Content Verifiable Credentials E2E-Test:

Testspecification Verifiable-Credentials-Android

Schlüssel	Beschreibung	Zusammenfassung	Manuelle Testschritte (Export)
TXR-6574	VER_CRED_VerifierApp_JWT_Verify_Valid_Credential_With_GW_Trustissuer	Verify a valid JWT which is signed by a public Issuer, contained in the Issuers List from Gateway.	**1** Action Scan a valid JWT **A** which has been signed by a public Issuer, contained in the Issuers List originating from Gateway. Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result JWT **A** is evaluated as valid by the Verifier App. The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6575	VER_CRED_WalletApp_JWT_Claiming_Credential_With_GW_TrustIssuer	Claim a valid JWT which is signed by a public Issuer, contained in the Issuers List from Gateway.	**1** Action Claim JWT **A** which has been signed by a public Issuer, contained in the Issuers List originating from Gateway. Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result JWT **A** is successfully claimed by the Wallet App. **2** Action Load the claimed JWT **A** Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6576	VER_CRED_Grant_Consent_For_Issuer_Not_In_TrustedIssuer_List_Verifier	Test object here is that the verifier aksks for consent before trusting an issuer which is not in the TrustedIssuer list and only after the consent is given, does the verifier add the issuer to its internal trust store.	**1** Action Update the trusted issuer list on VerifierApp Scan to verify the JWT Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer

			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action User grants consent. Data _ Expected Result A dedicated mechanism of the app is started that enables the user to provide the needed information for the issuer, for which a consent was granted.
			4	Action Scan the JWT **A** again. Data _ Expected Result The JWT is evaluated as valid.
TXR-6577	VER_CRED_Deny_Consent_For_Issuer_Not_In_TrustedIssuer_List_Verifier	The aim here is to test whether in the case that the user has denied consent for issuer which is not in the TrustedIssuer list the verifier app does NOT add issuer to its internal trust store.	**1**	Action Update the trusted issuer list on VerifierApp Scan to verify JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action

				User does NOT grant consent. Data _ Expected Result System acknowledges the deny of consent.
			4	Action Scan the JWT **A** again. Data _ Expected Result The system recognizes that the signature is still invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6651	VER_CRED_VerifierApp_JWT_NOT_verifying_Case_GW_Issuer_Public_Key_Does_Not_Match	The aim of this test case is to check that a JWT with a Gateway Issuer but a not matching public key does not get verified by the verifier app.	**1**	Action Scan to verify JWT **E** with a Gateway Issuer but a not matching public key. Data JWT **E** which was signed with a Gateway Issuer but a not matching public key. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6652	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_GW_Issuer_Public_Key_Does_Not_Match	The aim of this test case is to check that a JWT with a Gateway Issuer but a not matching public key does not get claimed by the wallet app.	**1**	Action Try to claim JWT **E** with a Gateway Issuer but a not matching public key. Data JWT **E** which was signed with a Gateway Issuer but a not matching public key. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6653	VER_CRED_Grant_Consent_For_Issuer_Not_In_TrustedIssuer_List_Wallet	Test object here is that the wallet asks for consent before trusting an issuer which is not in the TrustedIssuer list and only after the consent is given, does the wallet add the issuer to its internal trust store. Then, the DCC can be saved in the wallet app.	**1**	Action Update the trusted issuer list on Wallet App Attemt to claim the JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList

				Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action User grants consent. Data _ Expected Result A dedicated mechanism of the app is started that enables the user to provide the needed information for the issuer, for which a consent was granted.
			4	Action Try to claim the JWT again. Data _ Expected Result The JWT is claimed in the wallet app.
TXR-6654	VER_CRED_Deny_Consent_For_Issuer_Not_In_TrustedIssuer_List_Wallet	The aim here is to test whether in the case that the user has denied consent for issuer which is not in the TrustedIssuer list the verifier app does NOT add issuer to its internal trust store. As a consequence, the JWT cannot be claimed in the wallet app.	**1**	Action Update the trusted issuer list on the Wallet App. Scan the JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result

				_
			3	Action User does NOT grant consent. Data _ Expected Result System acknowledges the deny of consent.
			4	Action Try to claim JWT **A** again Data _ Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6666	VER_CRED_Verifier_Update_TrustedIssuerList_From_GW	The aim here is to test whether the verifier app's TrustedIssuerList can be manually updated.	1	Action Try to verify a JWT **Update_Test** with an Issuer from Gateway which was not updated in the IssuersList. Data JWT **Update_Test** signed by a GW Issuer which was not yet added to the IssuersList but will be added with the next update of the IssuersList from the gateway. Expected Result The JWT does not get verified. The system recognizes it as invalid and triggers an interactive mechanism for granting consent by the user.
			2	Action User denies consent. Data _ Expected Result System acknowledges the denial of consent.
			3	Action Manually update TrustedIssuersList, given internet connection. Data _ Expected Result Timestamp of TrustedIssuersList gets updated
			4	Action

				Try to verify the JWT Update_Test again. Data _ Expected Result The JWT gets verified.
TXR-6667	VER_CRED_Wallet_Update_TrustedIssuerList_From_GW	The aim here is to test whether the wallet app's TrustedIssuerList can be manually updated.	1	Action Try to claim a JWT **Update_Test** with an Issuer from Gateway which was not updated in the IssuersList. Data JWT **Update_Test** signed by a GW Issuer which was not yet added to the IssuersList but will be added with the next update of the IssuersList from the gateway. Expected Result The JWT does not get claimed. The system recognizes it as invalid and triggers an interactive mechanism for granting consent by the user.
			2	Action User denies consent. Data _ Expected Result System acknowledges the denial of consent.
			3	Action Manually update TrustedIssuersList, given internet connection. Data _ Expected Result Timestamp of TrustedIssuersList gets updated
			4	Action Try to claim the JWT **Update_Test** again. Data _ Expected Result The JWT gets claimed.
TXR-6713	VER_CRED_WalletApp_JWT_Claiming_Credential_With_Non_GW_TrustIssuer	Claim a valid JWT which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.	1	Action Claim JWT **B** which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.

				Data JWT **B**, which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Expected Result JWT **B** is successfully claimed by the Wallet App.
			2	Action Load the claimed JWT A Data JWT A, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6714	VER_CRED_VerifierApp_JWT_Verify_Valid_Credential_With_Non_GW_Trustissuer	Verify a valid JWT which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.	**1**	Action Verify JWT **B** which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Data JWT **B**, which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Expected Result JWT **B** is evaluated as valid by the Verifier App. The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6715	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_Invalid_Thumbprint	The aim of this test case is to check that a JWT with an invalid thumbprint does not get claimed by the wallet app.	**1**	Action Try to claim JWT **D** which has an invalid thumbprint. Data JWT **D** with an invalid thumbprint. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6716	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_Invalid_GW_Issuer_Missing_SpecialCase	A special case where the Gateway Issuer's key has not yet been delivered to the app. It gets manually added and after that it also arrives over the Gateway. Thus, this is the case where the same Gateway Issuer, which was originally missing both in the Issuer List from Gateway and on the local app's Database eventually got added to both lists in the following order: first it was manually added on the local DB List and then -- it got	**1**	Action Claim a JWT **SpecialCase** which was signed with a Gateway Issuer but the Issuer has not yet landed in the Issuer List over the Gateway. Data JWT **SpecialCase** which has been signed with a Gateway yIssuer but this Issuer has not yet landed in the Issuer List over the Gateway.

		added on the local DB List and then – it got with Issuer List update from the gateway.		Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action User grants consent for the Issuer to be added to the local Database. Data _ Expected Result The Issuer is added in the local Database.
			3	Action The User manually triggers an update of the Gateway issuer list or waits the configured amount of time until the Gateway Issuer List gets updated. Data _ Expected Result The above issuer is now added in both the Issuer List from the Gateway and in the local app's database's issuer list.
			4	Action Claim another JWT which was signed by the same Issuer. Data A JWT "SpecialCase 2* which was signed with the same issuer as in the above steps. Expected Result The wallet app claims the JWT as the issuer is known from both the Gateway List and from the local database's list.
TXR-6717	VER_CRED_VerifierApp_JWT_NOT_verifying_Case_Invalid_Thumbprint	The aim of this test case is to check that a JWT with an invalid thumbprint does not get verified by the verifier app.	**1**	Action Scan to verify a JWT **D** which has an invalid thumbprint. Data JWT **D** with an invalid thumbprint. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer

Testspecification Verifiable-Credentials-IOS

Schlüssel	Beschreibung	Zusammenfassung	Manuelle Testschritte (Export)
TXR-6574	VER_CRED_VerifierApp_JWT_Verify_Valid_Credential_With_GW_Trustissuer	Verify a valid JWT which is signed by a public Issuer, contained in the Issuers List from Gateway.	**1** Action Scan a valid JWT **A** which has been signed by a public Issuer, contained in the Issuers List originating from Gateway. Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result JWT **A** is evaluated as valid by the Verifier App. The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6575	VER_CRED_WalletApp_JWT_Claiming_Credential_With_GW_TrustIssuer	Claim a valid JWT which is signed by a public Issuer, contained in the Issuers List from Gateway.	**1** Action Claim JWT **A** which has been signed by a public Issuer, contained in the Issuers List originating from Gateway. Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result JWT **A** is successfully claimed by the Wallet App. **2** Action Load the claimed JWT **A** Data JWT **A**, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6576	VER_CRED_Grant_Consent_For_Issuer_Not_In_TrustedIssuer_List_Verifier	Test object here is that the verifier aksks for consent before trusting an issuer which is not in the TrustedIssuer list and only after the consent is given, does the verifier add the issuer to its internal trust store.	**1** Action Update the trusted issuer list on VerifierApp Scan to verify the JWT Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer

			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action User grants consent. Data _ Expected Result A dedicated mechanism of the app is started that enables the user to provide the needed information for the issuer, for which a consent was granted.
			4	Action Scan the JWT **A** again. Data _ Expected Result The JWT is evaluated as valid.
TXR-6577	VER_CRED_Deny_Consent_For_Issuer_Not_In_TrustedIssuer_List_Verifier	The aim here is to test whether in the case that the user has denied consent for issuer which is not in the TrustedIssuer list the verifier app does NOT add issuer to its internal trust store.	**1**	Action Update the trusted issuer list on VerifierApp Scan to verify JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action

				User does NOT grant consent. Data _ Expected Result System acknowledges the deny of consent.
			4	Action Scan the JWT **A** again. Data _ Expected Result The system recognizes that the signature is still invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6651	VER_CRED_VerifierApp_JWT_NOT_verifying_Case_GW_Issuer_Public_Key_Does_Not_Match	The aim of this test case is to check that a JWT with a Gateway Issuer but a not matching public key does not get verified by the verifier app.	**1**	Action Scan to verify JWT **E** with a Gateway Issuer but a not matching public key. Data JWT **E** which was signed with a Gateway Issuer but a not matching public key. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6652	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_GW_Issuer_Public_Key_Does_Not_Match	The aim of this test case is to check that a JWT with a Gateway Issuer but a not matching public key does not get claimed by the wallet app.	**1**	Action Try to claim JWT **E** with a Gateway Issuer but a not matching public key. Data JWT **E** which was signed with a Gateway Issuer but a not matching public key. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6653	VER_CRED_Grant_Consent_For_Issuer_Not_In_TrustedIssuer_List_Wallet	Test object here is that the wallet asks for consent before trusting an issuer which is not in the TrustedIssuer list and only after the consent is given, does the wallet add the issuer to its internal trust store. Then, the DCC can be saved in the wallet app.	**1**	Action Update the trusted issuer list on Wallet App Attemt to claim the JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList

				Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result _
			3	Action User grants consent. Data _ Expected Result A dedicated mechanism of the app is started that enables the user to provide the needed information for the issuer, for which a consent was granted.
			4	Action Try to claim the JWT again. Data _ Expected Result The JWT is claimed in the wallet app.
TXR-6654	VER_CRED_Deny_Consent_For_Issuer_Not_In_TrustedIssuer_List_Wallet	The aim here is to test whether in the case that the user has denied consent for issuer which is not in the TrustedIssuer list the verifier app does NOT add issuer to its internal trust store. As a consequence, the JWT cannot be claimed in the wallet app.	**1**	Action Update the trusted issuer list on the Wallet App. Scan the JWT **A** Data JWT **A** is signed by an Issuer not on the TrustedList Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action A mechanism for acknowledging the user's consent deision is started. Data _ Expected Result

				_
			3	Action User does NOT grant consent. Data _ Expected Result System acknowledges the deny of consent.
			4	Action Try to claim JWT **A** again Data _ Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6666	VER_CRED_Verifier_Update_TrustedIssuerList_From_GW	The aim here is to test whether the verifier app's TrustedIssuerList can be manually updated.	1	Action Try to verify a JWT **Update_Test** with an Issuer from Gateway which was not updated in the IssuersList. Data JWT **Update_Test** signed by a GW Issuer which was not yet added to the IssuersList but will be added with the next update of the IssuersList from the gateway. Expected Result The JWT does not get verified. The system recognizes it as invalid and triggers an interactive mechanism for granting consent by the user.
			2	Action User denies consent. Data _ Expected Result System acknowledges the denial of consent.
			3	Action Manually update TrustedIssuersList, given internet connection. Data _ Expected Result Timestamp of TrustedIssuersList gets updated
			4	Action

				Try to verify the JWT Update_Test again. Data _ Expected Result The JWT gets verified.
TXR-6667	VER_CRED_Wallet_Update_TrustedIssuerList_From_GW	The aim here is to test whether the wallet app's TrustedIssuerList can be manually updated.	1	Action Try to claim a JWT **Update_Test** with an Issuer from Gateway which was not updated in the IssuersList. Data JWT **Update_Test** signed by a GW Issuer which was not yet added to the IssuersList but will be added with the next update of the IssuersList from the gateway. Expected Result The JWT does not get claimed. The system recognizes it as invalid and triggers an interactive mechanism for granting consent by the user.
			2	Action User denies consent. Data _ Expected Result System acknowledges the denial of consent.
			3	Action Manually update TrustedIssuersList, given internet connection. Data _ Expected Result Timestamp of TrustedIssuersList gets updated
			4	Action Try to claim the JWT **Update_Test** again. Data _ Expected Result The JWT gets claimed.
TXR-6713	VER_CRED_WalletApp_JWT_Claiming_Credential_With_Non_GW_TrustIssuer	Claim a valid JWT which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.	1	Action Claim JWT **B** which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.

				Data JWT **B**, which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Expected Result JWT **B** is successfully claimed by the Wallet App.
			2	Action Load the claimed JWT A Data JWT A, which has been signed by a public issuer, contained in the Ussuers List originating from the Gateway. Expected Result The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6714	VER_CRED_VerifierApp_JWT_Verify_Valid_Credential_With_Non_GW_Trustissuer	Verify a valid JWT which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway.	**1**	Action Verify JWT **B** which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Data JWT **B**, which is signed by a non-Gateway Issuer, which was locally added in the app's database but does not exist in the trusted issuers list which came from the Gateway. Expected Result JWT **B** is evaluated as valid by the Verifier App. The JWT Data is displayed fully and correctly syntactically and semantically.
TXR-6715	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_Invalid_Thumbprint	The aim of this test case is to check that a JWT with an invalid thumbprint does not get claimed by the wallet app.	**1**	Action Try to claim JWT **D** which has an invalid thumbprint. Data JWT **D** with an invalid thumbprint. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
TXR-6716	VER_CRED_WalletApp_JWT_NOT_Claiming_Case_Invalid_GW_Issuer_Missing_SpecialCase	A special case where the Gateway Issuer's key has not yet been delivered to the app. It gets manually added and after that it also arrives over the Gateway. Thus, this is the case where the same Gateway Issuer, which was originally missing both in the Issuer List from Gateway and on the local app's Database eventually got added to both lists in the following order: first it was manually added on the local DB List and then -- it got	**1**	Action Claim a JWT **SpecialCase** which was signed with a Gateway Issuer but the Issuer has not yet landed in the Issuer List over the Gateway. Data JWT **SpecialCase** which has been signed with a Gateway yIssuer but this Issuer has not yet landed in the Issuer List over the Gateway.

		added on the local DB List and then – it got with Issuer List update from the gateway.		Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer
			2	Action User grants consent for the Issuer to be added to the local Database. Data _ Expected Result The Issuer is added in the local Database.
			3	Action The User manually triggers an update of the Gateway issuer list or waits the configured amount of time until the Gateway Issuer List gets updated. Data _ Expected Result The above issuer is now added in both the Issuer List from the Gateway and in the local app's database's issuer list.
			4	Action Claim another JWT which was signed by the same Issuer. Data A JWT "SpecialCase 2* which was signed with the same issuer as in the above steps. Expected Result The wallet app claims the JWT as the issuer is known from both the Gateway List and from the local database's list.
TXR-6717	VER_CRED_VerifierApp_JWT_NOT_verifying_Case_Invalid_Thumbprint	The aim of this test case is to check that a JWT with an invalid thumbprint does not get verified by the verifier app.	**1**	Action Scan to verify a JWT **D** which has an invalid thumbprint. Data JWT **D** with an invalid thumbprint. Expected Result The system recognizes that the signature was invalid and responds with a corresponding message or request for approval of the new issuer