



EBSI Conformance Test Report

Talao - Talao wallet 1.16.0

30/10/2023

DID

did:key:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3
MotwqUJxWNkPRAVtwNYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyG
s3UK7j3LPiqRRG632ka25mCxsPr

The terms and conditions applicable to this report are described in the Service Offering Description document available [here](#).

1.Summary of the report

This report certifies the conformance of Talao wallet 1.16.0 distributed by Talao to the EBSI specifications vV3.0.0 on 30/10/2023.

The results and details of the tests can be found hereunder:

[illegible]

CT_WALLET_CROSS_DEFERRED

Deferred Credential

As an issuer, I want to enforce the deferred flow for the deferred credential from the issuer side. This means that when a participant requests the deferred credential, it will go through a specific deferred processing flow, resulting in a delay of 5 seconds from the first Credential Request. By implementing the deferred flow, the issuer can introduce a deliberate delay in providing the deferred credential.

10/25/2023, 1:28:13 PM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/25/2023, 11:28:13
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_deferred", "data": {"did": "did:key:zYqN
vVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtw
NYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxs
Pr", "credential_offer_endpoint": "openid-credential-offer://", "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_CROSS_DEFERRED

Deferred Credential

As an issuer, I want to enforce the deferred flow for the deferred credential from the issuer side. This means that when a participant requests the deferred credential, it will go through a specific deferred processing flow, resulting in a delay of 5 seconds from the first Credential Request. By implementing the deferred flow, the issuer can introduce a deliberate delay in providing the deferred credential.

10/28/2023, 9:17:12 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:17:12
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_deferred", "data": {"did": "did:key:zYqN
vVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtw
NYzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxs
Pr", "credential_offer_endpoint": "openid-credential-offer://", "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_CROSS_IN_TIME

Initiate Cross-Device Credential Issuance

As an issuer, I want to ensure that the in-Time credential goes through the in-time flow from the issuer side. This means that when a participant requests the in-Time credential, it will be processed and made available synchronously, without any delays. By implementing this in-time flow, participants can seamlessly obtain the in-Time credential without experiencing any significant wait times or processing delays. The synchronous availability of the credential ensures a smooth and efficient user experience.

10/25/2023, 1:27:48 PM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/25/2023, 11:27:48
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_in_time", "data": {"did": "did:key:zYqNv
VCKW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtwN
YzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxsPr
", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - { }
```

CT_WALLET_CROSS_IN_TIME

Initiate Cross-Device Credential Issuance

As an issuer, I want to ensure that the in-Time credential goes through the in-time flow from the issuer side. This means that when a participant requests the in-Time credential, it will be processed and made available synchronously, without any delays. By implementing this in-time flow, participants can seamlessly obtain the in-Time credential without experiencing any significant wait times or processing delays. The synchronous availability of the credential ensures a smooth and efficient user experience.

10/28/2023, 9:17:28 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:17:28
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_in_time", "data": {"did": "did:key:zYqNv
VCKW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtwN
YzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxsPr
", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - { }
```

CT_WALLET_CROSS_PRE_AUTHORISED

Pre-authorised Credential

As an issuer, I want to enforce the pre-authorised flow for the Pre-Authorised credential from the issuer side. This means that the credential can only be issued if the participant has gained access through a pre-authorised code. By implementing the pre-authorised flow, the issuer ensures that participants can only obtain the Pre-Authorised credential if they have successfully authenticated and gained access through a pre-authorised code. This pre-authorised code serves as a secure and controlled mechanism to verify the participant's eligibility for the credential.

10/25/2023, 1:28:33 PM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m 10/25/2023, 11:28:33
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_pre_authorised", "data": {"did": "did:key
:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaswfr7A3MotwqUJxWNkP
RAVtwNYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25
mCxsPr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m
- {}
```

CT_WALLET_CROSS_PRE_AUTHORISED

Pre-authorised Credential

As an issuer, I want to enforce the pre-authorised flow for the Pre-Authorised credential from the issuer side. This means that the credential can only be issued if the participant has gained access through a pre-authorised code. By implementing the pre-authorised flow, the issuer ensures that participants can only obtain the Pre-Authorised credential if they have successfully authenticated and gained access through a pre-authorised code. This pre-authorised code serves as a secure and controlled mechanism to verify the participant's eligibility for the credential.

10/28/2023, 9:17:52 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m 10/28/2023, 7:17:52
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_cross_pre_authorised", "data": {"did": "did:key
:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaswfr7A3MotwqUJxWNkP
RAVtwNYzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25
mCxsPr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m
- {}
```


CT_WALLET_SAME_DEFERRED

Deferred Credential

As an issuer, I want to enforce the deferred flow for the deferred credential from the issuer side. This means that when a participant requests the deferred credential, it will go through a specific deferred processing flow, resulting in a delay of 5 seconds from the first Credential Request.

10/28/2023, 9:19:03 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:19:03
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_deferred", "data": {"did": "did:key:zYqN
vVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtw
NYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxs
Pr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_DEFERRED

Deferred Credential

As an issuer, I want to enforce the deferred flow for the deferred credential from the issuer side. This means that when a participant requests the deferred credential, it will go through a specific deferred processing flow, resulting in a delay of 5 seconds from the first Credential Request.

10/28/2023, 9:19:58 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:19:58
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_deferred", "data": {"did": "did:key:zYqN
vVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtw
NYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxs
Pr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_DEFERRED

Deferred Credential

As an issuer, I want to enforce the deferred flow for the deferred credential from the issuer side. This means that when a participant requests the deferred credential, it will go through a specific deferred processing flow, resulting in a delay of 5 seconds from the first Credential Request.

10/28/2023, 9:20:29 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:20:29
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_deferred", "data": {"did": "did:key:zYqN
vVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtw
NYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxs
Pr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_IN_TIME

In-Time Credential

As an issuer, I want to ensure that the in-Time credential goes through the in-time flow from the issuer side. This means that when a participant requests the in-Time credential, it will be processed and made available synchronously, without any delays.

10/28/2023, 9:18:32 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m 10/28/2023, 7:18:32
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_in_time", "data": {"did": "did:key:zYqNv
VcKw65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtwN
YzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxsPr
", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_IN_TIME

In-Time Credential

As an issuer, I want to ensure that the in-Time credential goes through the in-time flow from the issuer side. This means that when a participant requests the in-Time credential, it will be processed and made available synchronously, without any delays.

10/28/2023, 9:20:02 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:20:02
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_in_time", "data": {"did": "did:key:zYqNv
VcKw65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtwN
YzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxsPr
", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_IN_TIME

In-Time Credential

As an issuer, I want to ensure that the in-Time credential goes through the in-time flow from the issuer side. This means that when a participant requests the in-Time credential, it will be processed and made available synchronously, without any delays.

10/28/2023, 9:20:23 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m 10/28/2023, 7:20:23
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_in_time", "data": {"did": "did:key:zYqNv
VcKw65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkPRAVtwN
YzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25mCxsPr
", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m - {}
```

CT_WALLET_SAME_PRE_AUTHORISED

Pre-authorised Credential

As an issuer, I want to enforce the pre-authorised flow for the Pre-Authorised credential from the issuer side. This means that the credential can only be issued if the participant has gained access through a pre-authorised code. By implementing the pre-authorised flow, the issuer ensures that participants can only obtain the Pre-Authorised credential if they have successfully authenticated and gained access through a pre-authorised code.

10/28/2023, 9:19:56 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:19:56
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_pre_authorised", "data": {"did": "did:key
:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkP
RAVtwNYzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25
mCxsPr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m
- {}
```

CT_WALLET_SAME_PRE_AUTHORISED

Pre-authorised Credential

As an issuer, I want to enforce the pre-authorised flow for the Pre-Authorised credential from the issuer side. This means that the credential can only be issued if the participant has gained access through a pre-authorised code. By implementing the pre-authorised flow, the issuer ensures that participants can only obtain the Pre-Authorised credential if they have successfully authenticated and gained access through a pre-authorised code.

10/28/2023, 9:20:11 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:20:11
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_pre_authorised", "data": {"did": "did:key
:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkP
RAVtwNYzkY34NG4FnTek534NE3L8pWznH5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25
mCxsPr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m
- {}
```


CT_WALLET_SAME_PRE_AUTHORIZED

Pre-authorized Credential

As an issuer, I want to enforce the pre-authorized flow for the Pre-Authorised credential from the issuer side. This means that the credential can only be issued if the participant has gained access through a pre-authorized code. By implementing the pre-authorized flow, the issuer ensures that participants can only obtain the Pre-Authorised credential if they have successfully authenticated and gained access through a pre-authorized code.

10/28/2023, 9:20:32 AM

```
[conformance-v3]/conformance-v3(stdout) [32m[Conformance API v3] [39m [33mInfo [39m      10/28/2023, 7:20:32
AM [33m[CheckService] [39m [32mTest Data {"intent": "ct_wallet_same_pre_authorized", "data": {"did": "did:key
:zYqNvVCkW65RVEgatyVyHRuhQbLCsx28Fm46oSWmmz63e2CpyYEWbMgV9woTidmacaSwfr7A3MotwqUJxWNkP
RAVtwNYzkY34NG4FnTek534NE3L8pWznh5kYbp1GkqpDusQvBLqHVuBNLD39aa9BzRyGs3UK7j3LPiqRRG632ka25
mCxsPr", "credential_offer_endpoint": "openid-credential-offer://"}, "result": {"success": true} } End Test Data [39m
- {}
```