John Craddock Mar 2, 2022 10 min read

Introduction to the future of identity - DIDs & VCs

Updated: Mar 4, 2022

Update 02 March 2022: Please read the introduction to see what's changed. Part 2 and beyond are new content.

Part 1 in the series

With the Microsoft Azure AD Verifiable Credentials (VCs) issuer service available in your tenant, it's time to understand what VCs are and how they work with Decentralized Identifiers (DIDs). VCs and DIDs provide a new paradigm for identity, a true step into the future.

In this blog, I want to think about identity in general and then explain Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs). There are four blogs in the series, and by following the four blogs, you will learn how to issue your own DIDs and VCs using the new Microsoft service.

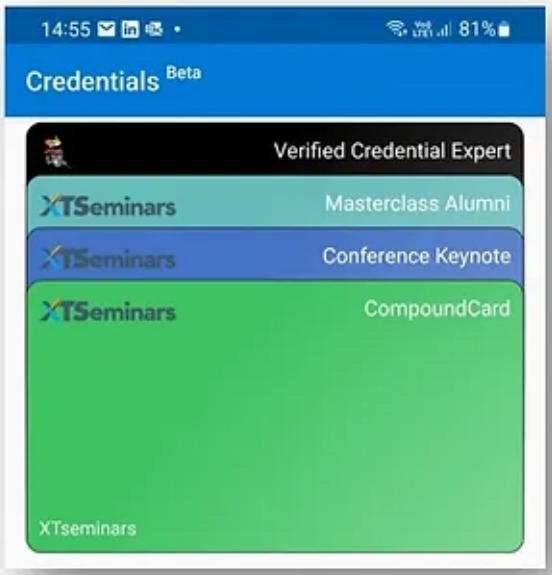
When I originally started writing this series of blogs, Microsoft provided an SDK and libraries that could be incorporated into an application to request the issuance and presentation of VCs. The libraries coded the cryptographic functions required to support VCs. Microsoft replaced the need for the libraries by implementing APIs to generate the necessary issuance and presentation requests. The APIs are also used for VC validation.

The APIs are the recommended way of working with VCs, and the SDK became depreciated at the end of 2021. I have now published three new blogs which show how the Microsoft Request Service REST API is used.

This first blog has not changed very much (apart from the intro), but blogs two, three and four are new.
The blogs in the series are:

1. Introduction to the future of identity - DIDs & VCs (this one)
2. Issuing and verifying Verifiable Credentials with the Microsoft Azure AD services
3. How to run the Microsoft Azure AD Verifiable Credentials sample app
4. Creating your own Azure AD Verifiable Credentials

Blog 4 will be published on 04/03/2022.
By the end of blog 4, you will be issuing your own VCs with claims taken from different sources

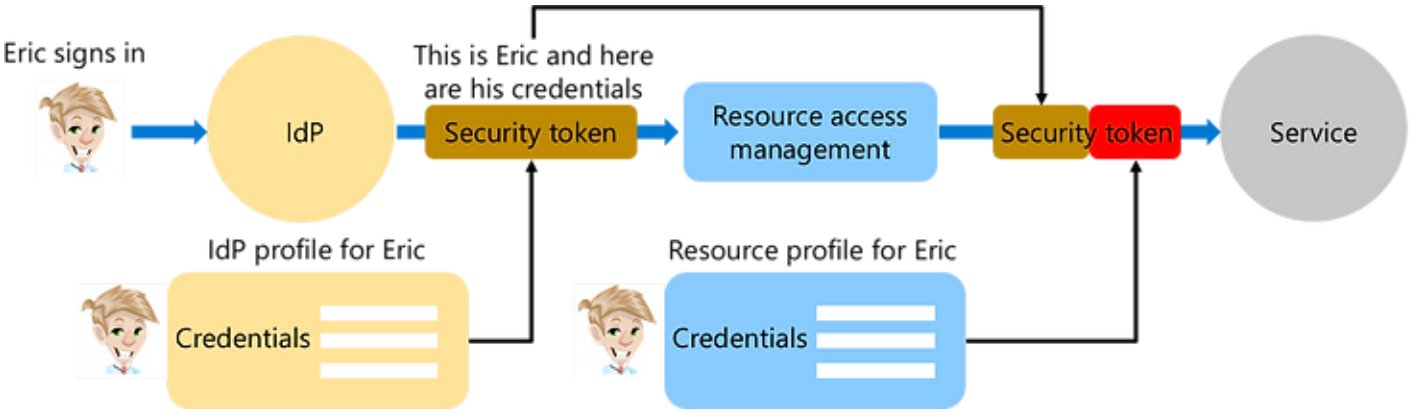


Different types of verifiable credentials

For now, read on for an introduction to DIDs and VCs.

Let's start by considering the question What is identity and access control?

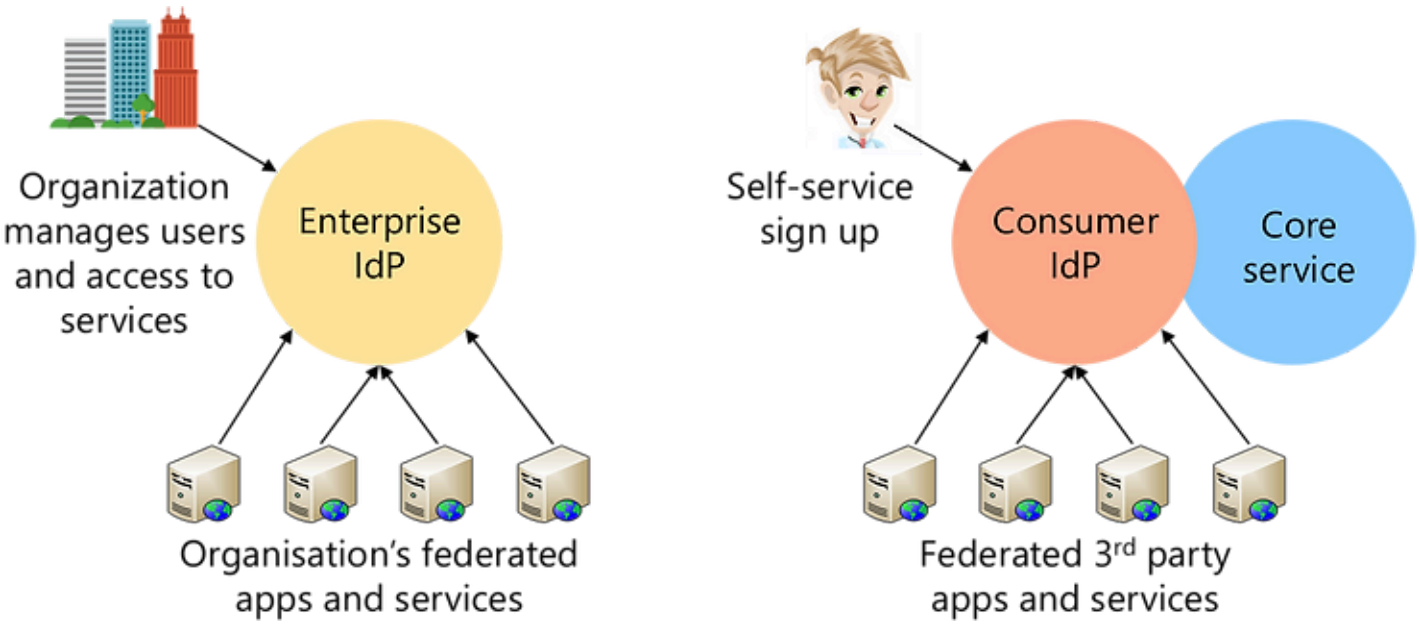
Identity is about knowing who someone is; the user is identified when they are authenticated. Access control manages access to resources based on the individual's credentials. I've used the term credential to refer to "known" information about the user. Traditionally an individual's credentials have been set by the systems that authenticate the user and/or control access to a resource.



Setting user credentials

Identity and access control could be governed by a single entity, such as a website authenticating users via its own accounts database and controlling access to the site's resources. Alternatively, a central identity provider (IdP) can authenticate the user, and individual resources implement access control based on the user's credentials.

If our systems use industry-standard authentication protocols such as SAML or OpenID Connect / OAuth 2.0, one organization can manage the authentication and another organization control access to the resources. This is a federated solution, and there are many industry players providing enterprise Identity and Access Management (IAM) services including, Microsoft, Okta, OneLogin, Ping Identity. These enterprise IAM services are full-featured, allowing an organization to manage their own users and application access. In addition to the enterprise IAM services, there is a range of consumer identity services such as Google(Gmail), Microsoft(MSA), Amazon, and Facebook. These consumer services allowing self-service sign up for the core service, for example, Gmail. However, they also provide federated sign-in for any resources that are configured to trust the IdP to authenticate users.



Enterprise versus consumer identity providers

Identity providers need to be able to uniquely identify a user and provide a method of authenticating that user. Traditionally this has been done with a username and password. When a user is registered with an IdP, the user will be identified by a username and password. When the user initially signs in, what do we know about them? Nothing. All the system knows is that the same user has returned. If it is required to know more information about the user, some form of identity proofing will need to be implemented. This could be as simple as verifying an email address by sending the user an email and requesting that they respond, through to asking the user to attend an interview with appropriate documents, and performing in-person verification.

Your digital identity

To live in the digital world, you will probably have a corporate user account and then accounts with multiple IdPs (Google, Facebook etc). We will have access to resources based on all these different accounts. Are we in charge of our identity? The answer to that is a BIG NO. When we leave an organization, we lose our corporate identity. A consumer IdP may go bust or block our account. Once again, we have lost our identity. In both cases, we have lost the ability to prove who we are to the different resources that trust the IdP. Of course, in a corporate world, it's a good

thing that when we leave, we can no longer access the organization's systems, but what about other systems that we signed up to using our work identity?

If the potential to lose your identity wasn't bad enough, the other major pain point is that you are constantly being asked to prove things about yourself. You open an online bank account. You have to prove who you are. Open a savings account with a different bank. Once again, you are sending off all the same documents as part of the required identity proofing. And it goes on and on.

Identity in the real-world

Do we continuously lose our identity as we move between jobs, countries, relationships, and hobbies in the real world? It may feel like it sometimes, but in reality, the answer is No. When we are born, we have biometric characteristics, our first set of credentials. Throughout our life, we gain other credentials—birth certificate, education achievements, driving licence, passport, marriage certificate and so on.

When we are stopped for speeding, how do we prove who we are to the police officer? We present our driving licence. Provided the photo matches and the driving licence has the correct hologram, the police officer may accept this as proof of who we are. For extra surety, she may well check that the licence has not been revoked. Once she knows the document is genuine, she can trust all the credentials it holds, age, address, and so forth.

Let's step back a moment and think about what's happened in this scenario. I had the license in my wallet and presented it to the police officer. She validated the licence by checking:

- It was about me by matching the photo ID
- That it came from an issuer she trusted by examining the document hologram
- Getting extra surety by checking for revocation

What my identity? My image, who can take away? No one!

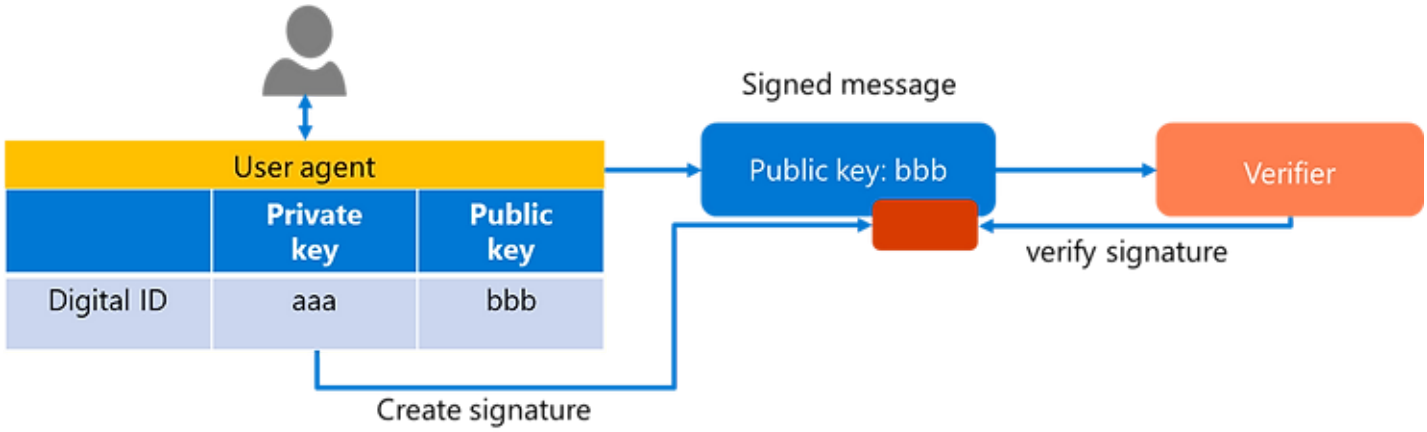
Digital identity mimics real-world identity

How would it be if we could digitally mimic the driving licence scenario? Well, now we can with DIDs and VCs.

My image is replaced with a globally unique digital ID which I generate and own. I will go into the details of this digital ID later in this blog; for now, let's keep it simple. Of course, being a mere human, I cannot generate globally unique IDs, so I leave that to my user-agent (also called a wallet). Microsoft has implemented a user-agent to do this in the Microsoft Authenticator App that you can install on your phone. This is the same app that does all those great things, such as multi-factor authentication and passwordless sign in to Microsoft 365/Azure AD. Thank you, Microsoft, for not cluttering up our phones with yet another app.

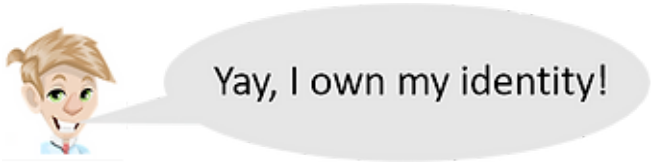
My globally unique ID consists of a cryptographic private/public key pair. The private key is securely held by the user-agent and never leaves the agent. Using the private key, my agent can digitally sign a message which includes the public key and send it to a recipient service. Using the public key (contained in the message), the recipient can validate the signature. The recipient now knows that the message's sender owns the private key. The private key is unique to the user and never leaves their possession. Consequently, the message has been signed by the user's

globally unique digital identity (private key). This can be confirmed by anyone using the user's globally unique associated public key.



Signing and verifying a message

This is a great start. The recipient service could store my public key and maybe some characteristics about my interaction with the service. When I return to the service, I supply another signed message, and the service can uniquely identify me as a returning user. We have just thrown away the need for usernames and passwords together with their associated vulnerabilities. We have also eliminated the requirements for an identity provider.



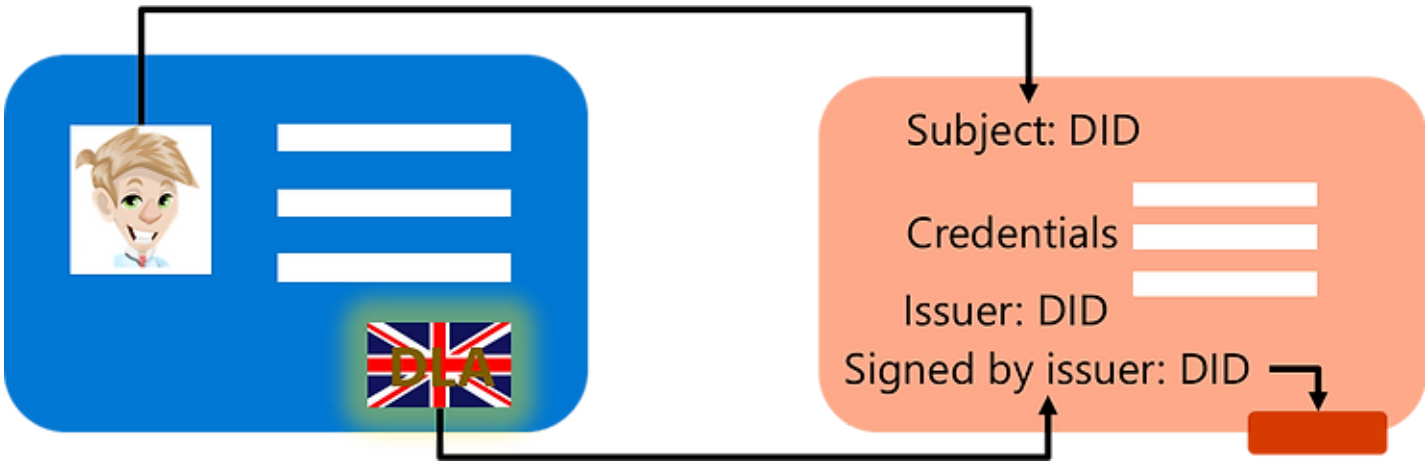
Eric now creates and owns his identity

The FIDO 2 standards are implemented in a similar way using public/private keys. Traditionally, the next step would be for the service to do some form of identity proofing and create a user profile that contains the appropriate credentials.

Back to the driving license and the police officer. Once I have passed my driving test, I can apply to the Driving Licence Authority (DLA) to issue a licence. The DLA will ask me for specific documents, including my test certificate, my photo, DOB, proof of address etc. Today they create a licence containing my photo to prove it belongs to me, appropriate credentials, and a hologram that confirms it was issued by the DLA.

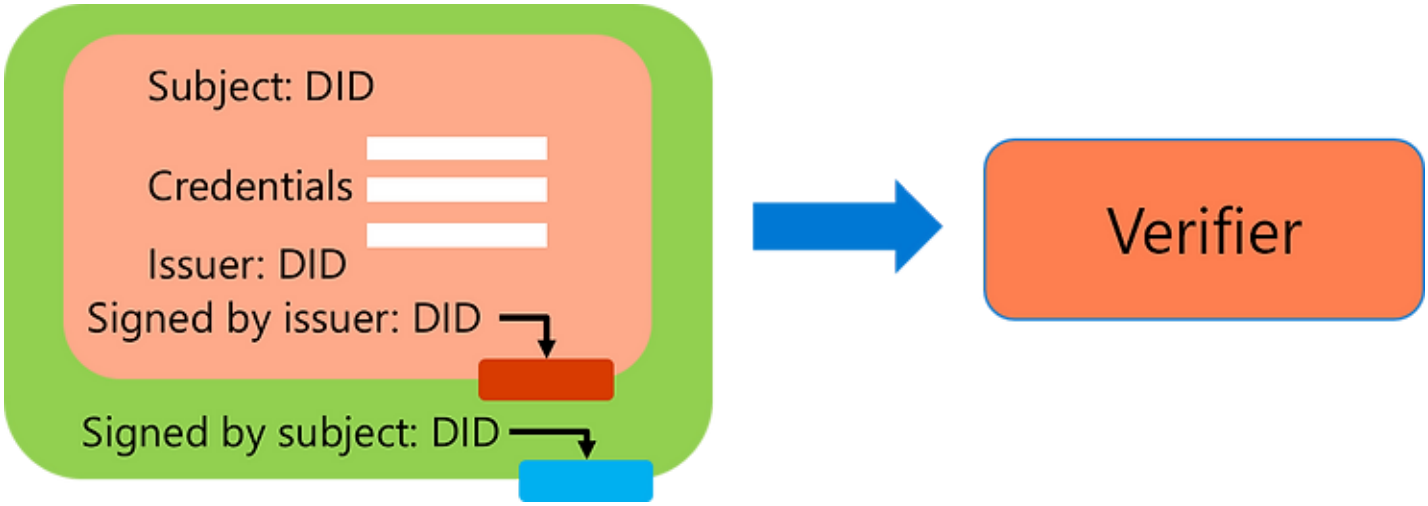
Let's go digital!

I apply for a licence, submitting my digital ID (public key) rather than supplying my photo. I furnish the other documents necessary (which could be done digitally). The DLA creates a digital record that includes my credentials, my digital ID (the subject), the DLA's digital ID (the issuer) and is signed by the DLA. I store this in my user-agent.



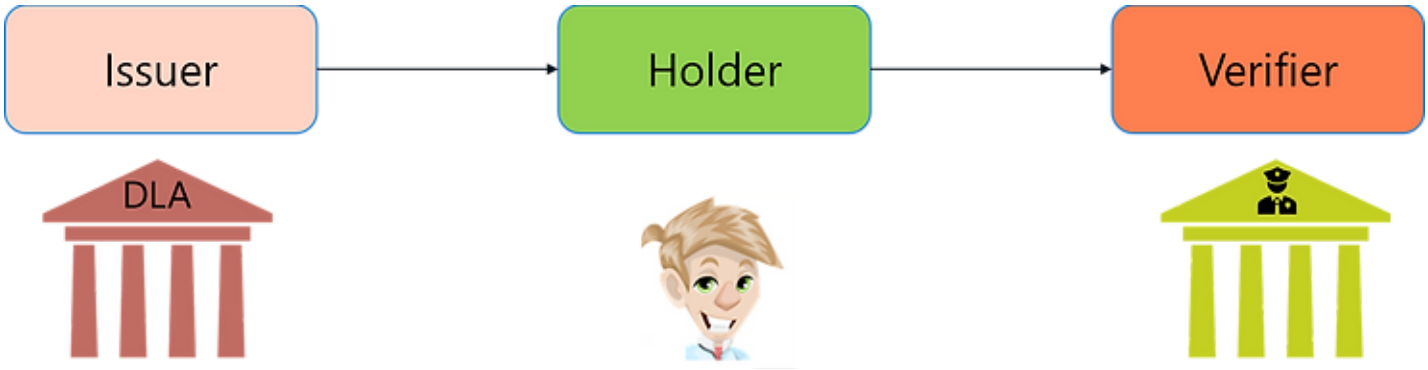
Physical to digital driving licence

The police officer questions my speeding (I am sure my Fiat 500 was going that fast!) she asks me to show her my driving licence. I unlock my phone (with a biometric) and use the agent to submit my driving licence to be verified. My agent digitally signs the submission using my private key, which proves I am the licence owner (only I have the private key, it has never left my wallet.)



Presenting the driving licence for verification

We now have the trio of issuer, holder and verifier, which is fundamental to verifiable credentials. A little like the three musketeers all supporting each other. “All for one and one for all, united we stand divided we fall.”



The three musketeers

The issuer, holder and verifier all have their own unique digital identity. I left out a bit of detail above. When the police officer asks me to submit my licence, her systems will make a presentation request which is digitally signed by the verification system. I then have the opportunity to decide if it is safe to submit the credentials to that system.

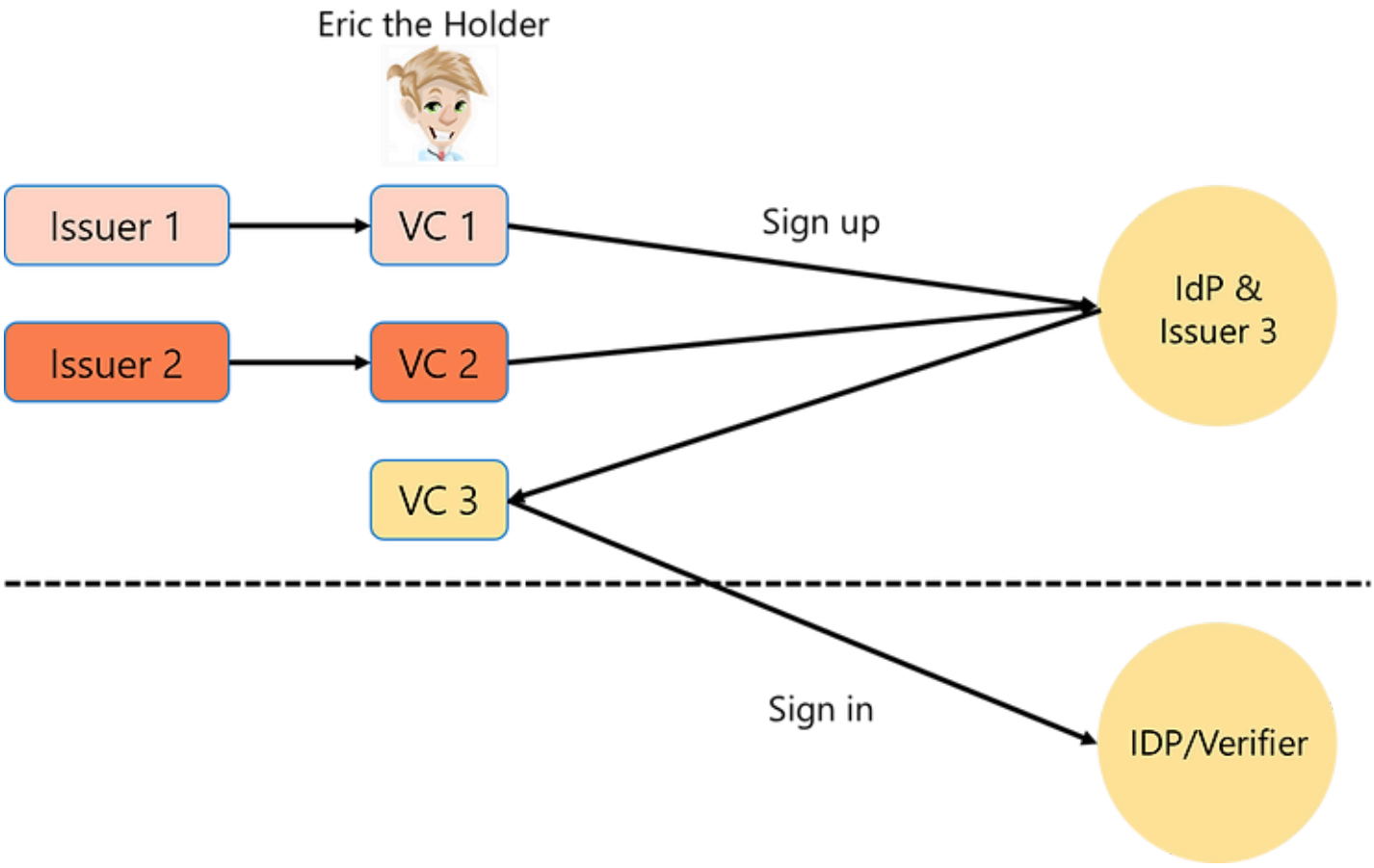
I have just introduced you to the world of verifiable claims (VCs). A key feature of VCs is that as the holder, I can submit a VC to multiple verifiers and provided they trust the issuer, they can all accept my credentials. This avoids me having to prove my credentials to numerous services. Prove the credentials once to an issuer and use them everywhere. Don't you just love that? I know I do.

If you are wondering how you trust the issuer, you will need to read the section below on DIDs

A world of endless possibilities

A verifier, via a presentation request, could ask for multiple VCs from different issuers or just a single credential. Using Zero-Knowledge Proof (ZKP), it will also be possible to prove to a verifier that you are over 21 without submitting your DOB.

Here's a thought, does our IdP need a user accounts database? When a user signs up to use our service, we make a presentation request to the user. The user submits the appropriate VCs from issuers we trust, and then we issuer a VC that proves the user is approved for access to our services. When the user subsequently signs in to our service, the VC that we issued is presented and we verify it.



Do we need to hold details of the user in an accounts database? I'll leave you to ponder that question.

This is an inspiring time for identity and open to ideas from many sources. You will be glad to know that there are emerging standards for VCs. If you want to know more, search the WC3 for verifiable claims https://www.w3.org/TR/?title=verifiable and the Decentralized Identity Foundation https://identity.foundation/

Microsoft has recently made its verifiable claims issuance service available in public preview. In my next blog, I will take you through issuing your own VCs and show you what's going on under the hood. If you want to get started now, a good starting point is here.

Before I leave this blog, I want to look at digital identity in more detail and, in particular Decentralized Identifiers (DIDs).

Decentralized Identifiers

I left you waiting for an answer as to how you could trust an Issuer. So here's it is.

The issuer, holder, and verifier are all signing messages with their private key. To validate the signatures, we need to know the member's public key, but we need more information than that. For example, we need information about the cryptographic algorithm used to sign the message and how we can validate the public key belongs to a particular entity.
All the relative information is contained in a DID document. Here is an extract from a DID document:

```
"publicKeys": [
  {
    "id": "sig_5eff18cb",
    "publicKeyJwk": {
      "crv": "secp256k1",
      "kty": "EC",
      "x": "r54B3SxP7aloHFKtkyoATKK1wFSPKFDZJ9OAnnophaY",
      "y": "omNmm9EduxdVWhO_sICJrsEMutAe6eFmjviG0eCo2dA"
    },
    "purposes": [
      "authentication",
      "assertionMethod"
    ],
    "type": "EcdsaSecp256k1VerificationKey2019"
  }
],
"services": [
  {
    "id": "linkeddomains",
    "serviceEndpoint": {
      "origins": [
        "https://learn.xtseminars.co.uk/"
      ]
    },
    "type": "LinkedDomains"
  }
]
```


You will see all the details of the public key and a service endpoint where the issuer has published a JSON document. https://learn.xtseminars.co.uk/.well-known/did-configuration.json.

This document binds the entity's DID document to the entity's domain. We now know that the DID belongs to learn.xtseminars.co.uk

When an entity creates a signed message, the DID document can be contained in the signed message or referenced. If it's referenced, then the DID document is stored on a Decentralized public ledger (blockchain). The reference is in the form:

- did:method:method-specific-identifier

The method identifies the underlying ledger technology and the method-specific-identifier uniquely identifies the DID document on the ledger. Microsoft is currently using the ION method to store their VC DIDs (more details in a later blog), and the DID is in this form:

- did:ion:EiBtLiugj5KjXko8o8Tczdg5KXN93Y3dn8TP5j6neJjpkw

Issuers and verifiers will almost certainly use linked domains to prove who they are. When it comes to a user in most situations, there is no need to publish their actual identifier. All the user needs to prove is that a VC was issued to them. The user's agent could create a new DID for every verifier that they interact with.

A user's complete identity is represented by all the VCs that contain one of the user's DIDs as the subject.

I can hear you thinking…. The VC was issued to a particular digital ID. How can a user present the VC when their digital ID has changed?

Read my next blog, where I will dive in under the hood.

Thank you for reading this blog, and stay tuned for the next one. Please let your friends and colleagues know about the blog via LinkedIn and Twitter, don't forget to include me, so I see it! twitter: @john_craddock and/or www.linkedin.com/in/johnxts

My next identity masterclasses for CET and EST are in March 2022. Why don't you join me for an action-packed week?

- Monday 7th - Friday 11th March 2022 9:00 - 17:00 CET
- Monday 14th - Friday 18th March 2022 8:00 - 16:00 EST

Full details at https://learn.xtseminars.co.uk