Microsoft Identity

Deep-dive into Microsoft Entra Verified ID

Christer Ljung | Product Manager, Verified ID

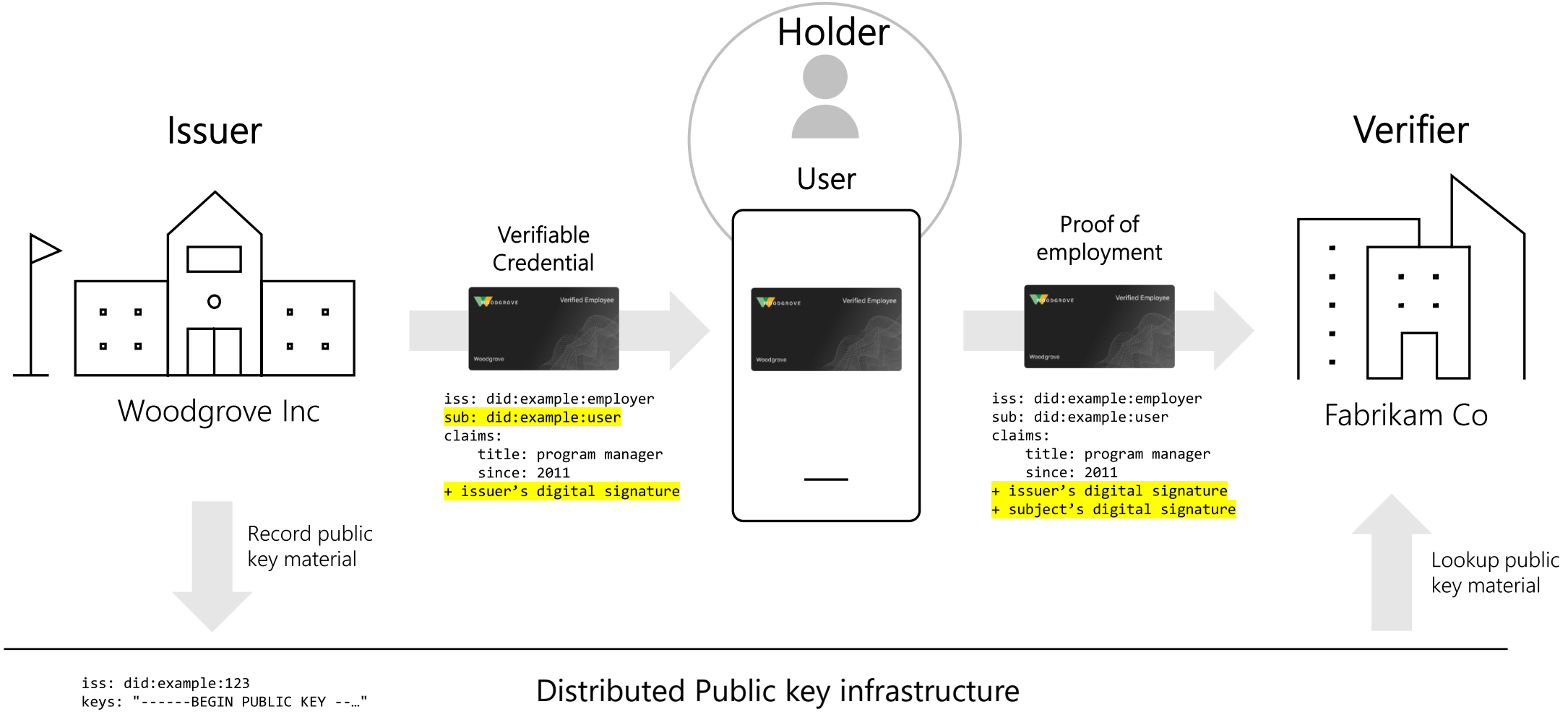
November 27, 2024

Agenda

- Decentralized Identity vs Traditional Identity
- What is Microsoft Entra Verified ID
  - How do I set it up?
  - How do I create, issue and verify credentials?
  - High assurance verification with Face Check
- APIs
- Docs & Samples
  - Generic, Entra ID, CIAM, B2C, Onboarding with TAP, Helpdesk

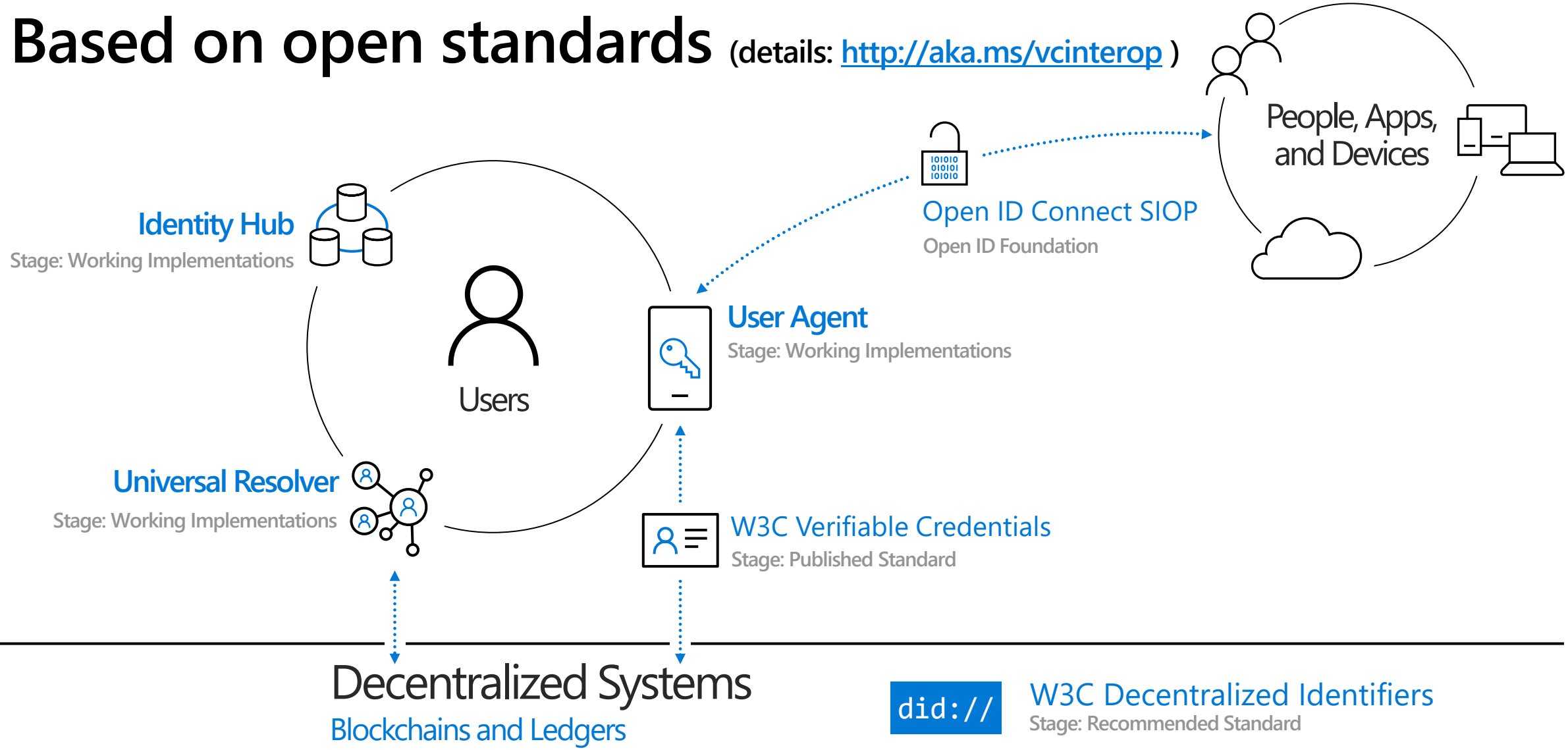
Decentralized Identity

The Three actors of Decentralized Identity



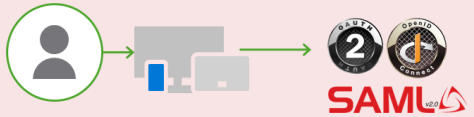
Microsoft

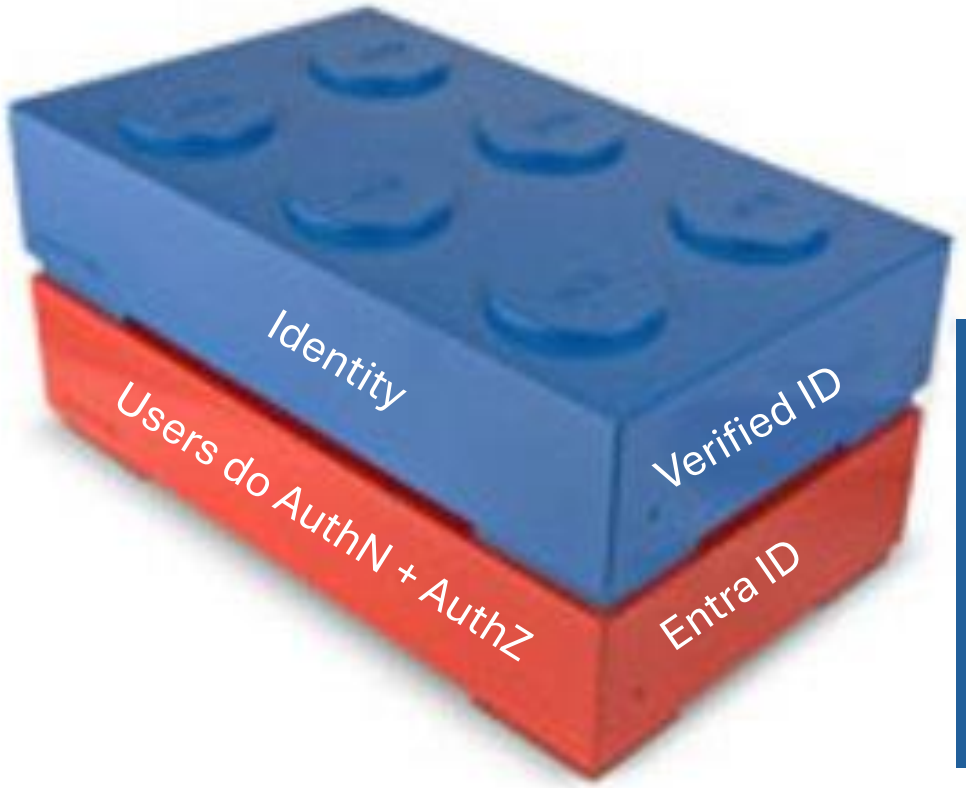
Based on open standards (details: http://aka.ms/vcinterop)



Decentralized Identity vs Traditional Identity

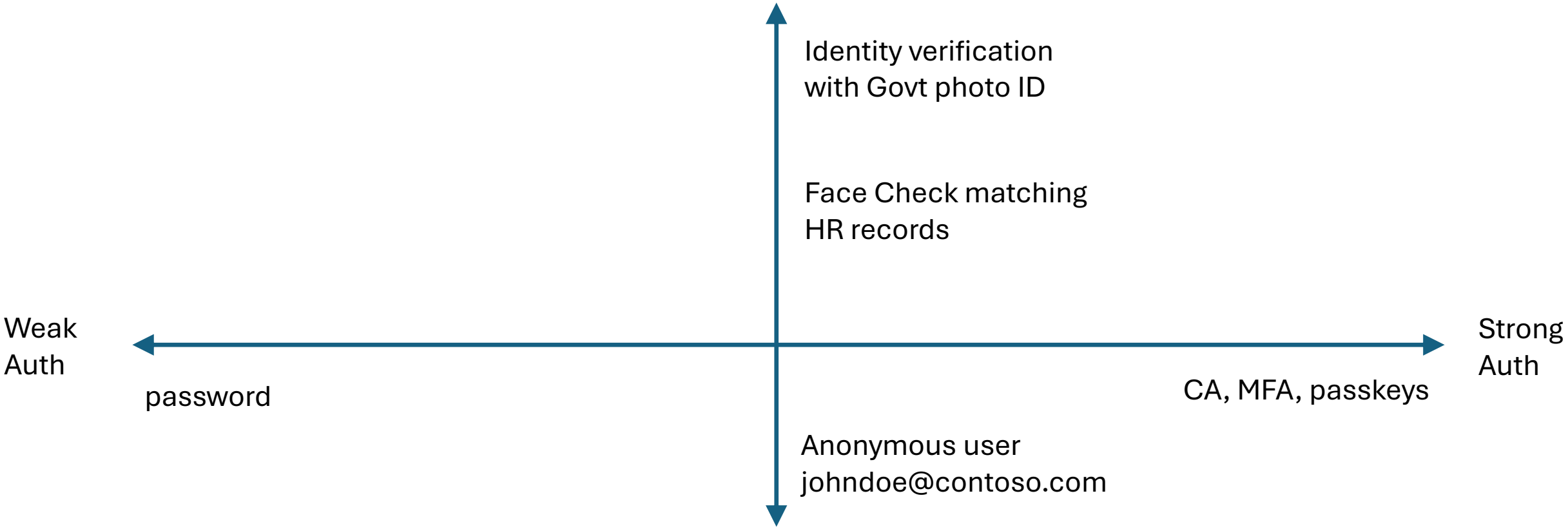
- Users authenticate themselves to a directory to get access to apps
- Who the user is opaque





- Identifies who the user really is beyond proof of possession of password
- John Doe is not opaque anymore
- Can be used inside and outside of Entra

Decentralized Identity vs Traditional Identity



Key interactions in need of greater trust & efficiency

Onboarding Shortcomings

81%

of employees feel overwhelmed by onboarding

Forbes: Why Better Onboarding Is Key To Improved Productivity

Access Challenges

75%

of the global population's personal data will be under privacy regulation by 2025

Gartner 2024 Top Five Trends in Privacy

Help Desk Vulnerabilities

$1.1 billion

Loses to impersonation scams in the U.S. in 2023

FTC Data Spotlight into impersonation scams

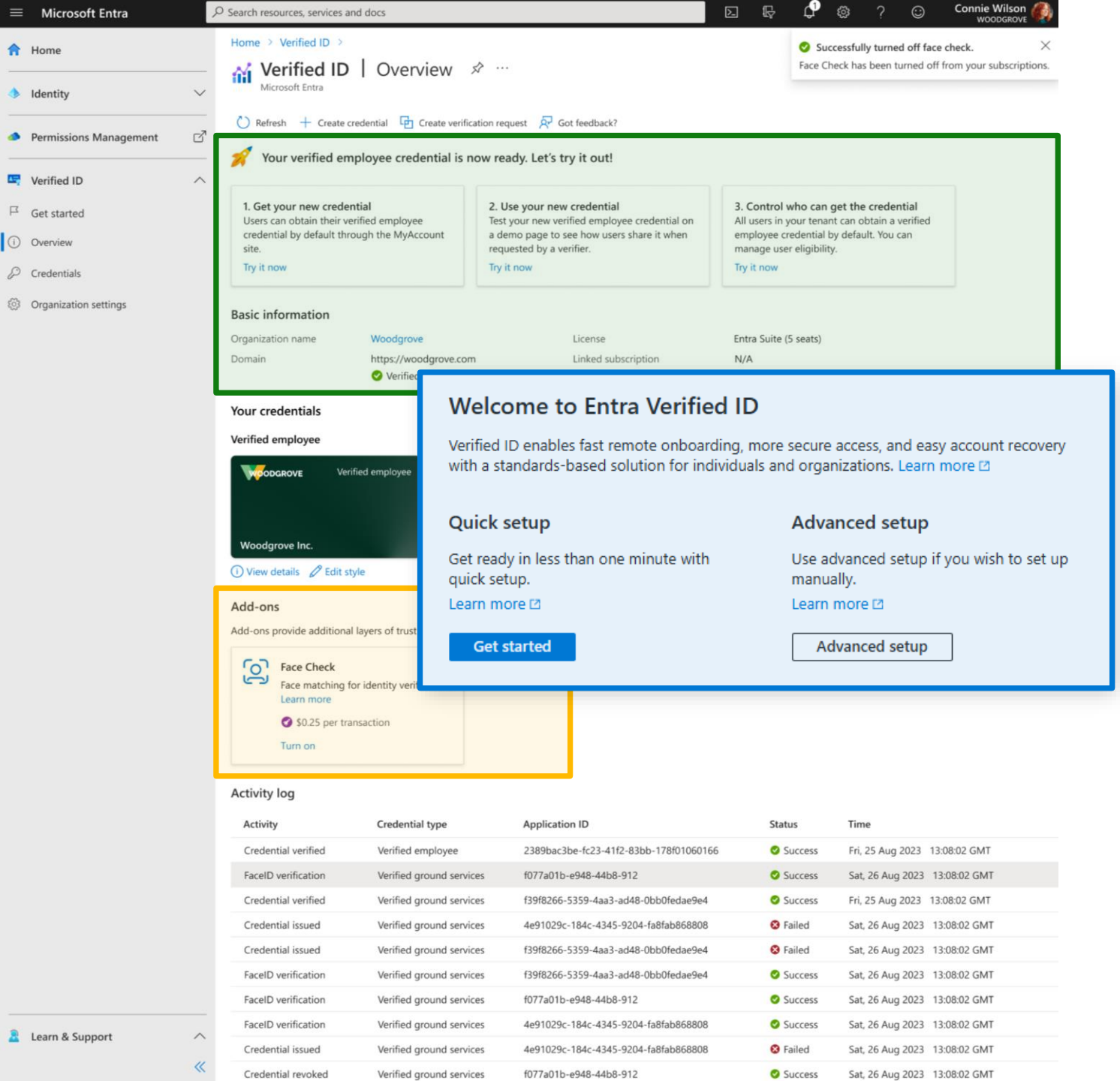
What's the impact?

Results in wasted time and lost productivity

Growing more important to know and trust who has access to your resources

Must close potential impersonation pathways

Microsoft Entra Verified ID



Protect trusted accounts from impersonation today

1. Get your Verified ID tenant ready to **issue and verify in minutes**
2. **Immediately issue and verify** employee and guest Verified IDs to selected users
3. Enable Face Check* with **one click**

* **Try Face Check for free** in Entra Suite trial

» Tutorials and resources to get started

Quick Setup

Admin follows a simple ***single step*** process to enable Entra Verified ID setup

- Global Administrator or the authentication policy administrator permission for the directory you want to configure. If you're not the Global Administrator, you need the application administrator permission to complete the app registration including granting admin consent.
- Requires the customer to be a M365 tenant with a **custom domain** (*yourdomain*.com), as an Entra ID setting.
- Shared signing key managed by Microsoft. No requirement of setting up Azure Key Vault or hosting JSON files on the web servers
- Lower RPS

Advanced Setup

Admin follows a simple ***multiple steps*** to enable Entra Verified ID setup

- Global Administrator or the authentication policy administrator permission for the directory you want to configure. If you're not the Global Administrator, you need the application administrator permission to complete the app registration including granting admin consent.
- Requires an **Azure Key Vault** to store your signing key(s).
- Requires a **webserver** for your domain to host DID Document JSON files containing public key information

Creating credentials (that can be issued later)

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Verified ID | Overview >

Create credential

Once the credential is created it will be a part of the Entra Verified ID network. Information including your company and domain name will be published so other organizations will be able to verify in their own tenant & application(s).
Learn more

Organization details

Organization: RG EU Directory

Linked domain: https://digzerolab.com/
Verified domain

Select a credential type

Credential types

- **Verified employee**
  Verified employee credential
- **Custom credential**
  Design your own credential from scratch.
- **Verified teacher (Coming soon)**
  A credential that contains the claims: name, first name, last name, title, email, pl
- **Verified student (Coming soon)**
  A credential that contains the claims: name, first name, last name, email, photo,

Next | Cancel

Create a new credential - Micro

https://portal.azure.com/?Microsoft_AAD_Decentralized 90%

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Verifiable credentials (Preview) > Issue credentials quickstart >

Create a new credential

Got feedback?

Create a custom credential that can be issued to your users. Learn more

Credential name *
A unique identifier for your credential, only displayed in the Azure Portal.

VerifiableCredentialExpert

Display file *
The display file describes the claims contained in the credential as well as the branding.

```
1  {
2      "locale": "en-US",
3      "card": {
4        "title": "Verified Credential Expert",
5        "issuedBy": "Microsoft",
6        "backgroundColor": "#000000",
7        "textColor": "#ffffff",
8        "logo": {
9          "uri": "https://didcustomerplayground.blob.core.
10         "description": "Verified Credential Expert Logo"
```

Learn how to create a display file

This is what users will see in the Authenticator app. The card branding, title and color come from the display file. The acceptance requirements, such as "sign in to your account" are covered by the rules file.

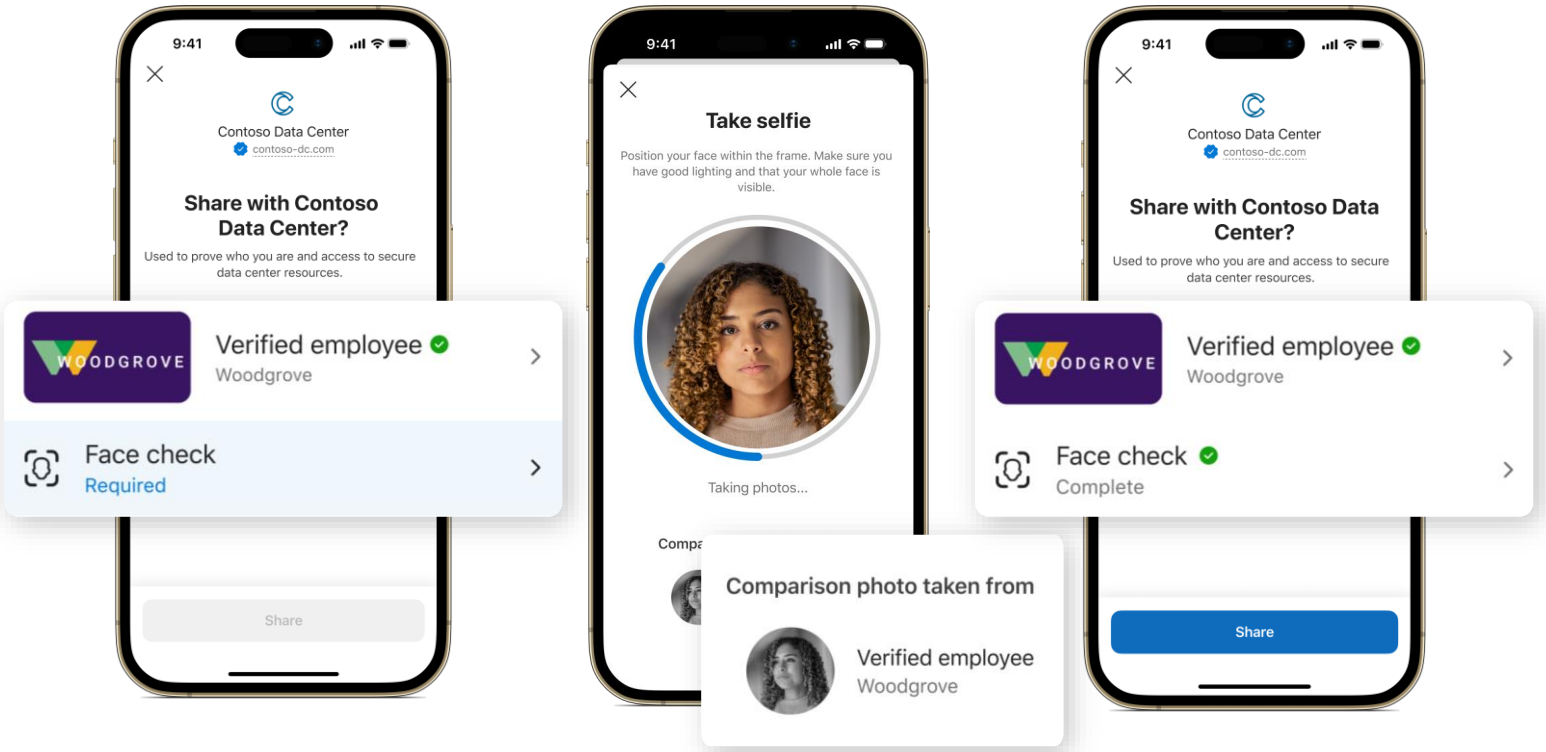
Rules file *
The rules file determines what the user needs to do to get the credentials. Include an index claim if you want to be able to search for the credential later.

```
1  {
2      "attestations": {
3          "idTokens": [
4            {
```

Microsoft Identity

Demo

GA: Real-time privacy respecting biometrics using Face Check



Present Verified ID and start Face Check

Take selfie for Face Check

Share valid Verified ID and Face Check for verification

High-assurance verification at scale

Build on trust established with Verified ID with an added layer of identity verification

Verify liveness in an instant

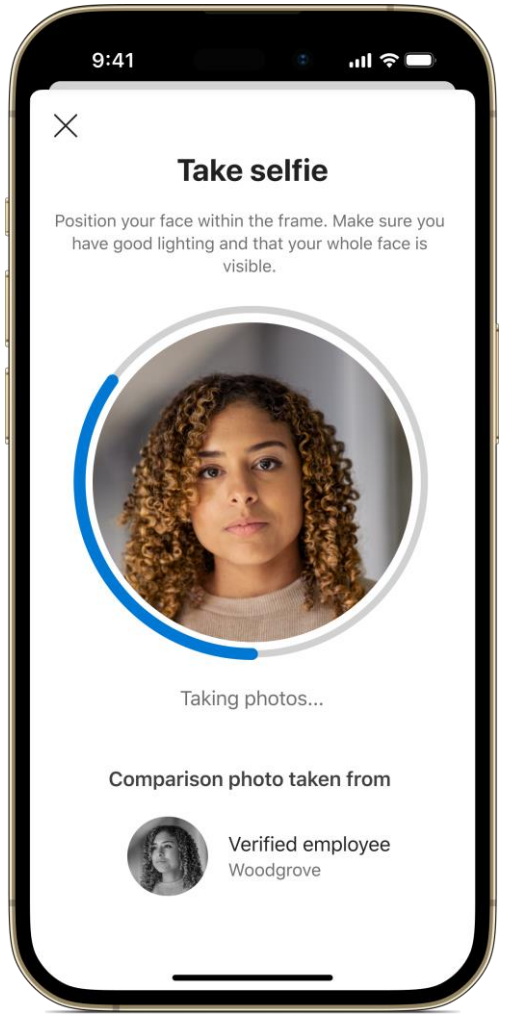
Make sure the right person is using a Verified ID in real time

Facial matching in any scenario

High assurance for onboarding, account recovery, or any Verified ID scenario

Verified Face check
Powered by Azure AI Vision API

1. User centric liveness match
2. Verifies that it is a real person in the live footage
3. Only a confidence in match is shared with the app
4. Match performed within the Verification App's cloud data boundary
5. Azure AI Vision service can detect a wide variety of spoofing techniques and conformant to ISO/IEC 30107-3 PAD (Presentation Attack Detection) standards as validated by iBeta level 1 and level 2 conformance testing



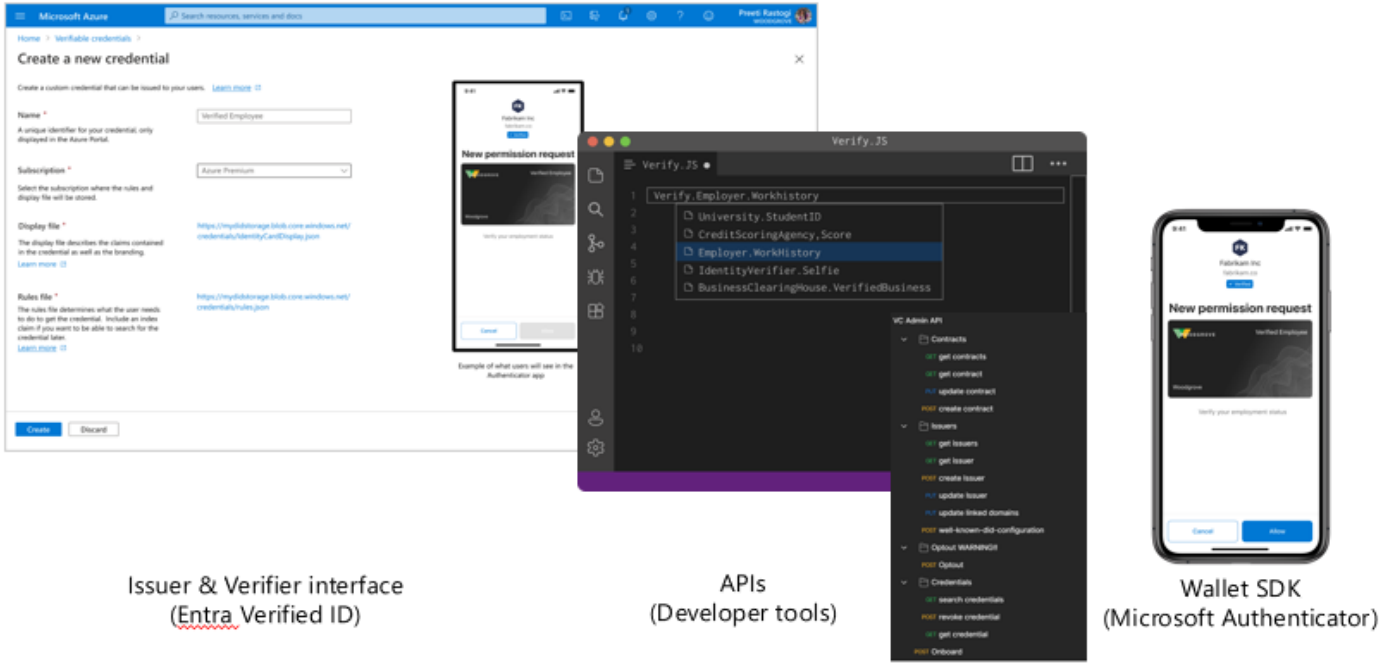
APIs

Entra Verified ID APIs

1. Request Service APIs for issue and verification
2. Admin APIs to perform admin functions and add such functions to your own custom build control panels
   1. Onboard
   2. Credential contracts
   3. Credential
3. Open-source library to integrate into branded app

Using the Microsoft Entra Wallet Library

Decentralized Identity Platform by Microsoft



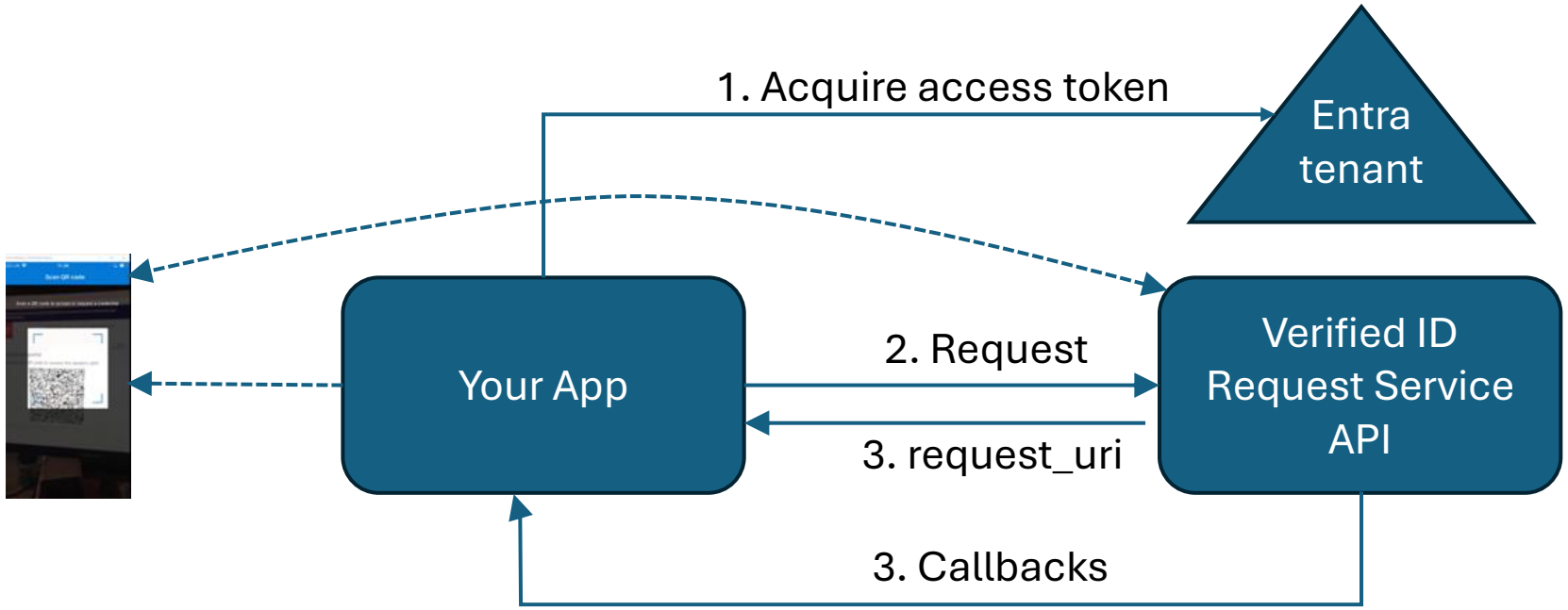
Issuer & Verifier interface
(Entra Verified ID)

APIs
(Developer tools)

Wallet SDK
(Microsoft Authenticator)

Request Service API model

- REST API
- Requires authentication from your Entra tenant
  - Your app needs and access token with permission to call APIs
- Your callback endpoints must be reachable from Azure Cloud platform



Request Service API – issuance request

POST https://verifiedid.did.msidentity.com/v1.0/verifiableCredentials/createIssuanceRequest
Content-Type: application/json
Authorization: Bearer <token>

```
{
  "authority": "did:web:verifiedid.contoso.com",
  "registration": { "clientName": “Your App’s name" },

  "callback": {
    "url": "https://contoso.com/api/issuer/myCallback",
    "state": “YOUR UNIQUE ID FOR THIS REQUEST",
    "headers": { "api-key": "OPTIONAL API-KEY for CALLBACK EVENTS“ }
  },

  "type": "VerifiedCredentialExpert",
  "manifest": "https://verifiedid.did.msidentity.com/v1.0/tenants/<tenant
id>/verifiableCredentials/contracts/<contract id>",

   "claims": {
    "given_name": "Megan",
    "family_name": "Bowen“
  },

  "pin": { "value": "1984", "length": 4},
```

Requestor’s authority (you)

Seen in the Authenticator

Your endpoint

Your id for this request

Your optional api-key

What are we issuing?

Claims for the VC

Optional pin code

Request Service API – issuance response

HTTP 201 Created
Content-Type: application/json


```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "url": "openid-vc://?request_uri=https://verifiedid.did.msidentity.com/v1.0/00001111-aaaa-2222-bbbb-3333cccc4444/verifiableCredentials/request/799f23ea-5241-45af-99ad-cf8e5018814e",

  "expiry": 1622227690
}
```


Verified ID's id

Request URI for wallet.

1. Shown as a QR code if cross device
2. Used as a deeplink if your-app is running on mobile

Request Service API – issuance callbacks

POST https://contoso.com/api/issuer/myCallback
Content-Type: application/json
api-key: OPTIONAL API-KEY for CALLBACK EVENTS

```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "requestStatus": "request_retrieved",

  "state": "YOUR UNIQUE ID FOR THIS REQUEST",

}
```

Verified ID's id

QR code scanned

Yours state passed in request

```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "requestStatus": "issuance_successful" || "issuance_error" ,

  "state": "YOUR UNIQUE ID FOR THIS REQUEST",

}
```

After wallet has completed issuance

Request Service API – presentation request

POST https://verifiedid.did.msidentity.com/v1.0/verifiableCredentials/createPresentationRequest
Content-Type: application/json
Authorization: Bearer <token>

```
{
  "authority": "did:web:verifiedid.fabrikam.com",
  "registration": { "clientName": “Your App’s name" },

  "callback": {
    "url": "https://fabrikam.com/api/verifier/myCallback",
    "state": “YOUR UNIQUE ID FOR THIS REQUEST",
    "headers": { "api-key": "OPTIONAL API-KEY for CALLBACK EVENTS“ }
  },
  "requestedCredentials": [
    {
        "type": "VerifiedEmployee",
        "acceptedIssuers": [ "did:web:verifiedid.contoso.com“ ],
        "configuration": {
          "validation": {
            "allowRevoked": false, "validateLinkedDomain": false },
            "faceCheck": { "sourcePhotoClaimName": "photo",
                           "matchConfidenceThreshold": 70 }
```

Same as issuance

What are we looking for?

Credential Type

Issued by whom?

Validations of the presented VC

If you want to include a Face Check

Request Service API – presentation response

HTTP 201 Created
Content-Type: application/json


```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "url": "openid-vc://?request_uri=https://verifiedid.did.msidentity.com/v1.0/00001111-aaaa-2222-bbbb-3333cccc4444/verifiableCredentials/request/799f23ea-5241-45af-99ad-cf8e5018814e",

  "expiry": 1622227690
}
```


Verified ID's id

Request URI for wallet.

1. Shown as a QR code if cross device
2. Used as a deeplink if your-app is running on mobile

Request Service API – presentation callbacks

POST https://fabrikam.com/api/verifier/myCallback
Content-Type: application/json
api-key: OPTIONAL API-KEY for CALLBACK EVENTS

```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "requestStatus": "request_retrieved",

  "state": "YOUR UNIQUE ID FOR THIS REQUEST",

}
```

Verified ID's id

QR code scanned

Yours state passed in request

```
{
  "requestId": "799f23ea-5241-45af-99ad-cf8e5018814e",

  "requestStatus": “presentation_verified" || “presentation_error" ,

  "state": "YOUR UNIQUE ID FOR THIS REQUEST",

  "verifiedCredentialsData": ... Next page ... (on presentation_verified)

}
```

After wallet has presented VC

Request Service API – presentation callbacks

POST https://contoso.com/api/verifier/myCallback
Content-Type: application/json
api-key: OPTIONAL API-KEY for CALLBACK EVENTS

```
"verifiedCredentialsData": [
{
    "issuer": "did:web:verifiedid.contoso.com",
    "type": [ "VerifiableCredential", "VerifiedEmployee" ],
    "claims": {
      “displayName": "Megan Bowen",
      ... etc ...
    },
    "credentialState": { "revocationStatus": "VALID" },
    "domainValidation": { "url": "https://contoso.com/" },
    "issuanceDate": "yyyy-MM-ddTHH:mm:ssZ",
    "expirationDate": "yyyy-MM-ddTHH:mm:ssZ"
}
],
```

Who issued the VC and what type is it?

Claims in the VC

State & metadata about the VC

Demo (if time)

Admin API

Admin API - Authority

Methods	Return Type	Description
Get Authority	Authority	Read properties of an authority
List Authority	Authority array	Get a list of all configured Authorities/verifiable credential services
Create Authority	Authority	Create a new authority
Update authority	Authority	Update authority
Delete authority	Authority	Delete authority
Generate Well known DID Configuration		Linked Domain verification
Generate DID Document		DID Document generation
Validate Well-known DID config		
Rotate Signing Key	Authority	Rotate signing key
Synchronize with DID Document	Authority	Synchronize DID document with new signing key

Admin API – Credential contracts

Methods	Return Type	Description
Get contract	Contract	Read properties of a Contract
List contracts	Contract collection	Get a list of all configured contracts
Create contract	Contract	Create a new contract
Update contract	Contract	Update contract

Methods	Return Type	Description
Get credential	Credential	Read properties of a Credential
Search credentials	Credential collection	Search a list of credentials with a specific claim value
Revoke credential		Revoke specific credential

Docs & Samples

Docs & Samples

- Docs – https://aka.ms/didfordevs
- Samples – https://aka.ms/vcsample

active-directory-verifiable-credentials-dotnet Public

Edit Pins | Unwatch 32

main | Branches | Tags | Go to file t | Add file | Code

cljung Added support for hashed pin code 9d46a01 · last month 277 Commits

.github	.github/PULL_REQUEST_TEMPLATE.md committed	3 years ago
.vscode	Fix launch.json .net7.0	last year
1-asp-net-core-api-idtokenhint	Added support for hashed pin code	last month
2-asp-net-core-api-user-signin	upd logo url	2 months ago
3-asp-net-core-api-b2c	upd troubleshooting	3 months ago
4-externalid-verifiedid	upd logo url	2 months ago
5-onboard-with-tap	chk EmployeeLeaveDateTime is set	3 months ago
6-woodgrove-helpdesk	upd troubleshooting	3 months ago


That's all Folks!
SUPERB wallpapers