Microsoft Entra verified ID

Verifiable Credentials, Microsoft style
TF-DLT, Sept 19, 2022

Peter Clijsters, Martin van Es, Niels van Dijk
SURF

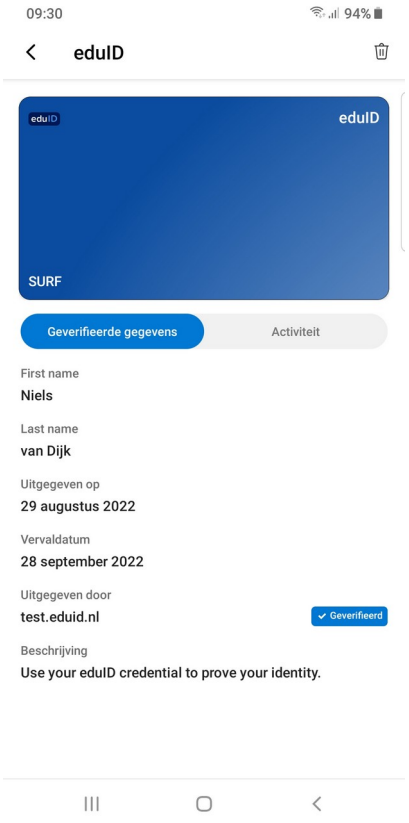
MS VC’s -Why

- VC issuence & verification capabiliteis are now part of Azure/Entra
- Many institutions have Azure in place already
- Can we leverage existing (Federated) AAI?
- Can we integrate incoming credentials?

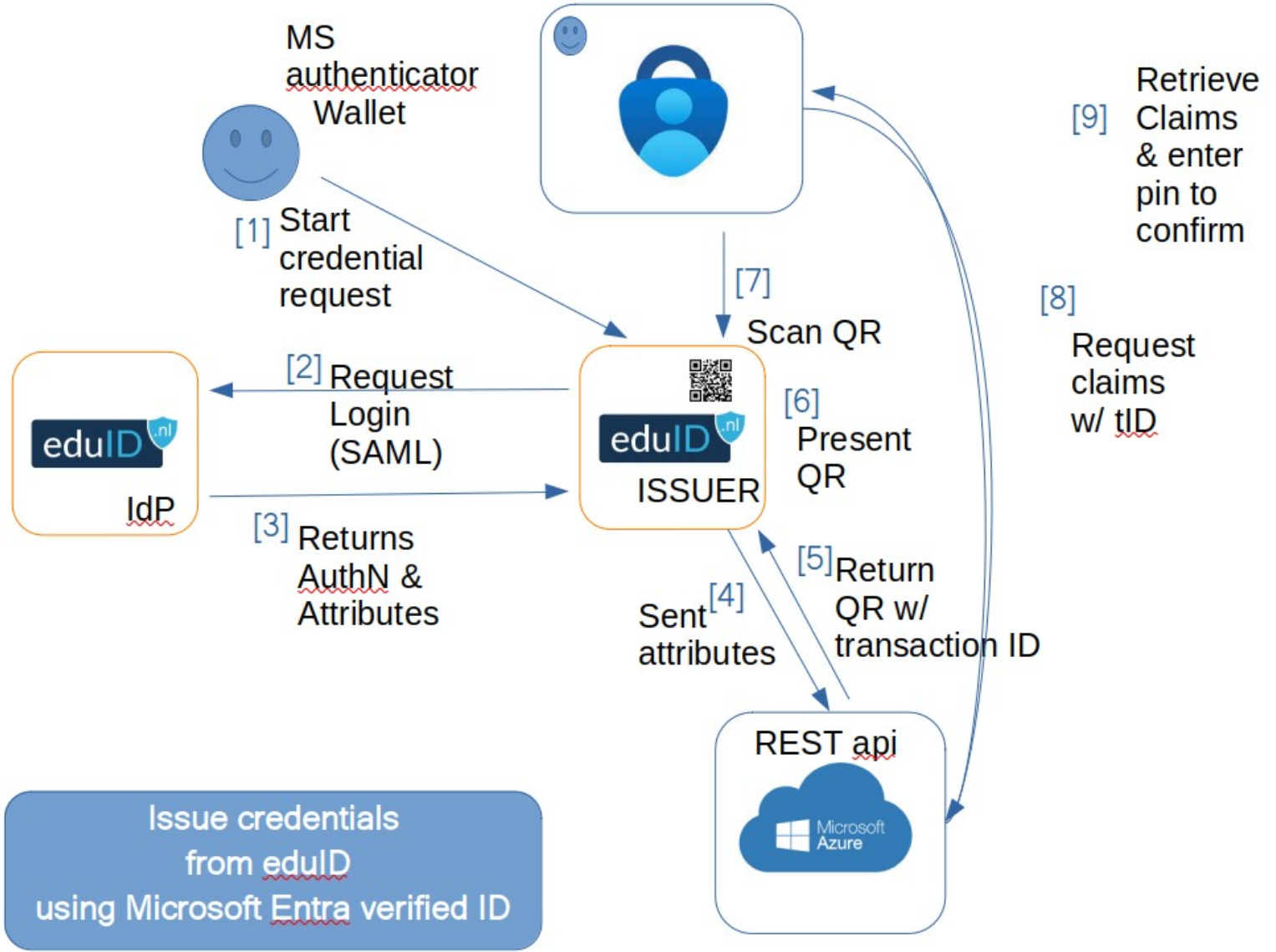
MS VC’s – Agenda

- How does it work?
- What are the requirements?
- What standards are involved?
- How is trust established?
- Findings and Conclusions?

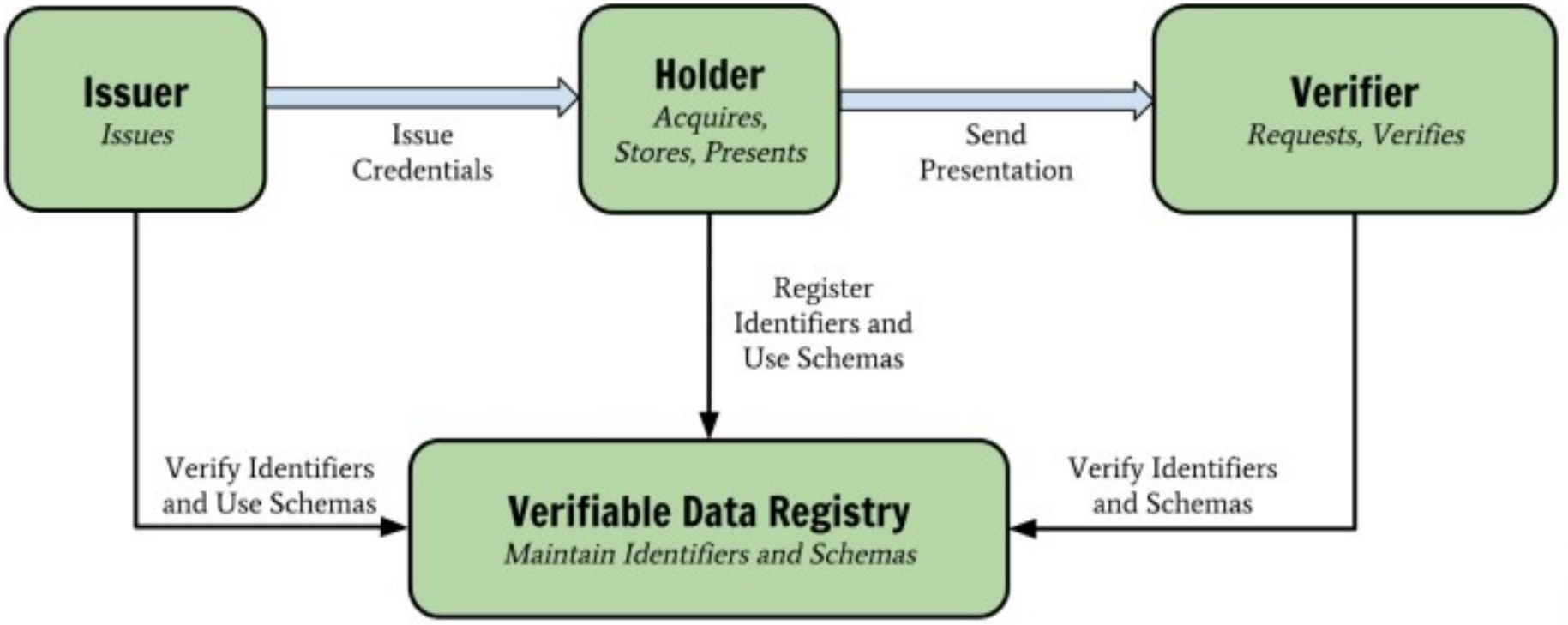
MS VC’s -Demo

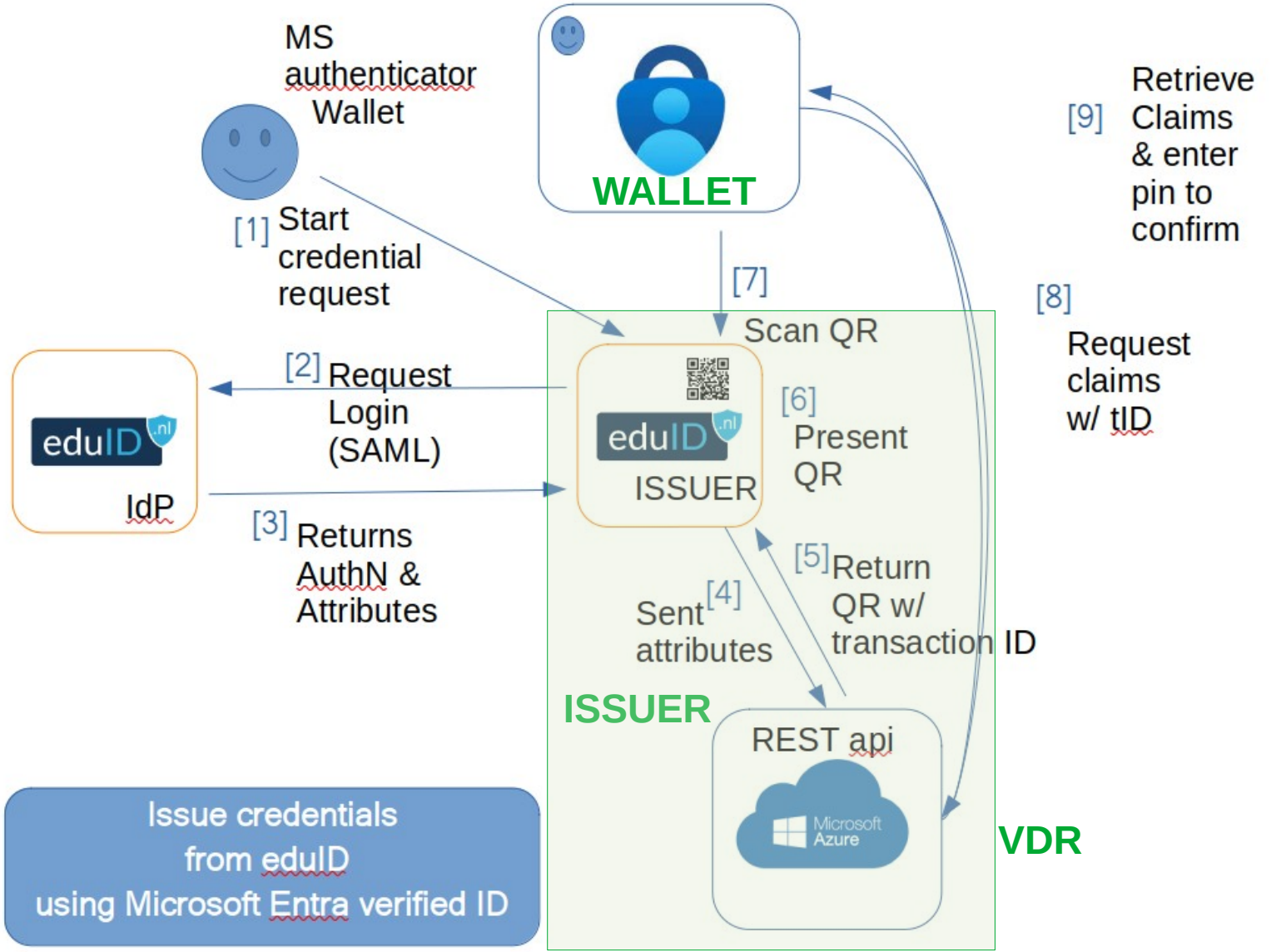

09:30
94%
eduID
eduID
eduID
SURF
Geverifieerde gegevens
Activiteit
First name
Niels
Last name
van Dijk
Uitgegeven op
29 augustus 2022
Vervaldatum
28 september 2022
Uitgegeven door
test.eduid.nl
Geverifieerd
Beschrijving
Use your eduID credential to prove your identity.

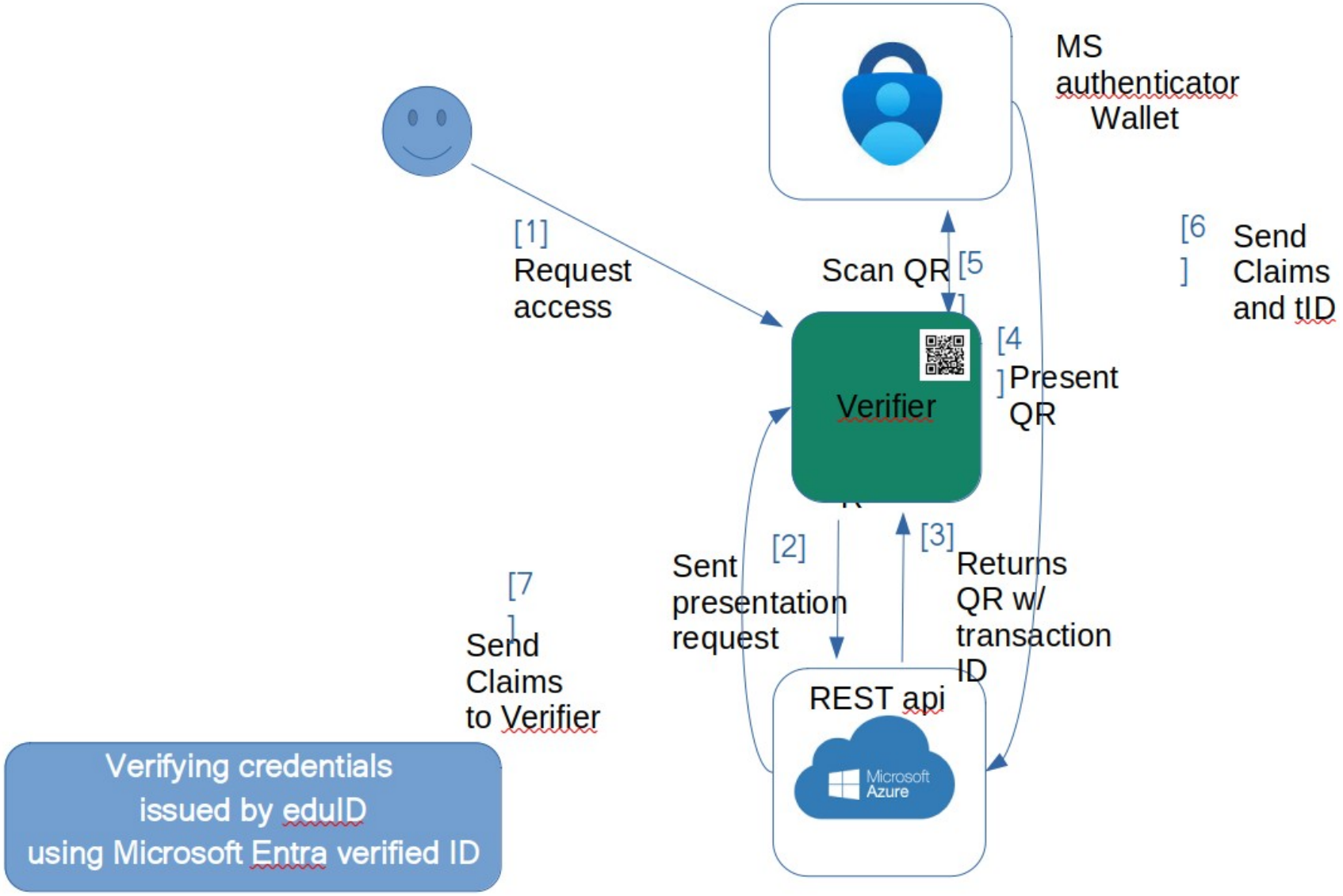

09:30
94%
eduID
eduID
eduID
SURF
Geverifieerde gegevens
Activiteit
VC
29 aug. 2022 15:49:17
U hebt deze geverifieerde id gedeeld met Verifiable Credential eduID Sample
VC
29 aug. 2022 15:26:19
U hebt deze geverifieerde id gedeeld met Verifiable Credential eduID Sample
VC
29 aug. 2022 15:09:22
U hebt deze geverifieerde id gedeeld met Verifiable Credential eduID Sample
S
29 aug. 2022 15:05:53
U hebt deze geverifieerde id ontvangen van van SURF


MS
authenticator
Wallet
[1] Start
credential
request
[2] Request
Login
(SAML)
[3] Returns
AuthN &
Attributes
eduID .nl
IdP
eduID .nl
ISSUER
[4]
Sent
attributes
[5] Return
QR w/
transaction ID
[6]
Present
QR
[7]
Scan QR
[8]
Request
claims
w/ tID
[9] Retrieve
Claims
& enter
pin to
confirm
REST api
Microsoft
Azure
Issue credentials
from eduID
using Microsoft Entra verified ID

W3C verifiable credential model




MS authenticator Wallet
WALLET
[9] Retrieve Claims & enter pin to confirm
[1] Start credential request
[7]
[8] Request claims w/ tID
Scan QR
[2] Request Login (SAML)
eduID .nl
IdP
eduID .nl
ISSUER
[6] Present QR
[3] Returns AuthN & Attributes
[5] Return QR w/ transaction ID
Sent attributes [4]
ISSUER
REST api
Microsoft Azure
VDR
Issue credentials from eduID using Microsoft Entra verified ID


MS
authenticator
Wallet
[1]
Request
access
Scan QR
[5
]
[6
]
Send
Claims
and tID
[4
]
Present
QR
Verifier
[2]
Sent
presentation
request
[3]
Returns
QR w/
transaction
ID
[7
]
Send
Claims
to Verifier
REST api
Microsoft
Azure
Verifying credentials
issued by eduID
using Microsoft Entra verified ID

MS VC’s - Requirements

- Azure tennant both for Issuers as well as Verifier
- Activate key storage and XYZ
- Issuer and/or verifier client lib (python/.net/java)
- Microsoft Authenticator

MS VC’s - Configuration

- SHOW AZURE backend

MS VC’s - Standards

- OIDC
- SIOP
- DID:WEB
- DID:ION

MS VC’s – Trust

- In the DID:Web implementation trust is established based on DNS, domain ownership and publishing .well-known file
- Azure as well as MS authenticator validate
- For DID:ION, Azure (and hence Microsoft) interacts with the VDR. It is at this point unknown if one can independently valdidate transactions against the ION ledger
- Relation between Issuer/Verifier client and Azure backend API is based on client key/secret
- Additional trust, beyond technical, is not possible except via credential issuace

MS VC’s – Findings

- ***Setup*** of Issuer or Verifier ***is super easy*** (< 1 hour)
- VC implementation does not in any way make use of backend AAI storage like Azure AD; all credentials must be presented to the API. ***Azure is only used as a ‘token translation’ proxy***.
- In DID:ION implementation Azure also interacts with the ledger (the wallet ***does not***)
- ***No selective release of credentials*** by user, only the entire presentation can be released
- Each issuer ***and verifier*** must have its own Azure tennant

MS VC’s – Conclusions

- Implementation is ***very tightly locked*** into the MS ecosystem, not only technically, but also from the trust perspective
- Verifier Azure tennant requirement is potential barrier
- ***Privacy and scalability*** of VC release looks challanging
- No controle over Credential usage by users
- Given that Azure ***already has federation capabilities***, the only clear benefit of verifiedID seems to be that ***it does not need upfront trust establisment***.
- No integrated provisioning path for Verifier