iXRPL

A Smart Contract-Powered, Self-Sovereign KYC Solution for the XRP Ledger

by

Scott Chamberlain, Richard Holland, and Ravin Perera

27 July 2021

1. Introducing iXRPL

1.1 We present iXRPL, a proof-of-concept for a decentralised self-sovereign KYC that runs on a global cluster of HotPocket nodes and uses Non-Fungible Tokens issued on the XRP Ledger.

1.2 Through a mobile App connected to the iXRPL smart contract, Users request verification of their identity from an identity verification service against official records. Once verified, the verification is "stamped" on the User's XRP Ledger Account with a Non-Fungible-Token, called a Human UUID, that uniquely identifies the verified individual. The User can then present their verified credentials, cross-checked against the XRP Ledger Account, to financial institutions to satisfy KYC requirements. Effectively, iXRPL "tokenises" the one-off cost of verifying your identity into a reusable asset.

1.3 All information is secured through end-to-end encryption. The Users always remain in full control of their identity documents. The solution is self-sovereign, secured, decentralised, and complies with Australian KYC safe harbour regulations and GDPR privacy requirements.

2. iXRPL's Components

2.1 iXRPL has the following components:

(a) **The User Mobile App:** The User uploads their identity documents and XRPL account information via the mobile app to get verified by iXRPL. The app also maintains the rolling encryption keys for the User's personal documents. When the User wishes to, they can grant access to their digital identity to a third-party service by scanning a QR code.

(b) **RoboNotary:** The RoboNotary is a software agent responsible for verifying the validity and truthfulness of the User's personal documents with the aid of one-time decryption keys sent by the User. Upon verification, the RoboNotary "stamps" the User's XRP Ledger account with the iXRPL digital identity called "Human UUID".

(c) **iXRPL Smart Contract:** iXRPL's service is implemented as a decentralized HotPocket smart contract. It is responsible for storing the User encrypted documents, performing encryption key roll over, and serving document access requests from notaries and consumer services when approved by the Users.

(d) **Human UUID:** This is a "unique mathematical identifier" assigned to each human being based on legal information contained in their identity documents. The Human UUID is generated by the RoboNotary which verifies the personal identity documents submitted by the User. Once assigned, it is immutable.

(e) **XRPL Account Stamping:** Upon generating a Human UUID for a User, the RoboNotary will also "stamp" the User's XRPL account with the generated Human UUID. This will be publicly visible to anyone and can be considered as a verification seal issued to the respective XRPL account by the RoboNotary. The Trust Lines feature provided by XRPL is used for this feature in iXRPL

(f) **End-to-End Encryption:** iXRPL uses strict encryption and key rollover policies to make the personal document storage as secure as possible. The encryption keys always stay with the User, making the personal documents stored in iXRPL inaccessible without the User's explicit consent. Whenever another party (the RoboNotary or a consumer service) needs to access stored documents, they get a one-time decryption key from the User. The moment the documents are accessed, iXRPL re-encrypts the documents with a rollover key rendering already-issued decryption key invalid. iXRPL also ensures to never to keep any encryption keys stored on the smart contract. They are always stored on the User's mobile app.

(g) **Consumer Service Integration:** Any consumer service can integrate with iXRPL and provide an end-user facility like "Login with iXRPL". They can display a QR code that is understandable by the iXRPL mobile app. With the QR code, the consumer service can specify what pieces of identity information they are interested in (e.g. birth year and family name). iXRPL mobile app conveys this information to the User in a friendly manner so the User can choose to only send the decryption keys for that specific set of information allowing for a fine-grained privacy control for the User.

2.2 Together, these components create a Decentralized Public Key Infrastructure ("DPKI") smart contract-based identity solution that puts the individual in control of their KYC'd identity and meets the design principles we established for our project.

3. How It Works

3.1 The high-level flow of iXRPL is as follows:

(a) **Get App:** The Users download the App and register in iXRPL by paying a fee in XRP on the XRP Ledger.

(b) **Upload ID Docs:** The Users upload their identity documents to iXRPL smart contract running on a HotPocket cluster.

(c) **Request Verification:** The Users send their identity documents to the RoboNotary to get them verified.

(d) **Identity Verification:** The RoboNotary uses a professional verification service (RapidID was selected for the pilot because they had a per-use fee model) to verify document records against government databases and a matching selfie.

(e) **Confirm Verification:** Upon verification from RapidID, the RoboNotary issues a signed trust stamp on the User's documents as well as the User's XRPL account trust lines (Human UUID).

(f) **Grant Access to Verified Docs:** The User can then provide access to the documents' details to a financial institution, cross-checked against XRPL account trust line (Human UUID).

(g) **Admit the User to Services:** Upon successful cross-checking, the financial institution allows the User to login to their service.

4. Document Encryption Details

4.1 To ensure our solution is secure and protects privacy of personal information, it employs elaborate encryption protocols as follows:

(a) Uploaded documents are first encrypted on the mobile app, and subsequently on the iXRPL smart contract in such a way that only recipients authorized by the User can decrypt the document contents.

(b) Each of the distinct personal details (Birth certificate, driver's license etc.) are encrypted independently of each other so that the User can grant granular decryption access as needed.

(c) Decryption keys are always kept in the possession of the User (stored locally on the mobile app), so cannot be accessed by the owner of any of the HotPocket nodes on which the iXRPL smart contract runs.

(d) Single use rolling decryption keys are used to ensure that:

(i) Only one authorized third-party can access the documents at any given time; and

(ii) The authorized third-party can access the documents only once.

4.2 The iXRPL mobile app in conjunction with the smart contract also facilitates (but does not mandate) the single-use decryption key transfer using end-to-end encryption.

5. Document upload flow

5.1 To upload an identity document, the following flow is implemented:

(a) The User encrypts the documents locally on the mobile app. (XSalsa20 symmetric key encryption is used in conjunction with a rolling key).

(b) The User uploads the encrypted documents. Upon upload, the iXRPL smart contract re-encrypts uploaded documents with a new rolling key.

(c) The new, contract-generated rolling key is sent back to the User and never stored on the contract.

(d) Decryption keys and rolling keys which can be used to decrypt each document component is signed with the User's Ed25519 private key and stored on mobile device secure storage.

5.2 The set of decryption keys and signatures (called key bundle) produced and stored on mobile device at step 4 are single-use keys.

6. Document verification process

6.1 The User authorizes the RoboNotary to verify their identity documents by sending decryption keys to the RoboNotary. Figure 1 outlines this process.

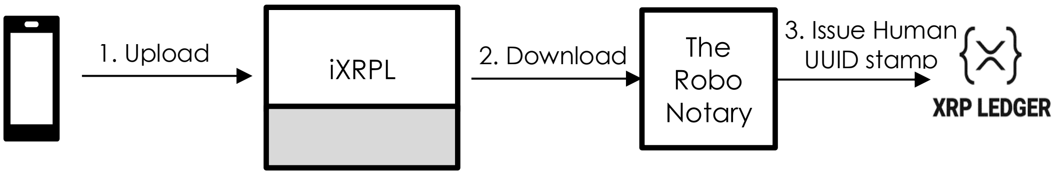

Figure 1-Document Verification Process

6.2 This is achieved as follows:

(a) The User encrypts the set of single-use decryption keys already stored on their device with the RoboNotary's well-known Ed25519 public key. The User can then choose to submit the encrypted key bundle to the smart contract to be delivered to the RoboNotary. The encrypted key bundle can be delivered to the RoboNotary via any unsecure channel.

(b) The RoboNotary decrypts the key bundle with the RoboNotary's Ed25519 private key.

(c) The RoboNotary verifies the ownership of the key bundle using the signatures included by the User inside the key bundle. This step also verifies that the User indeed owns the XRP Ledger account as well.

(d) The RoboNotary then downloads the encrypted documents from the smart contract using authorization token details included in the key bundle. It is important to note that:

(i) Decryption keys are never sent to the smart contract.

(ii) After download completes, the smart contract performs a single-use key rollover and re-encrypts the documents stored in smart contract with the new rolling keys. At this point, the keys currently possessed by the RoboNotary become invalid and hence cannot be used to access the documents ever again.

(iii) New rolling keys are sent to the User who then updates key bundle with new keys.

(e) The RoboNotary now decrypts the downloaded documents using the single-use symmetric keys it received from the User.

(f) It then submits document details to RapidID and gets them decoded and verified and uses the decoded personal data fields (first name, family name etc.) to generate the Human UUID.

(g) Using RapidID, it can perform following cross checks after decrypting the doc bundle:

(i) Extract textual personal data which will make up the Human UUID.

(ii) Validate the extracted text data with the government databases.

(iii) Match the selfie/video to the validated photo ID.

(h) The RoboNotary then prepares the verification proof and the RoboNotary attestation (which includes the Human UUID) and signs it with the RoboNotary's Ed25519 private key.

(i) The RoboNotary then encrypts the proof and attestation with the User's Ed25519 public key.

(j) The RoboNotary then sends encrypted proof and attestation to the User via the iXRPL smart contract (any unsecure channel can be used for this as well).

(k) Upon receipt of the encrypted verification proof and attestation, the User decrypts it with their Ed25519 private key.

(l) The User verifies the verification proof and attestation provided by the RoboNotary by checking the RoboNotary signatures.

(m) The User then encrypts and appends the verification proof to the documents stored on the iXRPL smart contract using similar rolling key upload encryption mechanism as above.

(n) Upon successfully appending the verification proof, the smart contract instructs the RoboNotary to issue the trust stamp containing the Human UUID to the User's XRPL account.

(o) Finally, the RoboNotary issues a "trust line" to the User's XRPL account.

7. Login with iXRPL

7.1 Once their identity has been verified, a User can use the iXRPL mobile app to scan a QR code presented by a third-party financial institution and grant them access to cross-check their XRPL account against the documents verified by the RoboNotary. The process is summarised in Figure 2.

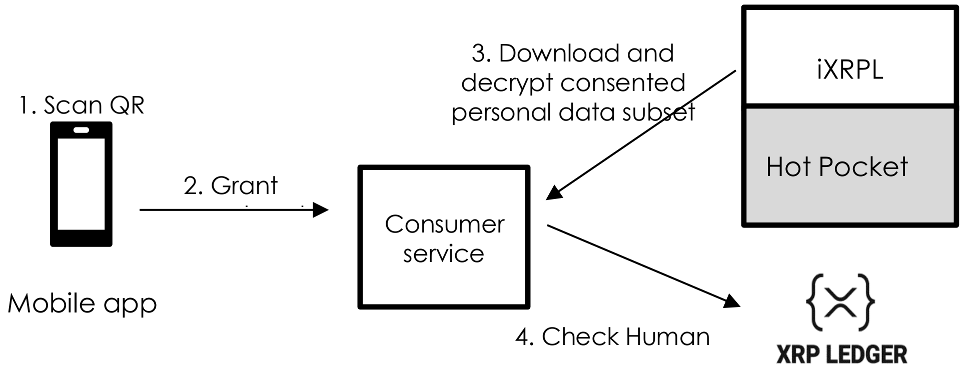

Figure 2-Login With iXRPL Process

7.2 It works as follows:

(a) The User chooses the granular personal data they wish to share with the third party.

(b) The User uploads the selected set of keys to the TLS-secure web endpoint indicated in the QR code on the financial institution's website.

(c) Upon receiving the decryption keys the financial institution will then download the authorized document components from the iXRPL smart contract. Again, it is important to note:

(i) Decryption keys are never sent to the smart contract.

(ii) After download completes, the smart contract performs a single-use key rollover and re-encrypts the documents stored in smart contract with the new rolling keys. At this point, the keys currently possessed by the financial institute becomes invalid and hence cannot be used to access the documents ever again.

(iii) New rolling keys are sent to the User who then updates key bundle with new keys.

7.3 The financial institution then decrypts and inspects the RoboNotary verification signatures contained in the documents against the Human UUID trust line signatures contained in the User's XRPL account.

7.4 Upon successful verification, the User is granted access to the financial institution's services.

8. iXRPL Meets Our Self-KYC Design Principles

8.1 iXRPL meets the design principles we established for our self-KYC solution:

(a) **Cryptographically Secure:** It uses a mix of cryptographic techniques to protect the User data, including a method for hiding private keys from the hosts on which the smart contract runs.

(b) **Self-Sovereign**: Once issued, the Users control their ID Token and their identity documents through their master private key and their one-use private key.

(c) **Appropriately Decentralised:** It uses the HotPocket smart contract platform which has the flexibility to be as centralised or decentralised as desired.

(d) **AML/KYC Compliant:** Identity is verified in a manner compliant with the safe harbour provisions of Australia's KYC regulations.

(e) **GDPR Compliant:** It meets or is capable of meeting GDPR privacy requirements in that all data is encrypted, all keys are controlled by the User, all express User permissions are obtained, and the User can delete their data as desired.

8.2 Points 8.1(c)-8.1(e) are further explained below.

Appropriately Decentralised

8.3 Our proof of concept relies upon a RoboNotary (software agent) to receive requests from Users, submit the documents to an identity service provider for verification, and to issue the Human UUID. This is obviously centralised, a necessary bootstrap unless and until identity service providers interface with the protocol directly.

8.4 Even with this level of centralisation the global HotPocket cluster delivers the benefits of a decentralised network, in two ways. First, it is more reliable and secure than a single server/client model. One or mode nodes can fail, and the service will remain available.

8.5 Second, the source code would be open-source and the RoboNotary is almost completely automated. This allows for a level of transparency and trust-minimisation that would not otherwise be achievable if the entire solution ran on a single server controlled by a single entity.

AML/KYC Compliant

8.6 Our proof of concept assumes the User will upload a passport and the RoboNotary will engage an identity verification service to verify this against the government's databases. Driver's licences are not supported by the proof of concept, but it can readily be enhanced to support them. Passports and driver's licences have standardised formats and so are so easily auto verified.

8.7 It would be preferable to include birth certificates in the list of accepted documents, especially since birth certificates have a unique certificate number and don't expire. However, birth certificates are problematic because they are not usually supported by identity verifiers. Unlike passports and driver's licences, there is no global standardisation for birth certificates. So automated checking of birth certificates requires country-by-country OCR programming.

GDPR Compliance

8.8 Our use of HotPocket and a DPKI structure enables our solution to be GDPR compliant. In particular:

(a) **User Sovereignty and Express Consent:** Every use of the User's identity data is initiated or approved by the User when they upload their documents or when they issue a one-use private key for third parties to access their documents. So, all necessary express consents about how their data is used and stored can be obtained from the User.

(b) **Right to Be Forgotten:** The User can delete their identity data at any time.

(c) **Privacy by Design:** All data is encrypted and cannot be decrypted without private keys the User decides to provide. Even the owner of the host node cannot access the encrypted data.

(d) **Data Minimisation:** The system only collects the data necessary to meet safe harbour provisions of the KYC laws and any keys a User provides to access their data can only be used once.

(e) **Consent to Cross-Border Transfers:** The User can expressly consent to their data being stored/processed overseas. If necessary, HotPocket clusters can be configured so that data from a given jurisdiction is only stored on Nodes that are also within that jurisdiction

(f) **Compliant Agreements:** Binding agreements with compliant privacy terms will exist between the User and the dApp/RoboNotary, the RoboNotary and the identity verification service, and the User and third-party service providers.

9. Working Demo

9.1 A demonstration of our proof of concept is available here and is summarised in Figure 3.

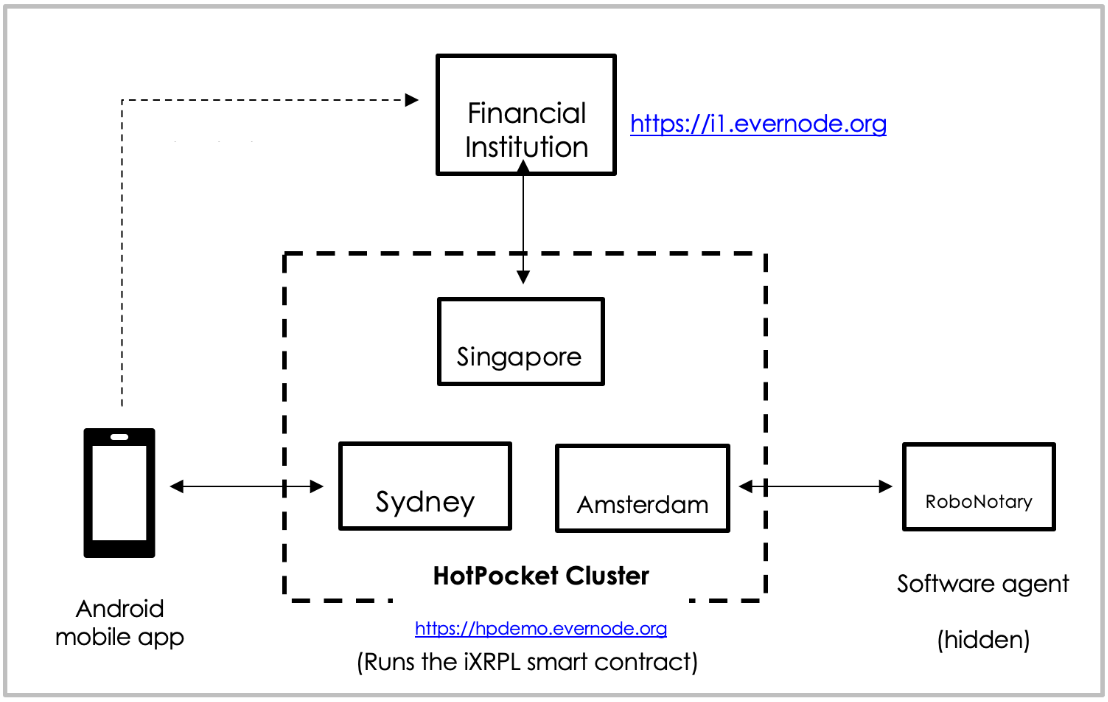

Figure 3-iXRPL Working Demo

9.2 As this is a demo only, we have made some compromises as follows:

(a) **Three Nodes:** The iXRPL smart contract only runs on three HotPocket nodes. The cluster can be viewed here.

(b) **Spoofed Identity Service Provider:** The link to RapidID has been spoofed because we do not want to pay a per-use fee for a live test, and we do not want to use real identity documents. In developing our proto-type, we were able to integrate real RapidID API using the RapidID-provided sample/mock documents.

(c) **Low Grade Human UUID:** The Human UUID generation has been set to low-sensitivity mode to conserve cloud hosting fees for the RoboNotary. The production-grade high-security sensitivity mode is designed to be taxing on the CPU to avoid brute-force attacks and takes around 10-15 seconds on a decent CPU.

(d) **XRP Ledger TestNet:** XRP Ledger TestNet is used for XRP transactions and trust line processing because we are not using real money/accounts in the demo setup.

(e) **Spoofed Financial Institution:** The financial institution's "login with iXRPL" website is, obviously, a mock site. It can be accessed via https://i1.evernode.org.

10. Issues and Enhancements

10.1 This is a proof of concept. Further enhancements to decentralisation, privacy, compliance, and digital inclusion are technically feasible in production.

Decentralisation Enhancements – Obsolete RoboNotary

10.2 The present solution is centralised around a RoboNotary that acts as the conduit between Users and identity service providers. This is a necessary bootstrap. At a very granular level, for example, somebody must sign the contractual agreements that give API access to the identity verification technology each identity provider supplies and pay their fees.

10.3 A better, more decentralised solution is possible, one that obviates the need for a RoboNotary. It is technically possible - and infinitely preferrable - for iXRPL to be a decentralised marketplace where each identity service provider runs a Node and provides its services directly to Users and issues the Human UUID. This structure would remove the need for a RoboNotary. Each verifier would certify its own verifications.

10.4 But this structure cuts across the business model for identity verifiers, based as it is on servicing financial institutions, not individuals. In the short to medium term, it is more feasible for the RoboNotary to handle the "novelty" involved in a blockchain-based self-KYC solution.

10.5 One factor that may accelerate service providers running their own Node and serving Users directly is for the overall solution to be a joint venture between identity verifiers and regulators/government.

Compliance Enhancements – Regulator Participation

10.6 KYC not only has a customer facing service dimension, but it also has a regulator facing law enforcement dimension. It is regulator's appetite for the User's personal data that drives financial institutions to collect and store the User's KYC data in the first place.

10.7 One potential decentralised design for iXRPL would be for law enforcement to be an active participant in the network. Broadly, this would work as follows:

(a) The iXRPL smart contract code could recognise a role called "law enforcement".

(b) The law enforcement role could be bestowed upon an Account by a supermajority of Nodes assuming Nodes are all run by identity service providers and/or participating financial institutions.

(c) The law enforcement role would have privileges allowing it to decrypt the identity data of any User within its jurisdiction, potentially with the User receiving simultaneous notification of that access.

(d) Such a system might be linked to national laws in that law enforcement's rights are dependent on it notifying the basis for its intrusion, or receiving approval from the relevant financial institution, if not the individual User.

10.8 This could result in a national self-KYC solution in which any financial institution and identity service provider can participate where Users retain full control over their identity documents, but regulators having privileged access to the identity of Users consistent with national laws.

Privacy Enhancements – Payment Routing

10.9 Stamping your XRP Ledger account with a Human UUID does not give the public access to your name and date of birth. The public only knows that XRP Ledger Account is controlled by a real individual whose identity has been verified. This information is probably useful. There are many use cases, such as social media accounts, and preventing cryptocurrency scams where it would be useful to know the entity behind the account is a real person even if their identity is not known.

10.10 However, there is a genuine privacy problem in that people who confirm your identity and then subsequently deal with your XRP Ledger Account can know with certainty that you are the owner of a particular account. This knowledge tends to be shared within crypto communities, so eventually your ownership of a particular account (and your cryptocurrency holdings) would become public knowledge.

10.11 A holistic, privacy-by-design solution would also give you a way to transact anonymously. So, while people might know to whom they are sending XRP, they would not know, or be able to match, the XRP Ledger Account that received their payment, and so be able to dox the owner. However, these solutions are non-trivial to design and implement without them being deemed money transmission services as the solutions all tend to involve peer-to-peer transactions being routed via other accounts.

Digital Inclusion Enhancements – More Identity Verification Options

10.12 Our proof of concept is not open to everybody for three reasons, all of which have technical solutions. First, it would only be open to people who hold both a driver's licence and a passport. To broaden access to the service we would need to use a wider range of identity documents. There is no technical barrier to doing so: it is a question of the technical requirements of each identity verifiers' systems.

10.13 Second, our proof of concept relies upon a Human UUID, a unique number computed from the Users verified identity information. But it is technically possible for two people to have the exact same names and date of birth and place of birth. In that rare-but-possible situation, one of the persons would be unable to create a new account but would be able to reverify and so take over the other person's existing identity. So, an alternative system will need to be developed for the system to operate at scale.

10.14 It would be preferrable for the Human UUID to be computed from data that also includes the identity document's unique IDs. However, passports and driver's licences are renewed regularly and given new numbers. The one document that isn't renewed is your birth certificate. But birth certificates have no standard format and so are not usually supported by the automated systems that identity verifiers use.

10.15 Finally, some jurisdictions rely on in person identity checks. Our proof of concept has the flexibility to permit such checks, but it would take time and effort to design a process fortified against fraud that was nonetheless user friendly.

11. Key Learnings

11.1 There are five learnings from the project to highlight.

(a) **Government Required:** To ensure trust, continuing government regulation of identity verifiers as "oracles" is crucial to a blockchain based identity solution. The identity must "get onto" the blockchain in some way and trusted oracles are an effective solution. The government should consider special regulations/standards for identity verifiers acting as oracles for tokenised identity verification.

(b) **Collaborative Marketplace:** To be truly decentralised, the service should be run as a collaborative marketplace curated by government and digital identity verifiers, obviating the need for Notaries as go-betweens. The government already has a regulated market for identity verification services to access government databases. It would be a natural and welcomed progression for that to evolve into a decentralised market running on a permissioned-but-public blockchain.

(c) **Being KYC'd Should Allow Hidden Transactions:** To meet privacy concerns, KYC'd individuals on public blockchains should be able to obscure their blockchain addresses. Allowing people to know who they are transacting with should give you the right to obscure the transacting accounts, without the solution being deemed a money transmission business or otherwise nefarious.

(d) **Regulators Should Participate:** To enable seamless integration with law enforcement and improve privacy compliance, regulatory authorities should consider publishing public keys that private services can incorporate into their blockchains to facilitate legitimate law enforcement access. This is something we intend to further explore in our research.

(e) **Possibly More Significant Than Realised:** Finally, a blockchain-base Self KYC solution is possibly more broadly useful than we previously realised. An account stamped with a digital seal that confirms it is owned by a real person, but not knowing who that person's identity, is *per se* a useful thing. It provides a middle ground between anonymity and being forced to use your real name. There are many situations where knowing an online account is controlled by a real person but not knowing who they are would be a useful institution, including for social media (and the internet in general), whistle-blower complaints, and bootstrapping decentralised communities. This is something we intend to further explore in our research.

12. Next Steps

12.1 This project is complete. We have achieved our goals a bult a proof of concept for a blockchain-based self-KYC solution that is self-sovereign, secured, decentralised, complies with Australian KYC safe harbour regulations, and meets GDPR privacy requirements.

12.2 However, based upon the success of the proof-of-concept we plan a more detailed paper setting out a vision for a blockchain-based Self-KYC solution for Australia.