DIGID Project
Technical Specifications

June 2021

Supported by:

Table of contents

1. Introduction

Cash distribution and digital identity

Humanitarian aid and beneficiary identity have become inextricably linked. With limited aid budgets, organisations need to make sure they reach as many people in need as possible. **Hence, aid has to be targeted to specific groups and there needs to be a mechanism to prevent "double-dipping".** This means that organizations delivering humanitarian aid need to collect personal data about their beneficiaries to assess whether they fit into the programme criteria and to re-identify them at the time of distribution. The latter requirement has led to an increased use of biometrics, especially fingerprints, in aid programmes.

The emergence and rapid expansion of cash transfer programmes has amplified that need. Cash disbursement, especially through mobile payment channels or bank accounts often require a Know-Your-Customer (KYC) procedure to be performed. In fact, the financial service providers (FSPs) used for distribution are usually required by law to perform an identity check. This implies **checking whether the beneficiary is the holder of a valid government-issued identity (i.e. legal identity).** Unfortunately, a significant proportion of beneficiaries of aid do not have a legal identity. This could be due to the fact that they have never been enrolled in a national identity scheme or because they have been forcibly displaced.

Even though **provisional functional identities** are established by humanitarian organizations for the purpose of aid delivery, the long term utility of such identities for the beneficiaries of aid remains questionable.

Being able to leverage these dispersed and siloed identities, along with other information such as cash transfer transaction history (and in the longer run training programs and language skills etc.) to continue availing services is crucial in building self-reliance for beneficiaries of humanitarian aid.

A **decentralized digital identity solution** could not only help humanitarian aid organizations identify and serve their beneficiaries better, but also **economically empower beneficiaries of humanitarian aid with a self-owned, coherent identity for the long run.**

Such a solution reinforces the spirit and priorities established by the **Grand Bargain**. In particular:

- Greater transparency (Workstream 1),
- Increase the use and coordination of cash-based programming (Workstream 3),
- Reduce duplication and management costs (Workstream 4),
- Improve joint and impartial needs assessments (Workstream 5), and
- A participation revolution: include people receiving aid in making the decisions which affect their lives (Workstream 6).

In addition, the accumulation of identity related data at humanitarian organisations creates concerns around privacy, data security and scope creep of existing applications. These provide an urgency to **decouple the identity management of humanitarian organisations from software that is intended for other purposes**. Also, the amount of

identity data that is actually stored and accessible by the organisation at any given time should be minimized and access should only be granted on a per need basis.

This project sets out to provide an open source digital identity platform based on decentralized identity technology that is inclusive and easily integratable by different humanitarian organisations. It will be piloted within the **framework of a cash transfer programme in Kenya in April 2021.**

The structure of the document is as follows:

- In the remainder of this section **(Section 1)**, we will lay out the **current process flows of cash distribution**,
- **Section 2 describes the DIGID decentralized identity platform,** starting from a self-sovereign identity scenario and explains how this project takes the limitations with regard to literacy, connectivity and device penetration into account to build an inclusive platform, and
- **Section 3 provides the technical details of the implementation**, including the high-level architecture and sequence diagrams.

Current Cash Distribution Processes ("As is" flows)

Beneficiary Identification & Registration

When a disaster or shock strikes a community, humanitarian organizations (such as KRCS) active in the area **visit the community to introduce themselves and the assistance program, as well as conduct the community selection process** to identify a preliminary list of eligible beneficiaries.

Once beneficiaries have been informed of their eligibility for assistance, KRCS staff proceed with their registration for the program.

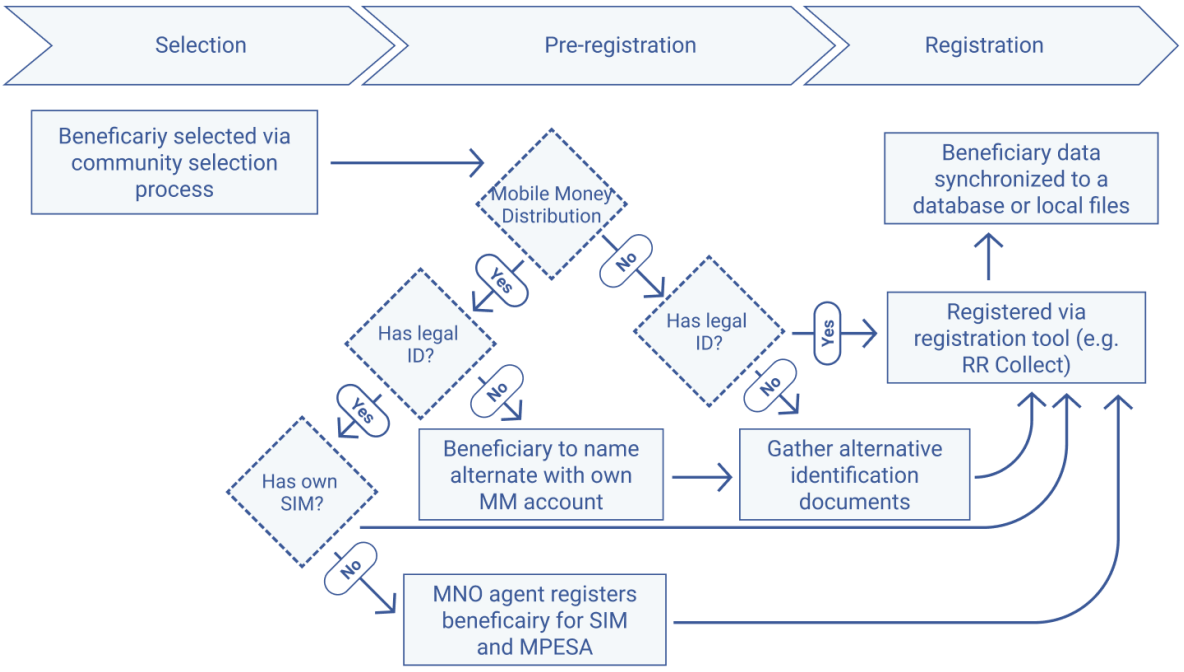
The registration process collects information from beneficiaries in order to:

1. **Establish the identification mechanism**, so the right beneficiaries are identified and enrolled in the program, and a form of identification is used to authenticate them when receiving their assistance;
2. **Validate eligibility of beneficiaries** (e.g., check against agreed targeting criteria which includes vulnerability);
3. **Gather other relevant data needed for the program** (e.g., shelter questions for shelter-related intervention);
4. **Additional data required to fulfill Know Your Customer (KYC) requirements** and to facilitate cash distribution (e.g. phone number and a valid account for mobile money) may also be collected.

Typically, **a data collection tool such as ODK/Kobo, RedRose Collect, or Excel** is used with a prepared questionnaire. In certain cases, basic pen and paper might be used and

later transcribed digitally. **Data collected after the registration process is typically stored in a database or in an Excel spreadsheet.**

Regardless of the modality of assistance (direct cash, voucher or mobile money), registration is done as follows:



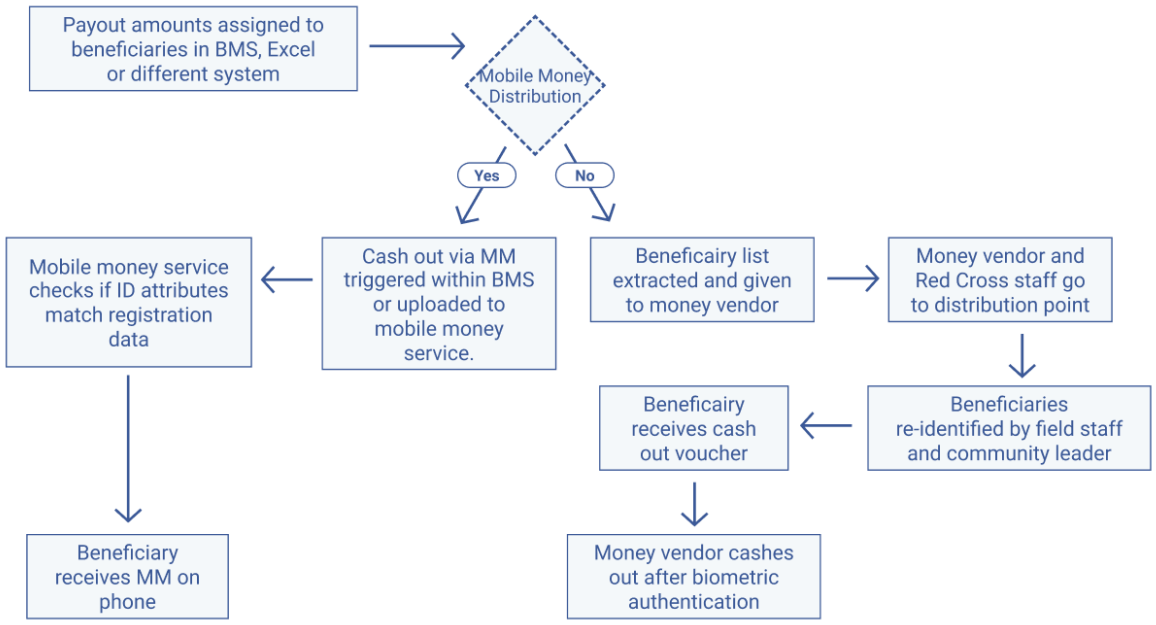
1. **The organisation prepares for data collection**, which includes the tool to use for registering beneficiaries (e.g. ODK/Kobo, RedRose RR Collect, Excel, or pen and paper) as well as the questionnaire or survey form.
2. **Beneficiaries are registered one by one** (via home visits or they are invited in a common area).
3. **Depending on the payment mechanism, KYC requirements, and program requirements, beneficiaries are asked questions about their identity** (e.g. names, age, legal ID, address or location), established if they match the vulnerability criteria and any additional program-related data is captured.
    a. **In the case of mobile money distribution, a phone number and a mobile money account (e.g., MPESA) will be required.**
        - In the c**ase where a beneficiary does not have a SIM card** registered in their own name, a proxy with a valid ID and SIM card, who is authorized to receive cash on their behalf is registered in addition to the details of the head of the household. **In certain cases, the organisation may negotiate with the Mobile Money Operator to register a limited use SIM card and MPESA account** to a beneficiary to receive one-off cash assistance. The KRCS is having these negotiations with Safaricom, for example.

b. **When distributing using money vendors or other modality,** if the beneficiary **does not have a legal ID, registration involves gathering alternative identity documents.** The exact nature of these documents largely depends on the context, especially in terms of geography. It can be a birth certificate, school enrollment certificate, a church membership or a letter from the community leader attesting they are a member of the community. Identity details are recorded and a picture of the document might be taken and uploaded. Biometrics might also be taken to allow for stronger authentication, when appropriate.

4. The **collected data is either uploaded into a database or saved in files** (e.g. Excel) locally on a computer.

Cash Transfer Distribution

Before the distribution, the programme officer assigns payout amounts to the beneficiaries, either in the beneficiary management system or manually.



The distribution flow then largely depends on which payment mechanism is being used.

a. **In the case of mobile money, cash-outs are triggered by submitting payment and KYC details to the mobile money service.** This can be done from within the data management system if it has an integration with the mobile money service, through an API, or through a web portal provided by the mobile money service. The beneficiaries then receive the cash on the mobile money account. The process is similar for other digital payments means.

b. **In the case of physical cash distribution, a beneficiary list along with the amount to be paid out is extracted and given to the money vendor.** The money vendor and Kenya Red Cross go together to the point of distribution. Beneficiaries first go to the Kenya Red Cross staff where they are identified and receive a voucher for the cash out. They then proceed to the money vendor to

retrieve their cash. In some cases, an additional authentication is performed at this point using the beneficiary's fingerprint.

2. DIGID Decentralized Identity platform

The case for self-sovereign identity

Given the persistent need to collect identity data for humanitarian programming, systems need to be designed and implemented that help **overcome some of the main challenges,** such as:

- **Improving security and privacy** and keeping an **audit trail of access** to identity data,
- Allowing the same and other authorised organisations to **request access to a beneficiary's data for specific purposes** and demonstrate interoperability between organisations,
- **Decoupling identity management from program management systems such as Red Rose** and offering a dedicated service that grants third-party applications like Red Rose access to beneficiary identity on an on-demand basis, and
- **Giving beneficiaries as much control as possible over their identity data.** Ideally, beneficiaries are in full control of their data and only provide access to specific organisations on a per-need basis.

Self-sovereign identity, or rather, the underlying technology has the potential to address these points and deliver additional benefits.

The key characteristics of self-sovereign identity include:

- **Verifiability**: The use of digital signatures ensures that the authenticity of the data that is shared by the beneficiary can be verified.
- **Data Standards:** Alignment with open data standards such as Verifiable Credentials and Decentralized Identifiers ensures that other humanitarian organisations and third-parties will be able to easily read and process data from beneficiaries due to standardized formats and processes.
- **User control:** Data cannot be shared without the digital signature of the beneficiary. In case the beneficiary is unable to sign themselves, a designated Guardian entity can do it on their behalf. This ensures that no data is accessed without consent.
- **Auditability:** As mentioned above, data that is shared by the beneficiaries comes with the beneficiary's (or their guardian's) signature. It also includes some metadata, e.g. the duration of the consent and the specific purpose. This reduces liabilities for humanitarian organisations and increases auditability.

SSI User Journey

Wallet Setup

User receives a **Sign Up SMS** with an onboarding link

User sets an **encryption PIN** for the keystore. The keystore is used to decrypt and create digital signatures

Once the wallet is created, we perform an **identity check** to make sure the user is who they say they are.

Now the user is **ready to receive data** from credential issuers

Consult and Populate Gravity Wallet

Log In

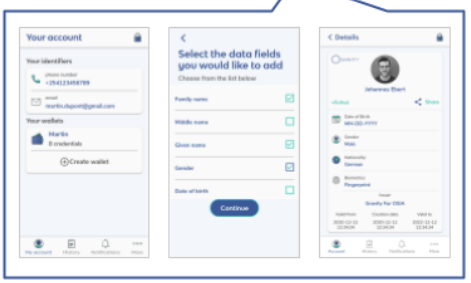
User logs in after authentication and accesses his Gravity Wallet

Self declare credentials

User may self declare credentials through the Gravity webapp to populate his wallet (e.g household size...)

Request credentials

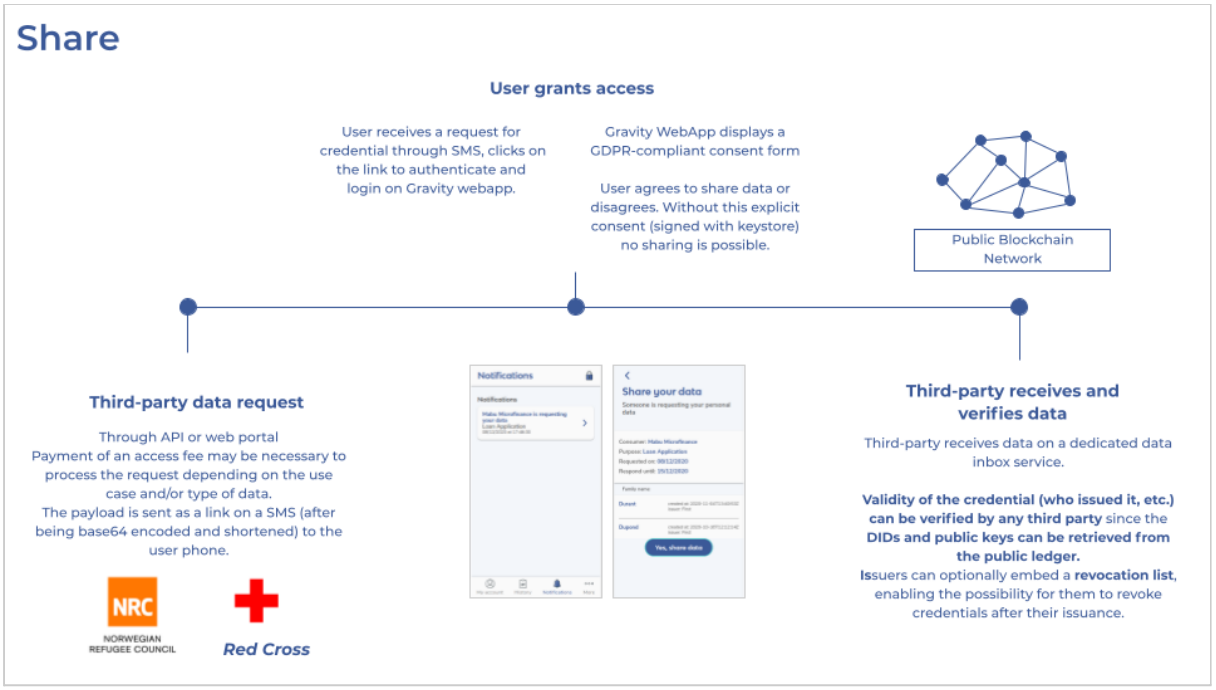
User fills in a form to request credentials from third parties to populate his wallet (e.g health condition, education certificates, ...).



Issue credentials

Third parties are able to create verifiable credentials either by accessing their **personal issuing service or by integrating an SDK**. Credentials are **encrypted** and send to the user's wallet so it is only readable by the user.

Red Cross



Implementing an SSI-based solution in the humanitarian aid setting could have longer term benefits for beneficiaries which go beyond the immediate benefits accruing from an ad hoc cash transfer (or other humanitarian aid) program. Beneficiaries, especially those who lack an official proof of identity, may leverage the identities established within the context of humanitarian aid programs to access mobile and financial services with service providers outside the humanitarian aid ecosystem.

DIGID Digital identity platform - SSI with Guardianship

Current barriers to full SSI adoption

Access to smartphones

Data sharing in an SSI protocol requires credentials and consents to be encrypted and signed on users' smartphones. This can make full SSI adoption difficult in the humanitarian aid context since beneficiaries of aid may typically belong to low income and vulnerable groups without smartphones. While this issue is mitigated by the guardianship model to a large extent within the scope of the DIGID project, it still remains a significant barrier to full SSI adoption.

Digital literacy

Sharing, requesting, viewing and updating credentials depends on the end-user/data subject's familiarity with and knowledge of digital platforms and channels. End-users may also need to be comfortable with additional concepts such as mobile money and, to some extent, data privacy and protection. Low levels of digital literacy could therefore prevent end-users from enjoying the full suite of advantages offered by an SSI protocol.

Unfamiliarity with the concept of digital identity

Conveying the concept of digital identity and its utility to end users may be difficult in the humanitarian context. As observed during the user consultations for the DIGID project, community members did not fully grasp the utility of a digital identity in itself and preferred having a physical identity credential (ID card). Unfamiliarity with the concept and its benefits could limit the adoption of such a solution by end users.

Key recovery

Digital signatures and encryption play a key role in decentralized identity. But they also bring added complexity. Operations that are performed on a centralized server in other systems are now performed on the devices of the respective entities that interact with the platform. This implies that these entities (including the beneficiaries) need to keep their keys safe and also be able to save a 12-word passphrase that is necessary to recover their keys if they lose them (in case they lose their device, for instance).

Interoperability between protocols

The full SSI adoption scenario is aimed at complete interoperability between different SSI protocols. This would allow for the sharing and reading of credentials between different SSI protocols without the need for additional integrations. Given that complete interoperability between different SSI protocols is still a work in progress, the possibility of a full SSI solution is limited for the time being.

Guardianship

Given the current barriers to the adoption of SSI, **this project takes a more pragmatic approach that delivers many of the features of SSI but accounts for the fact that most beneficiaries lack smart devices.** However, many do have access to basic phones and mobile phone coverage is improving.

The prevalence of basic phones allows for the implementation of important features like requesting access to identity data directly from the beneficiary who can respond to the request. However, **since it is not possible to store encryption keys on basic phones, the signing of the response has to be done by another entity**.

This entity that performs cryptographic operations on the beneficiary's behalf is called the Guardian.

Decoupling programme management and identity management

Many aid organisations have existing systems in place to handle enrollment and management of beneficiary data. Sometimes, these systems can be quite basic, with enrollment being done using a data collection tool like ODK and beneficiary data being managed in Excel sheets.

There are also very sophisticated tools that comprise a whole product suite. One such tool is Red Rose, which is used by the KRCS. Red Rose is a suite of tools and includes an enrollment mobile application for beneficiary registration, a web interface that allows to review beneficiary data, assign beneficiaries to programmes and aid (such

as cash-based aid) to beneficiaries. Red Rose is used by many different organisations worldwide that have enrolled more than 50 million beneficiaries on the platform.

Whatever the exact setup, these systems have not been designed for Identity Access Management as their core purpose.

Therefore, **one fundamental goal of the proposed system is to decouple identity from programme management systems and offer a dedicated service** that grants third-party applications access to beneficiary identity on an as-needed basis.

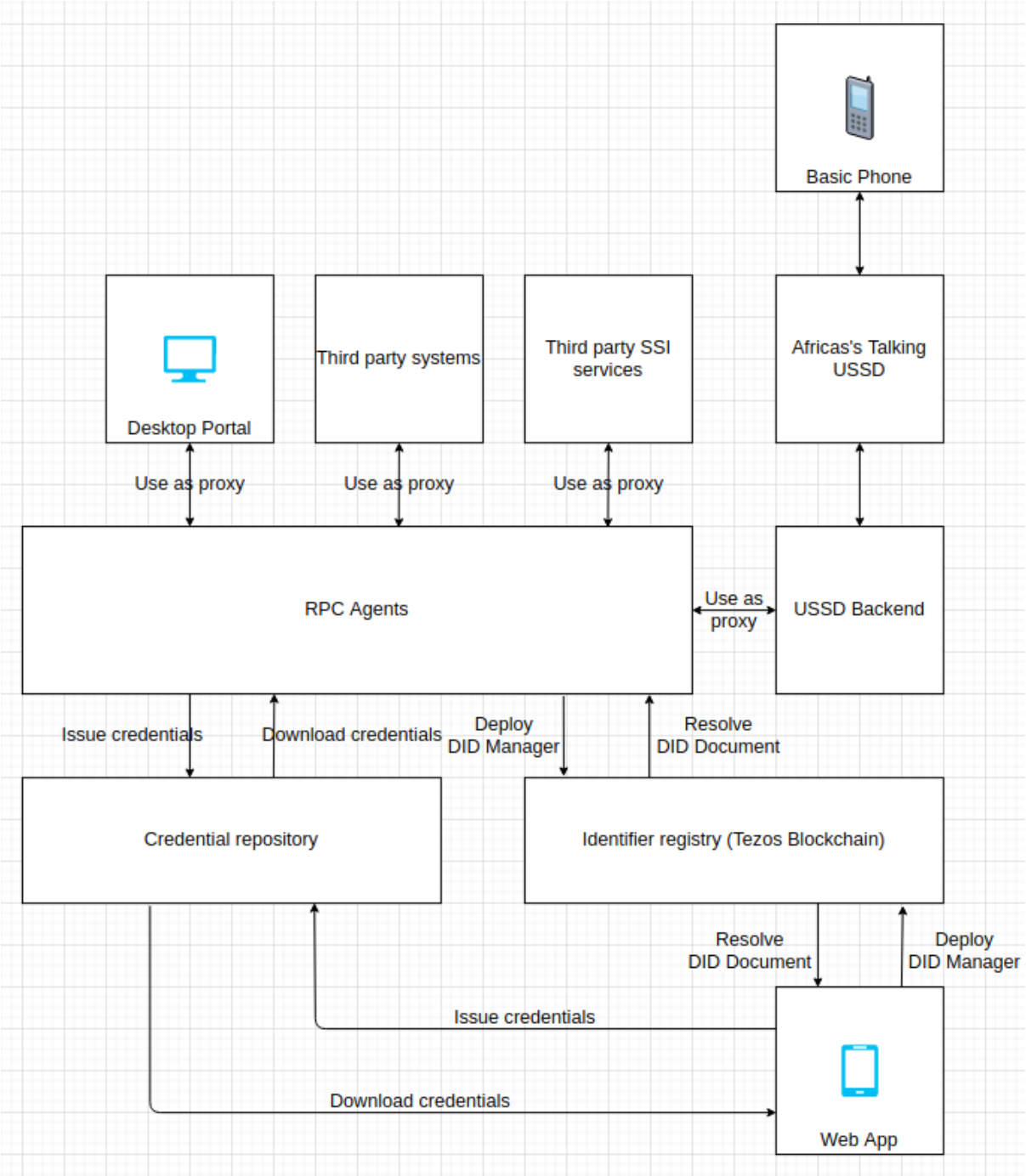
Leveraging existing systems

The DIGID platform is not seeking to replace existing tools for enrollment and programme management.

It is designed to integrate with those systems and leverage their core capabilities, that is enrollment, programme management, cash disbursement etc. This ensures that existing systems can remain to be used for their primary purpose and are leveraged as data sources for trusted credentials. It also **eases the adoption of decentralized identity by being less disruptive**.

3. Technical solution

High-level architecture



Ecosystem Components

Component	Description	Status
Backend Layer		
Core Package	The Gravity Core Package is a javascript implementation of the W3C DecentralizedIdentity and VerifiableCredential standards. This allows the creation of very low level components including W3C VerifiableCredentials, VerifiablePresentations and DID Documents, and provides cryptographic encryption and signature algorithms.	Open Source
RPC Agent	The agent is a server which can be used outside the DIGID ecosystem. It could be set up on customer infrastructure, or used by a guardian. This server exposes an API reachable via the public network (https requests). Each party able to host such an agent is then able to remotely, in a secure and close environment, perform high level operations including wallet creation, encrypted credential issuance and sharing, credential management and account recovery.	Open source
Credential Repository	Beneficiaries' credentials are stored on a decentralized credential repository. The credentials, which are encrypted by the issuers for the subjects, are split up and stored across different nodes. Multiple trusted entities can participate in storage. It is necessary to have secure cloud storage since the solution does rely on web applications instead of native applications and Guardianship also requires secure cloud storage.	Proprietary
Software Development Kit (SDK)	The SDK is a library developed by Gravity, running on the background of the PWA or the Gravity agent. It uses the core package in order to create low level objects and orchestrates the communications between the other services, such as the PWA, the RPC Agent, Verifiable Data Registry (Tezos Blockchain) and the Credential Repository. It is also responsible for managing the keys used for the decentralization authentication process.	Proprietary
Frontend Layer		
USSD Application	Basic phone users will interact with the solution via a USSD menu interface which relies on a PIN code for authentication. Basic phone users will therefore be able to perform consent & share, update & delete, view & verify operations as per requirements.	Open Source
Progressive WebApp	The App is accessible via smartphone and web browser. Smartphone users have the ability to encrypt and sign	Proprietary

(PWA)	credentials and consents directly on their phone. The App retrieves the received credentials from the Gravity credential repository. It receives presentation requests directly from relying parties, resulting in the credential being shared on to the requester's RPC Agent.	
Portal	An existing BMS can leverage the Gravity Core Package to build, sign, and encrypt credentials and send them directly to a beneficiary's wallet. However, in some cases this kind of integration might not be possible due to technical limitations. Therefore, such a portal needs to have direct connection with a remote RPC Agent.	Open Source
Third party software		
Africa's Talking USSD	The Africa's Talking API is used for the USSD menu to allow users with basic/feature phones to access the platform.	Open Source

DIGID Process Flows

User Types

User Type	Actions	Notes
Beneficiary (Smartphone, Basic Phone, No Phone)*	• Create digital identity • Manage digital identity • Share data/Authenticate using digital identity	Beneficiaries with smartphones can create their own digital identity by signing up to the platform themselves.
NGO/ Humanitarian organisation	• Issue credentials • Request credentials • Create and manage digital identity on behalf of beneficiary	NGOs serve as issuers of data which is requested by organisations such as FSPs. However, they can also request credentials. For example NGO A can request credentials issued by NGO B.
Financial Service Provider (FSP)	• Request credentials • Issue credentials	Both money vendors and mobile money/digital cash providers can use the DIGID Platform, making it suitable for different scenarios. FSPs are currently envisioned as requesting data to authenticate beneficiaries for cash disbursement.

		However, they may also issue credentials, for example confirming cash disbursement to a certain beneficiary.

Since device possession may vary among beneficiaries of aid, certain process flows may be different for smartphone users in comparison to those without a device. For simplicity it is useful to **distinguish between 3 user types** based on the type of device they possess:

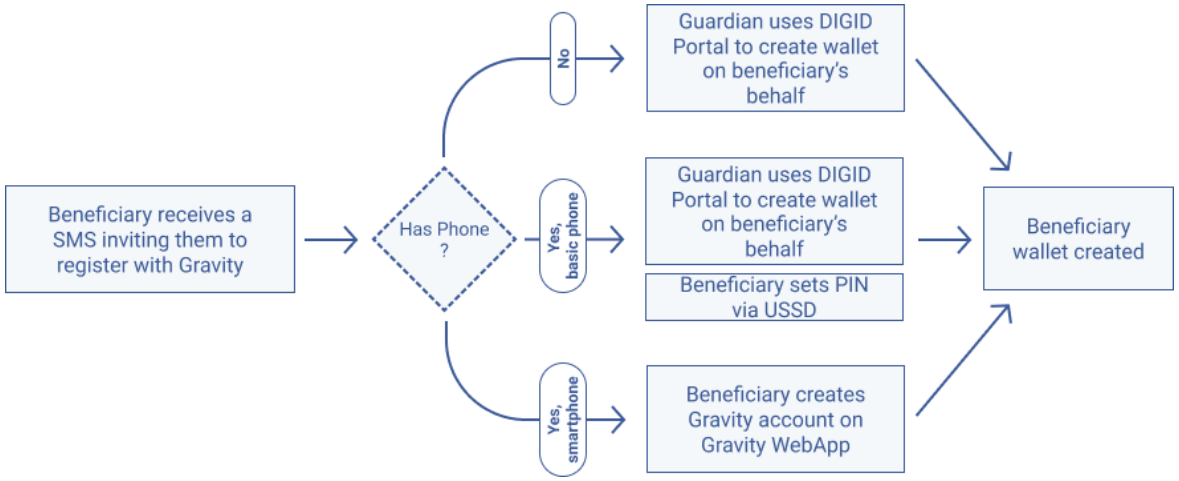
1. **Smartphone** (self-sovereign)
2. **Basic/feature phone** (guarded)
3. **No phone** (guarded)

According to DIGID functional requirements, all 3 user types should be able to interact with the digital identity system.

The following section outlines the main flows of interaction with the DIGID digital identity system.

1. Create a wallet

The wallet creation flow is independent of the programme registration. It can be performed at the same time as programme registration or at a later stage.



The diagram above depicts the wallet creation flow as follows:

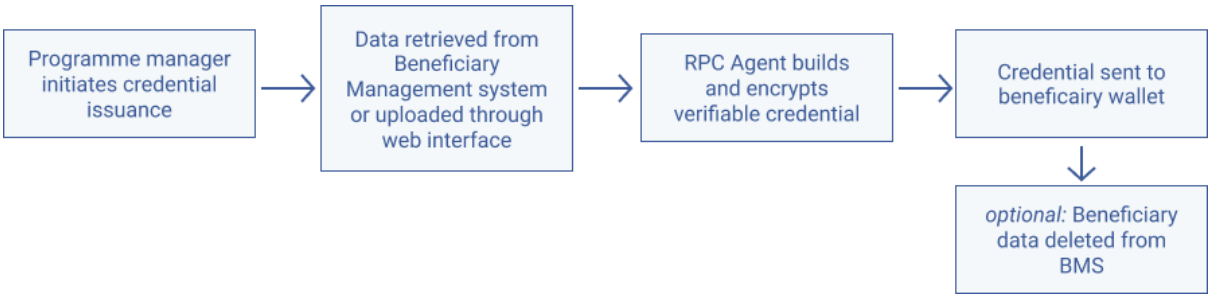
a. **Smartphone users:** If the beneficiary has a smartphone, they go through the account creation flow on the Gravity WebApp.

b. **Basic phone users:** A guardian entity can create a wallet on the users' behalf via the Gravity data sharing portal (or other dedicated interface). The beneficiary can then set their PIN via the USSD menu on their phone.

In both cases it is important that beneficiaries provide a number that has previously been registered by the humanitarian organisation and belongs to them.

c. **No phone:** If the beneficiary does not have any phone and/or is not able to access the internet, a guardian entity can create a wallet on the beneficiaries behalf using the Gravity data sharing portal or another dedicated app.

2. Issue Credential



Once wallets are created, organisations can issue credentials using programme registration data. Depending on the service used, the credential issuance can be done through one of the following methods:

- **Triggered from within a BMS (e.g. RedRose).** This is the case for the DIGID pilot, during which credential issuance is triggered from within Red Rose after data collection. Organizations using Red Rose can therefore opt for this method to avoid disrupting existing data collection and enrollment flows. This can also be done through other BMS in use by organizations, or
- **Issued via the Gravity Portal** using a simple CSV upload. The Portal is accessible via a web browser, or
- **Upload manually to the RPC Agent of the issuer through a web interface.** In this case, a deeper integration is required and will depend on the system provider to integrate that Agent.

The issuer's RPC Agent receives the relevant data fields from the third-party system. The SDK embedded within then builds, encrypts and signs the credentials. Next, the service sends the credentials to the beneficiaries secure cloud wallet on the decentralized storage system.

Users without device

If the beneficiary does not have any phone, the Guardian can create and download QR codes that contain claims and the digital signature of the issuers. The beneficiaries can then be provided with a **paper-based identity card that contains a QR code and an ID**

number that is assigned by the issuing organisation. Verifiable digital credentials are encoded in the QR code and can be read using the Gravity WebApp.

Putting it together: Wallet creation and credential issuance in practice

As soon as beneficiaries have completed the *Create Wallet* flow, credentials can be issued to their wallet using the *Issue credential* flow. This is the scenario most likely to be adopted for the DIGID pilot.

During the registration process, beneficiaries are shown how to create a wallet if they have a device, otherwise it is explained to them that they will receive a paper-based credential at the time of the first disbursement.

The beneficiaries then complete the *Create wallet* flow. They are also registered using the existing registration tool (RedRose Collect in the pilot scenario).

There is no need to create dedicated enrollment software for this project since the DIGID platform is agnostic as to the source of the credentials. They all come from an issuer.

Once the data is synchronized in RedRose, the data will be sent to the issuer's RPC Agent hosted on a KRCS controlled server. That Agent uses the data to build credentials, sign them, encrypt them, and send them to the beneficiary's wallet.

Beneficiaries can then view their credentials as follows:

- **Smartphone users:** Beneficiaires with smartphones can view the credentials using the beneficiary WebApp.
- **Basic/Feature phone users:** Beneficiaries with a basic phone can view their credentials by requesting to view them via the USSD menu. They will then receive an SMS containing an overview of their credentials.
- **No phone users:** As for the beneficiaries without a device, their Guardian can use a WebApp to generate QR codes containing a certain set of credentials along with signatures. They can print those and hand them over to the beneficiaries of whom they are the Guardian.
    - **In the pilot scenario, the Guardian will be KRCS,** who will then be able to hand laminated paper cards with a QR code and system generated beneficiary identity. These cards could be distributed in a similar fashion as smart cards are usually distributed during the first distribution.

3. Request and share credential

When an organisation requires access to a beneficiary's data, it sends a data sharing request. How this request is received and processed depends on whether the user is a self-sovereign smartphone user or a user under Guardianship with a basic phone or no device.

a. **Basic phone users:** Under Guardianship, a beneficiary with a basic or feature phone receives a data sharing request by SMS. They dial the dedicated USSD shortcode, enter their PIN and give their consent to share data with the relying

party. The USSD backend confirms the data sharing request to the Guardian service, which builds, signs and encrypts the presentation.

b. **Smartphone users:** In the self-sovereign case, the beneficiary receives a data share request by SMS that contains a link. The link takes them directly to the notification center where they can click to view the request including which data needs to be shared and for which purpose. The beneficiary provides consent by providing a PIN. The PIN unlocks the keys on the beneficiary's phone that are used to sign and encrypt the presentation that contains the credentials as well as metadata specifying the purpose and expiry date of data access rights.

In both cases, the relying party receives this data on their RPC Agent. The presentation mentioned above functions as an **access right "token" that can be used for auditability.**

c. **No phone users:** If the beneficiary has only a paper-based QR code, data requests can only be done in-person. In this case, the beneficiary shows the QR code to a verifier. The verifier uses the WebApp to scan the QR code and extract the credentials as well as the signatures. An additional authentication layer, such as a PIN code or biometric, can also be used to ensure that the QR code does indeed belong to that beneficiary.

Putting it together: Sharing credentials for cash distribution

The way the flow is implemented in practice may vary substantially depending on the distribution method.

Mobile money distribution

In the case of mobile money distribution, we can set aside the beneficiaries without the device. In this case, the data requests are done remote, following the flow outlined above.

The data is received on the RPC Agent of the requesting party, from where it can be retrieved to be fed into a BMS or used otherwise - if compliant with the *purpose* specified in the metadata of the access right token.

Physical cash distribution (Money Vendor)

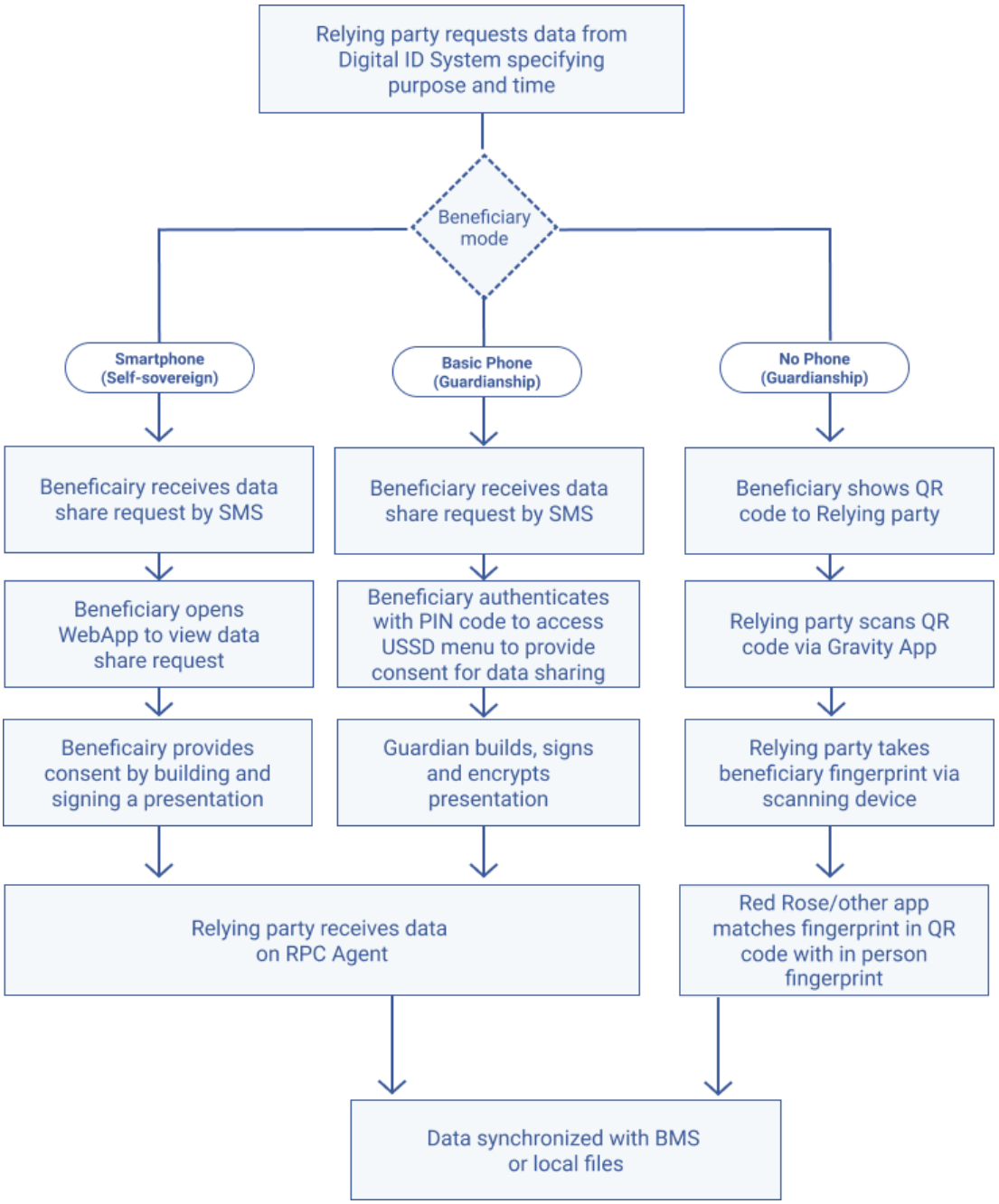
In the case of physical cash distribution,

- **No phone users:** Beneficiaries without a phone share their data through QR code scan as outlined above. The verifier can then use that data and input it into the data collection/beneficiary management software being used to trigger payouts via mobile money.
- **Smartphone users**: Beneficiaries with a smartphone can also create a QR code on the WebApp to get it scanned.
- **Basic phone users:** As for basic phone users, there are two options:

i. Receive QR codes that their Guardians generated for them;

ii. Alternatively, beneficiaries receive a data sharing request through SMS and consent to it through the USSD menu. The verifier/relying party then receives the presentation containing credentials on their RPC Agent. The field staff needs internet access to download credentials that the beneficiary shares through their basic phone on the organizations' RPC Agent.

The diagram below shows a sample flow for requesting and sharing credentials.



Using DIGID credentials for registration with another organisation

If an organisation registers beneficiaries, some of whom already have a DIGID wallet, the digital identity can be treated similarly to other “alternative identity documents” such as school certificates etc.

Sharing data would happen as described in the physical distribution scenario. For beneficiaries with basic and smartphones the organisation has the choice between

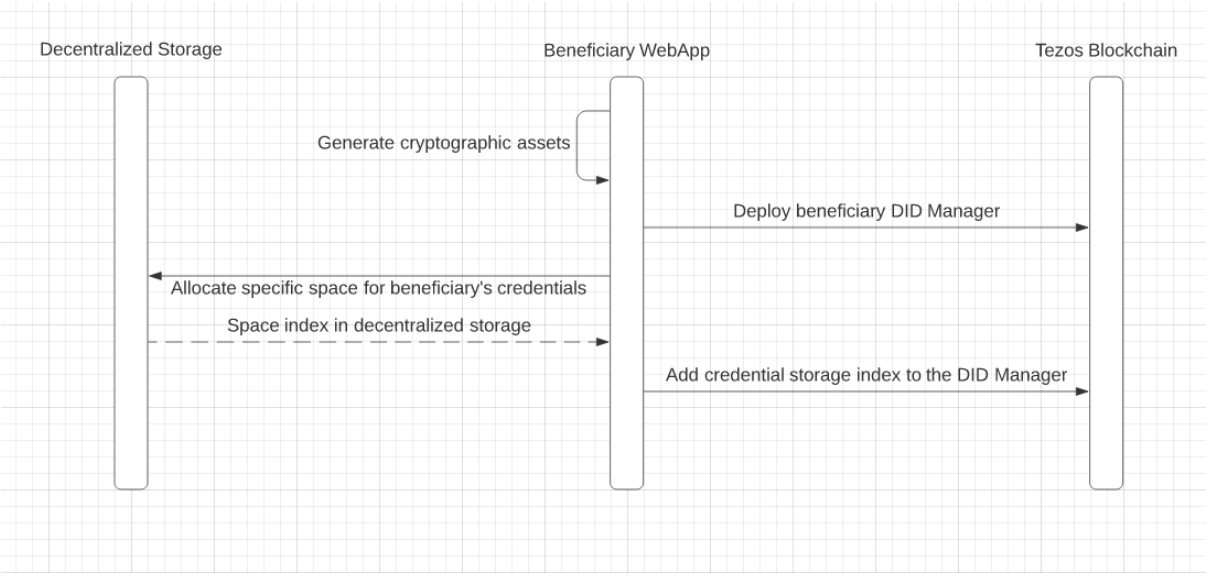
a. Reading data from QR codes and subsequently inputting it into the data collection tool being used

b. Only reference the DIGID identifier in the data collection tool and later retrieve the credentials via the RPC Agent.

Sequence Diagrams

Create a wallet

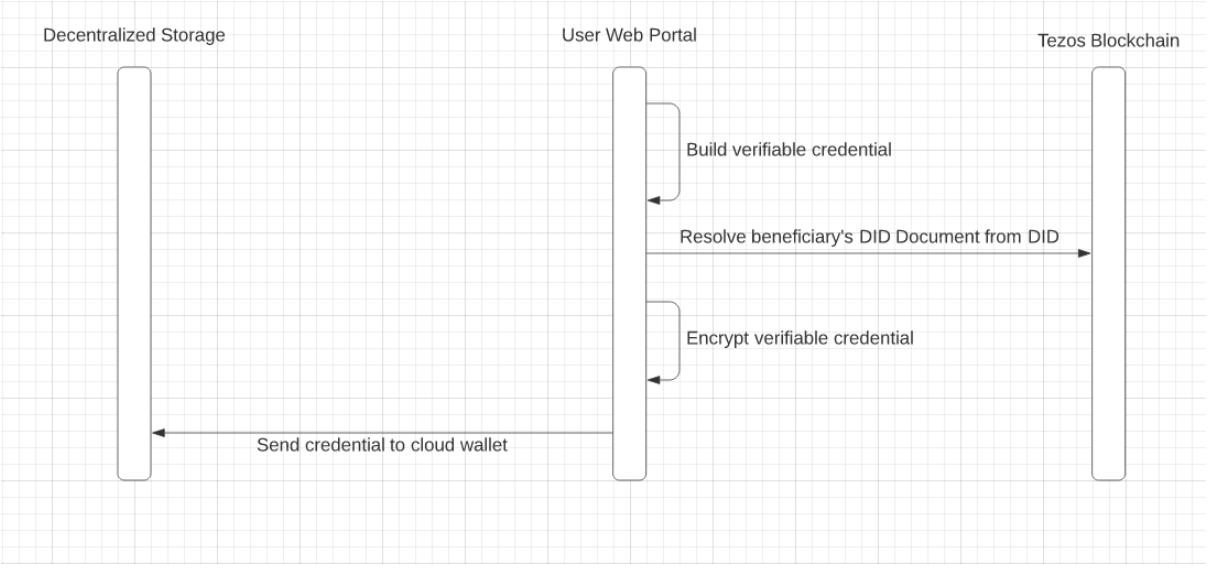
A beneficiary uses a smartphone to publish a DID Document on the Tezos Blockchain



Beneficiaries generate cryptographic assets directly from their smartphone. They can then initialize their DID manager, i.e. wallet manager, on the Tezos Blockchain and add a credential service to help issuers find their dedicated space on the decentralized storage.

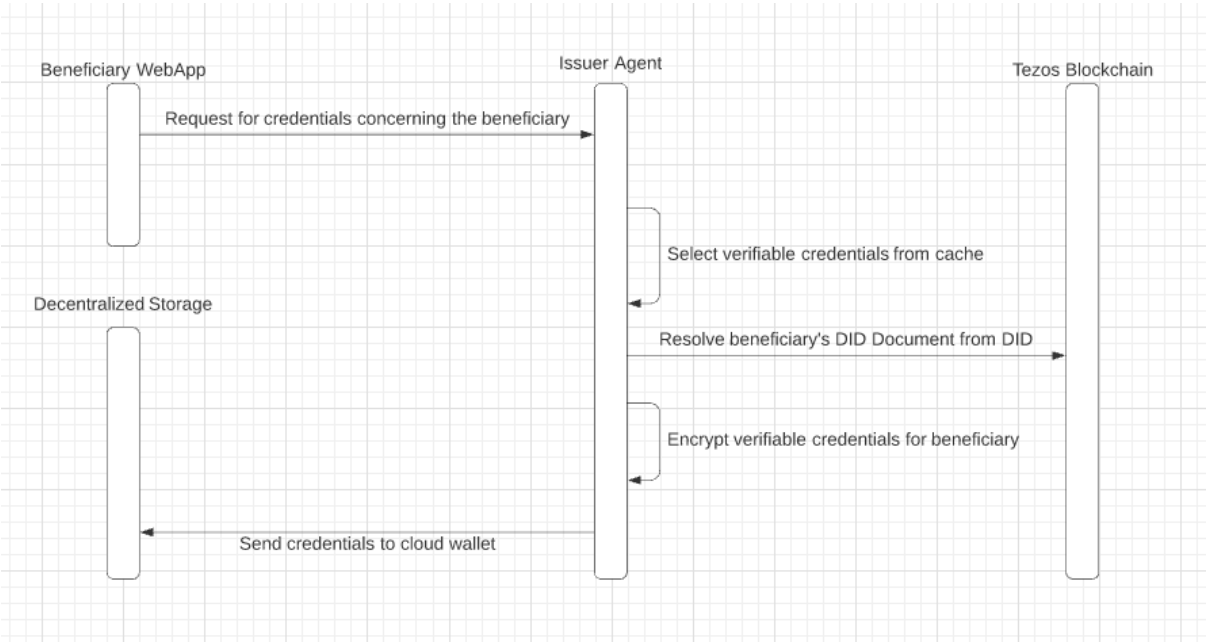
Issuing a credential

An issuer uses a web portal to issue a credential on a beneficiary



Issuers build credentials offline and download the DID Document beneficiaries. From there they extract cryptographic keys to perform end-to-end encryption and indices specifying beneficiaries' cloud wallets on the decentralized storage.

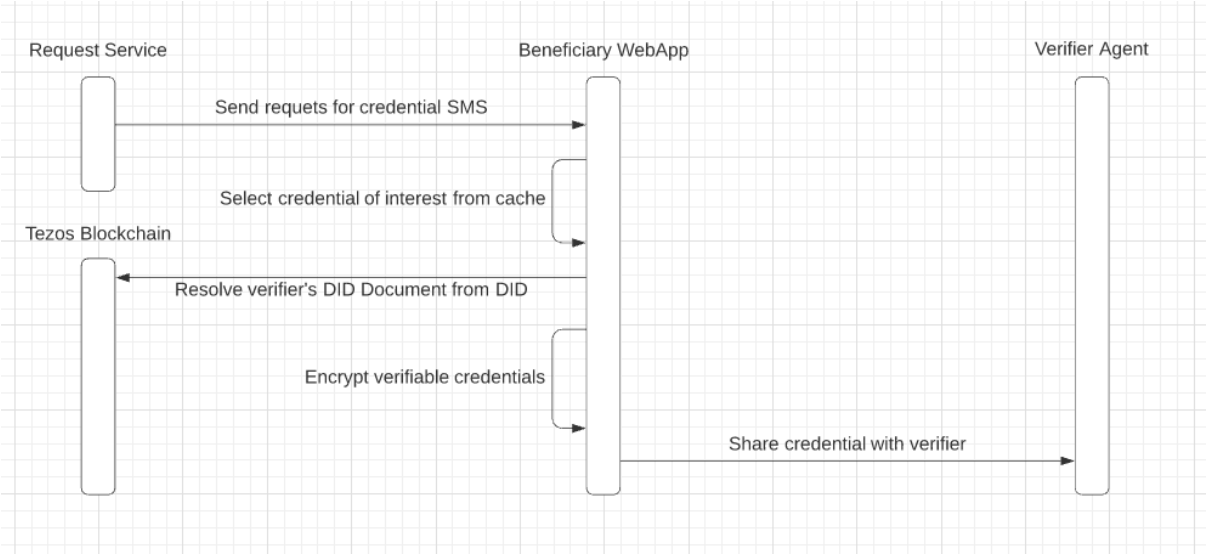
A beneficiary uses a smartphone to request an issuer to issue credentials



As an alternative, beneficiaries can request issuers to issue the credentials which concern them. This is done by having beneficiaries identifying themselves with issuers that enter the classic flow described above

Share a credential

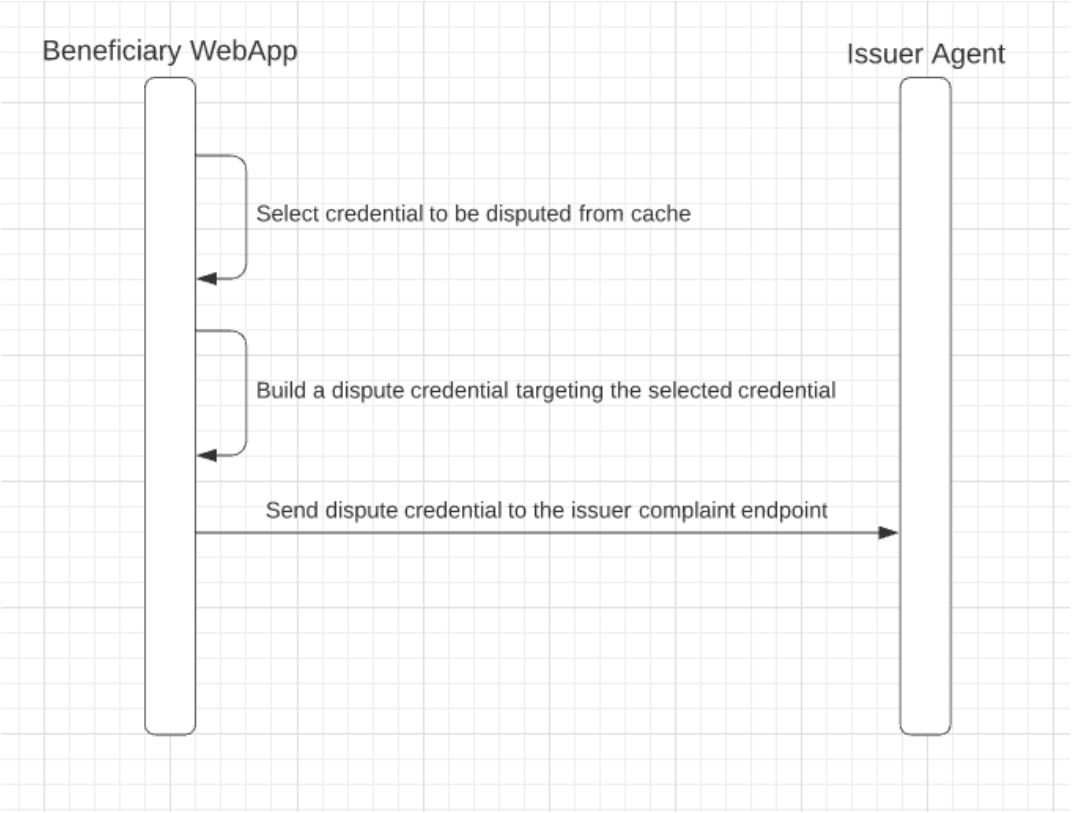
A beneficiary uses a smartphone to share a set of credentials



Beneficiaries receive SMSs requesting for credentials, reason of sharing and details about the verifier. They download related DID Documents to perform end-to-end encryption and send the result directly to the verifier Agent

Dispute a credential

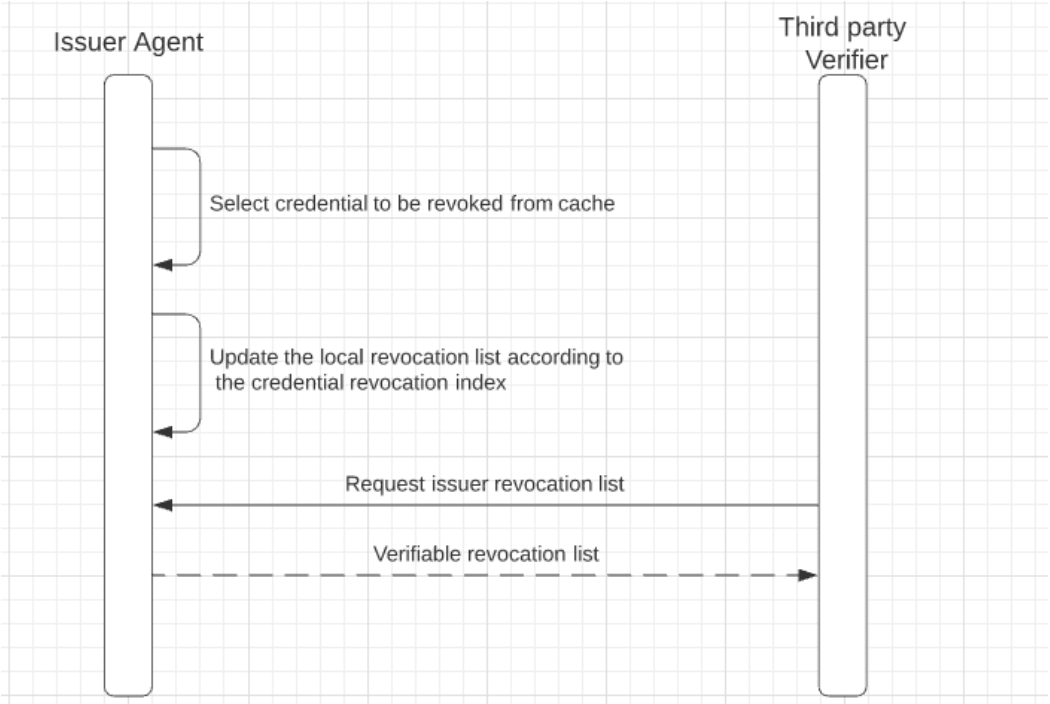
A beneficiary uses a smartphone to dispute a credential



Beneficiaries could dispute a wrong or outdated credential at any time. After retrieving the credential to dispute, they create the said dispute credential that must have a link to the credential it disputes. Beneficiaries then directly send it to the related issuer issuing service complaint endpoint. Disputing a credential is a crucial first step in allowing beneficiaries to request updates to that credential, depending on the issuer's policies and program requirements (see note).

Revoke a credential

An issuer revokes a previously issued credential



Issuers can revoke credentials they did previously issue (following a dispute for example). A revocation is possible using the revocation list standard. This involves that before issuing a credential, issuers must include a unique index (statically maintained and incremented by the issuer) that corresponds to its revocation status on the binary list. Next step is to publish this list so it is resolvable by any verifier who wants to check the revocation status of a requested credential. This technique has low computational cost and does not reveal the relation type between issuers and beneficiaries.

To update a credential, issuers first revoke the credential disputed by a beneficiary (see note).

Note: Updating Credentials

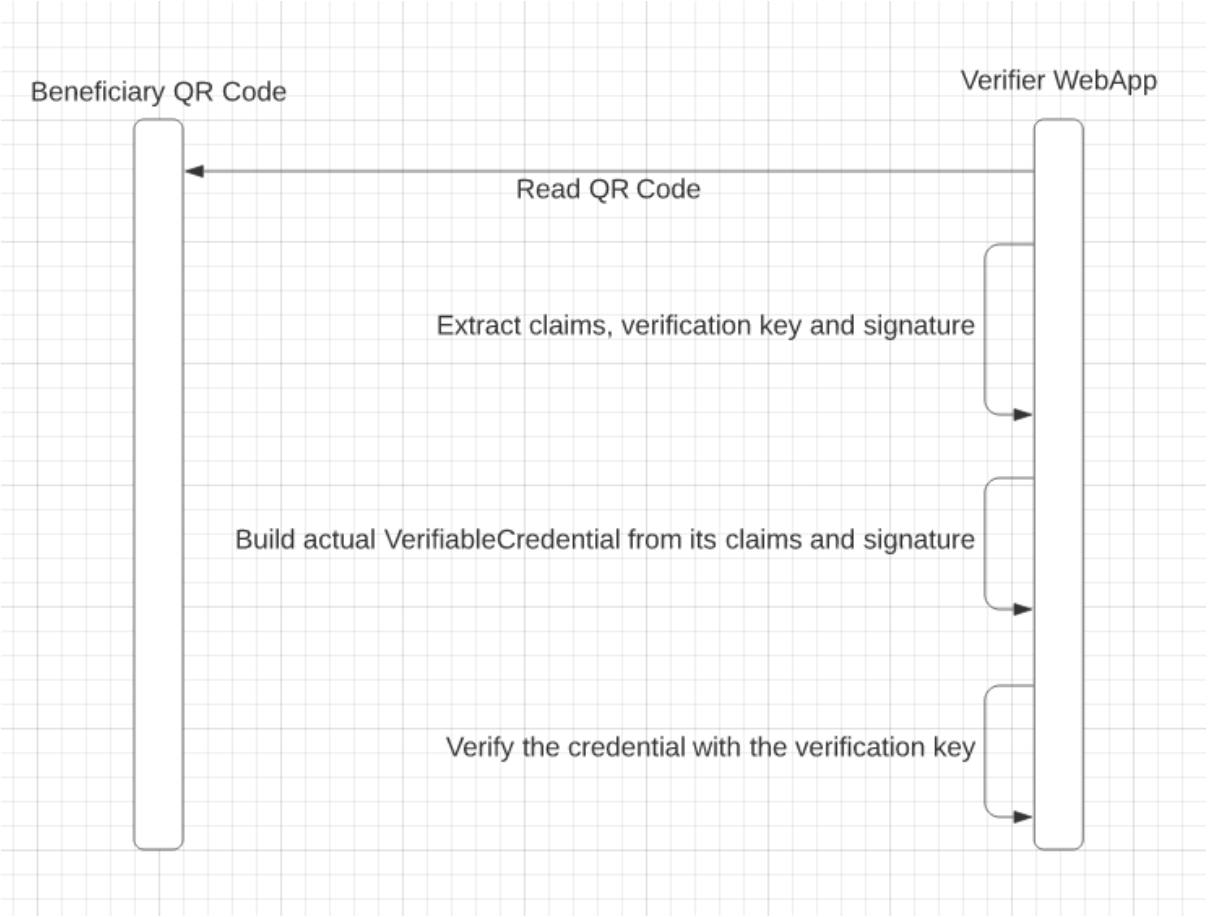
Beneficiaries may request updates to outdated or inaccurate credentials by first disputing that credential.

Issuers can then revoke that credential and issue a new credential that reflects updated beneficiary information accurately.

However, the exact process for updating credentials may depend on individual organizations' policies and program requirements, such as data accuracy and updating frequency.

Verify a credential on a QR Code

A verifier uses a smartphone to verify a credential embedded in a QR Code



Verifiers are able from the WebApp to read QR Codes that include raw claims. Those claims can, along with a cryptographic signature, be turned into actual credentials that are easily verifiable thanks to the verification key also encoded in the QR Codes.

API Documentation

The API documentation of the Agent can be accessed at https://docs.gravity.earth/agent**.** They outline which data fields are consumed by the service

Additional Notes

Choice of an USSD Menu as basic phone interface

Building an interface for basic phone users is challenging. The most common interface for basic phones is SMS messaging or USSD. For digital identity, SMS is not suitable, since it lacks an authentication mechanism.

USSD is a potential solution, since it allows to perform authentication using a PIN. In this case, if a beneficiary receives a consent request for instance, they would type a USSD shortcode. This would trigger a short menu that asks them to provide consent to the requesting organisation for a specific set of data. At the end of the menu, they would confirm using their PIN, similar to an MPESA transaction. This is a feasible option that is straightforward to implement and should therefore be considered the fallback option.

However, the use of a PIN creates more challenges for the typical beneficiary population. According to accounts from KRCS staff, people in rural areas as well as elderly and disadvantaged people are not familiar with the concept of a PIN. When they use MPESA they rely on the Safaricom agent who is part of their community. This agent will simply use the year of birth as their PIN so when the user comes with their ID card they know what to do.

Another challenge with USSD is the use of written language. At least in Kenya, many people don't learn to read their tribal language. And they will only be able to understand written Swahili or English if they went to school for a significant time. This is also one of the main learnings from the user consultation.

While initially an Interactive Voice Response (IVR) menu was in consideration as the choice of interface for users with basic/feature phones, this was not implemented for the DIGID pilot. This is because the introduction of the IVR menu required additional time for user consultations, prototyping and a full analysis of the data protection and privacy risks before testing with real beneficiaries of assistance. Given the DIGID project's focus on interoperability between different NGOs, an additional consideration was that voice authentication may not be a widely used authentication mechanism by other NGOs.

For these reasons, USSD was the interface chosen to be tested for the DIGID pilot.

Addition of Interactive Voice Response (IVR) as interface for basic/feature phones

Keeping in mind the challenges described above, an IVR menu may be a suitable interface to allow users to authenticate easily, without compromising on security. IVR's have recently been used in Somalia by various NGOs in cash transfer disbursement (see video here). In Kenya, Africa's talking provides APIs for Voice menus.

The flow should be kept as simple as possible, and beneficiaries authenticate by using the same phrase every time: "My voice is my password".

The IVR menu will become available on the DIGID Platform in the upcoming months.

QR Codes for offline environments

Identification in long-term cash programmes

Many cash transfer programmes run over several months or sometimes longer with regular distribution on a monthly or bi-monthly basis. **Everytime there is a distribution happening, beneficiaries have to re-identify.** This is especially inefficient in contexts

where people lack an official national identity and have been registered using alternative identity documents like a letter from a community leader or else.

Smartcards are a frequently used solution in long-term programmes. They provide a secure reliable identity for the duration of the programme. In addition they allow to securely store biometrics that can be used for authentication offline, or even on the card. **The main downside of smartcards is their price,** often around $2 per beneficiary, which can quickly add up to a significant amount.

Reusable humanitarian identity

In a crisis setting, be it a slow onset or an emergency scenario, different organisations provide aid. This could be due to a lack of sufficient coordination at a country level, or simply because they provide for different needs. **All of these organisations have to identify the same beneficiaires** and collect and identity and programme relevant data.

A humanitarian digital identity, based on credentials issued by the organisations could greatly improve the efficiency of the overall operation avoiding duplications and maximizing reach of a limited aid budget.

It would also increase the dignity of the beneficiaries, reducing the number of times they have to go through an identification exercise, and providing them with a token and a wallet that they can share with organisations on a per need basis.

Interoperability

An important argument in favour of decentralized identity is interoperability. Interoperability becomes relevant in a scenario where Alice, who has a decentralized identity wallet based on protocol A can share credentials with Bob, who might use a wallet based on protocol B. With full interoperability, Bob can easily decrypt, read, and verify Alice's credentials.

Why is interoperability important?

Interoperability is important because it reduces the risk of vendor lock-in and siloed solutions. The whole idea of decentralized identity for humanitarian aid is that Organisation A can issue credentials to a beneficiary who can then easily share it with Organisation B. Interoperability allows Organisation A and B to use different decentralized identity protocols for issuance and verification.

The current state of interoperability

Currently, almost all decentralized identity solutions follow a certain data standard called verifiable credentials. While this is an important piece, there is still some way to go to achieve full interoperability between protocols. For instance, protocols have different ways of creating connections and exchanging messages and credentials, public keys and schemas are stored on different ledgers and use different verification algorithms.

All these issues are being addressed in different working groups in the World Wide Web Consortium and the Decentralized Identity Foundation but it will presumably take several years to achieve interoperability.

What can we do today?

Still, it is important to ensure that any solutions that are being built for long-term (like the DIGID platform) are as interoperable as possible. While the DIGID project mainly leverages Gravity's identity stack and beneficiaries will be creating Gravity wallets during the pilot, we will **ensure that services that are at the intersection between the identity platform and third-parties are also able to issue and read credentials from other SSI platforms. As part of this pilot, we will integrate these services with TYKN's Ana platform which is built on Sovrin.** Ana is **fully interoperable with the 121 identity platform** which was also built by TYKN. Other services can be integrated as well but specific modules have to be built.

Interoperability test for DIGID

To prove the interoperability, a credential will be issued from Red Rose to a wallet based on the Ana platform as part of a controlled exercise. The code and output from the exercise will be provided as part of the implementation deliverables

Privacy & security

Data storage

The credential repository is a decentralized storage vault that stores beneficiaries' credentials. This is a maximally confidential storage protocol for the following reasons:

Authentication

Nodes on the repository come with a cryptographic authentication mechanism using the same verification keys stored on-chain. This way, the authentication to access this vault storage is as strong as the native Blockchain authentication.

Encryption

The repository only deals with encrypted data. This is the role of the issuer (resp. data subject) software to perform data encryption (resp. decryption), resulting in an end-to-end encryption.

In addition to that, the underlying technology itself comes with a native encryption layer so even a malicious nodes of the repository can't read the data stored

Redundant Array of Independent Disks (RAID)

The internal storing mechanism, namely RAID, splits the encrypted data across the multiple nodes and ensures redundancy. This way, even in the unlikely case of several nodes getting attacked, damaged or lost, users' information remains safe. Lastly, a reparation protocol is executable in order to reconstitute data.

This splitting algorithm along with the native encryption guarantees that nodes part of the credential repository only possess a few pieces of encrypted information and therefore are unable to read the data itself

Location of data

In order to store sensitive data from the beneficiaries, the **Credential Repository should ideally avoid** international transfers (as recommended by the DIGID Data Protection Impact Assessment). To do so, a private cloud decentralized storage is used and its belonging nodes exclusively hosted in Kenya for the DIGID project, in compliance with the Kenya Data Protection Regulation. **In future, organizations may continue to use the storage nodes in Kenya or decide to have one in their country of operation.**

Because of the underlying technology being public Blockchain, the **Verifiable Data Registry** has storage nodes across the world and therefore international transfers can not be avoided. However, this does not concern personal data as mentioned in below the section "Note on personal data storage and the blockchain".

Note on personal data storage and the blockchain

The following data is stored on the blockchain:

- **Authentication public key:** Used by remote parties verify a signature with an "authentication" purpose (in W3C terms), most frequently used during the sharing of credentials,
- **AssertionMethod public key:** Also for use by remote parties to verify signatures with an "assertionMethod" purpose (in W3C terms), usually used during credential issuance,
- **KeyAgreement public key:** Used by remote parties to compute the shared secret necessary for end to end encryption,
- **Link to the credential repository:** A link to a dedicated space on the credential repository which allows issuers to know where to send credentials post encryption, and
- **TZIP-16:** Standard that helps attach off-chain metadata to the DID manager, allowing for the inclusion of metadata views.

This means that no personal data/personally identifiable information is stored on the blockchain, upholding beneficiaries' Right to be Forgotten and safeguarding their privacy.

Data retention

Data is retained both in the decentralized identity system on users' digital identity wallets and on third parties' respective systems.

The data retention period for data within the decentralized identity system is **set and agreed upon between the implementing parties** (i.e. the vendor and individual third party organizations). This can be done by adding **maximum validity dates** in the metadata, after which it is deleted from the platform.

Data shared with third parties is stored on the RPC Agent hosted by the party itself. In this case, the **data retention period depends on the respective policies of the individual organizations/third parties**. Retention or deletion of data will also depend on third parties' respect of the date of expiration of consent objects.

Data protection principles

Our platform was built in line with the principles of **privacy-by-design** and **data minimisation** and is fully **compliant with the GDPR.** In particular, our platform is based on the following principles to ensure data privacy & protection:

Data minimisation: Only data for which machine-readable consent is given can be accessed by a relying party. Consents have an expiry date set by the user after which data is no longer accessible.

Zero-Knowledge Proofs: The solution allows users to share proofs with relying parties without the need to actually demonstrate the entirety of the information. This is particularly important for the humanitarian context given the sensitive nature of data regarding displaced persons.

Full transparency and visibility: Data subjects have full transparency on which kind of data is stored on the system, and the ability to request to view it at any time either via the WebApp or USSD menu.

Purpose limitation: Every consent request states a specific and explicit purpose of the data. The consent validity period is adjusted according to the needs for that purpose.

Storage limitation: Data is stored-exclusively on users' self-sovereign wallets. No other entity, including Gravity, is able to access the users' data. They are therefore in full control. Maximum validity dates can be added in the metadata after which it is deleted from the platform.

End-to-end encryption: Data is encrypted end-to-end using a shared secret encryption key between sender and receiver. Since it is end-to-end, the encryption / decryption is performed on client sides.

Despite the provisions for data protection and privacy set out above, the **USSD protocol has limitations** in this regard. Data channeled through the USSD channel is by default non-encrypted. This means that **breaking the GSM encryption could leave data vulnerable to leaks**. Additional protocols to mitigate this risk must be explored. However, this remains a limitation given that the USSD menu is a functional requirement within the scope of the DIGID project.

Data deletion

Beneficiaries can exercise their Right to be Forgotten at any time by requesting the deletion of their digital identity wallet through the Web App or USSD menu. Storage nodes within the system are built with an extra layer of server-side lease expiration and client-side lease renewal. Upon a beneficiary requesting the deletion of their wallet, the renewal mechanism stops and every piece of data concerning the client is then Garbage Collected.