



# **Using Verifiable Credentials to identify vulnerable customers in finance**

---

# Software Design Specification (SDS) Document

Version number	1.1
Date of version	03/08/2021
Author	Dave Horsfall <dave.horsfall@ncl.ac.uk>
Author job title	Research Software Engineer
Team	Aad van Moorsel <aad.vanmoorsel@ncl.ac.uk> Karen Elliot <karen.elliott@ncl.ac.uk> Kovila Coopamootoo <kovila.coopamootoo@ncl.ac.uk> Magdalene Ng <magdalene.ng@ncl.ac.uk> Tasos Spiliotopoulos <tasos.spiliotopoulos@ncl.ac.uk>

<b>Software Design Specification (SDS) Document</b>	<b>1</b>
<b>Introduction</b>	<b>5</b>
Document Purpose	5
Project Overview	5
Project Aims	5
<b>System Overview</b>	<b>6</b>
Description	6
Intended Usage	6
<b>Verifiable Credentials</b>	<b>7</b>
User Roles	7
Verifiable Presentations	7
<b>Decentralized Identifiers</b>	<b>8</b>
<b>Trust Model</b>	<b>9</b>
Selective Disclosure	10
Derived Credentials	10
The Principle of Data Minimization	10
<b>Technology Review</b>	<b>11</b>
Benefits of Verifiable Credentials	11
Criticism and Outstanding Issues	11
<b>Vulnerability in Finance</b>	<b>12</b>
National Vulnerability Scheme	13
Fair Treatment For All	13
Use Case	14
Using a current account with a physical disability	14
<b>Design Considerations</b>	<b>15</b>
Assumptions	15
Dependencies	15
General Constraints	15
Stakeholder Engagement	15
Key Stakeholder Feedback [anonymised]	16

<b>User Needs</b>	<b>17</b>
Holder	17
Verifier	17
Issuer	17
<b>Design Discussion</b>	<b>18</b>
Options for Data Model	18
Leverage information in existing verifiable credentials	18
Define a new credential type that contains vulnerability information	18
Infrastructure Options	18
Hyperledger indy/aries	19
Microsoft Azure	19
Digital Bazaar	19
Factom Harmony Integrate	19
Microsoft Azure	19
<b>System Architecture</b>	<b>20</b>
Architectural Strategies	20
Fullscreen Diagram	20
Deployment	20
<b>User Sequences</b>	<b>21</b>
Create a verifiable credential	21
Use a verifiable credential	23
<b>Data Design</b>	<b>24</b>
Credential Type	24
Data Model Design	24
JSON-LD Context	25
Publishing Location	25
Credential Rule File	26
Credential Display File	27
Credential Preview	29
<b>Issuer and Verifier Websites</b>	<b>30</b>
<b>User Workflows</b>	<b>31</b>
Issuance Flow	31
Presentation Flow	35



# Introduction

## Document Purpose

A software design specification is a document that describes a structured collection of requirements, and is created to facilitate analysis, planning, implementation, and decision-making in the development process. It defines how the software is expected to perform, and explains how a software application will be built to meet a set of technical requirements.

## Project Overview

The [Trustworthy Digital Infrastructure for Identity Systems](#) project is led by the Turing Institute and is funded through a grant from the Bill & Melinda Gates Foundation. Governments around the world are committed to supporting the roll out of national digital IDs, but there are privacy and security implications associated with scaling these systems at a national level. Responsible implementation of ID services is a critical enabler for financial inclusion; it enables access to services and enactment of civil rights.

## Project Aims

The Turing Institute project aims to enhance the privacy and security of national digital identity systems, with the ultimate goal to maximise the value to beneficiaries, whilst limiting known and unknown risks to these constituents and maintaining the integrity of the overall system.

The FinTrust team at Newcastle University has published a position paper that examines the potential of the combination of [Decentralized Identifiers](#) and [Verifiable Credentials](#) for the identification of vulnerable consumers in finance, and discusses possible implications that these technologies can have for the provision of tailored financial services and products. The UK financial regulator has identified the protection of vulnerable customers as a key priority for the industry and has published appropriate guidance for financial firms, strongly encouraging them to treat [vulnerable customers fairly](#). Four categories of characteristics are considered to constitute drivers of financial vulnerability with the latest report finding that 53% of UK adults show one or more of these characteristics. Our position paper proposes several ways in which a Verifiable Credential could be used to identify a potential vulnerability characteristic.

This document maps vulnerability characteristics to a Verifiable Credential implementation, and defines a user-centred design on which a sandbox environment can be developed and evaluated with end users and other stakeholders.

# System Overview

## Description

The system will provide a working implementation of the World Wide Web Consortium (W3C) standards for [Decentralized Identifiers \(DIDs\) v1.0](#) and [Verifiable Credentials Data Model 1.0](#) in a sandbox environment. The architecture will leverage services from Microsoft that facilitate the creation, storage and presentation of Verifiable Credentials (VCs) on the [Identity Overlay Network \(ION\)](#), which is a public DID overlay network. Two associated Node.js applications will be developed using the Microsoft VC Software Development Kit (SDK) that issue VCs to end users, and verify VCs from end users. The use case identified in our position paper explores guidance published in February 2021 by the [Financial Conduct Authority \(FCA\)](#). The finalised guidance highlights the actions firms should take to understand the needs of vulnerable customers to make sure they are treated fairly. Through engagement with stakeholders, and extensive attention to consumer needs, we have defined user-centered workflows that vulnerable users may encounter when trying to access financial services. We have defined a data model for a new verifiable credential type that maps drivers of vulnerability to new attributes in the verifiable credential, allowing the presentation of tamper-evident claims that cryptographically prove who issued them, but without the need to disclose the specific details of the vulnerability about which the claim is made.

## Intended Usage

The software prototype will be deployed and evaluated within the following environments:

- Interviews with industry experts
- End-user focus groups
- Academic workshops and seminars

# Verifiable Credentials

Verifiable Credentials are the electronic equivalent of the physical credentials that we possess today, such as credit cards, passports, driving licences and qualifications. The [data model for verifiable credentials](#) is a World Wide Web Consortium specification, first published 19 November 2019.

## User Roles

We define the roles of four core actors and the relationships between them in an ecosystem where verifiable credentials are expected to be useful.

Subject	An entity about which claims are made. Example subjects include human beings, animals, and things.
Holder	A role an entity might perform by possessing one or more verifiable credentials and generating verifiable presentations from them. Example holders include students, employees, and customers. In many cases the holder of a verifiable credential is the subject, but in certain cases it is not. For example, a parent (the holder) might hold the verifiable credentials of a child (the subject).
Issuer	A role an entity performs by asserting claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder. Example issuers include corporations, non-profit organizations, trade associations, governments, and individuals.
Verifier	A role an entity performs by receiving one or more verifiable credentials, optionally inside a verifiable presentation, for processing. Example verifiers include employers, security personnel, and websites.

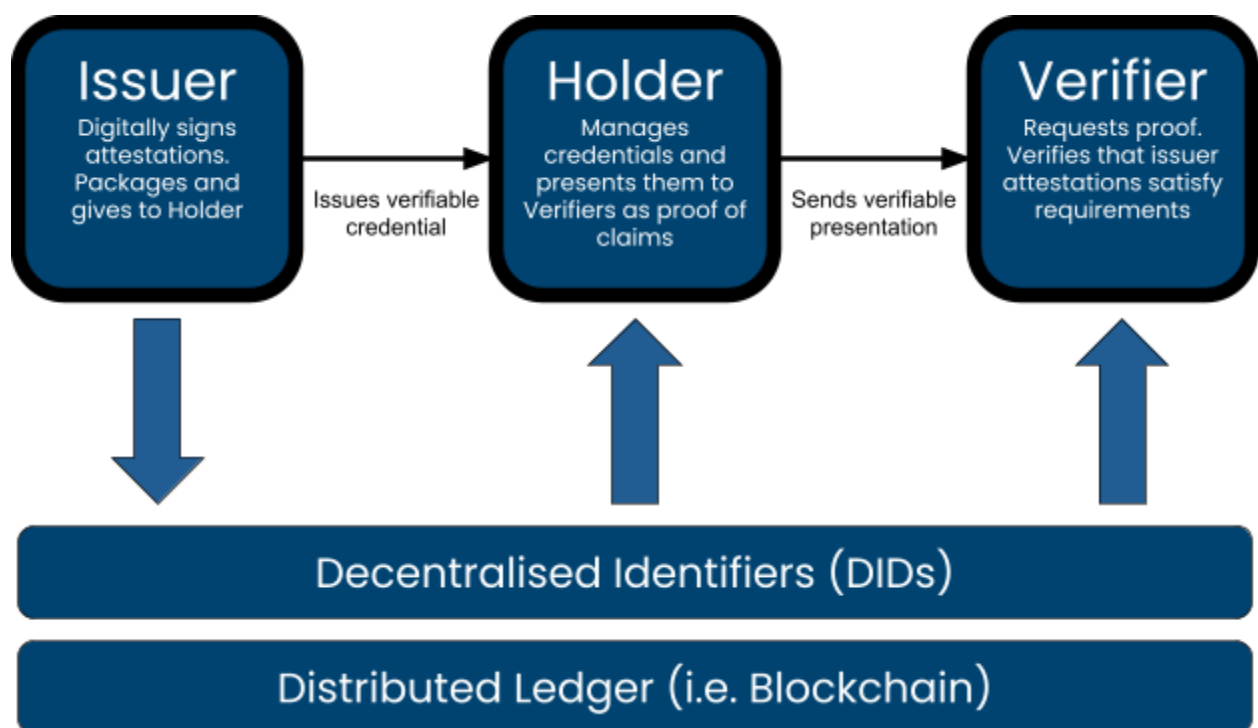
## Verifiable Presentations

A verifiable credential can represent all of the same information that a physical credential represents. The addition of technologies, such as digital signatures, makes verifiable credentials more tamper-evident and trustworthy than their physical counterparts. Holders of verifiable credentials can generate Verifiable Presentations and then share these presentations with verifiers to prove they possess verifiable credentials with certain characteristics. Verifiable Presentations are packages of evidence (either credentials, or data derived from one or more credentials) built by holders to satisfy a verifier's requirements. Verifiers learn with certainty which issuers have attested something by checking digital signatures against a verifiable data registry.



## Decentralized Identifiers

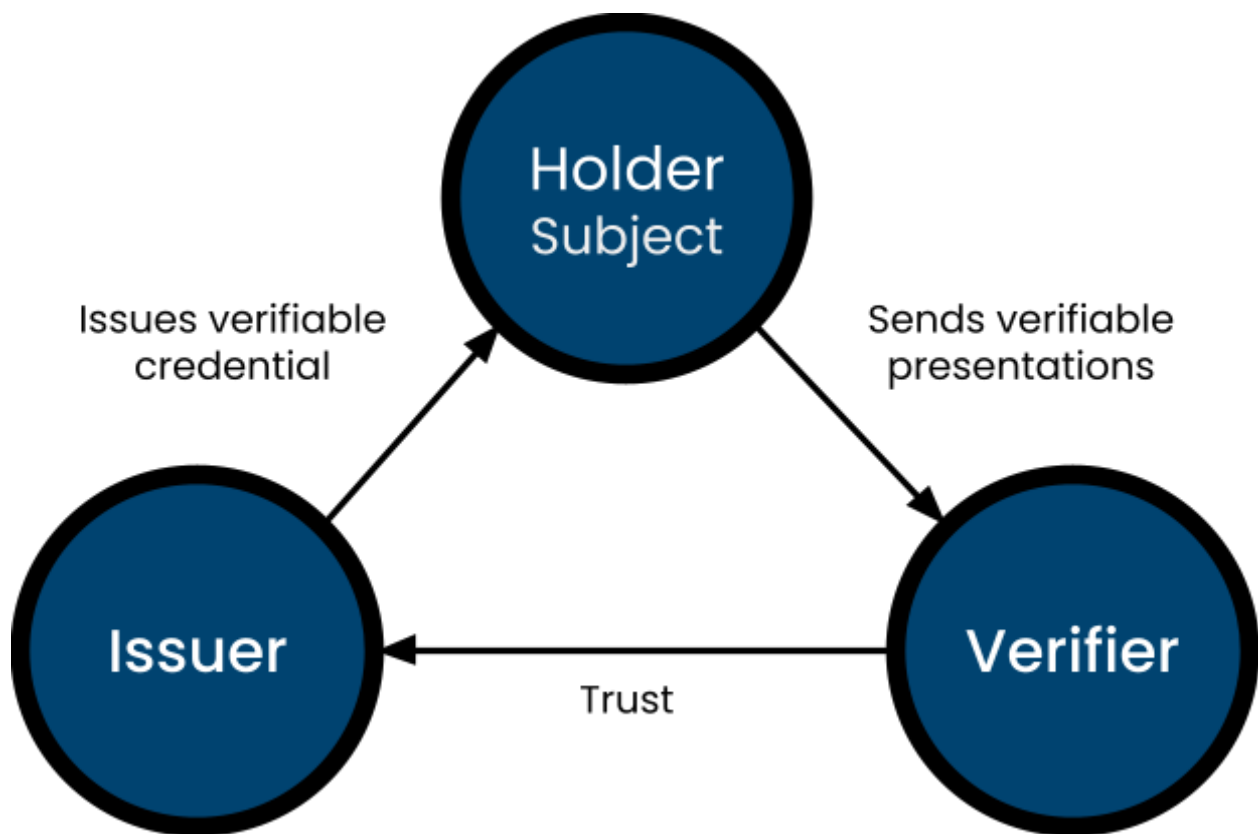
The attestations made by VCs are backed by another emerging W3C standard technology, [Decentralized Identifiers \(DIDs\)](#). DIDs are a type of identifier that enables a verifiable, decentralized digital identity, and are based on the Self-sovereign identity paradigm. They use industrial-strength, peer-reviewed cryptography, and the digital signatures that endorse them have documented algorithms for verification. [DID](#)-based URLs are used for expressing identifiers associated with [subjects](#), [issuers](#), [holders](#) and other machine-readable information associated with a verifiable credential.



Verifiable credentials are not dependent on DIDs and DIDs do not depend on verifiable credentials. However, many verifiable credentials implementations adopt DIDs, and software libraries implementing the specification need to resolve DIDs.

## Trust Model

The core actors and the relationships between them can be represented with a triangle of trust. Issuers create credentials, holders store them, and verifiers ask for proof based upon them.



All entities trust the verifiable data registry to be tamper-evident and to be a correct record of which data is controlled by which entities. The distributed ledger (i.e. Blockchain) establishes a chain of trust between stakeholders.

The holder and verifier trust the issuer to issue true (that is, not false) credentials about the subject, and to revoke them quickly when appropriate.

The holder trusts the repository to store credentials securely, to not release them to anyone other than the holder, and to not corrupt or lose them while they are in its care.

This trust model differentiates itself from other trust models by ensuring the issuer and the verifier do not need to trust the repository, and that the issuer does not need to know or trust the verifier.

## Selective Disclosure

The concept of selective disclosure means that a derived verifiable credential is formatted according to the verifier's data schema instead of the issuer's data schema, without needing to involve the issuer after verifiable credential issuance. This provides a great deal of flexibility for holders to use their issued verifiable credentials, and improves privacy.

## Derived Credentials

Verifiable presentations can either disclose the attributes of a verifiable credential, or satisfy derived predicates requested by the verifier. Derived predicates are Boolean conditions, such as greater than, less than, equal to, is in set, and so on.

## The Principle of Data Minimization

The principle of data minimization involves limiting data collection to only what is required to fulfill a specific purpose. This approach is required by regulation in multiple jurisdictions, including the Health Insurance Portability and Accountability Act (HIPAA) in the United States and the General Data Protection Regulation (GDPR) in the European Union.

# Technology Review

## Benefits of Verifiable Credentials

- Both verifiable credentials and verifiable presentations can be transmitted rapidly, making them more convenient than their physical counterparts when trying to establish trust at a distance.
- When presenting verifiable credentials, the verifier no longer has to contact the issuer to confirm the credential. This removes a significant administration burden from industries such as finance and healthcare in data verification costs. However, it is important to understand that verifiers still need to determine if an issuer is trustworthy or not.
- The holder keeps control and ownership over their identity and data. Holders choose what they want to disclose, and to whom they want to disclose it. They can share only the required information, without disclosing unnecessary and irrelevant data. For example, they can prove they are over 18 years old, without needing to disclose their exact age.

## Criticism and Outstanding Issues

- There has been considerable discussion in recent months about the risks of verifiable credentials, particularly in the context of [COVID-19 immunity passports](#). Much of this discussion [focuses on the legal, ethical and social](#) challenges when adopting the new technology.
- VCs, DIDs, and related APIs are largely built on web protocols such as HTTP and URLs, which generally presume internet connectivity and online use. The web standards on which VCs are based currently offer nothing to support offline usage.
- There is still active discussion and debate about the implementation mechanisms for selective disclosure in verifiable credentials. Most current solutions for privacy enhanced VCs are constructed on Hyperledger Aries, but Microsoft has chosen a different approach. In their recently [published paper](#), they describe an alternative scheme using [SNARKs](#).

# Vulnerability in Finance

The Financial Conduct Authority (FCA) has identified a vulnerable consumer as ‘somebody who, due to their personal circumstances, is especially susceptible to harm, particularly when a firm is not acting with appropriate levels of care’. The UK financial regulator has identified the protection of vulnerable customers as a key priority for the industry and has published appropriate guidance for financial firms, strongly encouraging them to treat vulnerable customers fairly. Four categories of characteristics are considered to constitute drivers of financial vulnerability—poor health, impact of life events, low resilience and low capability (Table 1). The latest report finds that 53% of UK adults show one or more of these characteristics.

**Table 1. The four key drivers of vulnerability and the types of characteristics of vulnerability they may cause**

Health	Life Events	Resilience	Capability
Physical disability	Caring responsibilities	Low or erratic income	Low knowledge or confidence in managing finances
Severe or long-term illness	Bereavement	Over indebtedness	Poor literacy or numeracy skills
Hearing or visual impairment	Income shock	Low savings	Low English language skills
Poor mental health	Relationship breakdown	Low emotional resilience	Poor or non-existent digital skills
Addiction	Domestic abuse		Learning impairments
Low mental capacity or cognitive impairment	People with non standard requirements such as people with convictions, care leavers, re		No or low access to help or support
	Retirement		

People with vulnerability characteristics are more likely to lack confidence in the financial industry, and this lack of trust and confidence can result in consumers not engaging with financial services and products, or failing to address their own financial needs. In order to address this problem and support vulnerable individuals more effectively, financial firms are encouraged to i) understand vulnerable customers by carrying out research and collecting vulnerability data, ii) train and develop their staff to embed the consideration of vulnerable

consumers, iii) consider the communication and information needs of vulnerable customers, iv) adapt customer service processes and systems to account for vulnerable customers, v) integrate the consideration of vulnerable customers into the product and service design process, and vi) monitor and evaluate the treatment of vulnerable customers.

## National Vulnerability Scheme

Considering the diverse range of vulnerability characteristics, we can explore several ways in which a vulnerable consumer might encounter barriers when trying to access financial services. Some vulnerability characteristics, such as a physical disability, could be attested through the National Health Service (NHS). Verifiable Credentials issued through national organisations such as the NHS could be trusted by financial firms when verifying claims however, other characteristics in Table 1, such as poor digital skills, are more difficult for a consumer to prove because the Verifier needs to recognise and trust the Issuer. In order to complete the trust triangle, and establish trust between the issuer and the verifier, it is suggested that a national vulnerability scheme is introduced to which accredited organisations can join, and assert claims about the vulnerability of individuals. This scheme would be transparently developed through engagement with the public, financial firms, and relevant organisations capable of issuing claims about vulnerabilities defined in Table 1. The scheme would be underpinned by a technical framework of new verifiable credential types, schemas, and URIs, allowing verifiers to confidently act upon vulnerabilities issued by accredited organisations.

## Fair Treatment For All

Through engagement with industry experts and other stakeholders, we understand that end-users in finance generally do not want to be identified as vulnerable. The characteristic of vulnerability is seen as negative. Users are sceptical about how commercial companies will use information about their vulnerability, and feel it may place them at a disadvantage in the context of financial services.

We propose a vulnerability scheme called “Fair Treatment for All”. Instead of focusing on vulnerability, which is important for the Verifier, we have focused on the user’s experience and considered their motivation to engage with the scheme. Users are more likely to identify with fairness, which is ultimately the objective of the FCA guidance. If users are able to identify with the benefits of the scheme they are more likely to engage.

## Use Case

In our system design, we will refer to one specific vulnerability criteria, to explore how such a system would work. We will consider how a consumer with a physical disability might interact with their bank to notify them about their vulnerability status, but without disclosing the nature of their disability.

### Using a current account with a physical disability

In this scenario, the core actors in the system are defined in Table 2.

Table 2. Use case actors		
Verifier	Bank	The Verifier is a financial firm providing services to a consumer.
Holder	Current account holder	The Holder is a consumer of services from a financial firm.
Issuer	NHS	The Issuer is an organisation accredited to the vulnerability scheme, that is capable of asserting the Subject meets one of the vulnerability criteria.

# Design Considerations

## Assumptions

In our system design we assume that:

- We can leverage existing architecture for the VC and DID implementations.
- The sandbox will be deployed in a public cloud provider.
- The Holder and the Subject are the same entity (i.e. the consumer about which verifiable claims are made will hold and assert the claim to the Verifier).

## Dependencies

The following dependencies are noted:

- Users will require a compatible client to request and present VCs. This will be a mobile phone application or a web application.
- The DID and VC implementation must comply with the W3C Standards for Verifiable Credentials Data Model 1.0 and Decentralized Identifiers (DIDs) v1.0.

## General Constraints

The project has several constraints under which the system design must operate:

- This project is limited to 9 months, and should be completed in July 2021
- This project has staff costs funded, of which 3 months is allocated for the development of this software system design.
- VCs and DIDs are built on web standards, meaning internet connectivity is required. Offline usage of the technology is not currently supported.

## Stakeholder Engagement

16/02/2021	<a href="#">Atom Bank</a>	Initial discussion about vulnerability and verifiable credentials.
25/02/2021	<a href="#">Yoti</a>	Initial discussion with Yoti about digital identity and vulnerability. Julie Dawson is Director of Regulatory & Policy.
26/02/2021	<a href="#">Turing Research Engineering Group</a>	Initial technical discussion with Turing RSE team about verifiable credentials and current implementations.
02/03/2021	<a href="#">Northumberland Community Bank</a>	Initial discussion with Northumberland Community Bank about vulnerability



04/03/2021	<a href="#">Warwick University</a>	Exploring synergies between projects with Carsten Maple's group at Warwick University.
19/03/2021	<a href="#">Turing Research Engineering Group</a>	Follow up discussion with Turing RSE team
22/03/2021	<a href="#">Yoti</a>	Follow up with Yoti and Jim Purves from the Post Office
08/04/2021	<a href="#">Atom Bank</a>	Interview with vulnerability champion, Kelly Robertshaw and Michael Sherwood

## Key Stakeholder Feedback [anonymised]

- Financial firms are concerned about storing personal data, and don't want the responsibility or burden of compliance. This might be a regulatory barrier for firms in practical identification of vulnerable customers. Verifiable Credentials solve this problem because personal data is not stored by the bank.
- A static snapshot of vulnerability isn't useful to financial firms. Firms need a way to automatically and periodically check the vulnerability status for consumers without needing to contact or get approval from the consumer.
- Need to show benefits to users so they adopt it. People may not want to be identified as vulnerable because there is a stigma associated with the term.
- Users raised concerns about banks not responding positively if they identified them as vulnerable (seeing vulnerability as a negative thing). Needs clear communication from the bank to the consumer to understand how being identified as vulnerable would result in a fairer service.
- Data minimisation is important and should be built into the data model. Presentation requests should also be able to request only the information that is required.
- Some organisations are trying to identify vulnerability through the aggregated data streams. This is firstly unreliable, but also potentially unethical. Allowing users to clearly present accurate information about vulnerability through verifiable credentials removes ambiguity.
- Concerns about ethics of large scale data aggregation of vulnerable citizens? Actually, by using DIDs and VCs this is somewhat mitigated because there isn't a central repository of all data, nor does a single organisation (other than the subject) store all claims relating to themselves.
- Issuer is the entity that makes the claim. We know from our trust model that the verifier must trust that the issuer is making true (not false) claims about the subject. Banks will need a safe framework to establish organisational trust with Issuers.

# User Needs

## Holder

The system must provide the following features for Holders:

1. **Store a claim.** It MUST be possible for the holder of a claim to store that claim in one or more credential repositories. It MUST also be possible for the holder to move a claim among credential repositories, and to do so without requesting a new claim from the claim issuer.
2. **Retrieve a claim.** It MUST be possible for a holder to select if and which appropriate credential should be sent to a verifier.
3. **Assert a claim.** It MUST be possible for the holder of a verifiable credential to assert the claim to a verifier, and restrict the amount of information exposed in a credential they choose to share. It also MUST be possible for the holder to limit the duration for which that information is shared.

## Verifier

The system must provide the following features for Verifiers:

- **Verify a claim.** It MUST be possible for a verifier to verify that the credential is an authentic statement of an issuer's claims about the subject. The verifying entity must have the capability to connect the issuer's identity to its credential identifier and the subject's identity to their identifier as indicated in the credential. The issuer's verification information, such as its public key, must be discoverable from the credential record and verifiably linked to the issuer. It MUST be possible to do this in an automated fashion, so that financial firms can periodically, and automatically identify vulnerable consumers, and allocate appropriate resources to respond appropriately.

## Issuer

The system must provide the following features for Issuers:

- **Issue a claim.** It MUST be possible for any entity to issue a verifiable credential.
- **Revoke a claim.** It MUST be possible for the issuer of a claim to revoke it, after which it will no longer satisfy verification procedures. For example, in the case of Income Shock, there would be a time constraint after which this claim would no longer be active.

# Design Discussion

## Options for Data Model

When considering how users might present information to financial firms about vulnerabilities, or conversely how financial firms might request information about vulnerability from users, we have identified two potential design options.

### **Leverage information in existing verifiable credentials**

Other types of verifiable credentials will contain information relevant to vulnerability. For example, a user may have been issued with a credential from the NHS that demonstrates diagnosis of a physical disability, or an individual may have a credential from an employer indicating they have recently been made redundant. By leveraging this type of existing information it would be possible for a financial firm to request specific credential attributes to understand a vulnerability without the user needing to obtain any additional verifiable credentials. This would remove a significant burden from the end-user. However, it would be difficult to define an existing and immutable set of credential types that contain attributes pertinent to vulnerability. Additionally, there may be some vulnerability characteristics to which no existing credential types attest. From a technical perspective, it may not be possible to easily construct a single presentation request that captures all vulnerability credential attributes that may be useful to a financial firm, nor offer the end-user a simple workflow that is easy to understand.

### **Define a new credential type that contains vulnerability information**

By defining a new credential type, we can construct a data model that contains all relevant information needed by financial firms to follow the guidance issued by the FCA. The new credential type can then be used by multiple organisations to issue new credentials concerning vulnerability characteristics of individuals, regardless of the nature of the vulnerability. Financial firms can then construct presentation requests to obtain only the information they need.

## Infrastructure Options

There are a number of working open source implementations for DIDs and VCs. The following is a summary of the options that were reviewed.

## Hyperledger Indy/Aries

DIDs, Verifiable Credentials and anonymous credentials

- [Identity focused/DID blockchain](#)
- [Library for integration with indy + VCs + anon creds](#)
- [P2P, DLT/Blockchain agnostic \(in theory\) identity focused library \(e.g. agent-agent\), that supports DIDs and VCs](#)

## Microsoft Azure

ION + VC sdks

- [DID project](#)
- [ION: Sidetree based DID implementation](#)
- [VerifiableCredentials-Verification-SDK-TypeScript](#)
- [VerifiableCredential-SDK-Android](#)

## Digital Bazaar

DID resolution + VCs

- [Selective DID Resolver Client](#)
- [Decentralized Identifier client CLI](#)
- [Verifiable Credentials JS Library](#)

## Factom Harmony Integrate

VCs + identity ecosystem (uses Factom did implementation)

- [Factom: Harmony Integrate](#)
- [factom-protocol/FIS](#)

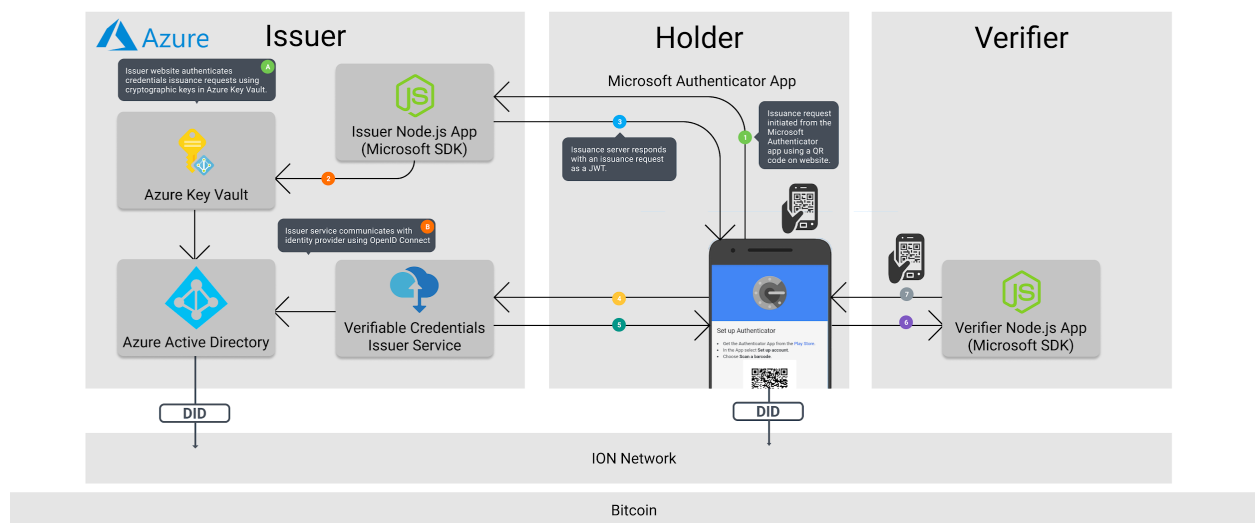
## Microsoft Azure

Due to the limited resources and time to complete the software prototype, we will select the Microsoft Azure service and infrastructure for our environment. The VC service in Azure was in private preview at the time of writing this document, but is due for public preview on 2nd April 2021.

# System Architecture

## Architectural Strategies

The software prototype will be built in Microsoft Azure using the [Verifiable Credentials preview service](#) which enters public preview on 2nd April 2021. The service sits on Microsoft's [Identity Overlay Network](#) (ION), which is a public, permissionless, decentralized DID overlay network that runs on top of Bitcoin, and leverages the Sidetree protocol.



[Fullscreen Diagram](#)



## Scalability

The scalability of software is an attribute of the system to increase its capacity based on demand. When considering how a system such as the one defined in this document might be rolled-out by large organisations or governments, there are several points in the architecture where scalability becomes important when considering risk.

## Decentralized Identity Systems (ION)

The Verifiable Credential system sits on top of a Decentralized Identity System, in our case Microsoft's ION. Scalability has been identified as a key challenge for DID systems using distributed ledgers. In the context of Blockchain, scalability can be defined as the ability of the network to maintain its processing capabilities while expanding the network ([Hileman and Rauchs, 2017](#)). The time between the initiation of a transaction and its addition to the block is

measured as roughly 10 minutes ([Croman et al, 2016](#)) which results in a throughput rate of 7 transactions/sec. One proposed solution to the scalability challenge is the implementation of a second layer protocol on top of the distributed ledger. This is the adopted solution for Microsoft ION, which is a public and permissionless DID system which utilizes a second layer protocol called SideTree. First outlined by [Alex Simons in 2018](#) with ambitions of providing a DID to 7.5 billion people, the first version of ION was launched in [early 2021](#).

## Node.js apps

The Node.js applications act as the interface with users to issue and verify credentials. The nature of these standalone applications mean they can be containerised, and scaled both up and out using standard techniques in cloud architecture.

## Issuing service

The Node.js applications operate through communication with several Azure services such as the Key Vault, Issuing Service and Azure Active Directory. Microsoft offers advice for designing scalable systems using Azure services. Assessing the scalability of specific Azure services is outside the scope of this project, but assessing scalability issues in the wider system may be addressed in future work.

## Deployment

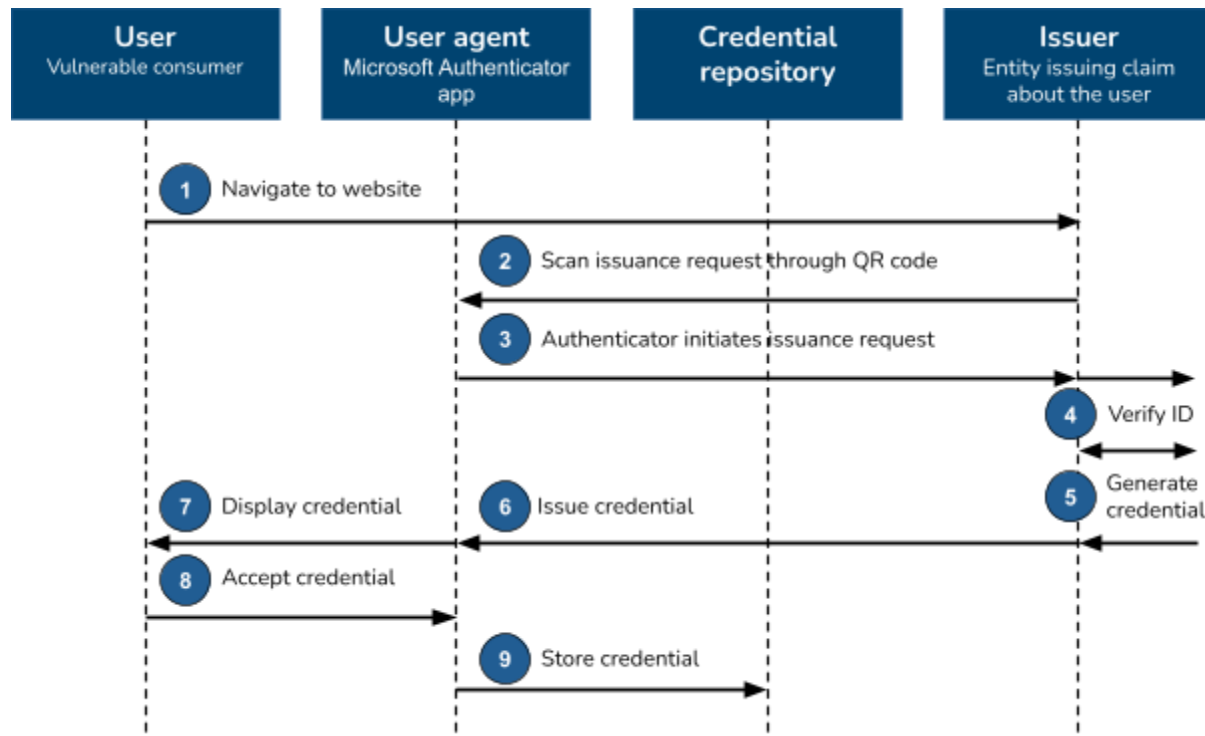
The deployment process for this infrastructure is as follows:

1. [Create and setup an Azure AD tenant](#)
2. [Connect to a compatible identity provider](#)
  - a. [Register the Issuer service in Azure AD](#)
  - b. [Customise claims in ID tokens](#)
3. [Create a credentials rules file](#)
4. [Create a credential display file](#)
5. [Build the issuer website](#)
  - a. [Verify issuance flow in Authenticator app](#)
6. [Build the verifier website](#)
  - a. [Verify presentation flow in Authenticator app](#)

# User Sequences

## Create a verifiable credential

In order to receive a verifiable credential, the following sequence is observed:



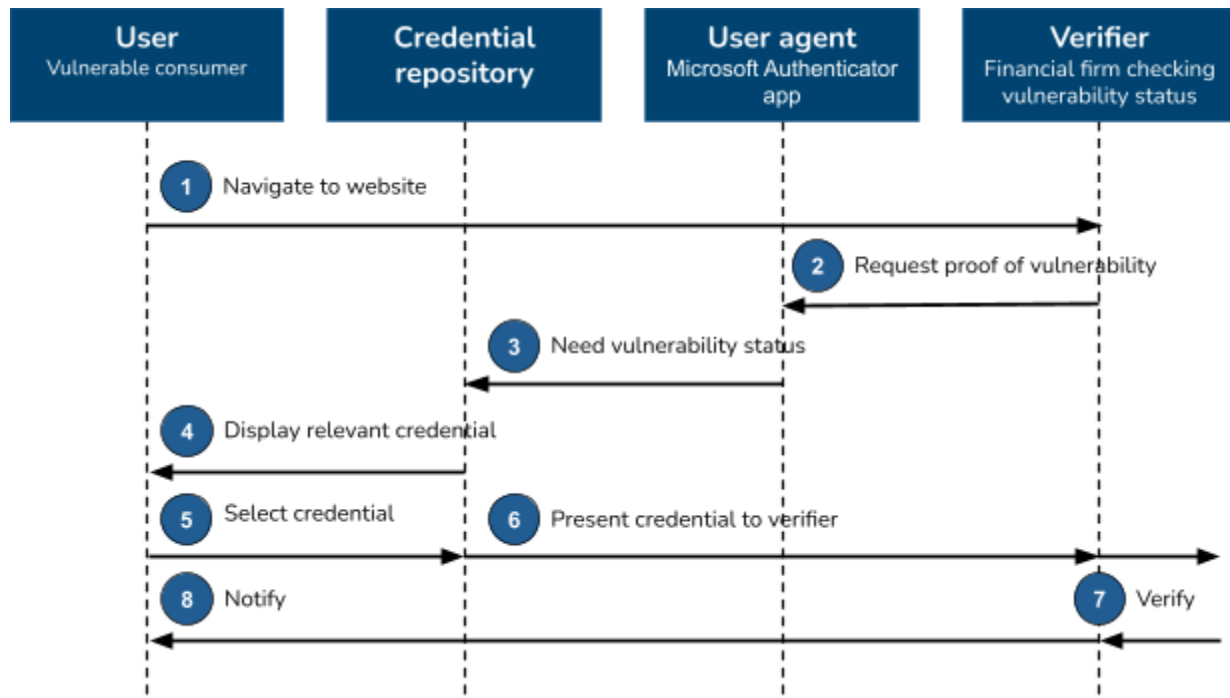
1. **User navigates to Issuer's website.** The user must first log in to the Issuer's identity provider. This allows the user to prove who they are before receiving their credential.
2. **Scan issuance request through QR code.** The user then scans a QR code which sends a credential issuance request to Microsoft Authenticator.
3. **Authenticator initiates issuance request.** Microsoft Authenticator will then send an HTTP GET request back to the issuer server to retrieve the details of the issuance request.
4. **Verify ID.** The Issuer then verifies the identity of the user and examines their systems to obtain any relevant vulnerability criteria.
5. **Generate Credential.** The issuer service then transforms these security tokens and their claims into Verifiable Credentials.
6. **Issue credential.** The credential is issued to the user agent.
7. **Display credential.** The details of the credential are displayed to the user through the user agent.
8. **Accept credential.** The user accepts the credential into the user agent.

9. **Store credential.** The credential is stored in the credential repository.



## Use a verifiable credential

In order to use verifiable credential, the following sequence is observed:



1. **User navigates to Verifier's website.** The user navigates to the Verifier's website.
2. **Request proof of vulnerability.** The Verifier asks for proof of vulnerability, which is initiated by the user scanning a QR code on the website. This sends a verifier request to Microsoft Authenticator.
3. **Need vulnerability status.** Microsoft Authenticator queries all user credentials for those that meet the vulnerability status requirements in the request.
4. **Display relevant credentials.** Any matching credentials are displayed to the user in the user agent.
5. **Select credential.** The user selects the credentials they would like to use.
6. **Present credential to verifier.** The user agent creates a verifiable presentation and sends this to the verifier.
7. **Verify.** The verifier verifies the claim.
8. **Notify.** The Verifier then notifies the user about the outcome of the verification, and can respond appropriately based on the vulnerability that has been identified.

# Data Design

## Credential Type

In order to assert claims about vulnerability criteria, a new credential [type](#) called **VulnerabilityStatusCredential** will need to be defined. We have engaged stakeholders in the financial industry who would verify these credentials to understand how they intend to request and consume them. To ensure interoperability of this credential, it will be necessary to work closely with related organizations to define credential types, schemas, and URIs for future use in the financial industry.

## Data Model Design

We will define a basic data model that maps a limited amount of information required for financial firms to provide support to vulnerable consumers in alignment with the FCA guidance. To achieve this, we will extend existing vocabulary already available on the web at [schema.org](https://schema.org). Although these vocabularies are not immutable, they are popular and have high inertia. We will extend the core vocabulary for <https://schema.org/Person>.

Table 3. <i>VulnerabilityStatusCredential data model</i>	
Person	<a href="https://schema.org/Person">https://schema.org/Person</a>
Vulnerability Classifier	Health
	Life Events
	Resilience
	Capability
Vulnerability Type	['Addiction', 'Bereavement', 'Caring responsibilities', 'Domestic abuse', 'Hearing or visual impairment', 'Income shock', 'Learning impairments', 'Low emotional resilience', 'Low English language skills', 'Low knowledge or confidence in managing finances', 'Low mental capacity or cognitive impairment', 'Low or erratic income', 'Low savings', 'No or low access to help or support',

	'Over indebtedness', 'People will non standard requirements such as people with convictions, care leavers', 'Physical disability', 'Poor literacy or numeracy skills', 'Poor mental health', 'Poor or non-existent digital skills', 'Relationship breakdown', 'Retirement', 'Severe or long-term illness']
Mobility Issues	Boolean
Deafness	Boolean
Blindness	Boolean

## JSON-LD Context

```
{
  "@context": [
    "https://vc.ncldata.dev/vulnerability/v1",
    "http://schema.org"
  ],
  "type": "Person",
  "vulnerability": {
    "classifier": "Health",
    "type": "Physical disability"
    "mobilityIssues": true,
    "deafness": false,
    "blindness": false,
  }
}
```

## Publishing Location

To allow people to read and discover the VC type on the Web, we will give it an URL like <https://vc.ncldata.dev/vulnerability-status-credential/v1>.

## Credential Rule File

The rules file is a JSON file that describes the properties of the verifiable credential, which will be defined as follows.

```
{
  "attestations": {
    "idTokens": [
      {
        "mapping": {
          "firstName": { "claim": "given_name" },
          "lastName": { "claim": "family_name" },
          "classifier": { "claim": "classifier" },
          "type": { "claim": "type" },
          "mobilityIssues": { "claim": "mobility_ssues" },
          "deafness": { "claim": "deafness" },
          "blindness": { "claim": "blindness" }
        },
        "configuration":
        "https://login.microsoftonline.com/00000000-0000-0000-0000-000000000000/v2.0/.well-known/openid-configuration",
        "client_id": "00000000-0000-0000-0000-000000000000",
        "redirect_uri": "vcclient://openid/"
      }
    ]
  },
  "validityInterval": 2592000,
  "vc": {
    "type": ["FairnessForAll"]
  }
}
```

## Credential Display File

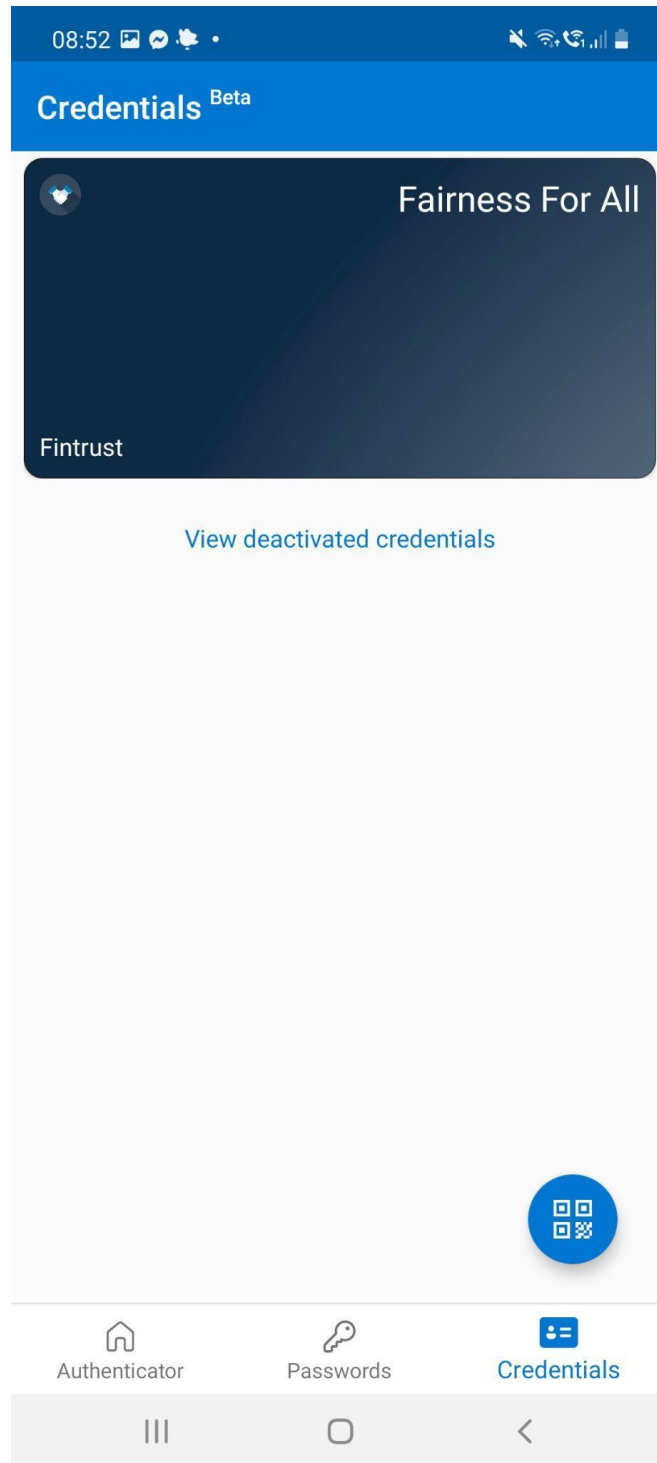
The display file is a JSON file that describes how the contents of the Verifiable Credential will be displayed in the Microsoft Authenticator app, and will be defined as follows.

```
{
  "default": {
    "locale": "en-US",
    "card": {
      "title": "Fairness For All",
      "issuedBy": "Fintrust",
      "backgroundColor": "#092642",
      "textColor": "#ffffff",
      "logo": {
        "uri":
"https://fintrust.blob.core.windows.net/public/FairnessForAllLogo.png",
        "description": "Fairness For All"
      },
      "description": "Use your verified credential to share details about your vulnerabilities."
    },
    "consent": {
      "title": "Do you want to get your Verified Credential?",
      "instructions": "Sign in with your account to get your card."
    },
    "claims": {
      "vc.credentialSubject.firstName": {
        "type": "String",
        "label": "First name"
      },
      "vc.credentialSubject.lastName": {
        "type": "String",
        "label": "Last name"
      },
      "vc.credentialSubject.classifier": {
        "type": "String",
        "label": "Classifier"
      },
      "vc.credentialSubject.type": {
        "type": "String",
        "label": "Type"
      },
      "vc.credentialSubject.mobilityIssues": {
        "type": "String",
```

```
    "label": "Last name"
  },
  "vc.credentialSubject.deafness": {
    "type": "String",
    "label": "Deafness"
  },
  "vc.credentialSubject.blindness": {
    "type": "String",
    "label": "Blindness"
  }
}
}
```

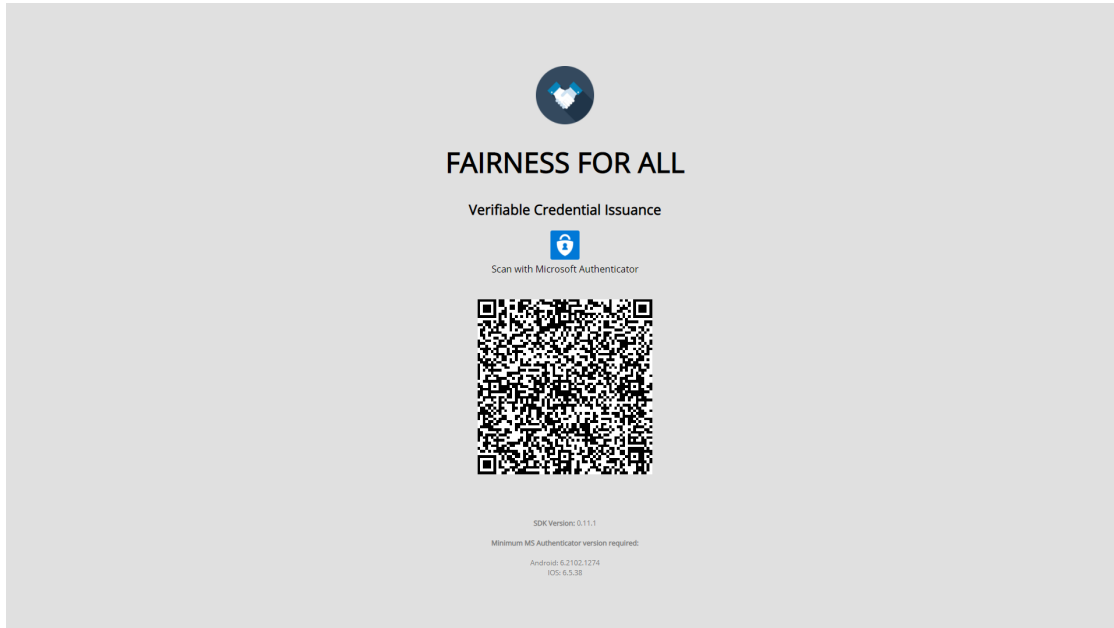
## Credential Preview

The credential preview in the Microsoft Authenticator app.

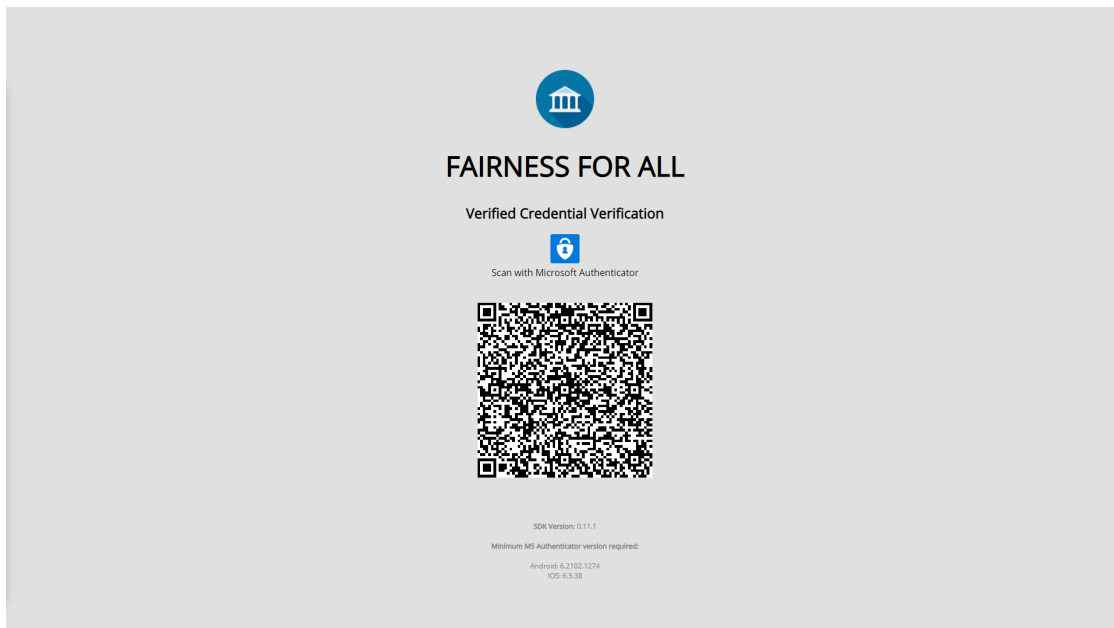


## Issuer and Verifier Websites

The issuer website will be set up in accordance with the [Microsoft documentation](#), and using the Verifiable Credentials SDK from the [Github repo](#).



The verifier website will be set up in accordance with the [Microsoft documentation](#), and using the Verifiable Credentials SDK from the [Github repo](#).



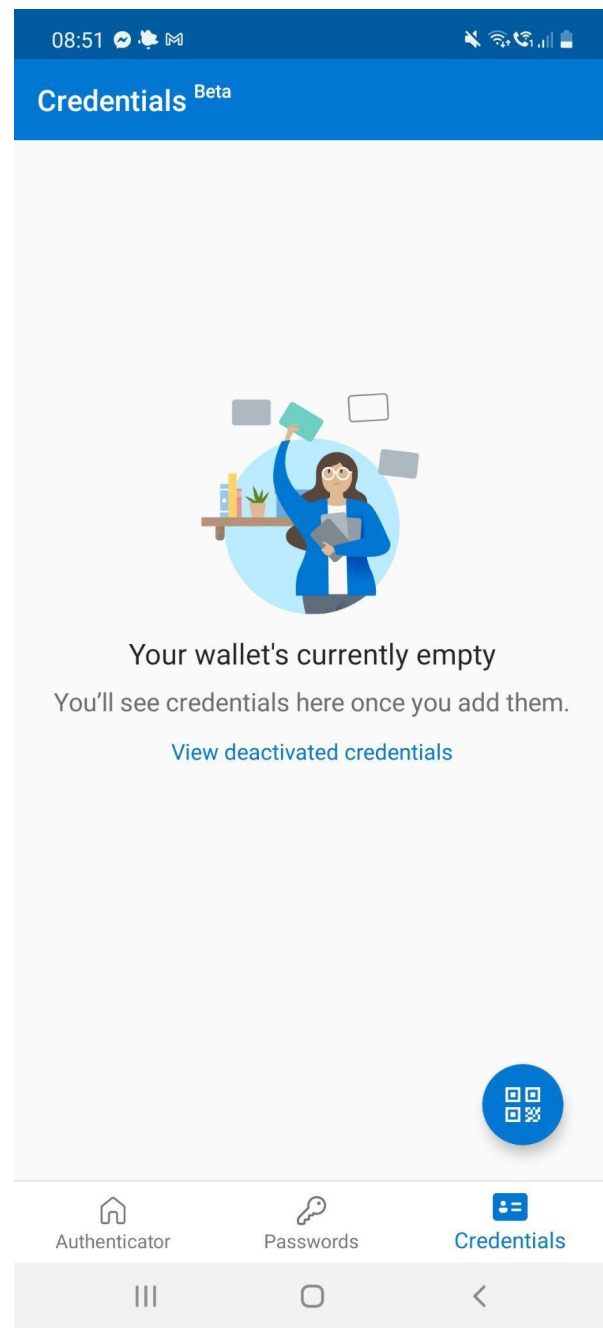


# User Workflows

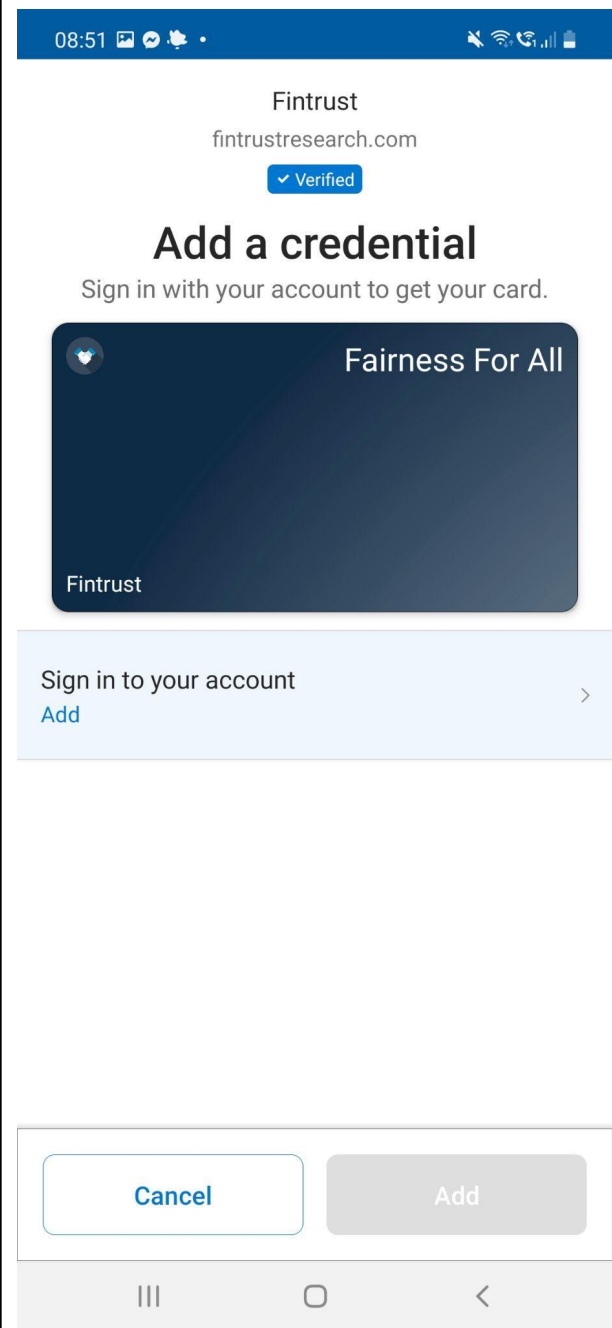
## Issuance Flow

When an issuance QR code has been scanned, the user will automatically be pushed through the issuance flow based on the rules file defined in the request. The following is an example of what the user will see in Microsoft Authenticator.



### Step 1 Open client to scan QR code





### Step 2 Preview card




### Step 3 Login to account

08:51  

login.microsoftonline.com  

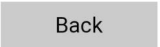

---


 **Microsoft**

## Sign in

Email address, phone number or Skype


[Can't access your account?](#)



 Sign-in options


---

[Terms of use](#) [Privacy & cookies](#) ...




### Step 4 Accept new card


08:52  


**Fintrust**  
fintrustresearch.com  




## Add a credential


Sign in with your account to get your card.

 **Fairness For All**

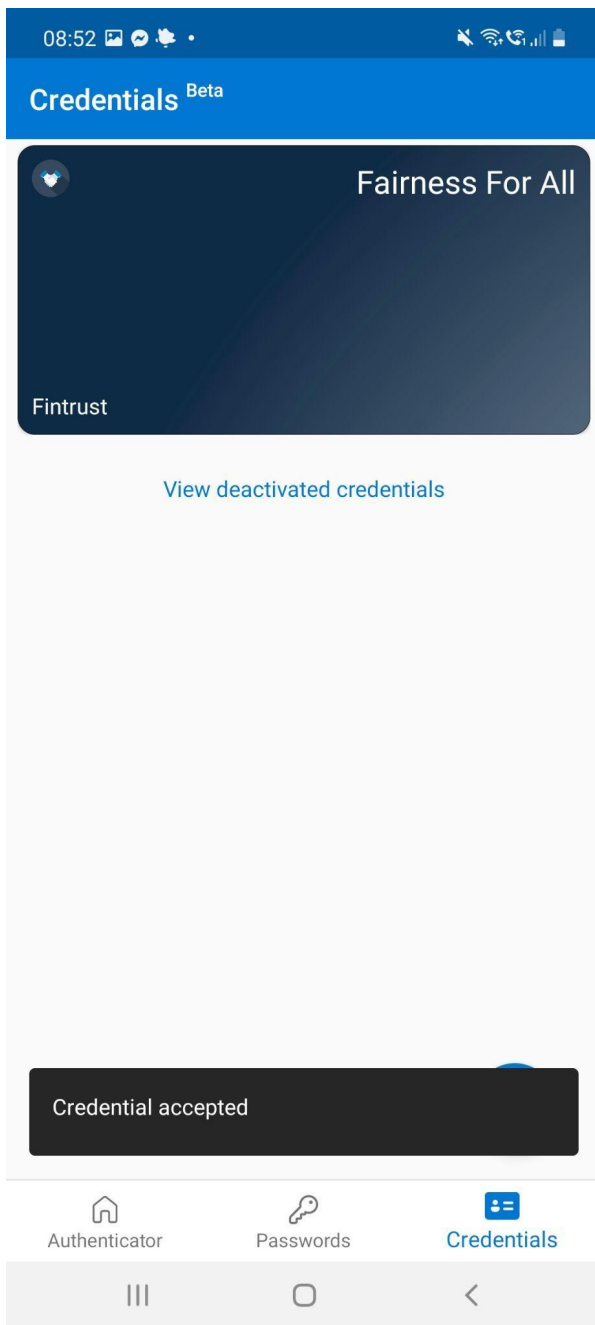


Signed in   
login.microsoftonline.com

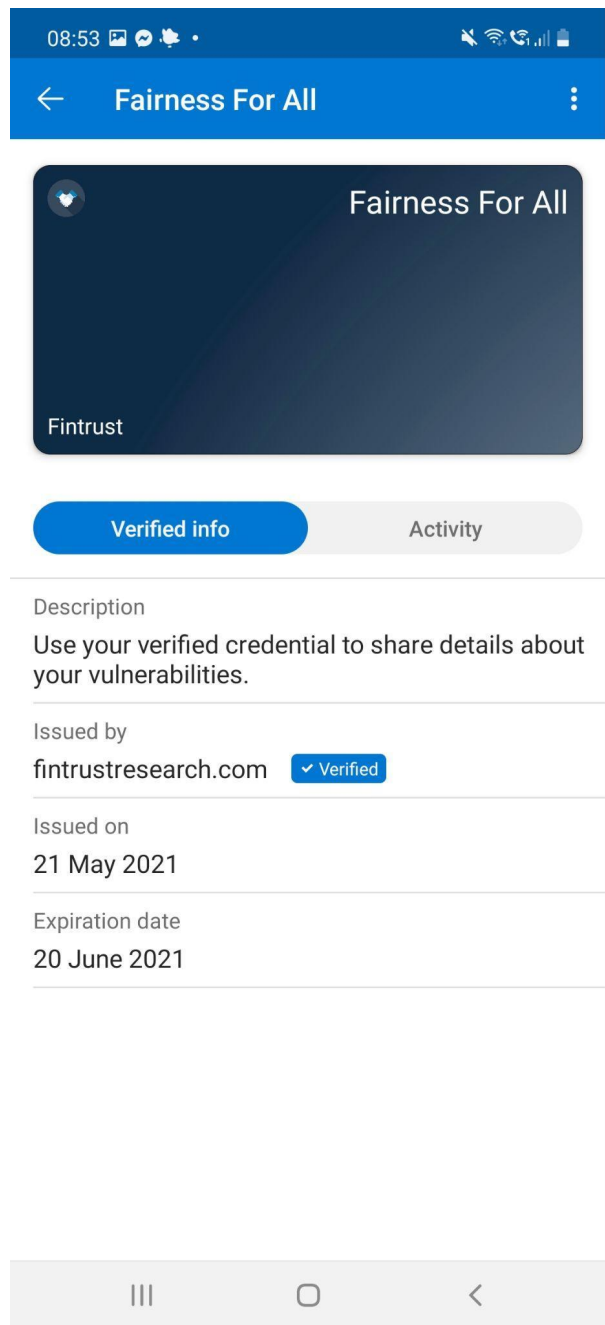
 



**Step 3** New card accepted into client



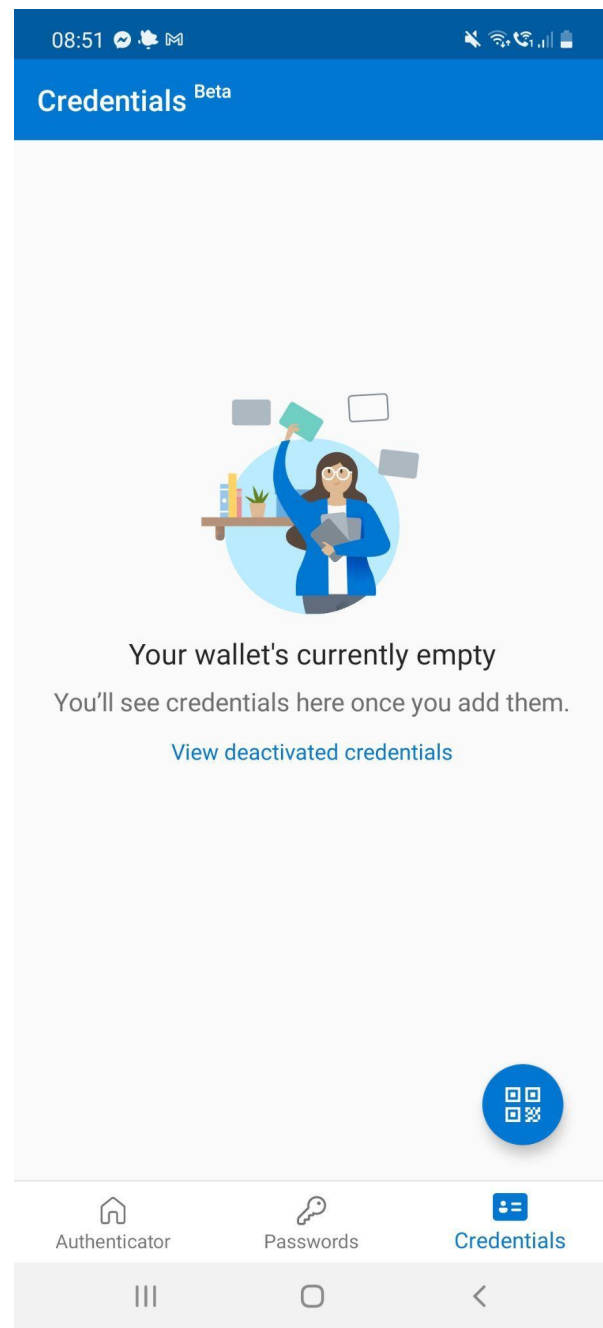
**Step 4** View credentials details



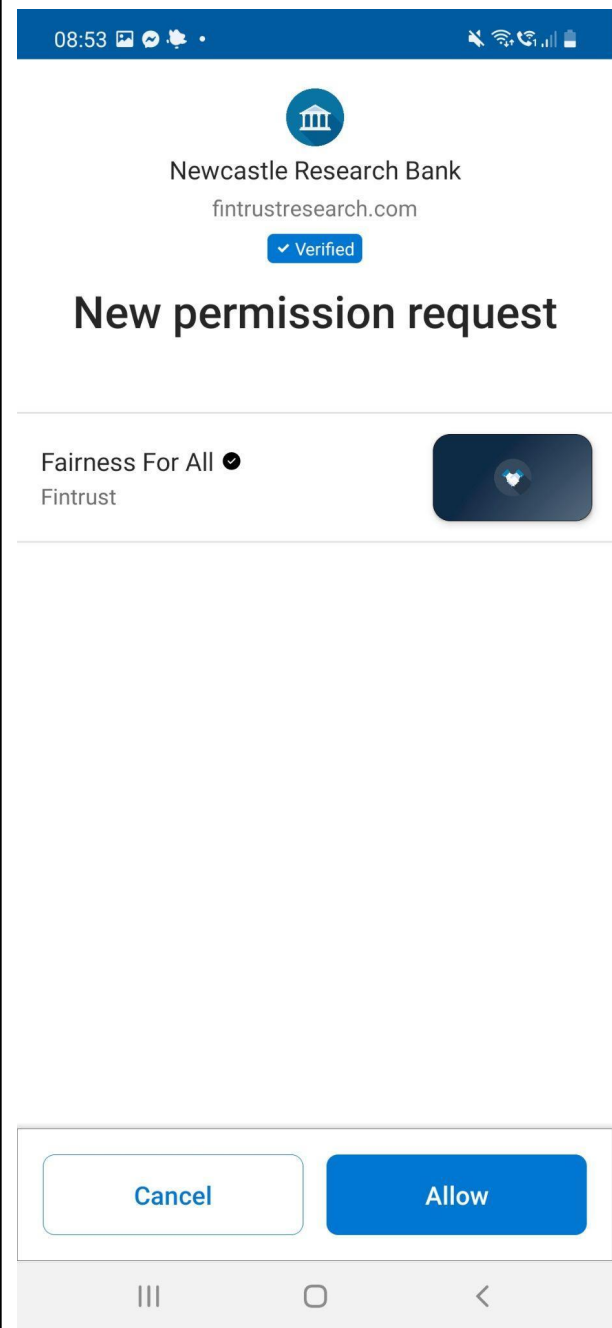
## Presentation Flow

After a user has successfully received a Verifiable Credential in Authenticator, the cards can be used to verify claims. To initiate a presentation flow, the user will scan a QR code on the verifiers website. The following is an example of what you should see on the Authenticator side.

**Step 1** Open client to scan QR code



**Step 2** Allow permission request



A successful presentation should always result in a receipt of the interaction, which is displayed in a list view in Microsoft Authenticator.

Step 3 View credentials details	Step 4 View receipt of presentations
