even though they are advertised multiple times per second. This means they too can 'miss' nearby devices creating a gap in contact payload data exchanged.

5.1.4 Decoding and storage of payload data

Depending on the operating mode of the app or wearable, the payload data may merely be recorded locally on the phone or processed in some form prior to recording. An advertising only based protocol, such as DP3T/GAEN, may just record the exposure token, for example, whereas a secure connection-based protocol may have to decrypt and verify at least part of the data prior to logging locally (decentralised, hybrid) or onward transmission (centralized).

Size of local storage is an issue. Logging every single raw piece of data in a verbose format such as JSON may lead to using too much storage either locally or for later transmission. The Herald Interoperability Standard, see [i.8], describes a general binary format that can be used to describe and encode multiple data payload formats, both when encrypted for transmission and after decryption. This is further discussed in clause 8.

It should also be noted that storage space for wearables is much reduced, perhaps as low as 512 KB. This may necessitate the routine upload of contact data to community, workplace or operating authority servers for temporary storage when connectivity is available.

5.1.5 Onward transmission of payload data

How and when data is onward transmitted may be a result of regulatory, privacy, or technical considerations. These are particularly acute for low-cost wearable devices or areas without the availability of more expensive devices or internet connections.

For a Bluetooth® LE only wearable (i.e. no 4G/5G support), when a compatible Bluetooth® LE beacon is detected these contacts may be automatically uploaded. This could be routinely during the day, or as someone leaves a place of work. The same issue may also be present in areas in developed countries with poor internet service, such as native communities in reservation areas, or for the homeless. This should be taken into account in the design of a payload sharing system.

The phone logging contacts will have a strategy on which data to upload once they are confirmed as ill, versus which data to store locally. In decentralized systems, for instance, there will be some form of Exposure Token shared with a DCTS app and transmitted to other devices and DCTSs in order to allow the accrual of risk to be calculated on other people's devices. For a large geographic area like the European Union this is challenging. For a future global system of exposure notification there are large issues to be resolved of bilateral data sharing, privacy, security, and proportionality of data sharing, as well as individual consent

Such a large system should be able to target the sharing of exposure tokens in a targeted manner. This does not only help to limit the data protection issues, but also lowers the overall load on the system, and prevents token clashes. As exposure tokens are often only 16 or 32 bytes, on a global scale - even with version 4 UUIDs - there are potential for clashes of tokens. This may lead to people being erroneously notified of exposure in Australia, for example, from a matching token for a different individual in France.

A common mechanism in use today is that of a 'routing header'. That is normally a country code and state code of the DCTS to be notified if the encountered individual has been exposed. Both ROBERT, Singapore's protocol, and the Herald Simple and Secured device payloads use this type of mechanism.

Walking around with a device that is, effectively, revealing nationality has obvious privacy concerns. For this reason, any system which uses such routing codes should encrypt the data being transferred between phones and ensure that the decryption of the routing code and any other identifying information not used purely for local exposure token matching can only be performed by a DCTS backend, and not by individual user devices.

In the EU the predominant system is currently DP3T/GAEN. The EU also has common data protection rules under EU GDPR. In this area it is feasible to federate all exposure tokens so they can be shared with all member states. For interoperability with systems beyond the EU, even those using a shared protocol like GAEN, a different mechanism of determining which country to share data with is required.

State DCTSs may decide to ask those who fall ill which countries they have travelled to recently. The relevant information can then be entered and allow transmission of tokens to the correct countries. This has privacy concerns of course, and any manual linking may fall to the wayside in the midst of an outbreak that requires a large amount of manual work by healthcare professionals already.

An alternative and automated approach is as follows. For payloads that do not log the country/state of the healthcare operator of the encountered devices' app, the current country/state the receiving device is loggein should be presumed and recorded against that data. If upon later upload this is found to be incorrect then those regional/national/supra-national gateways will need to validate tokens as being valid for their own DCTS app, where possible, in order to prevent a large number of exposure tokens from other areas being sent to their DCTS apps, thus increasing the load on the data networks and consumer data plan costs.

For this reason, a common device payload description format that describes payload type and version, country code, and state code for the operating authority of the app as described in Herald is useful to prevent over use of network resources. This is discussed in clause 8.

As for onward transmission format, it is currently the case that a particular app is tied to a specific public health authority's servers. This is necessitated by the need to register with that health service in order to upload and download contact information or receive exposure notifications.

Whilst this may mean that currently each DCTS app has a different set of contact upload services, the standardization of the data logging format and description of its data independent of the payload detected provides an opportunity for future efforts to standardize phone/wearable-to-healthcare system upload, and for inter-healthcare system exchange and interoperability. An example of this is described in the Herald International Interop standard draft; see [i.8].

The main gain in this approach is for international interoperability across borders as humanity returns to a new normal post national lockdowns. There will be a need to exchange notifications between healthcare providers of contacts from these countries that may have been exposed whilst travelling in order to detect, isolate, and prevent onward transmission.

This extended approach is discussed in clause 8. This clause shall only discuss issues as related to current EU systems - DP3T/GAEN and ROBERT.

5.2 Supporting multiple approaches today

Current DCTS protocols each have their own method of interaction (advertising or connection based) and are advertised with their own service IDs. A single application cannot adequately hope to support exposing their identity across all of the latest DCTS protocols available in all countries.

The two protocols in use today - DP3T/GAEN and ROBERT - have well known service IDs and so apps within the EU can be configured to recognize these services.

Until all apps converge to expose their payloads through a standard mechanism (as per clause 8), apps may scan for and detect existing token advertisements but would only advertise their own on the standard DCTS service.

There are three key elements to this type of interoperability:

1) Device-to-device interoperability by sharing a protocol that supports multiple payloads.

2) Describing the payload at each stage of its journey from source device, through exposed device, and between device-to-device payload exchange protocol, to an eventual list of exposure tokens for download. At different stages the same payload may have data decrypted and stripped out or annotated before onward transmission.

3) DCTS to DCTS backend integration allowing for secure exchange of exposure tokens and exposure information.

5.3 Supporting two protocols

Both advertising and connection-based exchange will need to be supported. DP3T/GAEN is advertising-based, whereas ROBERT can support both modes.

Clause 8 goes beyond this to discuss how to support any number of device-to-device exchange protocols via a common service ID and description format. Example data and service information is taken from the Herald Protocol which implements this approach today. This may be reassigned by future formal standards body publications.

5.4 Requirements and recommendations for Bluetooth® LE layer interoperability

5.4.1 Requirements

[IBL-01]: A DCT device shall not exceed the standard Bluetooth® Low Energy advertising PDU size for Bluetooth® Low Energy 4.0. Where multiple protocols need to be supported a single service that supports the transfer of multiple payloads, such as the Herald protocol, could be used to avoid breaching this limit.

[IBL-02]: Any DCT system which uses routing codes (e.g. country and state codes) shall encrypt this information for exchange in order to maximize the privacy of travellers.

[IBL-03]: When a DCT application receives payload data without a routing code in a given country, country and state code of that country, when available, shall be used for that payload.

NOTE: This minimizes the number of manual steps that need to be taken by travellers and contact tracers when a traveller falls ill, and maximizes the efficacy of contact tracing for those exposed to travellers.

5.4.2 Recommendations

A DCT protocol and application should support Bluetooth® devices back to Bluetooth® version 4.0 in 2010. This maximizes the number of devices that can run the DCT application and protects lower income communities.

A DCT protocol should support use on a dedicated low-cost wearable device.

NOTE 1: This helps protect those without expensive smartphones such as the elderly, young children, the homeless, indigenous traditional communities, and the developing world.

Implementors should consider using a protocol that provides encryption for data exchange between two devices in order to prevent relay & replay attacks and the later identification and tracking of ill people for whom exposure notifications have been distributed about.

Implementors should consider local contact risk scoring and the use of minimal device storage rather than logging every single RSSI reading or time of every advertisement for each contact.

NOTE 2: This allows low-cost wearables with restricted storage space to be used for DCT, maximizing the number of individuals that can be protected with such a system.

Implementors should consider moving beyond a single-protocol detection capability in their app, even if they only advertise a single payload format themselves, in order to maximize international interoperability as per clause 8 of the present document.

6 Interoperability between systems with a common design approach

6.1 Challenges of the Interoperability between pandemic contact tracing systems that have a common design approach

A detailed overview of interoperability challenges is presented in clause 4. The main challenges relevant to interoperability of DCTSs that have a common design approach are:

- Challenge IO-C1. Ability of DCTS interoperability system (Federation) to support partial or all-to-all replication among individual DCTS backends; see clause 4.3.1.
- Challenge IO-C2. Ability of a mobile application to subscribe to feeds from several DCTS backends; see clause 6.3.2. Note that this challenge can be relevant in the case partial replication across DCTS backends is used, which is in turn needed for scalability.
- Challenge IO-C3. Ability of the Federation to allow a Roaming User to prove authenticity of its positive diagnosis, provided by a Health Authority in a different jurisdiction to its home Health Authority in a confidential, secure and verifiable manner; see clause 4.1.1.4 and requirement HL-IO-04 in clause 5.11 of ETSI GS E4P 003 [1]. By requirement HL-IO-04, Roaming User's mobile application shall be allowed to upload relevant contact tracing data to its home Contact Tracing System backend.
- Challenge IO-C4. Automated certificate management among DCTS backends. All participants in the Federation (backends and possibly backend gateways) need to use secure and trusted communication mechanisms, e.g. using TLS, as well as digitally signing TLS feeds. These require public key infrastructure in place.

6.2 Interoperability between ROBERT systems

Clause 8 of [i.5] describes a partial replication solution for ROBERT systems which exploits the information contained in the ECC field.

The payload exchanged by users of the ROBERT DCTSs includes an 8-bit field (Encrypted Country Code, ECC), which encrypts a Country Code (CC), which is an 8-bit code that uniquely identifies a country in a group of federated backend servers. The ECC fields are generated by the backend servers for federation purposes, and can only be decrypted by federated backend servers; see clause 4 of [i.5]. The encryption/decryption process uses a Federation Key, K_G, which is an L-bit long key, with L larger or equal to 128 bits, shared between all servers of a federation; see clause 3 of [i.5].

When a user of the ROBERT DCTS system notifies a positive test, it uploads a *LocalProximityList* which contains the received payloads of HELLO messages broadcast by other users, together with timing information associated with each HELLO message; see clause 6.1 of [i.5]. The backend server decrypts the ECC fields of the received HELLO payloads in the *LocalProximityList*, to recover the Country Code (CC). If the CC is different from the server's country code and corresponds to a valid country code, the HELLO payload and the timing information is forwarded to the backend server in a federation that manages the corresponding country code messages, where the information is processed.

The use of ECC in ROBERT nicely solves challenges IO-C1 and IO-C2. Indeed, HELLO payloads and timing information can be forwarded to the backend server of the country of interest, where it will be processed as other HELLO payloads. The notions of partial and all-to-all do not apply here, being motivated by the absence of country information in the DP3T/GAEN solution.

Challenges IO-C3 and IO-C4 are not related to the DCT protocol but are generic public health policy issues. From this point of view, ROBERT can accommodate any solution to be designed.

6.3 Interoperability between DP3T/GAEN systems

6.3.1 Addressing Challenge IO-C1

As depicted in Figure 1, one critical task of the Federation is to replicate diagnosis keys across multiple DP3T/GAEN backends to enable notification of a Local about a Diagnosed Traveller. As discussed in clause 4.3.1, this can be in principle done in two ways:

- *Partial replication*, on a need-to-know basis. In this case, a Traveller shares with its home DCTS backend coarse-grained information about their travel patterns. In our example of Figure 1, Bob whose home DCTS backend is in Country B, upon Bob's positive diagnosis, shares with DCTS backend B that he travelled to Country A. This in turns allows Country B backend to transfer Bob's Diagnosis Keys to country A DCTS backend, which allows Alice to be informed.

- *All-to-all replication*, across a *cluster of* affiliated countries (e.g. EU countries). In this case, Bob would not need to upload his travel information to country B backend, but all data would be replicated across all backends belonging to a cluster.

As it is seen, addressing Challenge IO-C1, involves allowing GAEN/DP3T backends to exchange relevant Diagnosis Keys among themselves. In principle, this can be done in the following two ways, which can also be used in conjunction:

- *Peer-to-peer approach* in which decentralized pairwise point-to-point communication between DCTS backends is established. With the peer-to-peer approach, two DCTS backends periodically communicate between themselves and exchange relevant Diagnosis Keys. This approach has been proposed in literature which favors decentralized communication mechanisms; see [i.6] and [i.10]. Peer-to-peer approach can implement both partial and all-to-all replication; see Figure 4.

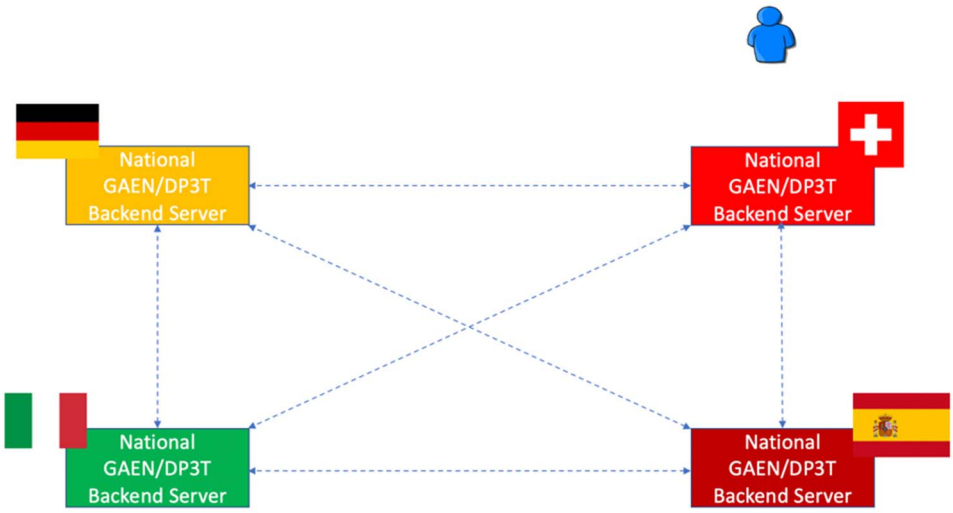


Figure 4: Peer to peer communication among GAEN/DP3T backend servers

- *Gateway approach*, in which a central trusted Federation Gateway Service (FGS) facilitates communication among individual DCTS backends. This approach is typically expected to be used to interconnect DCTS backends across jurisdictions which have close political relations, the example being European FGS (EFGS), deployed among EU country members which use the GAEN/DP3T system; see [i.11]. Like the peer-to-peer approach, the gateway approach based on an FGS can also implement both partial and all-to-all replication. The gateway approach is depicted in Figure 5. In principle, gateway approach can be combined with the decentralized approach.

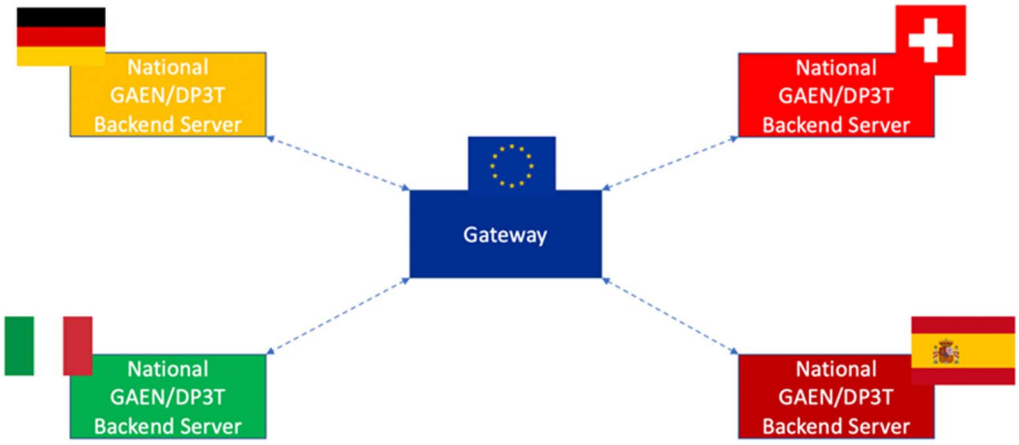


Figure 5: Gateway communication approach among GAEN/DP3T backend servers

In summary, both peer-to-peer and gateway approaches can implement both all-to-all and partial replication and can be chosen as architectures for solving Challenge IO-C1. The gateway approach is intuitively easier to deploy and less complex than the peer-to-peer approach [i.11]. However, the peer-to-peer approach seems more robust and potentially more appealing for geographical reasons (a central gateway covering all countries members in the world, e.g. members of United Nations, is conceivable but potentially challenging to deploy).

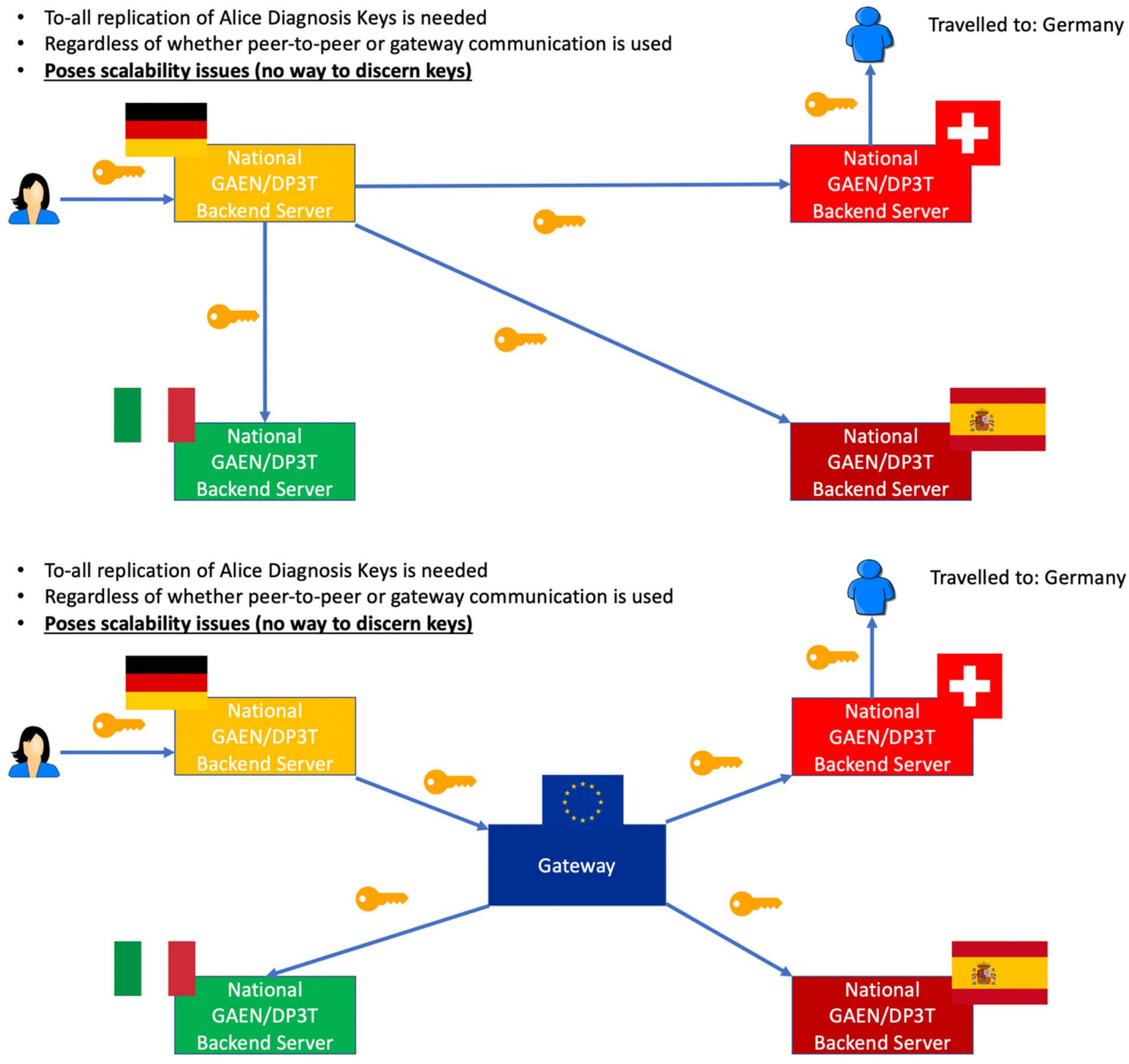
Other challenges (i.e. IO-C2, IO-C3, IO-C4) remain relevant regardless of the choice of the peer-to-peer vs gateway approach to DCTS backend communication. In the following, these challenges are reviewed in more details, while describing different approaches pertaining to peer-to-peer/gateway backend communication.

6.3.2 Addressing Challenge IO-C2

Challenge IO-C2 is tightly coupled with the need for the Federation to inform a Traveller of a positive diagnosis of a Local in case partial replication is used; see clause 4.3.2 and Figure 2.

Recall that this interoperability use case involves a Traveller Bob, which returns to his home country B after being in contact with Alice in country A, who later was diagnosed as positive. Unless Bob is able to listen to Diagnosis Keys feeds of country A, and if Bob can only listen to his home country B DCTS backend feed, then country B feed needs to fetch all Diagnosis Keys from Country A.

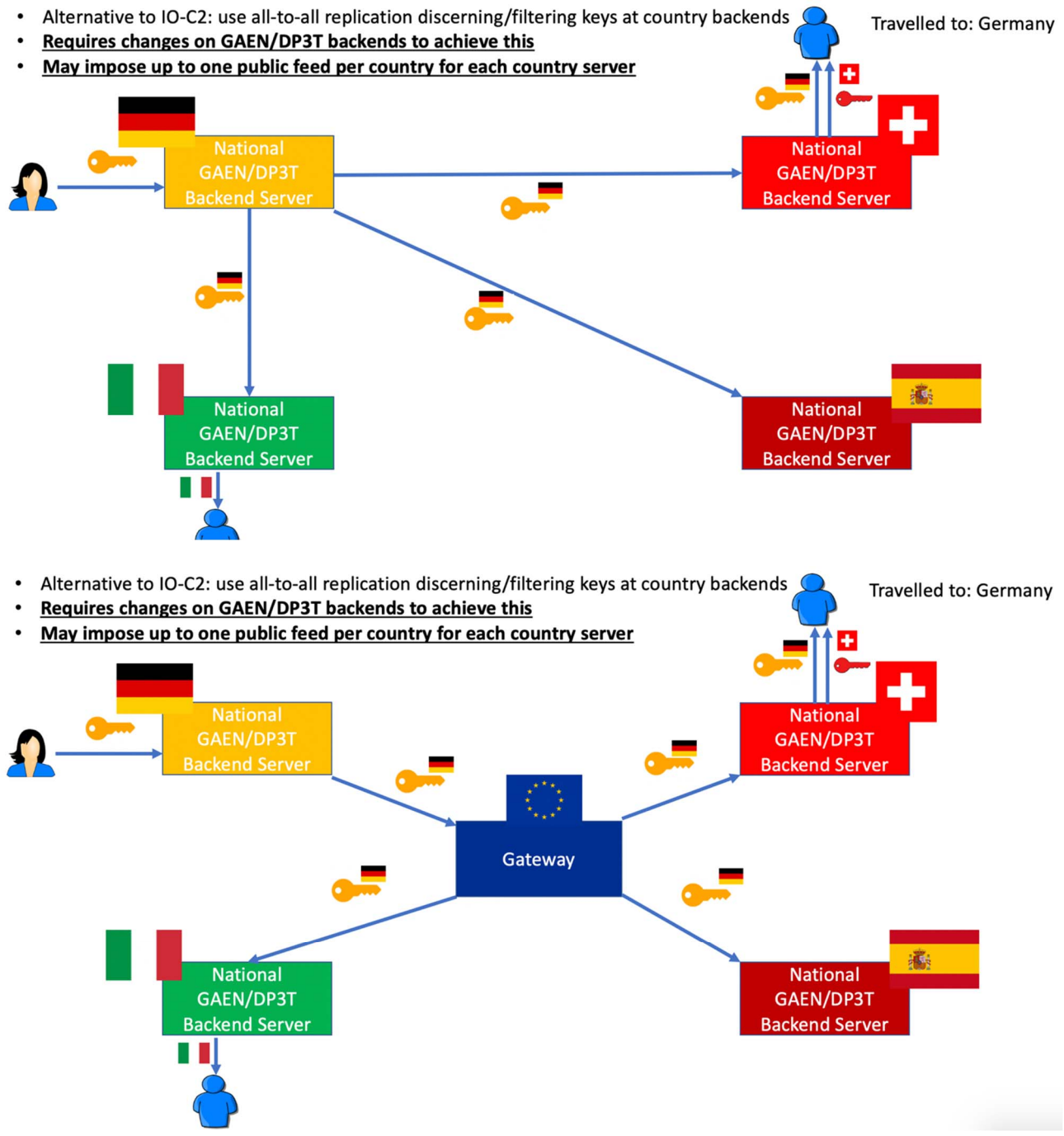
In case all-to-all replication is used among all countries in the Federation, this is not an issue. However, all-to-all replication is jeopardizing scalability; see ETSI GS E4P 008 [3]. All-to-all replication in this scenario, for both peer-to-peer and gateway approaches is illustrated in Figure 6.



NOTE: In case Bob, who travelled to Germany, listens only to the local Swiss Confederation backend, without providing futher information about his travel patterns to the Swiss Confederation backend, all-to-all replication is needed. In GAEN/DP3T Alice's device and her Diagnosis keys have no information about where Bob comes from.

Figure 6: All-to-all replication scenario, for both peer-to-peer and gateway approaches

It is possible for backend B to fetch all Diagnosis Keys from country A only when requested by user Bob (on-demand). However, this would violate the security and privacy requirements of a GAEN/DP3T solution in which a non-infected user never reveals its travel patterns to its backend, unless the User gets diagnosed/infected. To cope with scalability challenges, this approach also requires the local backend to discern different feeds of Diagnosis Keys that come from different countries. If this is not done and Diagnosis Keys are mixed into a single public feed, then scalability may also be jeopardized. This alternative approach is depicted in Figure 7.



NOTE: Here Bob, informs the Swiss Confederation server he is interested in both local Diagnosis Keys and German diagnosis keys. A user from Italy, Carlo, who did not travel, continues to receive only Italian Diagnosis Keys.

Figure 7: Alternative to IO-C2 that consists of combining to-all replication of Diagnosis Keys amongst GAEN/DP3T server backends, while tagging and discerning Diagnosis Keys feeds from different countries

For these reasons, the optimum solution in this case is to allow Bob's Mobile Application to simply listen (subscribe) to country A feeds for a certain number of days according to epidemiological parameters and then to allow Bob to simply unsubscribe from country A's feeds after a certain number of days following Bob's departure from country A. This is depicted in Figure 8.

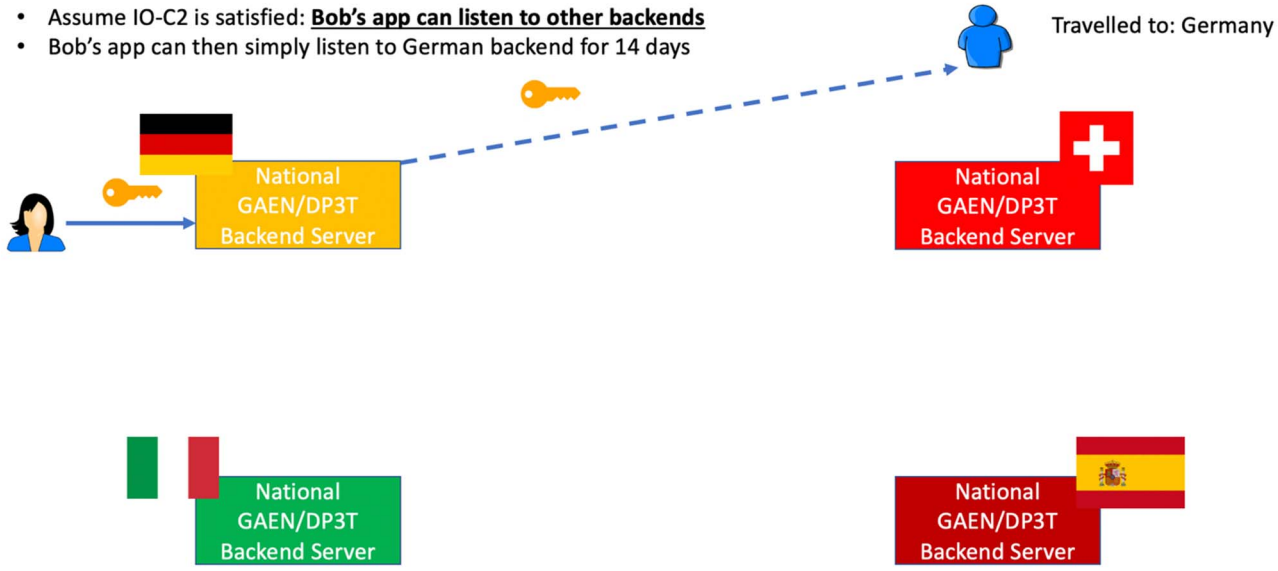


Figure 8: Interoperability in the Notification to a Traveller about infected Local use case, when Mobile Applications are allowed to listen to other backends

The Challenge IO-C2 is not addressed automatically by gateway backend communication systems such as Federation Gateway Systems (FGSs), unless an FGS employs continuous all-to-all replication across all backends. However, an FGS can be leveraged to assist Traveller's mobile application to identify and connect to a DCTS backend in a country the Traveller visits. This is depicted in Figure 9.

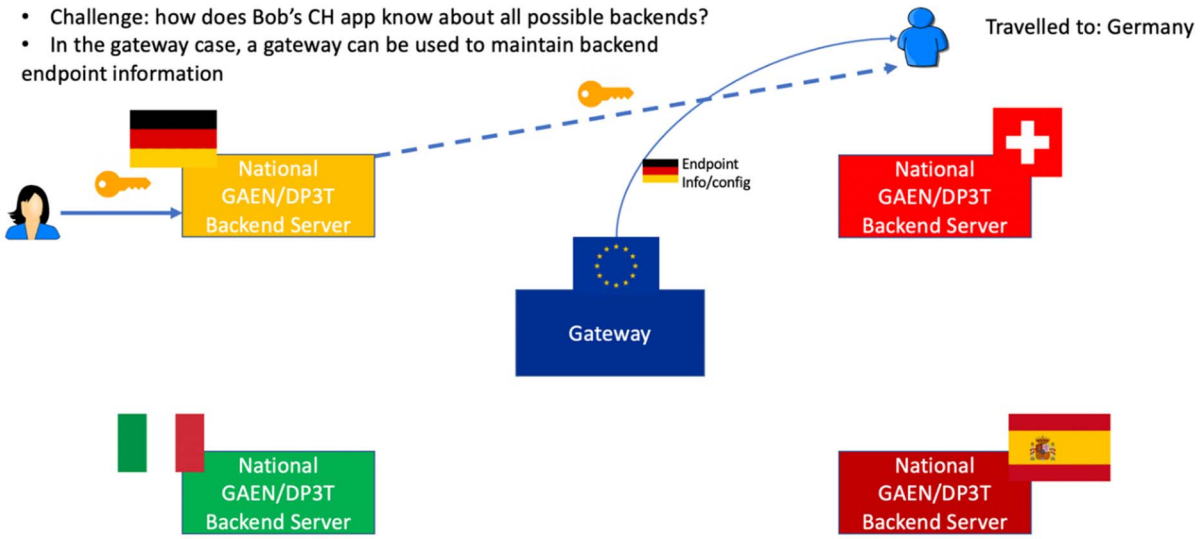
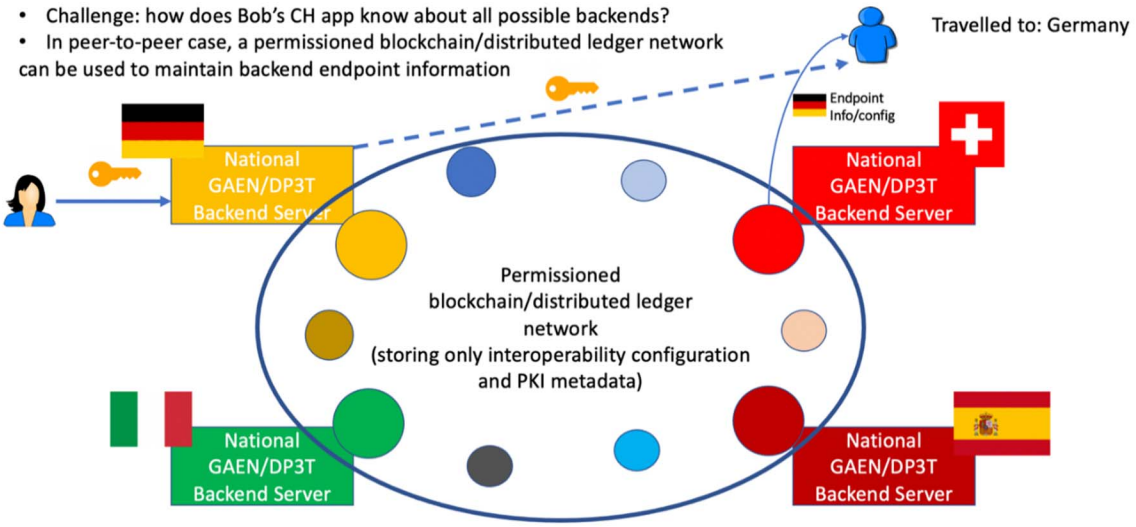


Figure 9: Federation Gateway Systems (FGS) based solution, the Gateway can assist users in finding other countries backends

In the case of peer-to-peer backend communication, Travellers should be able to securely identify and connect to GAEN/DP3T DCTS backends in the countries to where they travel. To this end, Travellers could use a decentralized permissioned block-chain network on which the identities, public keys and certificates of individual DCTS networks would be published. This approach is depicted in Figure 10.



NOTE: The consistent information about configuration of all backends which participate in the Federation needs to be fetched from repository. This can be achieved using a permissioned block-chain/distributed ledger network, which would hold only public information such as interoperability configuration and Public Key Infrastructure (PKI) metadata.

Figure 10: Peer-to-peer backend communication approach

As it is seen next, such a permissioned block-chain/distributed ledger network will also be useful in addressing challenges IO-C3 and IO-C4.

6.3.3 Addressing Challenge IO-C3

Challenge IO-C3: A Roaming User shall be able to prove authenticity of its positive diagnosis, provided by a Health Authority in a different jurisdiction to its home Health Authority in a confidential, secure and verifiable manner; see clause 4.1.1.4 and requirement HL-IO-04 in ETSI GS E4P 003 [1].

Recall that Challenge IO-C3 relates to a Traveller (Bob) who gets diagnosed by a Health Authority in another jurisdiction (Country T), other than his home Health Authority in country B.

In this case, to enable interoperability, Bob's Mobile Application shall be allowed to upload Diagnosis Keys to its home country B DCTS backend as if Bob was diagnosed by the Health Authority of country B. In GAEN/DP3T protocol however, the upload of Diagnosis Keys to a DCTS backend of country B is subject to Bob's Mobile Application obtaining Authorization Code from Health Authority of country B beforehand.

In this case it is necessary for Bob's Mobile Application to present the positive diagnosis results (Positive Test), performed by Health Authority of country T (e.g. Germany), to its home Health Authority in country B (e.g. Switzerland). This challenge is depicted in Figure 11.

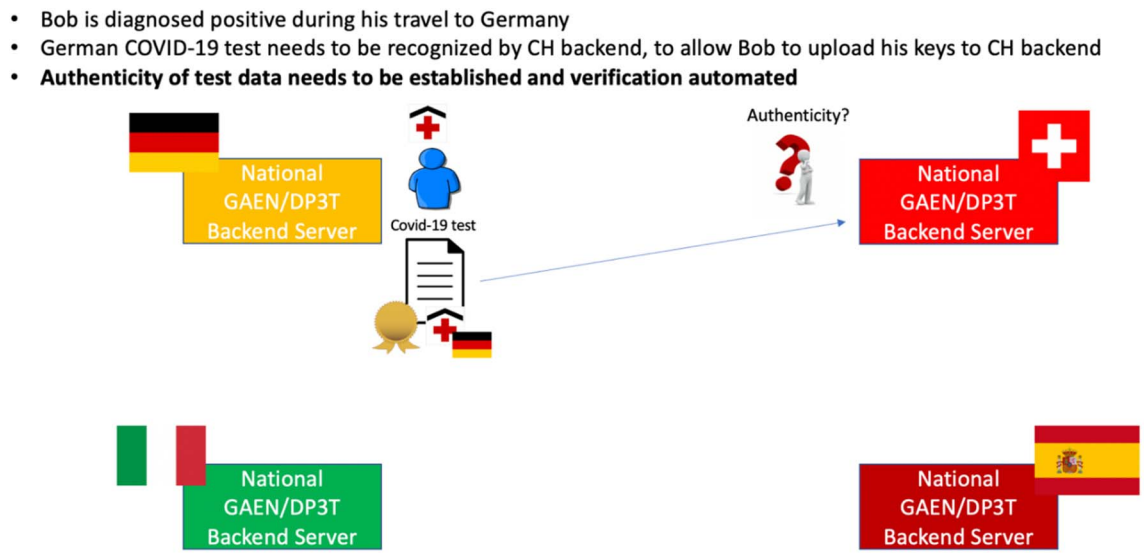


Figure 11: Scenario of a Roaming User

A roaming user shall be able to prove authenticity of his/her positive diagnosis, provided by a Health Authority in a different jurisdiction to his/her home Health Authority. In case of a Diagnosis of a Traveller in a foreign country, the interoperability system of a Federation shall satisfy requirements of HL-IO-04 and HL-IO-05.

To this end, it is necessary to establish an adequate certificate (public key) and trust infrastructure (e.g. based on Verifiable Claims) in which the Health Authority of country B (the Verifier) would be able to accept the test performed by the Health Authority of country T (Issuer) in a confidential, secure and verifiable manner. In other words, governance and lifecycle of Health Authority Certificates should be put in place. A logically centralized (in practice implemented as actually centralized or decentralized) repository of Health Authority Certificates is needed as depicted in Figure 12.

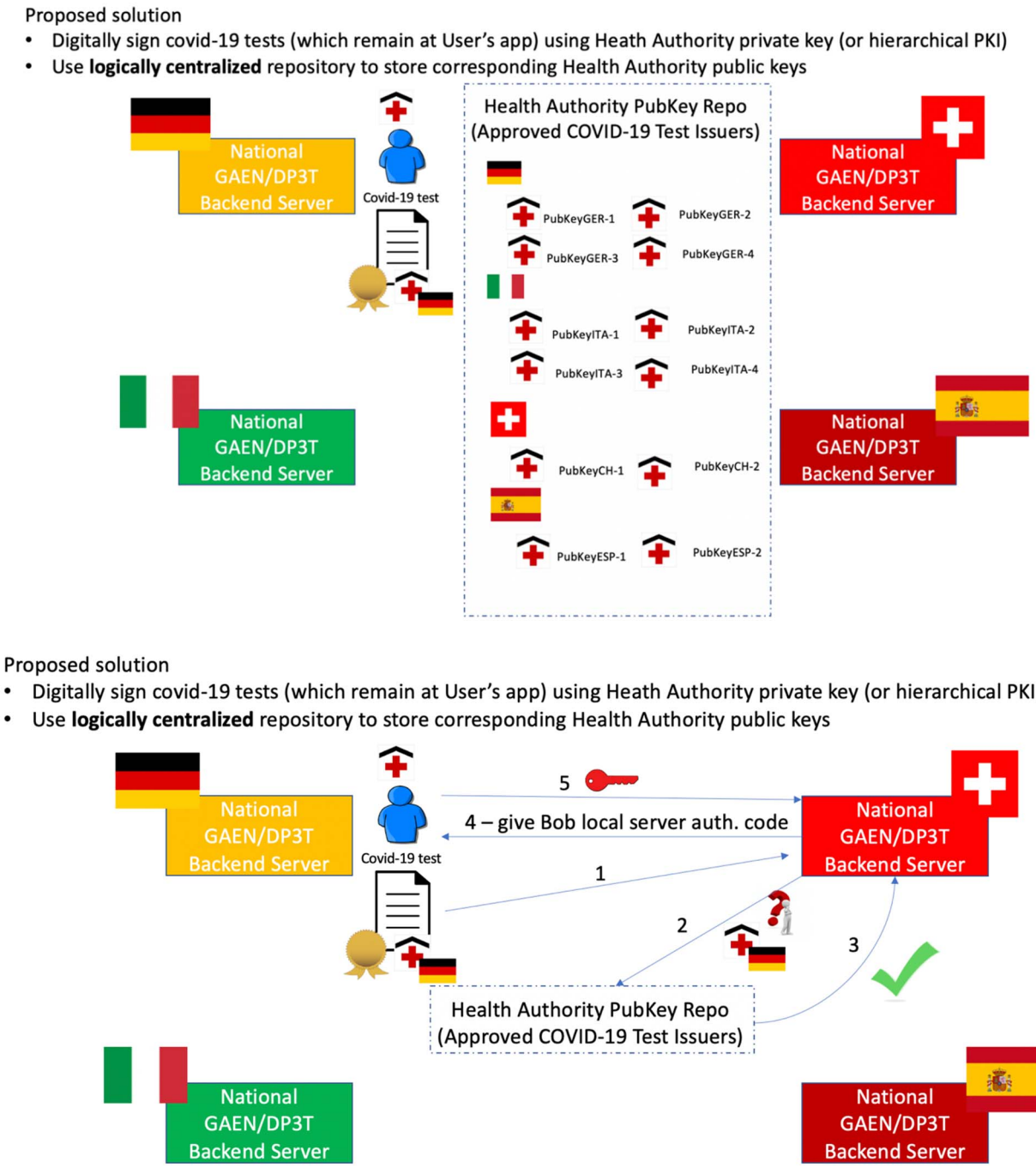
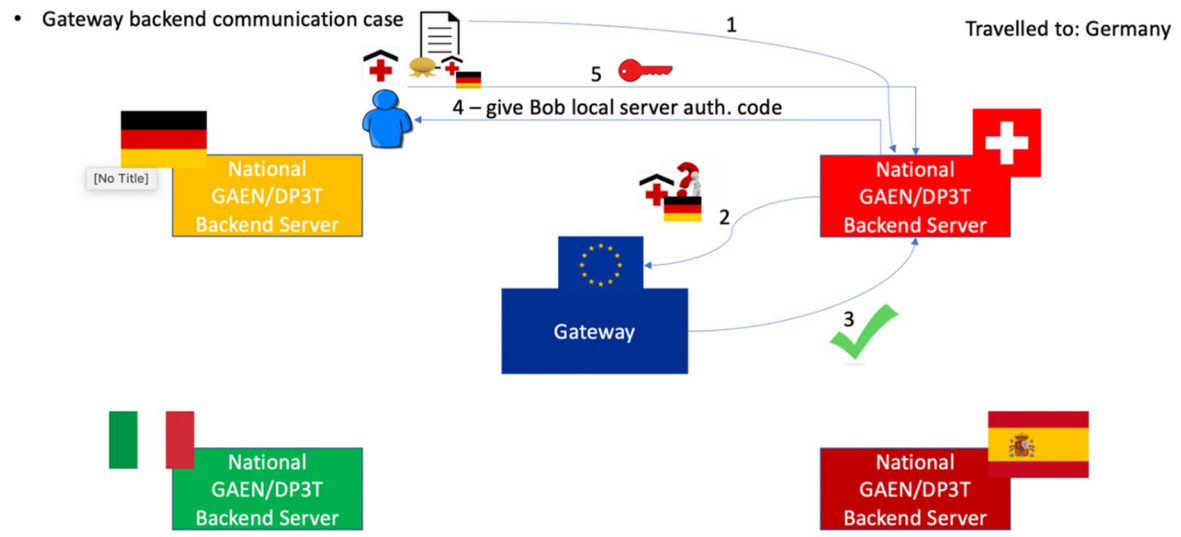


Figure 12: Logically centralized repository of Health Authority Certificates implemented as actually centralized or decentralized

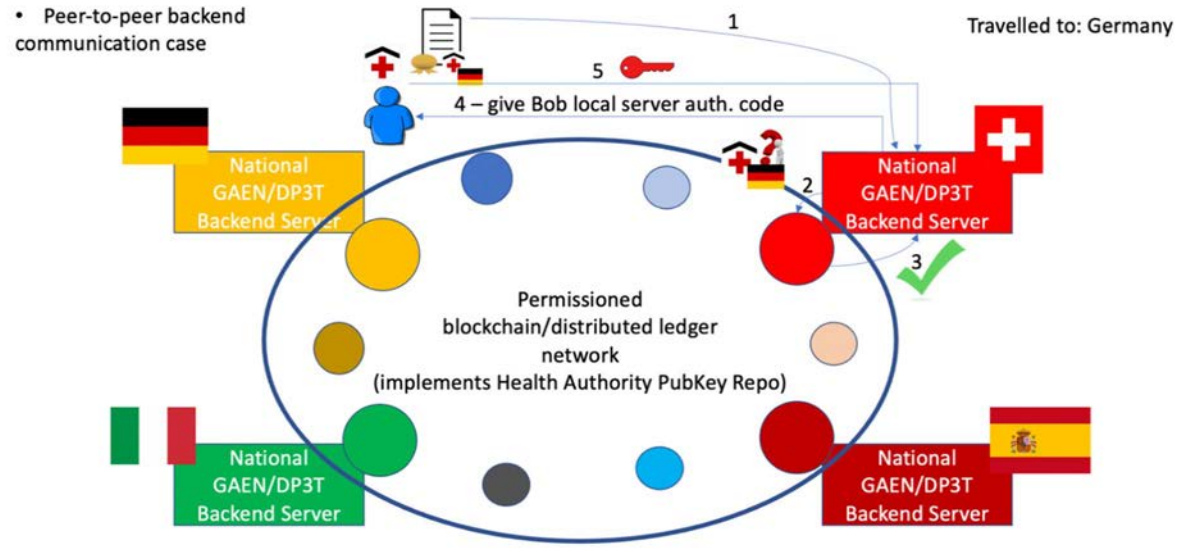
In case a gateway is used, an FGS can play the role of the trusted repository of public keys pertaining to issuers and play a trusted role in Health Authority Certificate governance and lifecycle. This is depicted in Figure 13. Currently, the proposal of European FGS neither provides nor foresees this functionality, see [i.11].



NOTE: Solution based on a FGS as a trusted repository of public keys pertaining to issuers and trusted role in Health Authority Certificate governance and lifecycle.

Figure 13: FGS as a trusted repository of public keys

In the case of decentralized peer-to-peer backend communication, a decentralized infrastructure for Verifiable Claims needs to be setup to address Health Authority Certificate governance and lifecycle. This can be done via dedicated permissioned block-chain-based system in which the role of trusted repository of public keys pertaining to Issuers would be decentralized and store on a block-chain. This is depicted in Figure 14.



NOTE: Solution based on a dedicated permissioned block-chain-based system in which the role of trusted repository of public keys pertaining to Issuers is decentralized and stored on a block-chain.

Figure 14: Permissioned block-chain-based system

In this case, the decentralized block-chain network would comprise Health Authorities of different jurisdictions. The block-chain network would store only Issuer public key certificates. Test results would only be stored at User's Mobile Applications and, potentially retained by Issuers and in any case would not be stored on the block-chain.

Note that this decentralized infrastructure could be reused in use cases far more general than DCTSs. For instance, it could be used to facilitate international travel, access to large events and so on, while fully respecting privacy.

6.3.4 Addressing Challenge IO-C4

Beyond Health Authority Certificate governance and lifecycle systems, the key to interoperability is a security architecture and interoperability certificate governance for the DCTS backend to the backend communication. This, DCTS Backend Certificate Governance and Lifecycle (DCTS BCGL) shall be put in place to allow different DCTS backends to securely communicate amongst each other.

In the case of centralized Federation Gateway Services, such as European FGS (EFGS), the certificate management pertaining to the DCTS backends can be done by an FGS. The document elaborated by the eHealth Network describing this functionality is available and is being put in place for European GAEN/DP3T DCTS backend interoperability; see [i.12]. In a nutshell, the DCTS BCGL function of EFGS is foreseen to manage TLS and public feed certificates pertaining to individual DCTS backends.

In the case of decentralized peer-to-peer DCTS backend communication, a similar functionality of DCTS BCGL shall be put in place, similar to EFGS DCTS BCGL functionality, albeit decentralized. Again, a promising technology for this purpose is a decentralized permissioned block-chain network which would store TLS and public feed certificates pertaining to individual DCTS backends, as well as other metadata required for secure communication among backends.

Regardless of the choice of a decentralized vs a centralized approach, or a hybrid combination thereof, see clause 6.3.5, BCGL should follow established recommendations and procedures such as European Interoperability Certificate Governance [i.12] to enable secure and trusted communication among DCTS backends.

6.3.5 Hybrid approach to interoperability mixing gateway and peer-to-peer approaches

Both the peer-to-peer approach and the gateway approach can address all interoperability challenges. While European FGS follows the gateway approach, extending this approach to a global, worldwide level may involve political and other challenges. Therefore, the DCTS interoperability Federation should support a hybrid approach, allowing for both gateway and peer-to-peer approaches. This is depicted in Figure 15.

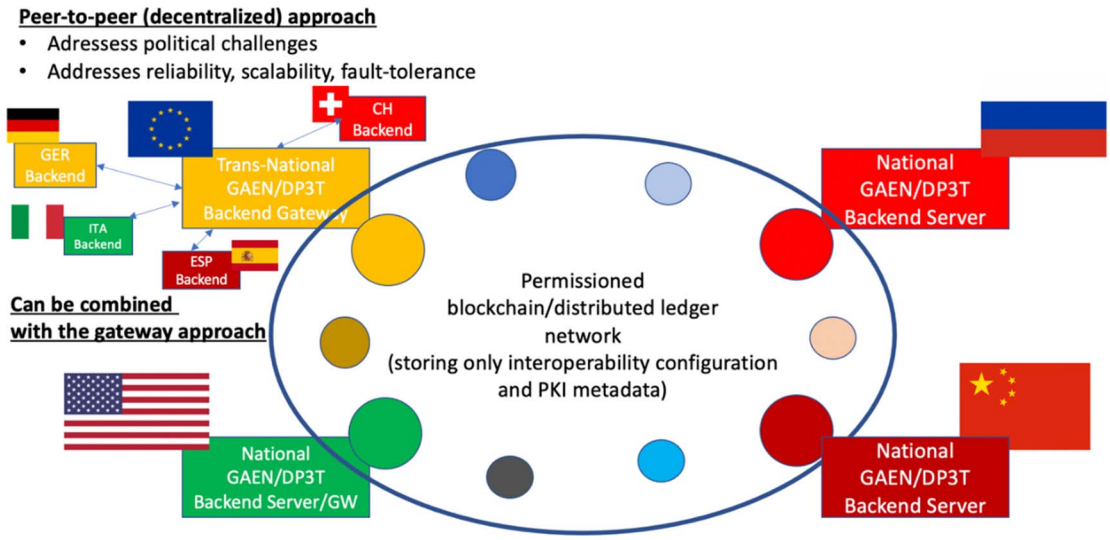


Figure 15: Hybrid approach to DCTS Federation interoperability combining peer-to-peer and decentralized approaches

6.4 Requirements for interoperability between systems with a common design approach

6.4.1 Requirements for interoperability between ROBERT systems

[IRI-01]: The ROBERT backend servers that are federated shall use an 8-bit field Country Code that uniquely identifies the geographical area (e.g. country, group of countries, region, or state).

[IRI-02]: The ROBERT backend servers that are federated shall share and use a common secret Federation Key, as described in clause 6.2 of the present document.

[IRI-03]: When a ROBERT backend server receives a LocalProximityList uploaded by one of its users, it shall decrypt the ECC fields for each entry of the list.

[IRI-04]: When a ROBERT backend server decrypts an ECC field and the CC corresponds to the geographical area (e.g. country, group of countries, region, or state) where another federated backend server is deployed, it shall securely forward the corresponding entry to the other federated backend server.

6.4.2 Requirements for interoperability between DP3T/GAEN systems

[IDI-01]: Federated DP3T/GAEN backends shall exchange relevant diagnosis keys, either using peer-to-peer, gateway or hybrid approaches, as described in clause 6.3.1 of the present document.

[IDI-02]: Federated DP3T/GAEN backends shall implement the exchange of relevant diagnosis keys using partial replication or all-to-all replication, as described in clause 6.3.1 of the present document:

- When the replication of diagnosis keys across multiple DP3T/GAEN backends is done by using partial replication, each DP3T/GAEN backend shall have coarse-grained information about travel patterns of diagnosed users.

NOTE: The DP3T/GAEN backends do not need to have information about travel patterns of diagnosed users in case of all-to-all replication.

[IDI-03]: When the federated DP3T/GAEN backends use partial replication, travellers shall be able to subscribe to the visited foreign federated DP3T/GAEN backends to receive diagnosis keys:

- When the system uses partial replication based on a gateway approach, the system should assist the traveller's mobile application to identify and connect to the corresponding DP3T/GAEN backend.
- When the system uses partial replication based on a peer-to-peer approach, the system could use a permissioned block-chain/distributed ledger network, which should hold only public information such as interoperability configuration and Public Key Infrastructure (PKI) metadata.
- When the federated DP3T/GAEN backends use all-to-all replication, the DP3T/GAEN backend shall deliver the required diagnosis keys upon request to their users.

[IDI-04]: If a user gets diagnosed in a roaming country, the user's mobile application shall be allowed to upload diagnosis keys to their home country DP3T/GAEN backend as if the user was diagnosed in his/her home country.

[IDI-05]: A roaming user shall be able to prove authenticity of his/her positive diagnosis delivered in a roaming country. To this end, governance and lifecycle of diagnosis certificates between countries should be put in place. The system should have a logically centralized repository of diagnosis certificates, which may be implemented using a decentralized (peer-to-peer), gateway or hybrid approach.

[IDI-06]: DP3T/GAEN Backend Certificate Governance and Lifecycle shall be put in place to allow different DP3T/GAEN backends to securely communicate amongst each other. This Backend Certificate Governance and Lifecycle should follow established recommendations and procedures such as European Interoperability Certificate Governance to enable secure and trusted communication among DP3T/GAEN backends.

[IDI-07]: To support regional interoperability, the DP3T/GAEN backends federation shall support at least one of the peer-to-peer or gateway approaches, and should support both to facilitate worldwide DP3T/GAEN interoperability.

7 Interoperability between systems with a different design approach

7.1 Challenges of the Interoperability between pandemic contact tracing systems that have a different design approaches

7.1.0 General considerations

Achieving interoperability between systems that have different design approaches is a difficult task, as the principles of operation and privacy properties are very different, meaning that an interoperability solution can lead to the situation in which users of one system suffer privacy vulnerabilities inherited from the other system. This situation can arise if the backend servers from ROBERT and DP3T/GAEN systems are directly connected.

In this clause, it is assumed that Alice and Bob have been in contact. Alice uses a DP3T/GAEN system, while Bob uses a ROBERT system. Both systems are federated, meaning that they can exchange information, such as lists of received or transmitted Bluetooth® LE payloads or encryption keys.

Two options for direct interoperation solutions are presented, discussing the inherited vulnerabilities that appear in these two cases.

7.1.1 Case A: DP3T/GAEN users log HELLO packets broadcast by ROBERT users

7.1.1.0 Assumptions

It is assumed that Alice stores a log with the payloads and timing information of the advertisement packets broadcast by ROBERT users (EBIDs); see clause 5.2 of [i.5], together with the timing information, exposure measurement, and ephemeral IDs broadcast by DP3T/GAEN users (EphIDs), see clause 2.1 of [i.2]. During her contact with Bob, Alice's device stores the payloads and timing information of the advertisement packets broadcast by Bob's device. Bob's device follows the usual operation of ROBERT DCTSs with the additional feature that it would notify the visited countries of his backend server in case of a positive test.

7.1.1.1 Case A1: A DP3T/GAEN user receives a positive test

If Alice receives a positive test, her device will send to its backend server the log of stored information during her contact with Bob (i.e. the same information as the one stored in the *LocalProximityList* used in ROBERT DCTS; see [i.5]). Alice's backend server is assumed to be federated with Bob's backend server, meaning that the server has access to the key K_G required to decrypt the ECC of Bob's EBIDs; see clause 6.2. Once Alice's backend server identifies Bob's Country Code, it will relay to Bob's backendserver the received information. Bob's backend server will use this information to evaluate his at-exposure risk, and Bob would be eventually notified of this condition, see Figure 16.

More specifically, the steps to follow would be:

1. Bob's device broadcasts EBIDs using Bluetooth® LE transmission.
2. Alice's device logs payloads and timing information of the EBIDs broadcast by Bob's device.
3. Alice receives a positive test and transfers her EphIDs to the DP3T/GAEN backend server.
4. Alice also transfers her log of Bob's information (i.e. a LocalProximityList) to the DP3T/GAEN backend server.
5. Alice's backend server decyphers the ECC using the key K_G it has as it is federated with Bob's ROBERT backendserver; see clause 6.2. It transfers Bob's EBIDs to Bob's backend server.
6. Bob's backend server decyphers the EBIDs and obtains Bob's ID.

7. Bob's backend server re-evaluates the risk scoring for Bob.
8. Bob is notified in case his risk is high.

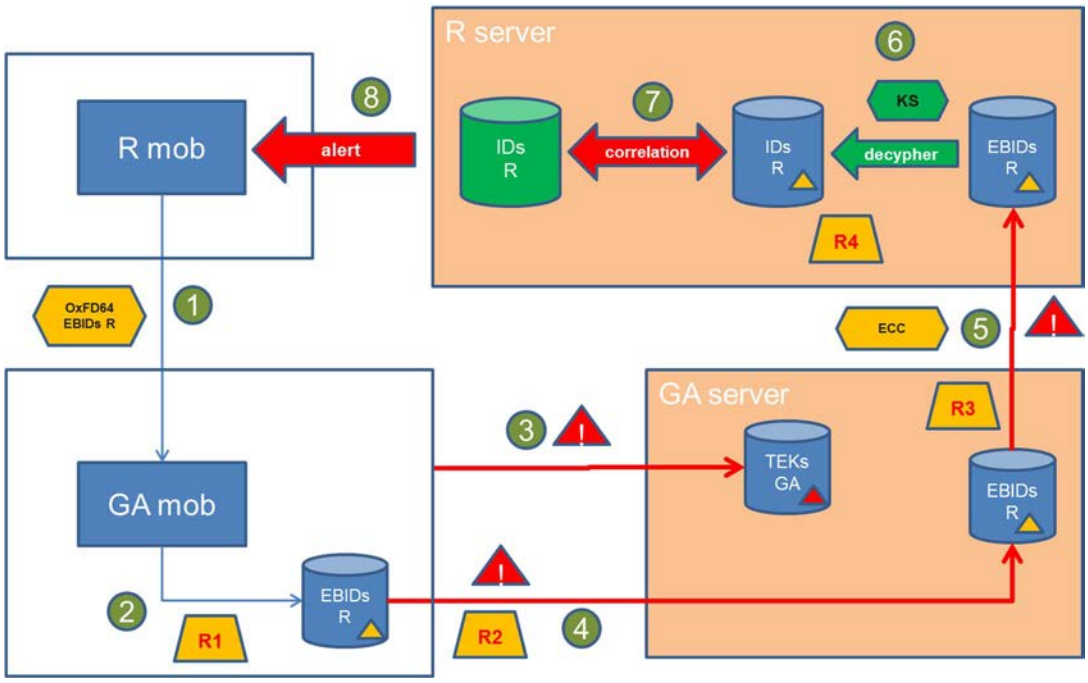


Figure 16: Case A1: A DP3T/GAEN user receives a positive test

7.1.1.2 Case A2: A ROBERT user receives a positive test

It assumed, as in the previous clause, that Alice's device stores a log with the payloads and timing information of the EBIDs broadcast by the ROBERT users, together with the timing information, exposure measurement, and ephemeral IDs broadcast by DP3T/GAEN users.

If Bob receives now the positive test, Bob's device will transfer a LocalProximityList to his backend server. The ROBERT backend server processes the information contained in the LocalProximityList as in the standard operation of ROBERT DCTSs. In addition, it would now relate Bob's information with his ID, and re-generate a list of EBIDs that has been used by Bob. Bob's backend server would now relay this information to Alice's backend server. The situation is here similar to the case of interoperability of DP3T/GAEN systems: Bob's backend server would require a list of countries visited by Bob to relay this information, and Alice backend server or device need to know whether it would fetch information coming from Bob's backend server. Alice's backend server would relay this information, meaning that Alice's device could evaluate her risk of infection using Bob's stored EBIDs; see Figure 17.

More specifically, the steps to follow would be:

1. Bob's device broadcasts EBIDs using Bluetooth® LE transmission.
2. Alice's device logs payloads and timing information of the EBIDs broadcast by Bob's device.
3. Bob's receives a positive test and Bob's device transfers the LocalProximityList to his backend server.
4. Bob's backend server obtains Bob's ID and re-generates the list of EBIDs used by Bob.
5. Bob's backend server determines to which backend servers it would relay this information. This information reaches Alice's backend server.
6. Alice's backend server distributes Bob's EBIDs to all the users of this system.
7. Alice's device receives the information and runs her risk scoring algortithm to determine whether she is at-risk.

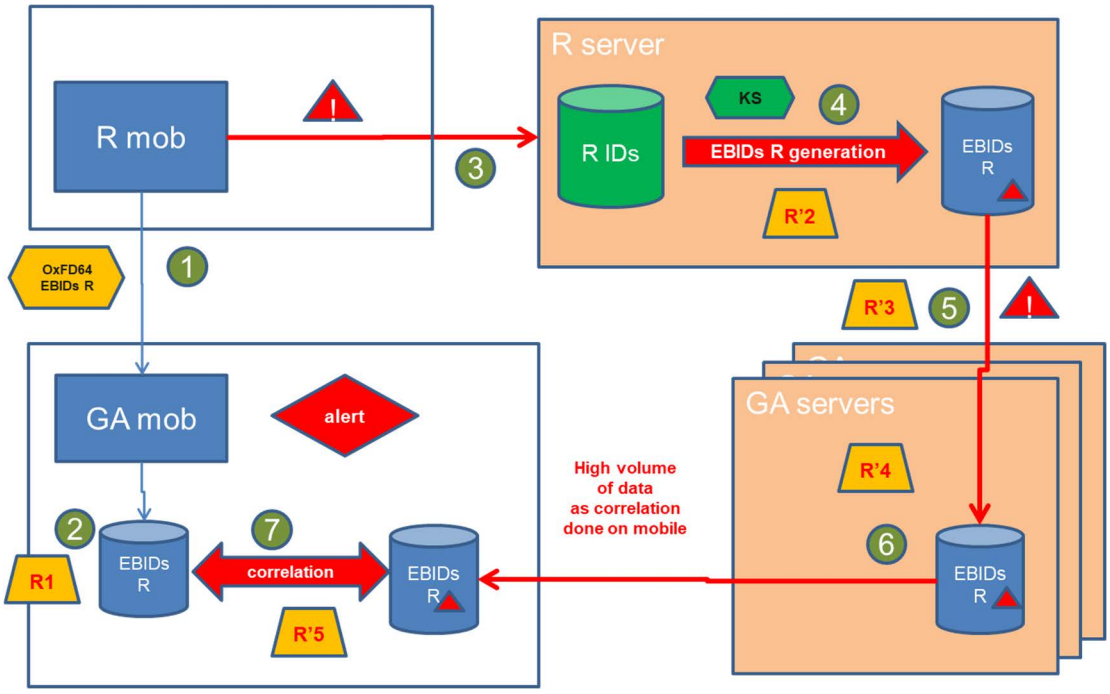


Figure 17: Case A2: A ROBERT user receives a positive test

7.1.1.3 Privacy risk for this interoperability scheme

A complete analysis of the privacy risks presented in this solution is outside the scope of the present document. However, it can be seen that in the case A2 a privacy risk for the users of the ROBERT system which was not present in the case of no interoperability: As the EBIDs of infected users are distributed to all users of the DP3T/GAEN system, an attacker has now the possibility of identification of infected ROBERT users, a risk vulnerability that is inherent to the DP3T/GAEN system, but which is not present in the standalone version of ROBERT; see clause 7 of ETSI GS E4P 008 [3]. Moreover, Bob uploads a list of visited countries, which is not necessary in the case of ROBERT DCTS with no interoperability.

7.1.2 Case B: ROBERT users log information broadcast by DP3T/GAEN users

7.1.2.0 Assumptions

It is assumed now that Bob stores a log with the ephemeral identifiers broadcast by DP3T/GAEN users (EphIDs), with timing information and strength of the receive signal. Bob also stores the EBIDs broadcast by ROBERT users as it is done in a normal ROBERT DCTS operation. During his contact with Alice, Bob's device would store some of the EphIDs broadcast by Alice's device. Bob's device will upload periodically the list of received EphID's to his backend server, together with timing information and strength of the received signal.

NOTE: This is done even if Bob does not receive a positive test. In addition, Bob includes a list of visited countries.

7.1.2.1 Case B1: A DP3T/GAEN user receives a positive test

If Alice receives a positive test, her device will send to its backend server the log of broadcast EphIDs. Alice's backend server, will relay these EphIDs to the corresponding ROBERT backend server. This implies either that Alice's adds a list of countries of interest, or that the ROBERT user somehow registers to receive this information and evaluate Bob's risk; see Figure 18. Once Bob's backend server receives this information, it compares this with the list of EphIDs uploaded by Bob, and evaluates whether he is at risk.

More specifically, the steps to follow would be:

1. Alice's device broadcasts EphIDs using Bluetooth® LE transmission.

2. Bob's device logs Alice's EphIDs.
3. Bob's device transfer Alice's logs to his backend server.
4. Alice receives a positive test and transfers her EphIDs to the DP3T/GAEN backend server.
5. Using the list of countries of interest, Alice's backend server determines the EphIDs to be relayed to Bob's backend server.
6. Bob's backend server receives this information and evaluates whether he is at high-risk.
7. Bob's backend server alerts Bob if he is at risk.

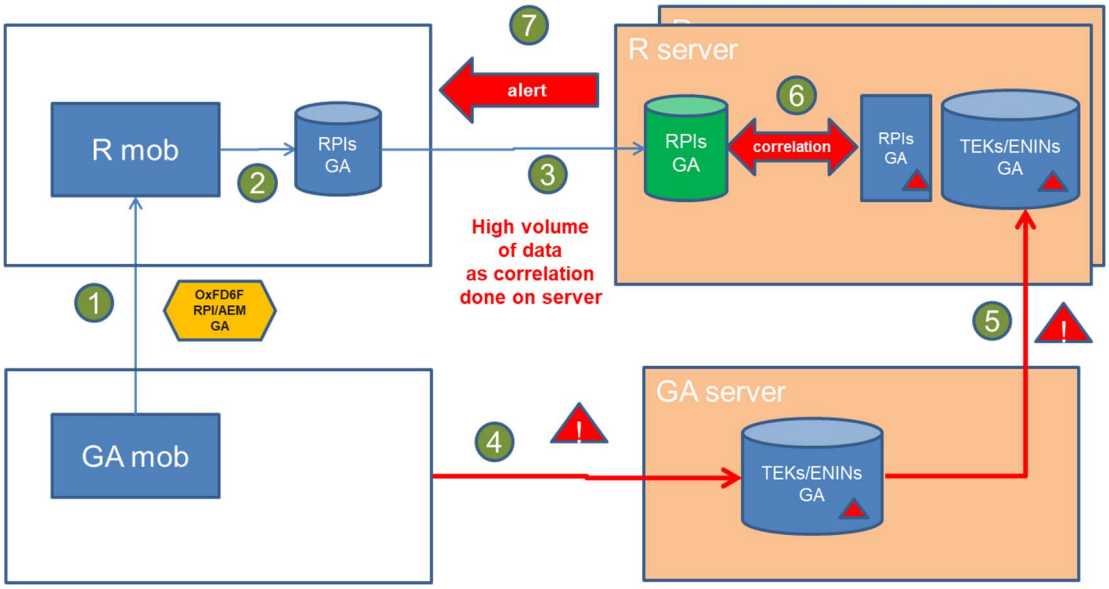


Figure 18: Case B1: A DP3T/GAEN user receives a positive test

7.1.2.2 Case B2: A ROBERT user receives a positive test

If Bob receives a positive test, he will send to its backend server the log of received EphIDs. Bob's backend server, will relay these EphIDs to the corresponding DP3T/GAEN backend server (the method used by Bob's backend server to identify the corresponding DP3T/GAEN backend server is not specified here). Alice's backend server relays the received list of exposed keys to all the devices. Alice's device will correlate this received list with its list of broadcast EphIDs, and determine her risk exposure; see Figure 19.

More specifically, the steps to follow would be:

1. Alice's device broadcasts EphIDs using Bluetooth® LE transmission.
2. Bob's device logs Alice's EphIDs.
3. Bob receives a positive test and transfer the received EphIDs to the DP3T/GAEN backend server.
4. Bob's backend server determines the EphIDs to be relayed to Alice's backendserver.
5. Alice's backend server receives this information relays it to the connected devices.
6. Alice's device correlates the EphIDs in the received list with her own broadcast EphIDs.
7. Alice's device determines her risk scoring, and alerts Alice if required.

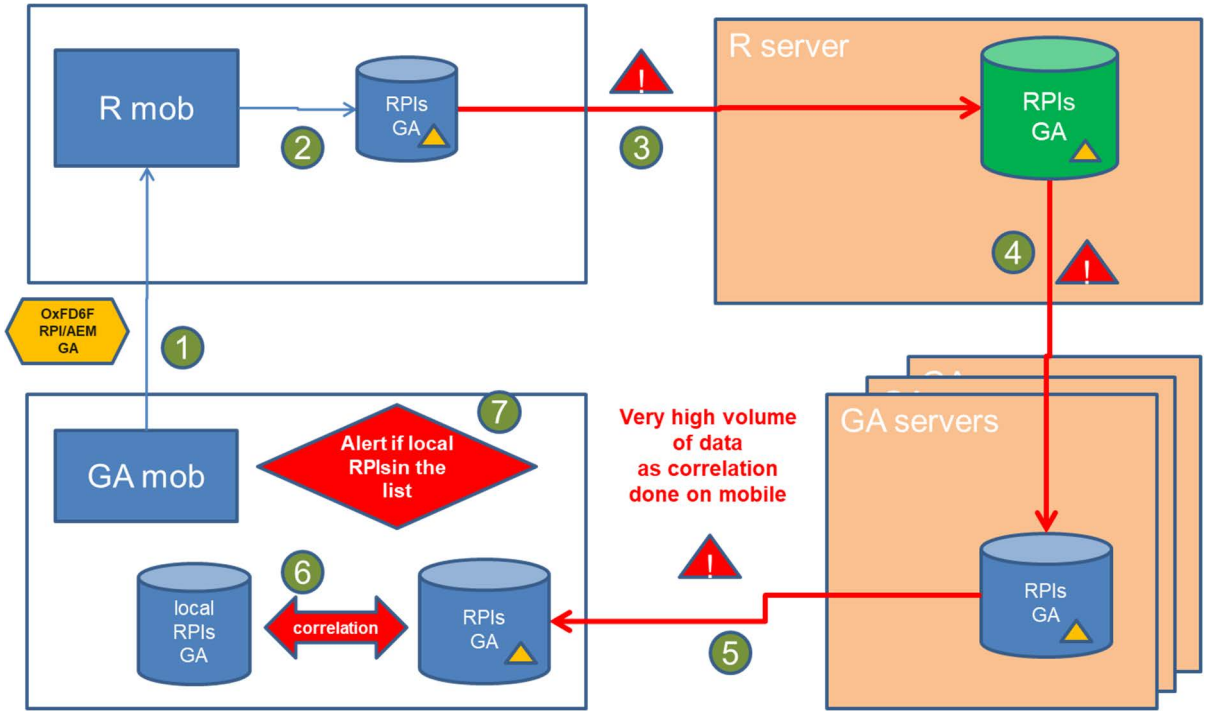


Figure 19: Case B2: A ROBERT user receives a positive test

7.1.2.3 Privacy risk for this interoperability scheme

A complete analysis of the privacy risks presented in this solution is also outside the scope of the present document. However, it can be seen that in the case B1, there is a privacy risk for the users of the DP3T/GAEN system who were not present in the case of no interoperability: As the EphIDs are correlated in the backend server of the ROBERT system, there is the risk of construction of contact graphs by the central server. This vulnerability is inherent to the ROBERT system, but which is not present in the standalone version of DP3T/GAEN; see [i.4]. Moreover, Bob uploads a list of visited countries, which is not necessary in the case of the ROBERT DCTS with no interoperability.

7.2 Interoperability between ROBERT and DP3T/GAEN+IDPT systems

7.2.0 General considerations

This clause describes a mechanism that allows the interoperability of users of ROBERT and DP3T/GAEN applications in which there are no changes in the privacy properties of each application.

7.2.1 Assumptions and notation

It is assumed that in the same geographic area there are users that can belong to any of the 4 different types of DCTSs: R, D, RI and DI. Applications RI and DI include a DCTS protocol, called Interoperable Digital Proximity Tracing (IDPT) protocol, allowing the interoperability between centralized and decentralized systems, without changing the privacy properties of these systems; see [i.7]. More specifically, it is assumed the following:

- Applications D implement the DP3T/GAEN protocol, without any further modification.
- Applications R implement the ROBERT protocol, without any further modification.
- Applications DI implement the DP3T/GAEN protocol in addition of the IDPT protocol. The risk score for these applications is the same as the one used in DP3T/GAEN, meaning that they are de-centralized applications.

- Applications RI implement the ROBERT protocol with an additional functionality. This includes new functionality in the backend server and the capacity of the nodes of processing the payload of incoming packets generated from the IDPT protocol. The risk score for these applications is the same as the one used in ROBERT, meaning that they are centralized applications.

It is assumed that applications R can interoperate by using the mechanism described in clause 6.2 of the present document, while applications D can interoperate by using the mechanisms described D in clause 6.3 of the present document.

Applications DI transmit packets using the same format of DP3T/GAEN, and they can interoperate amongst them, and with applications D, by using the standard mechanism of DP3T/GAEN. Applications DI transmit also packets using another format, as specified by the IDPT protocol.

Applications RI transmit packets using the same format of ROBERT, and they can interoperate amongst them and with applications R using the standard mechanisms of ROBERT. Applications DI can process received packets that use the format specified by the IDPT protocol, but they do not transmit packets using this format.

The mechanism described in the following clauses allow the interoperability of applications RI and applications DI without changing the basic properties of privacy of both applications.

To simplify the description of the mechanism, the following notation is introduced:

- For X = R, D, RI and DI, "X-device" is a mobile device that runs an instance of a X-type application, "X backend server" is the backend server used by an application X, while "X-user" refers to a user of an application X.
- As prescribed by ROBERT, the R, and RI backend servers generates ECC+EBID values, which will be called R-ECC+EBID and RI-ECC+EBID. Additionally, the transmitted in a beacon include the the fields "Time" and "MAC".
- As prescribed by DP3T, nodes D and DI generate EphID values, called D-EphID and DI-EphID respectively.
- DI-devices also generate EBID values called DI-EBID. DI-EBIDs have not attached an ECC field.
- IDPT requires the use of a relay, which is called DI-relay.

The R-ECC+EBID+Time+MAC, RI-ECC+EBID+Time+MAC, D-EphID, and DI-EphID values are 16-byte strings, transmitted in the payload of Bluetooth® LE packets.

I-EBIDs are 32-byte strings, and are transmitted using one of the two techniques:

- The technique described in clause 5.2.2.3.1 of ETSI GS E4P 006 [2], used in DESIRE.
- The techniques described in clauses 5 and 8 of the present document.

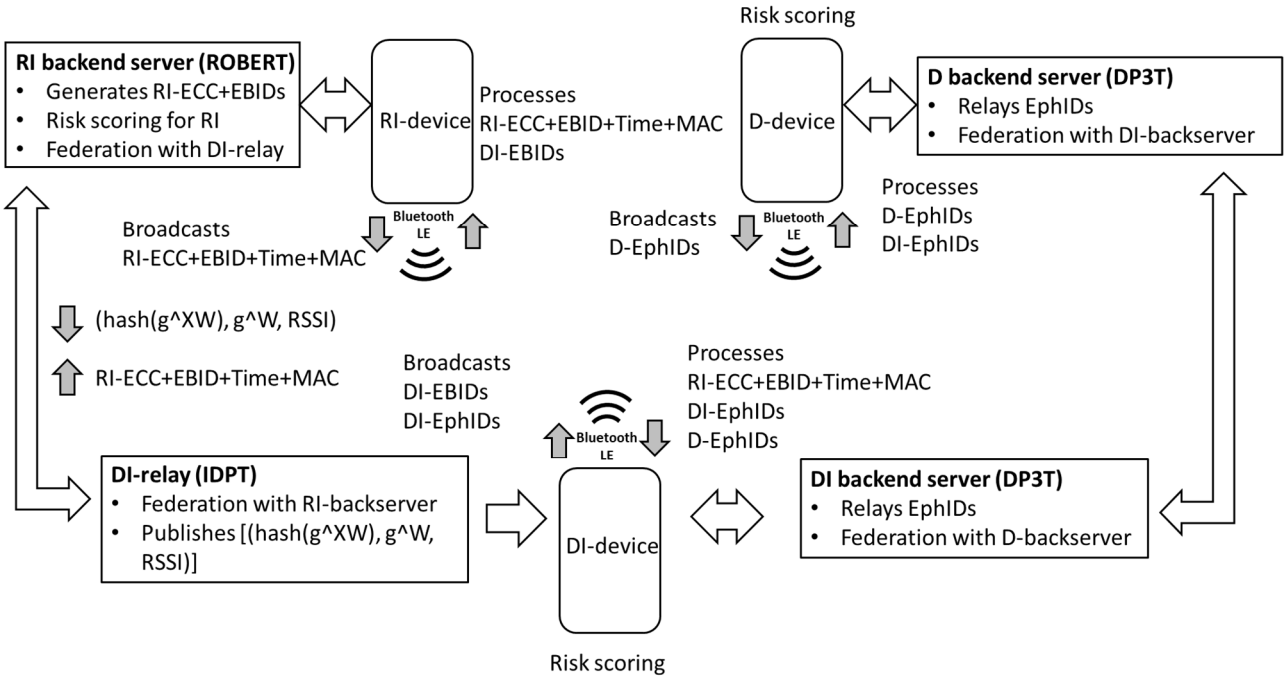
It is assumed that there are some metadata that distinguish the beacons emitted by different types of applications. More specifically, in the case of the app DI, it is assumed that it is possible to distinguish beacons carrying DI-EphIDs and DI-EBIDs; see clause 5 of the present document.

7.2.2 Backend servers and relays

Applications DI will require the use of a DP3T/GAEN backend server, and the use of a relay node, that it is called DI-relay. Its main functionalities are:

- Allow the exchange with the RI backend server of ephemeral identifiers of devices of RI-users and DI-users applications who are tested positive.
- Post lists of values that allow DI-devices check whether they were in exposure contact with a device of a user of the application RI who reported a positive test.

Applications RI will require the use of a ROBERT backend server, with the added functionality of allowing the exchange with the DI-relays of information uploaded by RI-users who were tested positive and were in contact with DI-users.



NOTE: Functional elements of the interoperability solution described in the present document in clause 7.2.

Figure 20: Functional elements of the interoperability solution

7.2.3 Ephemeral IDs generation

The RI backend server is responsible of generating the RI-ECC-EBIDs.

In applications DI, devices are the ones responsible of generating the DI-EphIDs, using the standard DP3T/GAEN mechanisms. These devices also generate another kind of ephemeral Bluetooth® IDcalled DI-EBID, as follows: each DI-device generates a sequence of secret numbers $\{X_n\}$ and computes a sequence $\{\text{I-EBID}_n\} = g\hat{}X_n$.

These $g\hat{}X_n$ values will be broadcast during an epoch of 15 minutes, and are stored in the DI-device for several days (for example, 14 days) where they are kept secret. These values will be used to generate a shared secret using a Diffie-Hellman-Merkle exchange. For simplicity, the scheme is described using the multiplicative DH notation, although the implementation would follow an instance of the discrete logarithm on elliptic curves; see [i.13].

7.2.4 Federation and backend server interconnection

For achieving interoperability with DI applications, the RI backend server should be able to identify the data to be forwarded to the DI-relay by another means, as DI-EBIDs are transmitted without an encrypted country code. This may be achieved by using the all-to-all replication mechanism described in clause 6.3. As discussed in clause 7.2.6, the list [(hash(g^XW), g^W, RSSI)] can include an additional k-bit hash of the g^X values, in order to reduce the computational cost in the DI devices when checking for matches for long lists.

7.2.5 Proximity Discovery and ephemeral ID processing

RI-devices and DI-devices that are in proximity interact as follows:

- RI-devices broadcast RI-ECC + EBIDs plus two fields "Time" and "MAC" on the payload of Bluetooth® LE beacons. DI-devices broadcast DI-EphID and I-EBID on Bluetooth® LE beacons.
- RI-devices store received RI-ECC+EBID+Time+MAC, R-ECC+EBID+Time+MAC (in case they are in proximity of R users), and DI-EBIDs.

- DI-devices store DI-EphIDs, D-EphIDs (in case they are in proximity of D users), RI-ECC+EBID+Time+MACs and R- ECC+EBID+Time+MACs (in case they are in proximity of R users, although this stored information would not be necessarily forwarded to the corresponding R-back end server). Note that DI-devices do not store DI-EBIDs.

This internal storage adds the reception epoch and additional information, such as the RSSI of the Bluetooth® LE signal.

7.2.6 Exposure Status notifications

For all four types of applications, when a user has a positive COVID-19 test, he/she receives an authentication token and voluntarily decides if he/she will use this token to report the application. If the user decides to report the positive COVID-19 test to the application, the following procedures are followed:

User of app DI reports a positive COVID-19 test

The application DI reacts as prescribed by the DP3T protocol, which means that devices of users of both DI and D apps are notified as it is assumed interoperability between the two applications. Again, devices apply a Risk Scoring algorithm to decide whether users should be notified.

In addition, the DI-device transfers to the DI-relay the received RI-ECC+EBID+Time+MAC and R-ECC+EBID+Time+MAC, which have been stored for several days (for example, 14 days). DI-devices do not transfer received DI-EBIDS to the DI-relay, as received DI-EBIDs are not even stored by DI-devices. This is not necessary since it is assumed that the other DI-users are notified using the DP3T/GAEN mechanism.

In other to avoid possibility of creating a contact graph by the DI-relay, the upload of the received RI-ECC+EBID+Time+MAC and R-ECC+EBID+Time+MAC should be done by using mechanism that ensure that the backend server cannot identify the device uploading the list, for instance using a mixnet, as it is also suggested for the case of DESIRE; see [i.9].

The DI-relay de-encrypts the field ECC from the received RI-ECC+EBIDs and forwards the information to the RI backend server. The RI backend server runs a Risk Scoring algorithm which decides which users of apps RI who were in contact with an DI-user who were tested positive are notified. It is assumed that DI-relays are also federated with R-backend servers, meaning that the received RI-ECC+EBIDs will also generate a valid ECC for these servers, and optionally would also be forwarded to the corresponding R backend server.

In other words, if an DI-user has a positive COVID-19 test, exposed devices of applications DI, D, and RI are notified.

Obviously, the DI-relay cannot de-encrypt the RI-EBID or R-EBID fields, since these EBIDs were generated by the backend server using a secret key K_S, meaning that there is no chance of creating contact graphs of users.

User of app RI reports a positive COVID-19 test

The application RI reacts as prescribed by the ROBERT protocol: it transfers the received RI-ECC+EBID+Time+MAC and R-ECC+EBID+Time+MAC and a processed value of DI-EBIDs (i.e. the LocalProximityList) to the RI backend server.

In order to anonymize the transfer of DI-EBIDs, and avoid the possibility of creation of proximity graphs in the RI-backserver and in the DI-relay, the RI-devices chose a random number W per received DI-EBID, and in the LocalProximityList transfer the tuples (hash(DI-EBID^W), g^W, RSSI)=(hash(g^XW), g^W, RSSI). i.e. including the RSSI values of the received packets with DI-EBIDS.

The RI backend server will separate these tuples (hash(g^XW), g^W, RSSI) from the RI-ECC+EBID+Time+MACs. The RI-ECC+EBID+Time+MACs and R-ECC+EBID+Time+MACs are processed locally by the RI backend server, as prescribed by the ROBERT protocol, which runs a risk scoring algorithm that decides which RI-users should be notified. It then transfers the list [(hash(g^XW), g^W, RSSI)] to the DI-relay.

In other to avoid possibility of reidentification by the RI-backend server, the upload of the list [(hash(g^XW), g^W, RSSI)] may be done by using mechanism that ensure that the backend server cannot identify the device uploading the list, for instance using a mixnet, as it also suggested in the case of DESIRE; see [i.9].

The DI-relay publishes a list of values [(hash(g^XW), g^W, RSSI)].

DI-devices periodically pull this list, e.g. twice daily. They compute hash((g^W)^X_n) = hash(g^X_nW) for the secret values stored in the sequence {X_n} that were generated by the device and all the values {g^W}. Then, they check if there is an intersection of the calculated values and some of the hashes in the list. They also determine the time epoch of the exposition.

The number of computations using the described mechanism would be 14*4*24*length([(hash(g^XW), g^W, RSSI)]), as every secret number X_n is checked against every element of the list. In order to avoid situations in which the overhead of this computation becomes too high, the RI devices can include a k-bit hash value of the received g^X (i.e. [(hash(g^XW), g^W, $hash_k$(g^X), RSSI)]. This can considerably reduce the amount of computations, as the values X_n which are checked for each element of the list are those values for which g^X_n produces the same hash as the value g^X (i.e. in average, it would be reduced by a factor 2^k). The value of k should be fixed for avoiding that only few values g^X give the same value of hash. If this situation happens, would mean that the overall amount of notified infections is too small, and a smaller value of k could be used without a too high computational overhead. Moreover, the RI devices could write dummy values for the hash values for which there is a small amount of notifications, thus slightly increasing computational overhead, but ensuring that the privacy would not be compromised.

Once the device determines the intersections, the application can run a risk scoring algorithm that decides whether the DI-user is notified. This algorithm is discussed in the next clause.

As a conclusion, it can be seen that if an RI-user has a positive COVID-19 test, exposed devices of applications R and DI are notified. Again, D-devices are not notified, as R and D do not interoperate.

7.2.7 Risk Scoring for IDPT

The algorithm that evaluates the risk scoring from exposures with RI-users who have reported a positive test would be based on the exposure time and proximity (obtained from the RSSI values). The specific form of this algorithm is left out of the scope of the present document.

7.3 Requirements for interoperability between systems with a different design approach

[IRD-01]: Applications that implement the DP3T/GAEN protocol in addition to the IDPT protocol (DI-applications) shall transmit packets using the same format as DP3T/GAEN, and packets using the format as the IDPT protocol.

[IRD-02]: Applications that implement the DP3T/GAEN protocol in addition to the IDPT protocol (DI-applications) shall be able to process packets that use the format specified by the DP3T/GAEN protocol and the format specified by the ROBERT protocol.

[IRD-03]: Applications that implement the ROBERT protocol in addition to the IDPT protocol (RI-applications) shall transmit packets using the format as ROBERT protocol.

[IRD-04]: Applications that implement the ROBERT protocol in addition to the IDPT protocol (RI-applications) shall be able to process packets that use the format specified by the ROBERT protocol and the format specified by the IDPT protocol.

[IRD-05]: DCTSs that support DI-applications shall use a DP3T/GAEN backend server and a DI-relay, as described in clause 7.2.2 of the present document.

[IRD-06]: DCTSs that support RI-applications shall use a ROBERT backend server with the added functionality of allowing exchange with DI-relays of information uploaded by RI-users who tested positive and were in contact with DI-users.

[IRD-07]: When a DI user notifies a positive COVID-19 test, the user's application shall upload their diagnosis keys to the DP3T backend server, as specified by DP3T protocol.

[IRD-08]: When a DI user notifies a positive COVID-19 test, the user's application shall upload the received and stored RI-ECC+EBID+Time+MAC and R-ECC+EBID+Time+MAC to the DI-relay. The latter procedure should be done by using a mechanism that ensures that the backend cannot identify the user of the device that uploads the list.

[IRD-09]: The RI backend servers and the DI relays that are federated shall use an 8-bit field Country Code, that uniquely identifies the geographical area (e.g. country, group of countries, region, or state).

[IRD-10]: The RI backend servers and the DI relays that are federated shall share and use a common secret Federation Key, as described in clause 7.2.4 of the present document.

[IRD-11]: When a RI backend server or a DI relay decrypts an EEC field and the CC corresponds to the geographical area (e.g. country, group of countries, region, or state) where another federated RI backend server is deployed, it shall securely forward the corresponding entry to the other federated backend server.

[IRD-12]: When an RI-user notifies a positive COVID-19 test, the user's application shall react as prescribed by the ROBERT protocol uploading the LocalProximityList to the RI-backend, and shall also upload the tuples (hash(DI-EBID^W), g^W, RSSI) (or optionally (hash(g^XW), g^W, $hash_k$(g^X), RSSI)) to the RI-backend.

[IRD-13]: The RI backend server shall identify the data to be forwarded to the DI-relay. The forwarding to the DI-relays uses the all-to-all replication mechanism described in clause 6.3 of the present document.

[IRD-14]: The DI-relay shall publish periodically a list of values. The format of this list of values should be either [(hash(g^XW), g^W, RSSI)] or [(hash(g^XW), g^W, $hash_k$(g^X), RSSI)].

[IRD-15]: DI-applications shall download the list published by the DI-relay to check, by using the mechanism described in clause 7.2.6 of the present document, if DI-users were in contact with an RI-user who reported a positive COVID-19 test.

8 Future harmonised interoperable contact tracing approaches

8.0 General considerations

This clause details the scenario where more than two protocols need to be supported. Upon the return of regular commuting and working arrangements it is conceivable that many DCTSs will have to interact across borders. Whereas previous clauses have dealt with the limited pair of systems currently in operation in the EU, in the future, worldwide interoperability should be considered. This clause describes a way to manage this interoperability in a standard manner, allowing all worldwide DCTSs to interoperate.

The approach is for a single general-purpose contact tracing Bluetooth® service to be advertised that supports both advertising and connection-based exchange, but is flexible enough to be used to describe and exchange DCTS payloads from a variety of systems through the use of a common DCTS description format. This would support centralized or decentralised approaches.

8.1 Additional interoperability challenges with more than two protocols

As mentioned in clause 5, there are a number of challenges with two-protocol systems. There are more when the need to support more than two protocols is considered. Namely:

1) The set of protocols and payloads to be supported cannot be predicted in advance.

2) It is practically impossible to co-ordinate software updates for all DCTS apps and backends throughout the world.

3) Including in every app the procedures to communicate with every other protocol and payload will likely lead to a loss in performance, and thus missed traced contacts, in all DCTS apps.

As a solution to this, it is proposed a single Bluetooth® LE service ID that supports both advertising-only and connection-based protocols, and allows the description of an arbitrary payload's content in a standard format. This is detailed in the following.

Protocols and payloads used beyond the EU include:

- Singapore created one of the first DCTS protocols which has been adapted by many countries. These share the same JSON format but are implemented in slightly different ways by each country, usually using their own service and characteristic UUIDs.
- Herald-based device-to-device payload exchange protocol in Australia and Canada in phones and wearables and beacons, supporting a range of payloads all sharing a common payload data format. The Herald protocol specifies a single set of Service and Characteristic UUIDs to facilitate interoperability and discovery. Australia's COVID application implementation of Herald currently uses its original custom device payload format over the Herald device-to-device payload exchange protocol with their own custom Service and characteristic UUIDs.
- Open source and proprietary wearable systems that shall not be considered here.

8.2 Bluetooth® device layer interoperability

8.2.0 General considerations

This clause describes device to device interoperability over Bluetooth®. In future other transports (e.g. UWB radio) could be supported. A common approach is shown for both advertising and connection-based protocols over Bluetooth®.

8.2.1 Advertising device payloads over a standard service

For advertising-based payloads that do not support secure exchange of tokens (e.g. DP3T/GAEN, Herald Simple payload) the device will expose the standard payload description through a common well-known manufacturer or service data area identifier, e.g. 0xFFFA. The same payload, and payload format, could also be exposed via a connection-based payload read characteristic on the standard device-to-device payload exchange service.

Whilst the advertising service data area would be using the same payload format, there are fields that could not be supported in such an exchange without a connection, e.g. replaying the remote inquiring device's TxPower or remote view of its signal strength (RSSI) as used for more accurate distance estimation. Those are necessarily only available in a connection-based approach.

Below is an example of a Herald Simple Payload v1.0 encoded as an advertising data area content. The full Advert PDU is shown in Table 1.

Table 1

Byte segments in Advert PDU	Explanation
02 01 1a	2 bytes data, Bluetooth® LE Flags (Connectable, Bluetooth® LE only)
02 0a 0c	2 bytes data, TXPower (12 dBm)
1a ff ff fa 10 3a03 0400 0f00 0f0e0d0c0b0a09080706050403020100	26 bytes data, manufacturer code fffa (Unassigned, used by Herald) 10 = device-to-device protocol header payload and version, Herald simple payload v1 3a03 = device-to-device protocol header country, 826 (UK) 0400 = device-to-device protocol header State, 4 (to be assigned by country) 0f00 = device-to-device protocol information remaining data length, 16 bytes 0f…00 =16-byte ephemeral identifier (similar to a DP3T/GAEN token)

Below is an example of a DP3T/GAEN token encoded in the same manner with a device-to-device protocol header, advertised by a mobile device, see Table 2.

Table 2

Byte segments in Advert PDU	Explanation
02 01 1a	2-bytes data, Bluetooth® LE Flags (Connectable, Bluetooth® LE only)
02 0a 0c	2-bytes data, TXPower (12 dBm)
03 03 6ffd	3-bytes data, complete 16-bit service UUID list, 6ffd is assigned to a specific vendor.
0bff4c00 1006 061a 396363ce 1824b4	11 bytes, vendor-specific manufacturer data area Vendor information (e.g. device type (Laptop, Phone, Smart TV, etc.), screen on/off, and other metadata)
17 16 6ffd 6a7b7defe811497244c5008ceefc57a4873ec9dd	23 bytes (17 is hexadecimal), 16 = service data, 6ffd = Device manufacturer service, remainder is the 20-byte DP3T/GAEN exposure token and metadata (16 byte 'chirp' and 4-byte metadata)

The above two examples can be encoded in a Herald International Interoperability description stored locally on the phone as follows in Table 3.

Table 3

Herald Interoperability byte data stored on the phone as a contact record	Explanation
(15) 10 3a03 0400 0f00 0f0e0d0c0b0a09080706050403020100	(Optional length, 21 bytes - only really needed when transmitting not locally storing) Herald Simple Payload. Same bytes as in advertisement
(17) 80 (3a03 0400) 6a7b7defe811497244c5008ceefc57a4873ec9dd	(Optional length, 23 bytes) 80 = DP3T/GAEN token binary representation (Optional routing code - 3a03 0400, UK, England - DP3T/GAEN does not include this data, but the phone could record the country/state it was in when it recorded the token.) Remainder is the DP3T/GAEN data as seen over the advertisement

Note that the country and state codes in the above examples have not been encrypted for ease of explanation. As mentioned in clause 5.1, this data, if present, should be encrypted over the wire, stored encrypted in the phone, and only be able to be decrypted by a DCTS backend.

It should be noted that a single 16-bit ID could be registered that would enable the above information - no matter which payload is being used - to be described and shared. This would allow for maximum interoperability and is the mechanism to be used in Herald Protocol for Advertising v1.4 in March 2021.

Using such a mechanism it would also be possible to advertise multiple payloads in the same service data area. Below is an example combining the above two advertisements in a single advert.

Table 4

Byte segments in Advert PDU	Explanation
02 01 1a	2-bytes data, Bluetooth® LE Flags (Connectable, Bluetooth® LE only)
02 0a 0c	2-bytes data, TXPower (12 dBm)
03 03 fffa	3 bytes data, complete 16-bit service UUID list, fffa = unassigned, used by Herald.
0bff4c00 1006 061a 396363ce 1824b4	11 bytes, vendor specific manufacturer data area Mobile device information (e.g. device type (Laptop, Phone, Smart TV), screen on/off, and other metadata)
30 16 fffa 14 80 6a7b7defe811497244c5008ceefc57a4873ec9dd 17 10 3a03 0400 0f00 0f0e0d0c0b0a09080706050403020100	48 bytes (30 is hexadecimal), 16 = service data, fffa = unassigned, used by Herald. First payload is 20 bytes (14 is hex), DP3T/GAEN token (80), and its token data. Second payload is 23 bytes (17 in hex), Herald Simple (10), and same fields as before.

Table 4 shows how the same data encapsulation mechanism can be used for over the air transmission and phone data storage and a space efficient and easy to parse binary format.

8.2.2 Connection based payloads over a standard service

For connection-based payloads that do not support secure exchange of tokens, their payload can be read from the standard device-to-device payload exchange service's read characteristic, or as a response to a write to the signal characteristic (for device's that do not support advertising themselves and have to write their payload in order to be 'seen').

This requires the exposure of a GATT service and characteristics. For example:

- A well-known Service UUID:
  - Could be a long unregistered 128-bit UUID or a registered 16-bit UUID.
  - A short 16-bit UUID may be registered in future with the Bluetooth® SIG.
  - For example, the Herald 128-bit UUID is 428132af-4746-42d3-801e-4572d65bfd9b.
- A well-known Read Payload (Exposure Token) characteristic UUID:
  - Could be a long unregistered 128-bit v4 UUID or a registered 16-bit UUID.
  - A short 16-bit UUID may be registered in future with the Bluetooth® SIG.
  - For example, the Herald 128-bit UUID is 3e98c0f8-8f05-4829-a121-43e38f8933e7.
  - Characteristic is read only, not write or notifiable.
- A well-known Write/Share payload and signal characteristic UUID:
  - Could be a long unregistered 128-bit UUID or a registered 16-bit UUID.
  - A short 16-bit UUID may be registered in future with the Bluetooth® SIG.
  - For example, the Herald 128-bit UUID is f617b813-092e-437a-8324-e09a80821a11.
  - Characteristic is write and notify only.
  - Write with response only, not write with no response (i.e. the OS does not support write without response).
  - Uses a 1-byte header for write message type (to allow multiple transmissions for secured exchange):
    - 0 - RSSI.
    - 1 - payload - may perform multiple calls in a secure exchange.
    - 2 - payload sharing - may perform multiple calls in a secure exchange.
    - 3 - immediate send (Arbitrary signalling data, needed by other distance peripherals):
      - Can be via write or notify.
      - Independently encrypted before exchange.

Multiple device payloads can be exchanged using the above read and write payload characteristics. The same data from the advertising mechanism mentioned in clause 5.1.3 could be exchanged. Also, multiple payloads' data can be exchanged in the same message. E.g. a Herald Secured device payload and a Singapore-style device payload sent at the same time.

Below is an example of a Singapore style device payload re-encoded for efficient exchange over the above mechanism. This can be seen in [i.8], Annex E.

Table 5

Data - As transmitted and stored on the phone as a contact record	Explanation
91 3a03 0400 1c00	Singapore-style system v2 binary reencoding, country code 826 (UK) state code 4 (state, e.g. England, to be assigned by the operating country), data length 28 bytes
0800 4142313233343536	ID string, 8 bytes UTF-8, 'AB123456'
40 01 c8	Received RSSI, 1 byte, -56
41 02 0a00	Received TxPower, 2 bytes, 12 dBm
42 09 6950686f6e372c32	Device model string, 9 bytes UTF-8, 'ModelAB1234'

8.2.3 Connection-based, with encryption

To prevent relay and replay attacks, and to allow the custody chain of a decentralised exposure token to be verified by the originating device with confidence, a connection-based exchange with key exchange pre-amble is necessary.

Such a mechanism is provided in the Herald Secured Payload as described in [i.8]. Whilst the whole exchange is out of scope of the present document, it is worth noting that such an exchange requires multiple payload writes, rather than using reads. The sequence of events is as follows:

- Initiator writes the device-to-device protocol header, identifying a secure exchange payload, and writes it's Diffie-Hellman-Merkle public information.
- The write response from the remote include the device-to-device protocol header of the response, followed immediately by its Diffie-Hellman-Merkle public information, followed by the encrypted payload.
- If the initiating device does not support advertising itself (and thus cannot rely on the other device requesting its payload in return), a second exchange may occur immediately, re-using the public key information from the remote but with a new public DH information from the initiating device, and the devices own encrypted payload. The response will be empty.
- All of the above occurs over the notify characteristic using the 'payload' message key.
- Due to the nature of such an exchange, if a device supported exposing two device payloads, separate writes would have to occur to exchange this information (as each may have a separate security procedure).

It should be noted that the above encryption is 'data in transit' encryption so that the communication is private between two phones, under the assumption that no active (i.e. modifying attacks like man-in-the-middle attacks) occur. The payload shared between them may be independently encrypted such that metadata can only be decrypted by the appropriate actor.

For example, the data passed via the above mechanism and decrypted by the receiving phone could be described as follows:

Table 6

Data stored on phone	Explanation
(XX) 20 YY	(Optional XX data length), 20 = Herald Secured payload, YY = remaining data length.
3d ZZ <encrypted bytes>	3d = data to be decrypted by the receivers DCTS, ZZ bytes.
01 01 49	Transmitter's RSSI for the receiver: 73.
42 09 <9 bytes>	Transmitter's phone model code "ModelAB1234"
20 04 5FD9147D	Time of the start of the contact event as recorded by this receiving phone - equivalent to 1608062077 (Tue 15 Dec 2020 ~19:54) - does not leave the phone, used for risk estimation.
21 04 5FD916D5	Time of the end of the contact event. Equivalent to 10 minutes after the start time) - does not leave the phone, used for risk estimation.

8.3 Device talking to its provider's DCTS backend

8.3.0 General considerations

This clause describes what happens when an owner of a device receives for instance a positive COVID-19 test, and how collected data described in a standard way is shared to the operating DCTS.

8.3.1 Uploading exposure information to a DCTS back-end

Whilst each current implementation has its own way to communicate data between a mobile app and the DCTS system back-end that operates it, this clause describes a standardized mechanism to upload exposure token information.

It is assumed that there are two DP3T/GAEN based app that needs to upload its own exposure tokens (decentralised) but also the exposure tokens for a Herald based system that it was received from other non- DP3T/GAEN devices.

The same standard description format can be used for describing both sets of information. Following on from previous examples in clauses 8.2.1 and 8.2.3, the receiving phone would upload the following to their DCTS back-end.

Table 7

Data bytes uploaded to DCTS back-end	Explanation
XX total length in bytes of this clause	Optional. This length is included as the upload may have many uploaded contact keys, and depending on the upload method the total length may need specifying first. (HTTP, for example, already has a Content-Length header, and so this field can be dropped for such a transport).
80 YY 43 0a <16 bytes>	80 = DP3T/GAEN, YY = total length 43 = Exposure daily key 0a = 16 bytes, and the bytes for this key.
… repeated clause as necessary to cover all days	There will be one exposure key per day uploaded by an individual who falls ill.
20 ZZ	Herald secured payload data of ZZ length in bytes.
3d WW <encrypted bytes>	Only the encrypted data intended for the DCTS is shared. WW = bytes length in hex.
22 04 12340000	Approximate contact start time - E.g. rounded down to the day the contact happened. Optional. Allows for epidemiological anonymous graph analysis.
Repeated clause as necessary to cover all contacts	Each contact exchange will have data.

The DCTS back-end now performs two actions:

1) Adds the DP3T/GAEN token to their in-country list of exposure keys.

2) Decrypts the non-DP3T/GAEN data to determine who to pass onward notification of exposure to.

Below is the data from the above Herald payload as decrypted by the DCTS back-end, and stored for later interpretation in the DCTS.

Table 8

Data bytes decrypted by DCTS back-end	Explanation
20 XX	Herald secured payload, XX = total length
YY 0a <bytes>	Persistent anonymous contact ID - allows building of an unattributed infection transmission graph, e.g. 16 bytes
ZZ 04 033a 0400	Routing code (Country 826 UK, State 4, state within UK) - This is the protocol to send the below encrypted exposure information on to
3f WW <bytes>	Encrypted data for decryption by the transmitter's device protocol. WW = bytes length in hex
22 04 12340000	Approximate contact start time - e.g. rounded down to the day the contact happened. Optional. Allows for epidemiological anonymous graph analysis. Based on Unix epoch seconds

The DCTS now has enough information to build an anonymous contact graph where only the person uploading the data is identified. This enables analysis within country of a transmission graph allowing analysis of the current R rate and spotting of asymptomatic spreaders in the anonymous graph. Of course, this data is optional. The user could, for example, choose only to share the encrypted exposure tokens rather than extended information describing the contact.

The receiver's (who has for instance received a positive COVID-19 test) DCTS now shares the exposure tokens with other DCTSs as described in the following clause.

8.4 DCTS backend interoperability

8.4.0 General considerations

In order to be able to pass on a notification of exposure to a device of a user in another country/state operated by a different DCTS, DCTS backends shall communicate using a shared mechanism for interoperability. Whilst this clause does not describe the low-level API required, it describe the principles and operations that are needed at a high level.

8.4.1 DCTS operating authority back end interoperability

At this point a DCTS app (DCTS B in our example) has exposure tokens for another country's (DTCS A) user's (User A) app and knows that its resident (User B) has fallen ill. DCTS B now choose to share this information with DCTS A.

To do this it should share the encrypted information (code 3f in the previous clause) intended for DCTS A, and may optionally share additional information. A prime example of this is the status of the ill individual (e.g. symptomatic and untested, or tested positive, or now tested as clear), and the disease with which they have tested positive (e.g. COVID-19, or a specific variant).

When received by DCTS A, the data can be decrypted, resulting in the following fields being available to DCTS A.

Table 9

Data decrypted by DCTS A	Explanation
19 0a <bytes>	Persistent anonymous v4 UUID (if shared) of the transmitter (User A), 16 bytes. DCTS A does not know this identifier as User A's phone never shares it with DCTS A. Allows for reverse country notification (E.g. if DCTS A analyses the data, and determines they need to notify this person's contacts of exposure - DCTS would now know to send this notification to DCTS B).
22 24 <bytes>	Exposure Service Token, 36 bytes - allows DCTS A to confirm that DCTS B is passing information definitely generated by DCTS A's DCTS app.
23 0a <bytes>	Exposure Confirmation Token, 16 bytes - Generated by User A's phone using credentials only known to that phone.
(optional extra data)	Any additional information shared, e.g. receiver's phone make/model for more accurate distance and risk estimation, or the date that the contact occurred.

That Traveller's DCTS (DCTS A) now adds the Exposure Confirmation Token to the download exposure token list for their app's users. The device for User A can match this token and any provided additional information (E.g. exposure date) to validate that it is a correct token generated by their device, thus completing the privacy and security checks, and preventing relay and replay attacks from hostile state actors.

The above mechanism works for payloads received by any DCTS protocol currently in use, and allows a standard way to share exposure information no matter which protocols and payload formats, or which subset of data, is in use worldwide. This is the mechanism described in the Herald International Interoperability document and reference implementation; see [i.8].

A full example including optional data is shown in Figure 21 (grey is encrypted and unreadable by this observer).

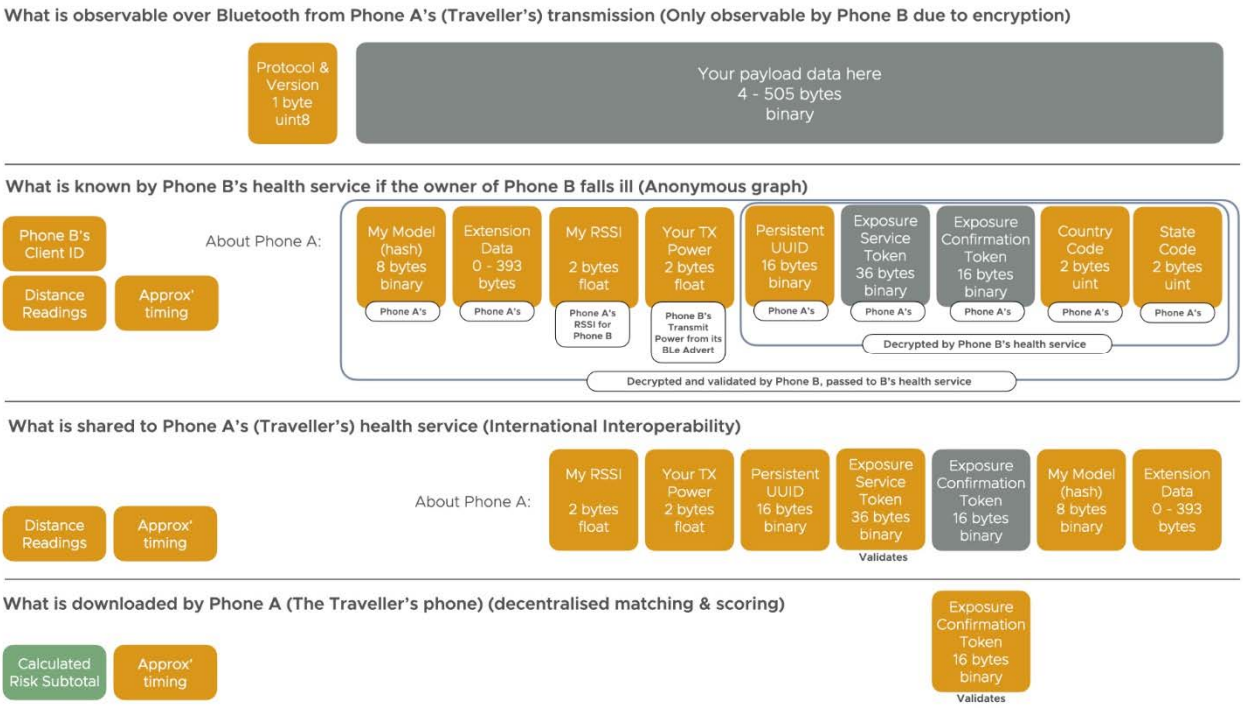


Figure 21: Flows of information

It should be noted that if the scenario were reversed and the Traveller (User A) not using DP3T/GAEN had fallen ill, he/she would forward the tokens observed (rather than their own daily keys) to his/her country's DCTS - as that app does not 'understand' DP3T/GAEN tokens itself - and the Traveller's DCTS (DCTS A) would forward this information to the local DCTS (DCTS B). (i.e. the country they were visiting). This would not allow notification in the local user's (User B's) country due to the way daily keys are uploaded only on becoming ill (Which is why any DP3T/GAEN app would need detection of 'other protocols' added, e.g. via Herald).

What this would allow though is The Traveller's DCTS (DCTS A) to provide a temporary 'callback identifier'. Should the local user (User B) fall ill and upload their daily keys, The Local DCTS (DCTS B) can identify the subset of countries for which such a daily key is relevant, and send those DP3T/GAEN keys to those countries, minimizing data exchange and potential loss of privacy. This is an edge case, as it would require both users to have fallen ill, but is useful in tracing the spread of a disease.

8.5 Migrating between protocols across application updates

There are several challenges when migrating between payload formats and protocols. These are listed below:

- Same-app software updates and backwards compatibility.
- Different versions of the same protocol in the same or different apps.
- Migrating between versions of the same payload in the same or different apps.
- Supporting multiple exposure token payload versions/types - as described on previous pages.
- Migrating between different protocols/payloads.

The approach of supporting multiple device-to-device protocols is the same approach as supporting multiple device payloads and device-to-device protocol updates for the same country. Using a standard description format and common device-to-device protocol, as described in previous pages, makes software updates and switching between device payload versions trivial - they all share a common data description format.

It has been shown in this clause that adopting a common DCTS service that is flexible enough to support a variety of device payloads allows for international interoperability across borders. It has been also shown that this can be done for very different device-to-device protocol approaches (decentralised, centralized) which maintaining security and privacy and without compromising one nations' user information by sending it to many other nations.

8.6 Requirements and recommendations for future harmonised interoperable contact tracing approaches

8.6.1 Requirements

[IHC-01]: The DCT application shall support a standard Bluetooth® Service UUID registration that is common across all countries for DCT service detection.

[IHC-02]: The DCT application shall support the binary standard multi-payload description format described in the present document to facilitate a single payload description format no matter the payload content.

[IHC-03]: The DCT application shall support encrypted communication of all payload data in order to prevent replay and relay attacks and the compromise of personal information (e.g. app provider's country and state code).

[IHC-04]: The DCT backend shall not transport any part of the data packet that contains personal information to another country or state except the ephemeral contact ID and the date of the contact event.

NOTE: The DCT system may share a risk score for the whole event. The DCT system may also share the status of the individual at the point in time for the contact event (e.g. tested positive, or if just before the test presumed positive via the epidemiology of the disease). This may include multiple risk scores and status information for diseases and strains of concern. (E.g. a SARS-Cov-2 strain, or an Ebola strain). This prevents the need to send exact time, duration, and distance information.

[IHC-05]: DCTS backend to backend communication shall be encrypted, e.g. using a mechanism such as mutual TLS.

[IHC-06]: A user's DCT application shall only communicate with another DCT backend if the user consents.

8.6.2 Recommendations

DCTS backend implementors should implement a single common set of interoperability services to facilitate interoperability between DCTS backends.

DCTS backend to backend communication should use the same general binary data format for contact event information as described in the present document.

For countries that do not support the Herald protocol in their DCT applications, a DCTS backend should allow the registering of contact tokens of interest by another DCTS backend so the user of that state's DCT application will receive notifications even if that format is not supported by the country they have visited (this is the 'callback identifier' mechanism mentioned earlier in the present document).

Annex A (informative): Matching with GS 'Requirements for Pandemic Contact Tracing Systems using mobile devices'

Table A.1

Interoperability requirement	System requirement (from ETSI GS E4P 003 [1])	Decentralized approach		Centralized approach	
		Applicable to DP3T/GAEN	Applicable to others	Applicable to ROBERT	Applicable to others
Bluetooth® LE Layer Interoperability					
[IBL-01]	**[HL-MA-03] [HL-IO-02]**	Yes		Yes	
[IBL-02]	**[HL-MA-04] [HL-SE-13] [HL-IO-02]**	Yes		Yes	
[IBL-03]	**[HL-MA-04] [HL-IO-02]**	Yes		Yes	
Interoperability between ROBERT systems					
[IRS-01]	**[HL-SE-13] [HL-IO-03]**			Yes	
[IRS-02]	**[HL-SE-13] [HL-IO-03]**			Yes	
[IRS-03]	**[HL-SE-13] [HL-IO-03]**			Yes	
[IRS-04]	**[HL-SE-13] [HL-IO-03]**			Yes	
Interoperability between DP3T/GAEN systems					
[IDG-01]	**[HL-IN-03] [HL-IO-03]**	Yes			
[IDG-02]	**[HL-PV-05] [HL-IO-03]**	Yes			
[IDG-03]	**[HL-IO-03]**	Yes			
[IDG-04]	**[HL-IO-04] [HL-IO-05]**	Yes			
[IDG-05]	**[HL-IO-04] [HL-IO-05]**	Yes			
[IDG-06]	**[HL-IO-04] [HL-SE-13] [HL-IO-05]**	Yes			
[IDG-07]	**[HL-IN-03] [HL-IO-03]**	Yes			
Interoperability between systems with a different design approach					
[IRD-01]	**[HL-IO-02]**	Yes		Yes	
[IRD-02]	**[HL-IO-02]**	Yes		Yes	
[IRD-03]	**[HL-IO-02]**	Yes		Yes	
[IRD-04]	**[HL-IO-02]**	Yes		Yes	
[IRD-05]	**[HL-IO-03]**	Yes		Yes	
[IRD-06]	**[HL-IO-03]**	Yes		Yes	
[IRD-07]	**[HL-IO-03]**	Yes		Yes	
[IRD-08]	**[HL-IO-03]**	Yes		Yes	
[IRD-09]	**[HL-MA-03] [HL-SE-13] [HL-IO-03]**	Yes		Yes	
[IRD-10]	**[HL-MA-03] [HL-SE-13] [HL-IO-03]**	Yes		Yes	
[IRD-11]	**[HL-MA-03] [HL-SE-13] [HL-IO-03]**	Yes		Yes	
[IRD-12]	**[HL-IN-03] [HL-IO-03]**	Yes		Yes	

Interoperability requirement	System requirement (from ETSI GS E4P 003 [1])	Decentralized approach		Centralized approach	
		Applicable to DP3T/GAEN	Applicable to others	Applicable to ROBERT	Applicable to others
[IRD-13]	**[HL-IN-03] [HL-IO-03]**	Yes		Yes	
[IRD-14]	**[HL-IN-03] [HL-IO-03]**	Yes		Yes	
[IRD-15]	**[HL-IO-03]**	Yes		Yes	
Interoperability between future harmonised digital contact tracing systems					
[IHC-01]	**[HL-MA-03] [HL-IO-02]**	Yes	Yes	Yes	Yes
[IHC-02]	**[HL-IO-02]**	Yes	Yes	Yes	Yes
[IHC-03]	**[HL-SE-12] [HL-IO-02]**	Yes	Yes	Yes	Yes
[IHC-04]	**[HL-PV-11] [HL-IO-02]**	Yes	Yes	Yes	Yes
[IHC-05]	**[HL-SE-08] [HL-SE-09] [HL-IO-03]**	Yes	Yes	Yes	Yes
[IHC-06]	**[HL-PV-02]**	Yes	Yes	Yes	Yes

History

Document history		
V1.1.1	May 2021	Publication