# Background

In the fall of 2015, the Rebooting the Web of Trust session produced a whitepaper[1] outlining a method of using blockchain distributed ledger technology to ground a new approach to public key management. In rethinking the web of trust, this distributed public key infrastructure removed any dependence on central authorities and established a method of rooting identity in a blockchain distributed database.

While this effort defined a way for everyone and everything to have a sovereign identity[2], it did not propose a means for using the distributed identity for resource creation and consumption. Nor did it propose a means of claims exchange that meets basic security and privacy requirements.

# Problem

The creation of a sovereign identity, one independent of any particular application, creates a shift in paradigm in regards to how digital security and privacy are maintained. In today's world, the application is the center of security and privacy. It controls the individual's identity, rights, and obligations. It controls the claims any individual asserts during any transaction.

As we reboot the web of trust, the center subtly shifts from application to individual. The individual still needs to provide the claims required by the application in order to get the requested services and the sovereign identity provides the root on which those claims are asserted.

However, a sovereign identity that stands alone is not valuable. It needs to be augmented with the proper claims, or attributes that expand on the base identity in order for that identity to be recognised by various applications as having rights and obligations regarding the creation and consumption of resources (assets), whether those assets are digital or physical.

The notion of a universal sovereign identity suggests that the identity is available to all that need it to perform their services. Such an identity needs to be public so that applications can begin the authentication and authorization processes necessary to allow access to assets. I say begin because applications will require differing levels of claims based on asset value, policies, and regulations. This creates a conundrum, how do you balance the need for public availability with proper security and privacy

[1] https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust/blob/master/final-documents/dpki.docx

[2] Sovereign Identity: someone who has independent authority over who they are

guarantees? Without those guarantees, the sovereign identity eventually becomes untrustworthy.

# Proposed Solution - Identity as a service

This paper proposes a solution where a combination of “on-chain” identity transactions, “off-chain” verifiable claims[3], smart contracts, and purpose-built software methods are combined to provide the proper claims while maintaining security and privacy. This combination, termed “identity as a service” is described below.

## Registration and key management

For this exercise, the function of the establishment of the sovereign identity base record, the generation and securing of master hierarchical deterministic (HD) keys, generation of child keys, key rolling, key delegation, and key recovery, while critical, are not within the scope of this whitepaper. The capabilities and scenarios contained assume these registration functions are available.

## Identity as a Service

There are four parties involved in identity as a service.

- Issuers - provide verifiable claims to people and organizations
- Curators - store and curate verifiable claims in a claims repository on behalf of people and organizations
- Consumers - request verifiable claims from people and organizations in order to give them access to protected resources
- People/ organizations - receive verifiable claims from issuers, store them at curators that they trust, and provide them to consumers in order to get access to protected resources

The following example shows the relationship between these parties.

1. Bob contacts the county clerk to get a series of verifiable claims about his birth (recorded name, date of birth, location of birth, witnesses, etc.)
2. As an **issuer** of verifiable claims, the county clerk gives Bob a non-repudiable set of statements about his birth event.
3. Bob stores these statements in a claims repository provided by his **curator**.
4. Bob wants to buy an age-restricted item. The business that can sell him the item requests, as a **consumer**, a verifiable claim as to his age.
5. Bob, as a **person**, retrieves the birth date claim from the repository at his **curator** and presents it to the **consumer** as evidence of age.

[3] https://www.w3.org/Payments/IG/wiki/Main_Page/ProposalsQ42015/VerifiableClaimsTaskForce

6. The **consumer** accepts the claim and allows the transaction.

## Claims repository

While the blockchain is the root of sovereign identity, the claims repository contains the branches and leaves. The claims repository can reside in distributed hash tables(DHT), relational database structures, or data graphs depending on the circumstance. For instance, a particular architecture for an identity blockchain implementation may use DHTs for its claims repository. Traditional identity management or access management (IDM, IAM) solutions may rely on established table structures to store claims. To connect these various claims repositories to a sovereign identity requires a purpose-built set of software methods, a curator agent.

## The Curator Agent

The Curator Agent contains several functions that support the connection between the Person and the Consumer. These are described as follows.

Listener capability - The function required to respond to messages sent to the agent. It contains:

- Signature verification - the agent uses the identity blockchain to verify that the message signature is correct.
- Message decryption - the agent uses one of its public/private key pairs to prepare the message content for processing.

Multi-factor Authentication capability - The function required to establish the level of authentication credentials outlined in a smart contract[4] as necessary to access the Resource Server. These can reflect any combination of the standard MFA techniques including something you know, something you are, and something you have.

Repository management - The function required to establish and maintain the verifiable claims needed to satisfy the smart contract which defines the actual exchange of information assets.

- verifiable claims retrieval - verifiable claims are associated with individuals, not particular services; as such claims are retrieved for presentation to the consumer by the Messaging capability.

[4] Smart contract - a shared piece of code where the parties involved in a swap of assets have confirmed their rights and responsibilities related to the asset. These smart contracts can be instantiated in the blockchain or in the person's and consumer's claims repositories as required by the particular implementation.

- verifiable claims creation - People receive and store verifiable claims from issuers through the agent that the issuer does not need to trust.

Messaging capability - The function required to build the message payload required by the Resource Server according to the smart contract that defines the transaction.

- Smart contract management - the agent negotiates, instantiates, and maintains the smart contract between the Client and the Resource Server.
- Message construction - using the smart contract logic, the agent retrieves verifiable claims from the repository to satisfy the contract and constructs the message to be sent to the Resource Server.

Broadcast capability - The function required to prepare the message for transmission.

- Key generation/ selection - the agent generates the public/private key pairs as required for messaging signing.
- Encryption - the agent encrypts the payload.
- Message signing - the agent signs the message using the sovereign id and public key.

# Examples

The use of the sovereign identity and the verifiable claims repository is illustrated in these simple examples. The identity as a service function separates the authentication and authorization functions from application code and properly places those functions as part of a utility that verifies the individual for all services. Services no longer have to develop proprietary security and privacy capabilities. Individuals no longer have to maintain a multitude of credentials. Trust is appropriately distributed across the ecosystem.

## Financial transaction

For a financial transaction, the authentication and authorization criteria could vary based on the value of the transaction. For instance, a balance inquiry might require a simple set of checks; password authentication and account number associated with the sovereign ID match the account details held by the bank. A smart contract covering those details could be executed by the curator agent.

For transactions of higher value, such as withdrawals, there can be a different smart contract requiring different verifiable claims including second form identifiers and transaction limitations.

Each of these situations are managed using the curating agent to provide the appropriate credentials at the appropriate time.

### Refugee health care

In this case, the refugee may not have any form of identity established, but is in need of services to support life.

- The first thing required is to establish an anchor point for all of the other credentials and identifiers. This is a sovereign ID.
- Next verify something you are; pick a biometric that can be used to connect an individual to their sovereign ID.
    - The simplest and most reliable is a biometric identifier that can be registered, verified, and used by an issuing NGO service.
    - This relationship between the verifiable claim (biometric) and sovereign ID is established by the curating agent.
- Now, at a basic level, healthcare data can begin to be captured. When the individual shows up for treatment, biometric matching technology can be used to identify the sovereign ID to which the health records need to be attached. With no other identifying information beyond the scan, the health status and treatment plans can be used to support the wellbeing of the individual.

All of these transactions are connected and managed by the Agent using a single sign-on attribute, in this case a single verifiable claim, a biometric scan.

## Conclusion

The creation of a sovereign identity not tied to any specific application or service requires that we think about security and privacy in a different way. No longer are the verifiable claims associated with the application services; they are now tied to the individual through his/ her sovereign ID. This allows us to think about different ways to manage identity and resource access. Using a Person, Consumer, Issuer, and Curator paradigm, we can now create a trusted “identity as a service” capability that simplifies security and privacy across the Web. Services no longer have to develop proprietary security and privacy capabilities. Individuals no longer have to maintain a multitude of credentials. Trust is appropriately distributed across the ecosystem.