EUROPEAN CENTRAL BANK
EUROSYSTEM

PONTES Pilot Mandatory Test Cases

applicable to PONTES Pilot Central Banks, Market Participants and Market DLT Operators

Version: 1.2
Status: Final (updatable)
Date: 24/07/2026

EUROPEAN CENTRAL BANK
EUROSYSTEM

Version Control

Date	Version	Comment
21/05/2026	0.1	Initial version produced by ECB testing team (TQR)
02/06/2026	1.0	Clean version incorporating NCB comments
29/06/2026	1.1	Added new test case PP-DLTO-05. References to Read Write modes/profiles were removed from the XvP test cases. Clarified that the unhappy path for XvP is agnostic for the settlement method (direct RTGS or Cash Token) used;
10/07/2026	1.2	Added a new test case for direct RTGS payment. Clarified that performing the XvP (i.e. DvP or PvP) test cases is not mandatory for Market Actors (Market Participants and Market DLT Operators) using Pontes Pilot only for Wholesale Payments. It is recalled that, other than for the DvP and PvP functionality via the Hash-Link protocol, the Eurosystem bears no responsibility for the coordination of settlement on the market DLT platform. Re-arranged the numbering of the test cases in line with the functionality.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Contents

Contents

EUROPEAN CENTRAL BANK
EUROSYSTEM

EUROPEAN CENTRAL BANK
EUROSYSTEM

1 Introduction

1.1 Approach

This document defines the set of mandatory test cases which must be passed by an NCB, Market Participant or Market DLT Operator before starting operations on Pontes Pilot in the production environment.

The test cases are organised per type of participant. For Market Participants and Market DLT Operators, certification is granted by the responsible NCB.

The responsible NCB may waive the need to execute some test cases depending on the settlement method (i.e., direct T2 RTGS or Cash Token settlement) used by the Market Participant / Market DLT Operator.

The responsible NCB may waive the need to execute XvP test cases provided the Market Participant / Market DLT Operator uses Pontes Pilot only for Wholesale payments. If Market Participants / Market DLT Operators decide to use Pontes Pilot for other use cases than Wholesale payments, executing the XvP test cases will be mandatory. It is recalled that, other than for the DvP and PvP functionality via the Hash-Link protocol, the Eurosystem bears no responsibility for the coordination of settlement on the market DLT platform.

Test cases are described in U2A mode wherever the test can be executed in U2A mode. However, if a test case can also be executed in U2A and A2A mode, the actor can decide on whether to execute the test case in A2A or U2A mode for certification purposes.

The below tables provide an overview of the mandatory test cases to be performed by the market actors, provided that the responsible NCBs grants exemptions based on the applicable settlement method and use-case planned to be used by the market actor.

Indicative overview for mandatory test cases for Market Participants:

Settlement method	**Use-Case**	**Test cases to be executed**
Direct settlement in T2 RTGS	XvP (DvP or PvP) functionalities	PP-MP-05 Complete an XvP transaction with direct RTGS settlement and retrieve execution key (A2A) PP-MP-06 Fail to complete an XvP transaction within timeout and retrieve cancellation key (A2A)
Direct settlement in T2 RTGS	Payment functionalities	PP-MP-03 Perform direct RTGS payment to another Participant
Settlement with Cash Tokens	XvP (DvP or PvP) functionalities	PP-MP-01 Fund own DCW from own RTGS DCA PP-MP-04 Complete an XvP transaction with token settlement and retrieve execution key (A2A)

		PP-MP-06 Fail to complete an XvP transaction within timeout and retrieve cancellation key (A2A) PP-MP-07 Query balances and transactions history for owned Dedicated Cash Wallets
Settlement with Cash Tokens	Payment functionalities	PP-MP-01 Fund own DCW from own RTGS DCA PP-MP-02 Perform wallet-to-wallet cash token payment from own DCW to another Participant's DCW PP-MP-07 Query balances and transactions history for owned Dedicated Cash Wallets

Indicative overview for mandatory test cases for Market DLT Operators:

Settlement method	**Use-Case**	**Test cases to be executed**
Direct settlement in T2 RTGS	XvP (DvP or PvP) functionalities	PP-DLTO-03 Complete an XvP transaction with direct RTGS settlement on behalf of two Market Participants and retrieve execution key (A2A) PP-DLTO-04 Fail to complete an XvP transaction within timeout and retrieve cancellation key on behalf of two Market Participants (A2A, any settlement method)
Direct settlement in T2 RTGS	Payment functionalities	PP-DLTO-05 Perform direct RTGS payment on behalf of a Market Participant to another Market Participant
Settlement with Cash Tokens	XvP (DvP or PvP) functionalities	PP-DLTO-02 Complete an XvP transaction with token settlement on behalf of two Market Participants (A2A) PP-DLTO-04 Fail to complete an XvP transaction within timeout and retrieve cancellation key on behalf of two Market Participants (A2A, any settlement method)
Settlement with Cash Tokens	Payment functionalities	PP-DLTO-01 Perform wallet-to-wallet cash token payment on behalf of a Market Participant to another Participant's DCW (A2A)

1.2 Test case template

The table below describes the elements included in the test case template.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test Case ID	*Unique ID of the test case, referencing the applicable type of actor (NCB, MP, DLTO)*
Test case name	*Descriptive name of the test case*
Relevant for (actor)	*NCB / Market Participant / Market DLT Operator*
Input mode	*U2A or A2A (the user chooses one of the two input modes corresponding to the mode that will be used in production)*
Mandatory	*Yes (subject to possible exemptions or specific conditions to be considered)*
Detailed Description	*Short description followed by the required steps to perform the test case, first in U2A and then in A2A (where applicable)*
Preconditions/Details	*Preconditions required to carry out the test case*
Expected results	*The expected and required outcome of the test case in U2A and A2A (if applicable)*
Test evidence	*Description of the required test evidence for the U2A and A2A execution (if applicable)*
Relevant Documentation	*References to the applicable functional specifications, user manuals, or other supporting documentation*
Related profiles/privileges	*Required profiles/privileges to carry out the test case*

2 Test cases

2.1 Test cases for National Central Banks

PP-NCB-01 – Configure in LRDM a Participant along with its T2 account reference and Dedicated Cash Wallet

Field	Description

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test Case ID	PP-NCB-01
Test case name	Configure in LRDM a Participant along with its T2 account reference and Dedicated Cash Wallet
Relevant for (actor)	National Central Bank (NCB) – acting as DCW Manager and responsible NCB for the Market Participant being onboarded
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to onboard a new Market Participant in the Pilot solution by configuring, in its Local Reference Data Management (LRDM) component, the three linked reference data objects required to operate: (i) the Market Participant entity, (ii) the T2 account reference linking the participant to its T2 RTGS DCA, and (iii) the Dedicated Cash Wallet (DCW) owned by the participant. Each configuration is submitted to the four-eyes principle on reference data. **U2A** *Preliminary steps in CRDM (out of scope of this test case but required as a prerequisite):* NCB User 1 (with privileges on CRDM) verifies on CRDM that the Party BIC matches the Party BIC defined in CRDM for the Market Participant, that the T2 Account Number is an existing and valid T2 RTGS DCA owned by the Market Participant, and that an Authorised Account User (AAU) of type *Direct* is in place between the T2 Account Number and the Pilot Participant BIC. NCB User 1 sets up on CRDM a Direct Debit Mandate allowing the Pilot Party (Payee) to debit the T2 Account Number. *Step 1 – Create the Market Participant entity (Initiator – NCB User with Referential Read Write profile):* 1. Authenticate to the Pilot solution GUI using credentials and the double authentication factor. 2. Select the **Global** tab from the main sidebar menu, then the **Participants List** sub-tab. 3. Click on the **Add a Market Participant** button (visible to Dedicated Cash Wallet Managers with a Referential Read Write profile). 4. Fill in the following fields: • **Participant BIC**: the Pilot Participant BIC (BIC11 – Authorised Account User BIC of the T2 RTGS account used

EUROPEAN CENTRAL BANK
EUROSYSTEM

	as the unique identifier in the Pilot). • **Long Name**: long name of the Market Participant. • **Short Name**: short name of the Market Participant. • **Roles**: select *DCW Owner* (only allowed value for Market Participants). • **Valid From** (mandatory) and **Valid To** (optional): beginning and end of the participant's validity. 5. Click on the **Submit** button. The new entity is queued in the **Pending Validations – Reference Data** sub-tab. *Step 2 – Approve the Market Participant entity (Approver – second NCB User with Referential Read Write profile, different from Step 1 User):* 6. Authenticate to the Pilot solution GUI. 7. Select the **Pending Validations** tab, then the **Reference Data** sub-tab. 8. Select the *Entities* filter and locate the newly created Market Participant entry. 9. Click on the **Approve** button (or, if a correction is needed, click on the **To be reviewed** button to send the item back to the initiator). *Step 3 – Create the T2 account reference (Initiator – NCB User with Referential Read Write profile):* 10. Select the **T2 Account References** tab, then either the **Create a T2 Account Reference** sub-tab or the **Managed** sub-tab followed by clicking on the **Create a T2 account Reference** button. 11. Fill in the following fields: • **T2 account Reference**: the T2 RTGS Account number of the Market Participant (alphanumeric, up to 34 characters). • **Responsible NCB**: the unique identifier of the NCB performing the configuration (set automatically or selected from the drop-down). • **Entity Link**: select the Pilot Participant BIC of the Market Participant approved in Step 2 (the T2 RTGS account holder). 12. Click on the **Submit** button. The new T2 account reference is queued in the **Pending Validations – Reference Data** sub-tab. *Step 4 – Approve the T2 account reference (Approver – second NCB User with Referential Read Write profile):* 13. Authenticate to the Pilot solution GUI. 14. Select the **Pending Validations** tab, then the **Reference Data** sub-tab. 15. Select the *T2 RTGS account References* filter and locate the pending item. 16. After verification of the associated T2 RTGS account data (including the T2 RTGS account number and the linked entity), click on the **Approve** button. *Step 5 – Create the Dedicated Cash Wallet (Initiator – NCB*

EUROPEAN CENTRAL BANK
EUROSYSTEM

	User with Referential Read Write profile, acting as DCW Manager): 17. Select the **Dedicated Cash Wallets** tab. 18. Click on the **New Wallet** button. 19. Fill in the following fields: • **Wallet ID suffix**: free text (alphanumeric, hyphen and underscore only, max 17 characters). The system will automatically prepend the prefix WCCAAABBBBBBBBBBB where *W* is the fixed Wallet identifier, *CC* the country code, *AAA* the currency (EUR), and *BBBBBBBBBBB* the owner BIC of the wallet. • **Short Name**: select the Short Name corresponding to the Entity owning the wallet (the Market Participant approved in Step 2). • **T2 RTGS account reference**: select from the drop-down the T2 account reference approved in Step 4. • **Valid From** and **Valid To**: optional, automatically filled if left blank. 20. Click on the **Submit for validation** button. The new DCW is queued in the **Pending Validations – Reference Data / Wallets** sub-tab. *Step 6 – Approve the Dedicated Cash Wallet (Approver – second NCB User with Referential Read Write profile):* 21. Authenticate to the Pilot solution GUI. 22. Select the **Pending Validations** tab, then the **Reference Data** sub-tab. 23. Select the *Wallets* filter and locate the pending DCW. 24. Click on the **Approve** button. *Step 7 – Verification:* 25. Navigate to the **Global** → **Participants List** sub-tab and verify that the new Market Participant is listed with status *Active*, the expected roles, and the configured validity period. 26. Navigate to the **T2 Account References** → **Managed** sub-tab and verify that the T2 account reference is visible and linked to the Market Participant. 27. Navigate to the **Dedicated Cash Wallets** tab and verify that the DCW is listed under the Market Participant with an *Active* status, zero balance, and a valid link to the T2 account reference.
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator, with the *DCW Manager* role. • Two distinct NCB users with the **Referential Read Write** profile are available to perform the four-eyes principle (initiator and approver). Note that, where the institution wishes to leverage the same approver across the three reference-data items, this is acceptable provided each approver is different from the initiator at each step. • The following Market Participant data has been gathered:

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Party BIC, Pilot Participant BIC, Short Name, Long Name, T2 Account Number, Settlement-in-cash-token flag, DvP/PvP flag, Administrator Email. • **CRDM prerequisites** are satisfied: – The Pilot Participant BIC is configured in CRDM as the Party BIC of the Market Participant. – The T2 Account Number is an existing and valid T2 RTGS DCA owned by the Market Participant. – An Authorised Account User (AAU) of type *Direct* is in place between the T2 Account Number and the Pilot Participant BIC. – A Direct Debit Mandate has been set up in CRDM to allow the Pilot Party (Payee) to debit the T2 Account Number. • The NCB Administrator User has previously created the two Referential Read Write users (out of scope of this test case). • All users hold valid (non-expired) credentials and double-authentication factors. • The test is executed within an open Business window.
Expected results	**U2A** • The Market Participant entity is successfully created in LRDM, approved through the four-eyes principle, and appears in the **Global** → **Participants List** sub-tab with status *Active*. • The T2 account reference is successfully created in LRDM, approved through the four-eyes principle, and appears in the **T2 Account References** → **Managed** sub-tab with the correct linked entity (the Market Participant). • The Dedicated Cash Wallet is successfully created in LRDM, approved through the four-eyes principle, and appears in the **Dedicated Cash Wallets** tab with status *Active*, owner = Market Participant BIC, manager = NCB BIC, currency = EUR, available balance = 0, and a valid link to the T2 account reference. • The configuration is consistent: the DCW is linked to the T2 account reference, which is in turn linked to the Market Participant entity. • The Market Participant is now eligible to receive its first funding request (see PP-E2E-001) and to operate on its DCW.
Test evidence	**U2A** • Screenshot of the Market Participant creation screen (Step 1) showing the entered Participant BIC, Short Name, Long Name, Role *DCW Owner*, and validity dates. • Screenshot of the **Pending Validations – Reference Data** sub-tab showing the Entity item in *Pending Approval* status, followed by a screenshot of the same item in *Approved* status. • Screenshot of the T2 account reference creation screen (Step 3) showing the entered T2 account number, responsible NCB, and Entity Link. • Screenshot of the **Pending Validations – Reference Data**

EUROPEAN CENTRAL BANK
EUROSYSTEM

	sub-tab showing the T2 account reference item in *Pending Approval* status, followed by a screenshot of the same item in *Approved* status. • Screenshot of the DCW creation screen (Step 5) showing the entered Wallet ID suffix (with the auto-generated prefix visible), Short Name, and T2 RTGS account reference. • Screenshot of the **Pending Validations – Reference Data / Wallets** sub-tab showing the DCW item in *Pending Approval* status, followed by a screenshot of the same item in *Approved* status. • Screenshot of the **Global → Participants List** sub-tab showing the new Market Participant *Active*. • Screenshot of the **T2 Account References → Managed** sub-tab showing the T2 account reference and its linked entity. • Screenshot of the **Dedicated Cash Wallets** tab showing the newly created DCW with all key attributes (Wallet ID, Owner, Manager, Available Balance, Valid From/To, T2 account reference).
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.4 *Local Reference Data Management*; §3.4.1 *National Central Banks and ECB*; §3.4.2 *Market Participants*; §3.4.3 *T2 account reference*; §3.4.4 *Dedicated Cash Wallets*; §6.4.2 *Market Participant – Onboarding* (CRDM prerequisites and Pilot Configuration Steps 1 & 2); Table F *List of reference data features in U2A and A2A*. • PONTES Pilot – User Handbook v1.0: §3.2.8 *T2 Accounts References* (sub-tabs); §3.2.9 *Global* (Participants List); §4.3.1 *Dedicated Cash Wallet creation*; §4.3.1.1 *T2 Account Reference creation*; §4.3.1.2 *Creation of a Dedicated Cash Wallet*; §4.7 *Pending Validation (Four-eyes process)*; §4.7.1 *Approval of Reference Data*; §4.8.2 *Participants List* (incl. §4.8.2.2 *Adding a new Market Participant*).
Related profiles/privileges	• User profile: **Referential Read Write** – for both initiator and approver across all three reference-data creations

PP-NCB-02 – Configure in LRDM a Market DLT Operator to instruct on behalf of a Market Participant

Field	Description
Test Case ID	PP-NCB-02

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test case name	Configure in LRDM a Market DLT Operator to instruct on behalf of a Market Participant
Relevant for (actor)	National Central Bank (NCB) – acting as the responsible NCB both for the Market Participant (via Whitelist) and, where applicable, for the eligibility of the Market DLT Operator
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to configure a Market DLT Operator (MDLTO) so that it is enabled to instruct on behalf of one of its Market Participants via A2A access. The end-to-end configuration involves three reference data objects in LRDM: (i) the creation of the Market DLT Operator on top of an existing Market DLT Platform (assumed pre-registered by the Service Providers acting as Operator); (ii) the Whitelisting linking the Market DLT Operator to the Market Participant; and (iii) the creation of a user with the *External User* profile, linked to the MDLT Operator, allowing the MDLTO to instruct in A2A on behalf of the Market Participant. Note: this test case assumes the responsible NCB executing the test is both the NCB responsible for the eligibility of the Market DLT Operator (steps 1–4) and the NCB responsible for the Market Participant being whitelisted (steps 5–10). In cross-jurisdiction scenarios, steps 1–4 are executed by the NCB of the MDLT Operator and steps 5–10 by the NCB of the Market Participant – see the example in SDD §2.3. **U2A** *Prerequisites (out of scope of this test case but required):* – The target **Market DLT Platform** (MDLT Platform) must already be registered in the Pilot solution by the Service Providers acting as Operator (see User Handbook §4.9.1 *Adding a new network*). The Market DLT Platform reference data is global and visible to all NCBs. – The Market Participant to be whitelisted must already be configured in the LRDM of its responsible NCB (see PP-NCB-01). – The Market DLT Operator institution has provided to the responsible NCB the following data (per SDD §6.4.1): Market DLT Operator ID, Market DLT platform Name, Market DLT platform Owner Name, and email address for the A2A user.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 1 – Create the Market DLT Operator (Initiator – NCB User with Referential Read Write profile, acting as DCW Manager): 1. Authenticate to the Pilot solution GUI using credentials and the double authentication factor. 2. Select the **External Network** tab from the main sidebar menu, then the **Network List** sub-tab. 3. Locate the target Market DLT Platform in the list and click on the **View Operators** link in the row of that platform. 4. Click on the **New Market DLT Operator** button. 5. In the pop-up window, fill in the only editable field: • **Market DLT Operator ID suffix**: free-text identifier (recommended BIC or LEI of the institution operating as Market DLT Operator). The system will automatically prepend the Market DLT Platform ID to produce the final ID in the format [Market DLT platform ID]-[Market DLT Operator ID]. 6. Click on the **Submit** button. The new Market DLT Operator is queued in the **Pending Validations – Reference Data** sub-tab. *Step 2 – Approve the Market DLT Operator (Approver – second NCB User with Referential Read Write profile, different from Step 1 User):* 7. Authenticate to the Pilot solution GUI. 8. Select the **Pending Validations** tab, then the **Reference Data** sub-tab. 9. Filter on the Market DLT Operator pending entries and locate the newly created MDLT Operator. 10. Click on the **Approve** button. The Market DLT Operator becomes visible to all NCBs. *Step 3 – Create the A2A External User for the Market DLT Operator (NCB User with Administrator profile):* 11. Authenticate to the Pilot solution GUI. 12. Select the **User Administration** tab. 13. Click on the **New User** button. 14. Fill in the following fields (per User Handbook §4.1.1): • **Username**: alphanumeric string respecting the naming convention P + country code + DLT BIC/LEI + free text. The Username must match the Common Name used to create the user certificate (see SDD §6.3.5). • **User email**: email address of the representative of the application connecting in A2A on behalf of the Market DLT Operator. • **Entity**: select the newly created Market DLT Operator ID as the entity to which the user belongs. • **Profile**: select **External User** (the only profile applicable to a Market DLT Operator A2A user). 15. Click on the **Submit** button. The user receives a confirmation email to activate the account and configure its credentials, terms & conditions, and certificate.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 4 – Create the Whitelist linking the Market DLT Operator to the Market Participant (Initiator – NCB User with Referential Read Write profile, acting as DCW Manager for the Market Participant): 16. Select the **External Network** tab, then the **Network List** sub-tab. 17. Locate the target Market DLT Platform and click on **View Operators**, then locate the Market DLT Operator approved in Step 2 and click on the **View participants** link to access the list of whitelisted participants for that Operator. 18. Click on the **Whitelist a participant** button. 19. In the pop-up window, fill in the editable fields: • **Participant BIC**: the Pilot Participant BIC of the Market Participant to be whitelisted (must be a Market Participant managed by the connected NCB). • **Valid From**: beginning of the whitelist validity. • **Valid To**: end of the whitelist validity (optional – leave empty for no end date). 20. Click on the **Submit** button. The new Whitelist is queued in the **Pending Validations – Reference Data** sub-tab. *Step 5 – Approve the Whitelist (Approver – second NCB User with Referential Read Write profile, different from Step 4 User):* 21. Authenticate to the Pilot solution GUI. 22. Select the **Pending Validations** tab, then the **Reference Data** sub-tab. 23. Select the *Whitelists* filter and locate the pending whitelist item. 24. Click on the **Approve** button. The list of whitelisted participants is updated; the entry becomes effective once its Valid From date is reached. *Step 6 – Verification:* 25. Navigate to the **External Network** → **Network List** → **[target platform]** → **View Operators** screen and verify that the new Market DLT Operator is listed with status *Active* and *Blocking Status = Not blocked*. 26. Click on **View participants** for the Market DLT Operator and verify that the Market Participant appears in the list of whitelisted participants with the configured validity dates. 27. Navigate to the **User Administration** tab and verify that the External User is listed, linked to the Market DLT Operator entity, with the *External User* profile, and that the user has activated the account (email verified).
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) acting as Operator • The Service Providers (SPs) have already registered the target **Market DLT Platform** in the Pilot solution (only Operator with Operational Read Write profile can do this –

EUROPEAN CENTRAL BANK
EUROSYSTEM

	User Handbook §4.9.1). • The Market Participant to be whitelisted has been previously configured in LRDM with status *Active* (see test case PP-NCB-01). • Two distinct NCB users with the **Referential Read Write** profile are available to perform the four-eyes principle on reference data creations (initiator and approver). The approver of a given item must be different from its initiator. • One NCB user with the **Administrator** profile is available to create the External User. • The following data is available for the Market DLT Operator: Market DLT Operator ID, Market DLT platform Name, Market DLT platform Owner Name, and A2A user email address. • A user certificate (with Common Name matching the Username) has been prepared for the Market DLT Operator's A2A user, in line with SDD §6.3.5 *Lifecycle of Digital Certificate*.
Expected results	**U2A** • The Market DLT Operator is successfully created in LRDM, approved through the four-eyes principle, and appears in the **External Network → Network List → [platform] → Operators** view with the final ID in the format [Market DLT platform ID]-[Market DLT Operator ID] and *Blocking Status* = *Not blocked*. The Market DLT Operator is visible to all NCBs (global scope). • The External User is successfully created, linked to the Market DLT Operator entity, with the **External User** profile, and successfully activates the account through the email confirmation process. • The Whitelist linking the Market DLT Operator to the Market Participant is successfully created in LRDM, approved through the four-eyes principle, and appears in the list of whitelisted participants for the Market DLT Operator with the configured validity dates. The Whitelist is visible only within the responsible NCB's data scope. • The Market DLT Operator is now technically enabled, once the External User has completed activation and the Valid From date of the Whitelist has been reached, to: – instruct payments, PFoDs, DvPs and PvPs in A2A on behalf of the whitelisted Market Participant (per SDD §2.2.7); – act as the actor responsible for the asset leg of DvP/PvP transactions involving the Market Participant (per SDD §3.4.7).
Test evidence	**U2A** • Screenshot of the Market DLT Operator creation pop-up (Step 1) showing the entered ID suffix and the pre-filled Market DLT Platform ID. • Screenshot of the **Pending Validations – Reference Data** sub-tab showing the Market DLT Operator pending entry,

EUROPEAN CENTRAL BANK
EUROSYSTEM

	followed by a screenshot of the same item in *Approved* status. • Screenshot of the **User Administration** tab showing the External User creation screen (Step 3) with the entered Username, email, Entity = Market DLT Operator ID, and Profile = *External User*. • Screenshot of the **User Administration** list showing the External User with status *Enabled* and *Email verified* = *Yes* after activation. • Screenshot of the Whitelist creation pop-up (Step 4) showing the entered Participant BIC and validity dates. • Screenshot of the **Pending Validations – Reference Data** sub-tab showing the Whitelist pending entry, followed by a screenshot of the same item in *Approved* status. • Screenshot of the **External Network** → **Network List** → **[platform]** → **Operators** view showing the new Market DLT Operator. • Screenshot of the **View participants** screen for the Market DLT Operator showing the Market Participant in the whitelisted participants list with the configured Valid From / Valid To dates.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.7 *External entity instructing on behalf*; §2.3 *Relationship between actors* (incl. configuration example with MDLTO-A and PBX/PBY); §3.4.5 *Market DLT platforms*; §3.4.6 *Market DLT Operators*; §3.4.7 *Whitelists*; §6.4.1 *Market DLT Operator – Onboarding* (Pilot Configuration Steps 1 to 3); §6.4.2 Step 5 *Whitelist configuration*; Table F *List of reference data features in U2A and A2A*. • PONTES Pilot – User Handbook v1.0: §3.2.10 *External Network*; §4.9 *External Network*; §4.9.1 *Adding a new network (Market DLT Platform)*; §4.9.2 *Adding a new Market DLT Operator*; §4.9.3 *Whitelisting of a participant for a Market DLT Operator*; §4.1.1 *Adding a new user*; §4.7 *Pending Validation (Four-eyes process)*; §4.7.1 *Approval of Reference Data*.
Related profiles/privileges	• User profiles required: – **Referential Read Write** – for both initiator and approver of Market DLT Operator and Whitelist creations – **Administrator** – for creation of the External User

PP-NCB-03 – Query balances and transactions history for managed Dedicated Cash Wallets

Field	Description

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test Case ID	PP-NCB-03
Test case name	Query balances and transactions history for managed Dedicated Cash Wallets
Relevant for (actor)	National Central Bank (NCB) – acting as Dedicated Cash Wallet Manager for the Market Participants within its jurisdiction and, if applicable, as DCW Owner for its own DCWs
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to query and verify, in the Pilot solution GUI, the balances and transaction history of the Dedicated Cash Wallets (DCWs) under its scope. The scope of an NCB connected as Dedicated Cash Wallet Manager covers (i) its own DCWs (if any) and (ii) the DCWs of the Market Participants for which it is the responsible NCB. The test exercises three complementary views in U2A: (a) the Dashboard (aggregated activity counters); (b) the Dedicated Cash Wallets list with per-wallet balances and details; and (c) the Transactions tab with the full transaction history, including detailed drill-down and Excel export. **U2A** *Step 1 – Authentication:* 1. The NCB user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Pilot Read Write** profile (or, for read-only verification, the **Pilot Read Only** profile). *Step 2 – Dashboard query (aggregated view):* 2. Select the **Dashboard** tab from the main sidebar menu. 3. Verify that the Dashboard displays the counters relevant to the connected NCB's scope (total amount and number of transactions for the current Business Date, including: *Outstanding*, *Minted*, *Burnt*, *Funded*, *Defunded*, *Settled*, *Unsettled*). 4. For each counter, hover over the information icon (i) to view the additional context/description in the pop-up box. 5. Note that the Dashboard refreshes automatically as transactions complete throughout the business day.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 3 – Dedicated Cash Wallets balance query (list view): 6. Select the **Dedicated Cash Wallets** tab from the main sidebar menu. 7. Verify that the displayed list contains all DCWs within the NCB's scope: its own DCWs (if any) and the DCWs managed for its Market Participants. 8. Click on the **Columns** button and select the columns relevant for the query, at minimum: • **Wallet ID** • **Owner** (BIC of the DCW Owner) • **Manager** (BIC of the DCW Manager) • **Available Balance** • **Currency** • **Country Code** • **Blocking Status** • **Main Wallet Flag** • **Valid From / Valid To** • **Last modified** 9. Click on the **Apply** button to confirm the column selection. 10. Use the **Filters** to narrow the result set (e.g. filter by Owner BIC or by Country Code) and verify the filtered list. 11. Click on a specific **Wallet ID** to open the wallet detail view; then click on **Details** to access the complete wallet attributes, including PoA Given To, PoA Validity Dates, PoA Max Amount, and Token details. 12. (Optional export) Click on the **Export** button to download an Excel file containing the current list with the applied filters and selected columns. *Step 4 – Transactions history query (list view):* 13. Select the **Transactions** tab from the main sidebar menu. 14. Click on the **Filters** button and apply the desired filters (per SDD §3.6.2.c): • **Transaction type**: *Transfer*, *Issuance*, *Redemption*, *Payment*, *Payment-XvP*, or *All* • **Transaction status**: *Accepted*, *Settled*, *Unsettled*, *Rejected*, *Pending settlement*, or *All* 15. Click on **Apply** to refresh the displayed list. 16. Click on the **Columns** button and select the columns relevant for the query, at minimum: • **Transaction ID** / **Request ID** • **Type** / **Status** • **Amount** / **Currency** • **Debited Wallet ID** / **Debited Owner Entity ID** / **Debited Manager Entity ID** • **Credited Wallet ID** / **Credited Owner Entity ID** / **Credited Manager Entity ID** • **Creation date** / **Acceptance Timestamp** / **Settlement Timestamp** / **Settlement Business Date** • **Instructing Entity ID** • **Correlation ID** 17. Click on **Apply** to confirm the column selection.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	18. Click on a specific **Transaction ID** to open the transaction detail view and verify the lifecycle timeline (Creation, Validation, Settlement), the From/To Wallet IDs and Manager BICs, the Initiator and Approver usernames. 19. (Optional export) Click on the **Export** button to download an Excel file containing the current list with the applied filters and selected columns. *Step 5 (Optional) – Cross-check between balance and transaction history (consistency check):* 20. Pick one DCW from the list of Step 3 and note its **Available Balance**. 21. Navigate back to the **Transactions** tab, apply filters on the selected wallet (using the appropriate filter on Debited/Credited Wallet ID) and on status *Settled*. 22. Verify that the sum of credits minus debits across the displayed settled transactions for that wallet reconciles with the **Available Balance** noted in Step 20, taking into account any pending or unsettled transactions.
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) acting as Operator • At least one Market Participant managed by the NCB has been onboarded via PP-NCB-01 and holds an active DCW. • Some funding/defunding/transfer/payment activity has been generated on the DCWs in scope (e.g. PP-E2E-001 has been executed at least once), so that the Dashboard counters and Transactions list contain non-empty data. • The NCB user holds the **Pilot Read Write** profile (or **Pilot Read Only** for read-only verification). • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window.
Expected results	**U2A** • **Dashboard**: All NCB-scope counters are displayed and refresh in real time. The aggregated values correspond to the activity of the DCWs managed by the connected NCB. • **Dedicated Cash Wallets tab**: The list correctly shows all DCWs within the NCB's scope (own + managed) and **no DCWs** outside that scope. For each wallet, the *Available Balance*, *Manager BIC*, *Owner BIC*, *Country Code*, *Blocking Status*, *Main Wallet Flag*, *Valid From/To* and *Last modified* are correctly displayed. The detail view (after click on Wallet ID → Details) shows all wallet attributes including PoA Given To, PoA Validity Dates, PoA Max Amount, and Token details. • **Transactions tab**: The list correctly shows all transactions impacting the DCWs within the NCB's scope (debit and credit legs). Filters on Type and Status correctly reduce the displayed set. The detail view (after click on Transaction ID)

EUROPEAN CENTRAL BANK
EUROSYSTEM

	correctly displays the transaction lifecycle timeline, the From/To wallets and managers, the Initiator and Approver. • **Excel exports** (where used): The downloaded files reflect the applied filters and selected columns, with all SDD §3.6.2.b/c fields populated. • **Cross-check**: The Available Balance of a wallet is consistent with the net of credits and debits of its settled transactions. • **Data scope enforcement**: No DCWs, transactions, or balances belonging to Market Participants of other NCBs are visible.
Test evidence	**U2A** • Screenshot of the **Dashboard** tab showing the NCB-scope counters. • Screenshot of the **Dedicated Cash Wallets** list with selected columns and applied filters, showing at least one own DCW (if any) and at least one managed DCW. • Screenshot of the **Dedicated Cash Wallet detail / Details** view for a representative wallet, with all attributes visible. • Excel export of the Dedicated Cash Wallets list (or Funding & Defunding extraction) opened locally, showing the expected fields (Wallet ID, Owner, Manager, Available Balance, etc. – per SDD §3.6.2.b). • Screenshot of the **Transactions** tab list with selected columns and applied filters. • Screenshot of the **Transaction detail** view (after click on a Transaction ID) showing the lifecycle timeline (Creation, Validation, Settlement). • Excel export of the Transactions list opened locally, showing the expected fields (per SDD §3.6.2.c). • Screenshot or note evidencing the reconciliation between the wallet *Available Balance* and the sum of settled credit/debit legs for that wallet.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.2 *Dedicated Cash Wallet Manager* (visibility on balances and transactions history related to managed DCWs); §3.6.2 *Reporting and data extraction* — sub-sections §3.6.2.b *Dedicated Cash Wallets balance query*, §3.6.2.c *Transactions for a Dedicated Cash Wallet query*; §3.4.4 *Dedicated Cash Wallets* (data model). • PONTES Pilot – User Handbook v1.0: §3.2.1 *Dashboard*; §3.2.3 *Dedicated Cash Wallets*; §3.2.5 *Transactions*; §4.2 *Dashboard Screen*; §4.3.2 *Dedicated Cash Wallets view*; §4.11 *Monitoring* (incl. §4.11.1 *Monitoring: key general considerations*, §4.11.2 *Monitoring T2 account references*, §4.11.3 *Monitoring Dedicated Cash Wallets*, and the duplicate-numbered §4.11.2 *Monitoring Transactions*).

EUROPEAN CENTRAL BANK
EUROSYSTEM

Related profiles/privileges	• User profiles: – **Pilot Read Write** – preferred profile for this test case – **Pilot Read Only** – equally valid for the read-only nature of this test case – **Referential Read Write / Read Only** also grant read access to the Dedicated Cash Wallets tab (per UHB §3.2.3) but are not the primary target profiles

PP-NCB-04 – Block a Market Participant using the blocking status attribute

Field	Description
Test Case ID	PP-NCB-04
Test case name	Block a Market Participant using the blocking status attribute
Relevant for (actor)	National Central Bank (NCB) – acting as Dedicated Cash Wallet Manager and responsible NCB for the Market Participant to be blocked
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to block one of the Market Participants under its jurisdiction by setting its *Blocking Status* attribute to *Blocked*. The blocking action is **immediately effective** and **is not submitted to the four-eyes principle** (per SDD §3.4.9). Once a Market Participant is blocked, all its Dedicated Cash Wallets are automatically blocked as well, preventing any cash-token debit or credit on those wallets and preventing the settlement of any new Direct Settlement transactions involving the blocked participant. **U2A** *Step 1 – Authentication:* 1. The NCB user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Referential Read Write** profile (per UHB §4.12.2.1).

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 2 – Identify the target Market Participant: 2. Select the **Global** tab from the main sidebar menu, then the **Participants List** sub-tab. 3. (Optional) Use the search/filter functions to locate the target Market Participant by its Pilot Participant BIC, Short Name or Country Code. 4. Verify, in the **Blocking Status** column, that the target Market Participant currently displays a **blue padlock icon (open)** – indicating that (i) the action on the blocking status is permitted (i.e. the connected NCB is the responsible NCB / DCW Manager for that participant) and (ii) the participant is currently *Unblocked*. *Step 3 – Block the Market Participant:* 5. Click on the **blue padlock icon (open)** in the row of the target Market Participant, under the *Blocking Status* column. 6. A confirmation pop-up window appears asking to either confirm or cancel the blocking action. 7. Click on the **Confirm** button. The blocking action is immediately effective – no four-eyes approval is requested. *Step 4 – Verification on the Participant:* 8. Remain on the **Global** → **Participants List** sub-tab and verify that the icon in the *Blocking Status* column for the target Market Participant has changed to a **plain blue padlock (closed)**, indicating that the participant is now *Blocked*. *Step 5 – Verification on the associated Dedicated Cash Wallets:* 9. Navigate to the **Dedicated Cash Wallets** tab. 10. (If not displayed by default) Click on the **Columns** button, enable the **Blocking Status** column, and click on **Apply**. 11. (Optional) Apply a filter on Owner BIC to restrict the list to the wallets owned by the blocked Market Participant. 12. Verify that **all** the DCWs owned by the blocked Market Participant now display the *Blocked* status in the *Blocking Status* column. The wallet-level blocking is automatic and reflects the participant-level blocking status. *Step 6 – (Optional) Verification of the blocking effect on transactions:* 13. As a Pilot Read Write user (initiator), attempt to submit a transaction involving the blocked Market Participant (e.g. a funding request on one of its DCWs, or a wallet-to-wallet transfer crediting/debiting one of its DCWs). 14. Verify that the system displays a **warning message** indicating that the operation will be ineligible for settlement (per UHB §4.12.2.2.2). 15. (If pursued) Submit the operation and verify that it is indeed not settled (ineligible). *Step 7 – (Optional) Verification of the blocking effect on new*

EUROPEAN CENTRAL BANK
EUROSYSTEM

	DCW creation: 16. As an NCB user with Referential Read Write profile (DCW Manager), attempt to create a new DCW for the blocked Market Participant (per PP-NCB-01, Step 5). 17. Verify that the new wallet, once approved, is created with the status *Blocked* by default (per UHB §4.12.2.2.1).
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator, with the *DCW Manager* role for the Market Participant to be blocked. • The target Market Participant has been previously onboarded in LRDM with status *Active* (see PP-NCB-01) and is **currently unblocked**. • The Market Participant **owns at least one Dedicated Cash Wallet** (per UHB §4.12.2.2: *"only Participants owning Dedicated Cash Wallets can be blocked"*). • The NCB user holds the **Referential Read Write** profile (per UHB §4.12.2.1). • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window. • (For optional Step 6) An additional NCB or Market Participant user with **Pilot Read Write** profile is available to attempt a transaction involving the blocked participant.
Expected results	**U2A** • The blocking action is **immediately effective** with no four-eyes approval step. • The Market Participant's *Blocking Status* changes from *Unblocked* to *Blocked* and is reflected in the **Global → Participants List** sub-tab (closed padlock icon). • **All** Dedicated Cash Wallets owned by the blocked Market Participant are automatically switched to *Blocked* status in the **Dedicated Cash Wallets** tab. • Any subsequent attempt to submit a new transaction involving the blocked Market Participant (cash-token settlement or Direct Settlement) is flagged as ineligible for settlement (warning message displayed at submission; transaction remains unsettled). • Any subsequent DCW creation request for the blocked Market Participant results in a DCW created with *Blocked* status. • The blocking action is auditable: the change is reflected in the participants list and the wallets list (per SDD §3.4.9).
Test evidence	**U2A** • Screenshot of the **Global → Participants List** sub-tab *before* the blocking action, showing the target Market Participant with the open blue padlock icon (Unblocked).

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• Screenshot of the confirmation pop-up window displayed after clicking on the blue padlock icon. • Screenshot of the **Global** → **Participants List** sub-tab *after* the confirmation, showing the target Market Participant with the closed (plain) blue padlock icon (Blocked). • Screenshot of the **Dedicated Cash Wallets** tab filtered on the Owner BIC of the blocked Market Participant, showing all its DCWs in *Blocked* status. • (Optional Step 6) Screenshot of the warning message displayed when attempting to submit a transaction involving the blocked Market Participant. • (Optional Step 7) Screenshot of the newly created DCW for the blocked Market Participant showing *Blocked* status from creation.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.4.9 *Market Participant Blocking / Unblocking* (immediately effective, not 4-eyes, effects on DCWs and on Direct Settlement); §3.4.2 *Market Participants* (data model – blocking status); §3.4.4 *Dedicated Cash Wallets* (data model – wallet blocking status); Table F *List of reference data features in U2A and A2A* (Participant – Block: 2-eyes ✓ / 4-eyes No). • PONTES Pilot – User Handbook v1.0: §4.12 *Blocking and unblocking participants*; §4.12.1 *Blocked and unblocked statuses* (incl. §4.12.1.1 *Participant status* – padlock icon semantics, §4.12.1.2 *Dedicated Cash Wallet status*); §4.12.2 *Blocking a participant* (incl. §4.12.2.1 *Blocking capability*, §4.12.2.2 *Participant blocking*, §4.12.2.2.1 *Dedicated Cash Wallet management of a blocked participant*, §4.12.2.2.2 *Transactions involving a blocked participant*); §3.2.9 *Global*; §4.8.2 *Participants List*.
Related profiles/privileges	• User profile: **Referential Read Write** – required to execute the blocking action (per UHB §4.12.2.1)

PP-NCB-05 – Unblock a Market Participant using the blocking status attribute

Field	Description
Test Case ID	PP-NCB-05
Test case name	Unblock a Market Participant using the blocking status attribute

EUROPEAN CENTRAL BANK
EUROSYSTEM

Relevant for (actor)	National Central Bank (NCB) – acting as Dedicated Cash Wallet Manager and responsible NCB for the Market Participant to be unblocked
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to unblock one of its Market Participants that is currently in *Blocked* status, by setting its *Blocking Status* attribute back to *Unblocked*. The unblocking action is **immediately effective** and **is not submitted to the four-eyes principle** (per SDD §3.4.9). Once a Market Participant is unblocked, all its Dedicated Cash Wallets are automatically switched back to *Unblocked* status as well, restoring the participant's ability to debit and credit cash tokens on those wallets and to participate in Direct Settlement transactions. This is the symmetric counterpart of PP-NCB-04. **U2A** *Step 1 – Authentication:* 1. The NCB user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Referential Read Write** profile (per UHB §4.12.3.1). *Step 2 – Identify the target blocked Market Participant:* 2. Select the **Global** tab from the main sidebar menu, then the **Participants List** sub-tab. 3. (Optional) Use the search/filter functions to locate the target Market Participant by its Pilot Participant BIC, Short Name or Country Code. 4. Verify, in the **Blocking Status** column, that the target Market Participant currently displays a **plain blue padlock (closed)** icon – indicating that the participant is currently *Blocked* and that the action on the blocking status is permitted (i.e. the connected NCB is the responsible NCB / DCW Manager for that participant). *Step 3 – Unblock the Market Participant:* 5. Click on the **plain blue padlock (closed)** icon in the row of the target Market Participant, under the *Blocking Status* column. 6. A confirmation pop-up window appears asking to either confirm or cancel the unblocking action. 7. Click on the **Confirm** button. The unblocking action is immediately effective – no four-eyes approval is requested.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 4 – Verification on the Participant: 8. Remain on the **Global → Participants List** sub-tab and verify that the icon in the *Blocking Status* column for the target Market Participant has changed back to an **open blue padlock**, indicating that the participant is now *Unblocked*. *Step 5 – Verification on the associated Dedicated Cash Wallets:* 9. Navigate to the **Dedicated Cash Wallets** tab. 10. (If not displayed by default) Click on the **Columns** button, enable the **Blocking Status** column, and click on **Apply**. 11. (Optional) Apply a filter on Owner BIC to restrict the list to the wallets owned by the unblocked Market Participant. 12. Verify that **all** the DCWs owned by the unblocked Market Participant now display the *Unblocked* status in the *Blocking Status* column. The wallet-level unblocking is automatic and reflects the participant-level unblocking status (per UHB §4.12.3.2: *"all associated Dedicated Cash Wallet are automatically switched to an Unblocked status. The unblocking action is also reported to the corresponding wallets to ensure full synchronisation."*). 13. **Specific check for DCWs created during the Blocked period (if any):** if one or more DCWs were created for this Market Participant while it was in *Blocked* status (which, per UHB §4.12.2.2.1, would have been created already in *Blocked* status), verify that those DCWs are also now in *Unblocked* status. *Step 6 – (Optional) Verification of the unblocking effect on transactions:* 14. As a Pilot Read Write user (initiator), submit a regular transaction involving the unblocked Market Participant (e.g. a funding request on one of its DCWs – see PP-E2E-001, or a wallet-to-wallet transfer). 15. Verify that **no warning message** about ineligibility for settlement is displayed at submission and that the transaction proceeds normally through the standard four-eyes validation flow. 16. Verify that the transaction reaches the *SETTLED* status (per the relevant settlement workflow), confirming that the Market Participant is fully operational again. *Step 7 – (Optional) Verification of new DCW creation:* 17. As an NCB user with Referential Read Write profile (DCW Manager), create a new DCW for the unblocked Market Participant (per PP-NCB-01, Step 5). 18. Verify that the new wallet, once approved, is created with the status *Unblocked* (i.e. the default behaviour is restored).

EUROPEAN CENTRAL BANK
EUROSYSTEM

Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator, with the *DCW Manager* role for the Market Participant to be unblocked. • The target Market Participant has been previously **blocked** (see PP-NCB-04) and is **currently in the *Blocked* status**. This is the mandatory entry condition for PP-NCB-05. • The Market Participant owns at least one Dedicated Cash Wallet (the same constraint as for blocking applies by symmetry; in any case, blocked wallets need to exist in order to verify that they are properly unblocked). • The NCB user holds the **Referential Read Write** profile (per UHB §4.12.3.1). • Per UHB §4.12.3.1, *"Participants allowed to unblock are the same one as the one allowed to block"* – the connected NCB user is therefore authorised to unblock any Market Participant under the NCB's jurisdiction. • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window. • (For optional Step 6) An additional NCB or Market Participant user with **Pilot Read Write** profile is available to attempt a transaction involving the unblocked participant.
Expected results	**U2A** • The unblocking action is **immediately effective** with no four-eyes approval step. • The Market Participant's *Blocking Status* changes from *Blocked* to *Unblocked* and is reflected in the **Global → Participants List** sub-tab (open padlock icon). • **All** Dedicated Cash Wallets owned by the unblocked Market Participant are automatically switched to *Unblocked* status in the **Dedicated Cash Wallets** tab, including any DCWs that were created during the blocked period. • Subsequent transactions involving the unblocked Market Participant (cash-token settlement or Direct Settlement) proceed normally through the standard validation and settlement workflows. • Subsequent DCW creation requests for the unblocked Market Participant result in DCWs created with the default *Unblocked* status. • The unblocking action is auditable: the change is reflected in the participants list and the wallets list (per SDD §3.4.9).
Test evidence	**U2A** • Screenshot of the **Global → Participants List** sub-tab *before* the unblocking action, showing the target Market Participant with the plain blue padlock (closed) icon (*Blocked*). • Screenshot of the confirmation pop-up window displayed after clicking on the closed padlock icon. • Screenshot of the **Global → Participants List** sub-tab *after*

	the confirmation, showing the target Market Participant with the open blue padlock icon (*Unblocked*). • Screenshot of the **Dedicated Cash Wallets** tab filtered on the Owner BIC of the unblocked Market Participant, showing all its DCWs in *Unblocked* status (including any DCWs that were previously in *Blocked* status as a side-effect of the participant blocking, and any DCWs created during the blocked period). • (Optional Step 6) Screenshot of a transaction involving the unblocked Market Participant reaching *SETTLED* status without ineligibility warning. • (Optional Step 7) Screenshot of a newly created DCW for the unblocked Market Participant showing *Unblocked* status from creation.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.4.9 *Market Participant Blocking / Unblocking* (immediately effective, not 4-eyes, effects on DCWs and on Direct Settlement); §3.4.2 *Market Participants* (data model – blocking status); §3.4.4 *Dedicated Cash Wallets* (data model – wallet blocking status); Table F *List of reference data features in U2A and A2A* (Participant – Unblock: 2-eyes ✓ / 4-eyes No). • PONTES Pilot – User Handbook v1.0: §4.12 *Blocking and unblocking participants*; §4.12.1 *Blocked and unblocked statuses* (incl. §4.12.1.1 *Participant status* – padlock icon semantics, §4.12.1.2 *Dedicated Cash Wallet status*); §4.12.3 *Unblocking a participant* (incl. §4.12.3.1 *Unblocking capability*, §4.12.3.2 *Participant unblocking*); §3.2.9 *Global*; §4.8.2 *Participants List*.
Related profiles/privileges	• User profile: **Referential Read Write** – required to execute the unblocking action (per UHB §4.12.3.1)

PP-NCB-06 – Block a Market DLT Operator using the blocking status attribute

Field	Description
Test Case ID	PP-NCB-06
Test case name	Block a Market DLT Operator using the blocking status attribute

EUROPEAN CENTRAL BANK
EUROSYSTEM

Relevant for (actor)	National Central Bank (NCB) – acting as the **responsible NCB** for the Market DLT Operator to be blocked (i.e. the NCB that onboarded the MDLT Operator per PP-NCB-02)
Input mode	U2A
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to block, in U2A, a Market DLT Operator (MDLTO) for which it acts as responsible NCB by setting its *Blocking Status* attribute to *Blocked*. The blocking action is **immediately effective** and **is not submitted to the four-eyes principle** (per SDD §3.4.10). Unlike Market Participant blocking (PP-NCB-04), a Market DLT Operator does **not own any Dedicated Cash Wallets**; therefore, blocking an MDLTO does **not** cascade to any DCW. Instead, the effect of blocking an MDLTO is on **transactions instructed by it on behalf of Market Participants** and on **DvP / PvP transactions where the MDLTO is referenced as responsible for the asset leg** (per SDD §3.4.10 effects list). **U2A** *Step 1 – Authentication:* 1. The NCB user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Referential Read Write** profile. *Step 2 – Locate the target Market DLT Operator:* 2. Select the **External Network** tab from the main sidebar menu, then the **Network List** sub-tab. 3. Locate the Market DLT Platform on which the target Market DLT Operator is registered and click on the **View Operators** link in the row of that platform. 4. In the list of Market DLT Operators displayed, locate the target Market DLT Operator (filter or search by Market DLT Operator ID or Responsible NCB if needed). 5. Verify, in the **Blocking Status** column, that the target Market DLT Operator currently displays **Not blocked** (open padlock icon, by analogy with the Participants List per UHB §4.12.1.1) – indicating that (i) the connected NCB is the responsible NCB of the MDLTO and the action is permitted, and (ii) the MDLTO is currently *Not blocked*. *Step 3 – Block the Market DLT Operator:* 6. Click on the **blue padlock icon (open)** in the row of the target Market DLT Operator, under the *Blocking Status*

EUROPEAN CENTRAL BANK
EUROSYSTEM

	column. 7. A confirmation pop-up window appears asking to either confirm or cancel the blocking action. 8. Click on the **Confirm** button. The blocking action is immediately effective – no four-eyes approval is requested. *Step 4 – Verification on the Market DLT Operator:* 9. Remain on the **External Network → Network List → View Operators** screen and verify that the icon in the *Blocking Status* column for the target Market DLT Operator has changed to a **plain blue padlock (closed)**, and that the *Blocking Status* value is now *Blocked* (per SDD Table L: possible values *Blocked* / *Not blocked*). 10. Verify that the same blocking status is reflected in the Market DLT Operators query (UHB §4.9.2: *Blocking Status* column) and, where consulted, in the Reporting/extraction queries (per SDD §3.6 if MDLTO extractions are in scope). *Step 5 – Verification of the effect on transactions (Optional but recommended):* 11. Identify or set up a Market Participant *MPA* that is whitelisted with the blocked MDLTO (per PP-NCB-02). 12. As an External User of the blocked MDLTO (A2A profile, instructing on behalf of MPA), or alternatively as a user simulating such an instruction in U2A where supported, attempt to submit a transaction of each of the following types on behalf of *MPA*: • Payment with cash token • Payment with Direct T2 RTGS settlement • Payment Free of Delivery (PFoD) • DvP / PvP 13. Verify that each of the attempted transactions is **rejected** by the Pilot solution (per SDD §3.4.10 effects list). 14. As a separate verification, identify or set up a Market Participant *MPB* that is whitelisted with the blocked MDLTO and that holds a DvP/PvP for which the blocked MDLTO is referenced as responsible for the asset leg. Verify that the DvP/PvP is **rejected** by the Pilot solution (per SDD §3.4.10 second effect). *Step 6 – Verification of the absence of effect on DCWs:* 15. Navigate to the **Dedicated Cash Wallets** tab. 16. Confirm that the DCWs of the Market Participants whitelisted with the blocked MDLTO are **not** affected by the MDLTO blocking (they remain in *Unblocked* status, unless they were independently blocked for other reasons such as PP-NCB-04). This confirms the difference in behaviour vs. PP-NCB-04 (where MP blocking cascades to its DCWs).
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• The target Market DLT Operator has been previously onboarded by the connected NCB (i.e. the connected NCB is the **Responsible NCB** of that MDLTO – per PP-NCB-02 Step 1) and is currently in *Not blocked* status. Per SDD §3.4.9 paragraph 1, *"A Market DLT Operator can only be blocked or unblocked by its responsible NCB"*. • At least one Market Participant under the same NCB has been **whitelisted** with this Market DLT Operator (per PP-NCB-02 Steps 4–5) and has an active External User for the MDLTO (per PP-NCB-02 Step 3), enabling the Step 5 verifications. • The NCB user holds the **Referential Read Write** profile. • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window.
Expected results	**U2A** • The blocking action is **immediately effective** with no four-eyes approval step. • The Market DLT Operator's *Blocking Status* changes from *Not blocked* to *Blocked* and is reflected in the **External Network → Network List → View Operators** view (closed padlock icon / *Blocked* value) and in the Market DLT Operators query/extraction. • Any subsequent attempt by the blocked MDLTO (acting on behalf of a Market Participant) to submit any of the following transaction types is **rejected** by the Pilot solution: Payment with cash token, Payment with Direct T2 RTGS settlement, Payment Free of Delivery (PFoD), DvP and PvP. • Any DvP or PvP instructed by any Market Participant where the blocked MDLTO is referenced as responsible for the asset leg is **rejected** by the Pilot solution. • DCWs of Market Participants whitelisted with the blocked MDLTO are **not** affected by this blocking action and remain in their pre-existing blocking status. • The blocking action is auditable: the change is reflected in the Market DLT Operators list (per SDD §3.4.10).
Test evidence	**U2A** • Screenshot of the **External Network → Network List → View Operators** screen *before* the blocking action, showing the target Market DLT Operator with open padlock icon (*Not blocked*). • Screenshot of the confirmation pop-up window displayed after clicking on the padlock icon. • Screenshot of the **External Network → Network List → View Operators** screen *after* the confirmation, showing the target Market DLT Operator with closed (plain blue) padlock icon (*Blocked*). • Screenshot or extract from the Market DLT Operators query

EUROPEAN CENTRAL BANK
EUROSYSTEM

	(UHB §4.9.2) showing the *Blocking Status* column with value *Blocked* for the target MDLTO. • (Optional Step 5) For each rejected transaction type attempted by the blocked MDLTO on behalf of a Market Participant: screenshot of the rejection notification and/or copy of the rejected message (with the rejection reason indicating the MDLTO blocking). • (Optional Step 5) Screenshot or copy of the rejection notification for a DvP/PvP instructed by a Market Participant where the blocked MDLTO is referenced as responsible for the asset leg. • Screenshot of the **Dedicated Cash Wallets** tab for one of the whitelisted Market Participants confirming that its DCWs remain in *Unblocked* status.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.4.9 *Market Participant Blocking / Unblocking* (introductory paragraph stating the responsible-NCB-only rule for MDLTO); §3.4.10 *Market DLT Operator Blocking / Unblocking* (immediately effective, not 4-eyes, effects on transactions); §3.4.6 *Market DLT Operators* (data model – *BLOCKING STATUS* attribute, Table L); §3.5.4 *Direct T2 RTGS Payment instructed by Market DLT Operator on behalf of Market Participant*; Table F *List of reference data features in U2A and A2A* (Market DLT Operator – Block: 2-eyes ✓ / 4-eyes No). • PONTES Pilot – User Handbook v1.0: §3.2.10 *External Network*; §4.9 *External Network* (incl. §4.9.2 *Adding a new Market DLT Operator* – describes Market DLT Operator list display with *Blocking Status* column); §4.12 *Blocking and unblocking participants*.
Related profiles/privileges	User profile: **Referential Read Write** – required to execute the blocking action

PP-NCB-07 – Unblock a previously blocked Market DLT Operator using the blocking status attribute

Field	Description
Test Case ID	PP-NCB-07
Test case name	Unblock a previously blocked Market DLT Operator using the blocking status attribute

EUROPEAN CENTRAL BANK
EUROSYSTEM

Relevant for (actor)	National Central Bank (NCB) – acting as the **responsible NCB** for the Market DLT Operator to be unblocked (i.e. the NCB that onboarded the MDLT Operator per PP-NCB-02 and that previously blocked it per PP-NCB-06)
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to unblock, in U2A, a Market DLT Operator (MDLTO) for which it acts as responsible NCB and that is currently in *Blocked* status, by setting its *Blocking Status* attribute back to *Not blocked*. The unblocking action is **immediately effective** and **is not submitted to the four-eyes principle** (per SDD §3.4.10). Unlike Market Participant unblocking (PP-NCB-05), the unblocking of a Market DLT Operator does **not** trigger any cascade on Dedicated Cash Wallets (as a Market DLT Operator does not own any DCWs). The effect of unblocking is to **restore the MDLTO's ability to instruct transactions** on behalf of whitelisted Market Participants and to **restore the eligibility of DvP / PvP transactions** referencing the MDLTO as responsible for the asset leg. This is the symmetric counterpart of PP-NCB-06. **U2A** *Step 1 – Authentication:* 1. The NCB user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Referential Read Write** profile. *Step 2 – Locate the target blocked Market DLT Operator:* 2. Select the **External Network** tab from the main sidebar menu, then the **Network List** sub-tab. 3. Locate the Market DLT Platform on which the target Market DLT Operator is registered and click on the **View Operators** link in the row of that platform. 4. In the list of Market DLT Operators displayed, locate the target Market DLT Operator (filter or search by Market DLT Operator ID or Responsible NCB if needed). 5. Verify, in the **Blocking Status** column, that the target Market DLT Operator currently displays **Blocked** (plain blue padlock / closed icon, by analogy with the Participants List per UHB §4.12.1.1) – indicating that the MDLTO is currently *Blocked* and that the connected NCB is the responsible NCB authorised to perform the action.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 3 – Unblock the Market DLT Operator: 6. Click on the **plain blue padlock (closed)** icon in the row of the target Market DLT Operator, under the *Blocking Status* column. 7. A confirmation pop-up window appears asking to either confirm or cancel the unblocking action. 8. Click on the **Confirm** button. The unblocking action is immediately effective – no four-eyes approval is requested. *Step 4 – Verification on the Market DLT Operator:* 9. Remain on the **External Network → Network List → View Operators** screen and verify that the icon in the *Blocking Status* column for the target Market DLT Operator has changed back to an **open blue padlock**, and that the *Blocking Status* value is now *Not blocked* (per SDD Table L: possible values *Blocked* / *Not blocked*). 10. Verify that the same blocking status is reflected in the Market DLT Operators query (UHB §4.9.2: *Blocking Status* column) and, where consulted, in the Reporting/extraction queries (per SDD §3.6 if MDLTO extractions are in scope). *Step 5 – Verification of the restored ability to instruct transactions (Optional but recommended):* 11. Identify the same Market Participant *MPA* used in PP-NCB-06 Step 5 (or another Market Participant whitelisted with the unblocked MDLTO). 12. As an External User of the unblocked MDLTO (A2A profile, instructing on behalf of MPA), or alternatively as a user simulating such an instruction in U2A where supported, submit a regular transaction of each of the following types on behalf of *MPA*: • Payment with cash token • Payment with Direct T2 RTGS settlement • Payment Free of Delivery (PFoD) • DvP / PvP 13. Verify that each of these transactions is **no longer rejected** due to MDLTO blocking and proceeds normally through the relevant settlement workflow (any rejection observed at this stage should have an unrelated business reason, not the blocking status of the MDLTO). 14. As a separate verification, set up a DvP/PvP referencing the unblocked MDLTO as responsible for the asset leg and verify that it **proceeds normally** through the relevant settlement workflow. *Step 6 – Verification of the absence of impact on DCWs:* 15. Navigate to the **Dedicated Cash Wallets** tab. 16. Confirm that the DCWs of the Market Participants whitelisted with the unblocked MDLTO are **not** affected by the unblocking action (their *Blocking Status* remains unchanged

EUROPEAN CENTRAL BANK
EUROSYSTEM

	from before – consistent with the absence of cascade behaviour observed in PP-NCB-06 Step 6).
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator. • The target Market DLT Operator has been previously onboarded by the connected NCB (i.e. the connected NCB is the **Responsible NCB** of that MDLTO – per PP-NCB-02 Step 1) and **has been previously blocked** (see PP-NCB-06) so that it is **currently in the *Blocked* status**. This is the mandatory entry condition for PP-NCB-07. Per SDD §3.4.9 paragraph 1, *"A Market DLT Operator can only be blocked or unblocked by its responsible NCB"*. • At least one Market Participant under the same NCB has been **whitelisted** with this Market DLT Operator (per PP-NCB-02 Steps 4–5) and has an active External User for the MDLTO (per PP-NCB-02 Step 3), enabling the Step 5 verifications. • The NCB user holds the **Referential Read Write** profile. • Per the symmetry rule established by UHB §4.12.3.1 for Participant unblocking, *"Participants allowed to unblock are the same one as the one allowed to block"* – which is assumed to apply to MDLTOs as well (i.e. the responsible NCB with Referential Read Write profile). • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window. • (For optional Step 5) An External User of the unblocked MDLTO is available to attempt the verification transactions.
Expected results	**U2A** • The unblocking action is **immediately effective** with no four-eyes approval step. • The Market DLT Operator's *Blocking Status* changes from *Blocked* to *Not blocked* and is reflected in the **External Network → Network List → View Operators** view (open padlock icon / *Not blocked* value) and in the Market DLT Operators query/extraction. • Subsequent transactions instructed by the unblocked MDLTO on behalf of a whitelisted Market Participant (Payment with cash token, Payment with Direct T2 RTGS settlement, PFoD, DvP/PvP) proceed normally through the relevant settlement workflows. • Subsequent DvP and PvP transactions referencing the unblocked MDLTO as responsible for the asset leg proceed normally through the relevant settlement workflows. • DCWs of Market Participants whitelisted with the unblocked MDLTO are **not** affected by this unblocking action and remain in their pre-existing blocking status (no cascade behaviour, consistent with PP-NCB-06).

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• The unblocking action is auditable: the change is reflected in the Market DLT Operators list (per SDD §3.4.10).
Test evidence	**U2A** • Screenshot of the **External Network** → **Network List** → **View Operators** screen *before* the unblocking action, showing the target Market DLT Operator with closed (plain blue) padlock icon (*Blocked*). • Screenshot of the confirmation pop-up window displayed after clicking on the padlock icon. • Screenshot of the **External Network** → **Network List** → **View Operators** screen *after* the confirmation, showing the target Market DLT Operator with open padlock icon (*Not blocked*). • Screenshot or extract from the Market DLT Operators query (UHB §4.9.2) showing the *Blocking Status* column with value *Not blocked* for the target MDLTO. • (Optional Step 5) For each transaction type successfully submitted by the unblocked MDLTO on behalf of a Market Participant: screenshot of the successful submission notification and/or copy of the settled message. • (Optional Step 5) Screenshot or copy of the successful settlement notification for a DvP/PvP referencing the unblocked MDLTO as responsible for the asset leg. • Screenshot of the **Dedicated Cash Wallets** tab for one of the whitelisted Market Participants confirming that its DCWs' *Blocking Status* is unchanged.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.4.9 *Market Participant Blocking / Unblocking* (introductory paragraph stating the responsible-NCB-only rule for MDLTO); §3.4.10 *Market DLT Operator Blocking / Unblocking* (immediately effective, not 4-eyes); §3.4.6 *Market DLT Operators* (data model – *BLOCKING STATUS* attribute, Table L); §3.5.4 *Direct T2 RTGS Payment instructed by Market DLT Operator on behalf of Market Participant*; Table F *List of reference data features in U2A and A2A* (Market DLT Operator – Unblock: 2-eyes ✓ / 4-eyes No). • PONTES Pilot – User Handbook v1.0: §3.2.10 *External Network*; §4.9 *External Network* (incl. §4.9.2 *Adding a new Market DLT Operator* – describes Market DLT Operator list display with *Blocking Status* column); §4.12 *Blocking and unblocking participants*.
Related profiles/privileges	User profile: **Referential Read Write** – required to execute the unblocking action

EUROPEAN CENTRAL BANK
EUROSYSTEM

PP-NCB-08 – Perform wallet-to-wallet cash token payment on behalf of a Market Participant (contingency, without PoA)

Field	Description
Test Case ID	PP-NCB-08
Test case name	Perform wallet-to-wallet cash token payment on behalf of a Market Participant (contingency, without PoA)
Relevant for (actor)	National Central Bank (NCB) – acting as **Dedicated Cash Wallet Manager** for the Market Participant being served. The NCB acts on behalf of the Market Participant in a contingency situation (e.g. the Market Participant is unable to instruct the payment itself); the action does **not** require any Power of Attorney (PoA) configuration.
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for an NCB to instruct, **on behalf of one of its Market Participants** and in a contingency situation, a wallet-to-wallet cash token Payment debiting a DCW owned by that Market Participant (and managed by the NCB) and crediting a DCW owned by another Market Participant. The NCB's authority to do so derives from its **DCW Manager role** (SDD §2.2.2: *"The instruction of all transactions related to managed DCWs on behalf of their Market Participants (funding & defunding requests, cash token transfers, payments, issuances, redemptions)* ***in case of contingency***") and **not** from a Power of Attorney (PoA) configuration; no PoA setup is required for this scenario. The payment is submitted following the standard **4-eyes principle** applicable to wallet-to-wallet cash token Payments in U2A (per SDD Table Q). **U2A** *Step 1 – Authentication (User 1 – NCB user with Pilot Read Write profile):* 1. The NCB User 1 authenticates to the Pilot solution GUI using credentials and the double authentication factor. *Step 2 – Create the Payment request:* 2. Select the **Transactions** tab from the main sidebar menu.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	3. Click on the **New transaction** button (top right of the screen). 4. Select **Payments** as the transaction type (per UHB §4.4.2). 5. Fill in the following fields: • **Debited Dedicated Cash Wallet ID**: select from the drop-down list the DCW of the Market Participant on whose behalf the NCB acts. Since the NCB is the DCW Manager, the drop-down lists both NCB-owned DCWs and DCWs of all Market Participants under the NCB's jurisdiction. The available balance of the selected DCW is displayed under the drop-down. • **Credited Dedicated Cash Wallet ID**: enter or select the DCW owned by the receiving Market Participant. • **Credited Dedicated Cash Wallet Manager BIC**: enter the BIC of the manager of the credited DCW (may be auto-deduced from the Credited DCW ID). • **Amount**: enter the amount of the payment (must be ≤ Available Balance of the debited DCW). • **Currency**: automatically pre-filled (EUR). • **Today / Later**: select **Today** for immediate settlement (current Business Date is set by default), or **Later** with a specific Intended Settlement Date for a future-dated payment. • **Payment Reference**: optional 30-character free text for reconciliation / correlation purposes. 6. Click on the **Submit** button. The payment request is submitted to the 4-eyes validation flow and appears in the **Pending Validations – Transactions** sub-tab with status *PENDING APPROVAL*. *Step 3 – Approve the Payment (User 2 – NCB user with Pilot Read Write profile, different from User 1):* 7. The NCB User 2 authenticates to the Pilot solution GUI. 8. Select the **Pending Validations** tab, then the **Transactions** sub-tab. 9. Locate the payment in *PENDING APPROVAL* status and click on the **Approve** button (per UHB §4.7.2). The payment details are displayed in read-only mode. If a correction is needed, the User 2 can click on **To be reviewed** to send the item back to the initiator (UHB §4.7.5). 10. Click on the **Approve for creation** button (or equivalent confirmation button) to validate the payment. A pop-up confirms the approval. *Step 4 – Verification of settlement:* 11. Navigate to the **Transactions** tab. 12. Filter on the Payment created in Step 2 and verify that the transaction status evolves to *Settled* (per SDD §3.6.2.c / Transaction status values). 13. Click on the Transaction ID to open the transaction detail view and verify: • Debited Wallet ID corresponds to the DCW selected in Step 5 (the MP's DCW);

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• Credited Wallet ID corresponds to the DCW of the receiving Market Participant; • Amount and Currency are correct; • Initiator = NCB User 1 (Username); • Approver = NCB User 2 (Username); • Settlement Timestamp is populated and Settlement Business Date is the expected one. *Step 5 – Verification of balances:* 14. Navigate to the **Dedicated Cash Wallets** tab. 15. Verify that the *Available Balance* of the debited DCW (MP-A) has decreased by the amount of the payment. 16. Verify that the *Available Balance* of the credited DCW (MP-B) has increased by the amount of the payment.
Preconditions/Details	• The NCB is set up in the Pilot solution by the Service Providers (SPs) as Operator, with the *DCW Manager* role. • At least two Market Participants are configured under the NCB's jurisdiction (per PP-NCB-01): the **debited Market Participant (MP-A)**, on whose behalf the NCB acts, and the **credited Market Participant (MP-B)** (may be under the same or a different NCB). • MP-A holds an active DCW with **sufficient cash token balance** to cover the payment amount (the DCW has been previously funded, e.g. via PP-E2E-001). • MP-B holds an active DCW under its own NCB. • Neither MP-A nor MP-B is **blocked** (per PP-NCB-04 / PP-NCB-05 semantics); both DCWs are *Unblocked*. • Both DCWs are within their *Valid From / Valid To* period. • A **contingency situation** justifies the NCB instructing the payment on behalf of MP-A (per SDD §2.2.2 and §3.5.4.a: NCB instruction on behalf is restricted to contingency). The operational evidence of the contingency justification is out of scope of the system but should be documented in the test execution log. • **No Power of Attorney (PoA) configuration is required**: the NCB's authority derives from its DCW Manager role. The *PoA Given To* attribute of the debited DCW is **not** set (or, if set, is irrelevant to this scenario since the NCB is not acting as PoA holder). • Two distinct NCB users with the **Pilot Read Write** profile are available to perform the 4-eyes principle (initiator and approver). • Both users hold valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window.
Expected results	**U2A** • The Payment request is successfully created by NCB User 1 and approved by NCB User 2 through the 4-eyes process.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• The payment is settled on the Pilot solution: – the DCW of MP-A (debited) is decreased by the payment amount; – the DCW of MP-B (credited) is increased by the payment amount; – the payment status reaches *Settled*. • The transaction is recorded with the NCB users (Initiator = User 1, Approver = User 2) as the operating users, while the **economic counterparties** are the two Market Participants (Debited Owner Entity ID = MP-A's BIC, Credited Owner Entity ID = MP-B's BIC). • No PoA configuration was used or required for the operation. • The transaction is visible: – in the **Transactions** tab of the NCB user (as Manager of MP-A's DCW); – in the **Transactions** tab of MP-A (as Owner of the debited DCW), if MP-A regains connectivity; – in the **Transactions** tab of MP-B (as Owner of the credited DCW). • The transaction appears in the cash-token Transactions extraction (SDD §3.6.2.c) with type *Payment* and status *Settled*.
Test evidence	**U2A** • Screenshot of the **Transactions → New transaction → Payments** screen (NCB User 1) showing the entered Debited DCW ID (MP-A's DCW), Credited DCW ID (MP-B's DCW), Credited DCW Manager BIC, Amount and Payment Reference. • Screenshot of the **Pending Validations – Transactions** sub-tab showing the payment in *PENDING APPROVAL* status. • Screenshot of the approval pop-up displayed to NCB User 2, followed by the confirmation pop-up after approval. • Screenshot of the **Transactions** tab showing the payment in *Settled* status. • Screenshot of the **Transaction detail** view showing Debited Wallet ID, Credited Wallet ID, Amount, Currency, Initiator (NCB User 1), Approver (NCB User 2), Settlement Timestamp and Settlement Business Date. • Screenshot of the **Dedicated Cash Wallets** tab showing MP-A's DCW with the decreased balance. • Screenshot of the **Dedicated Cash Wallets** tab (or, if executed in parallel by MP-B, screenshot from MP-B's session) showing MP-B's DCW with the increased balance. • Excel export of the cash-token Transactions extraction (per SDD §3.6.2.c), filtered on the Operation ID, evidencing the full transaction details.

Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.2 *Dedicated Cash Wallet Manager* (role granting the authority to instruct on behalf in contingency); §2.2.3 *Dedicated Cash Wallet Owner*; §3.4.4 *Dedicated Cash Wallets* (PoA attributes); §3.5.4.a *Payment with cash tokens* (workflow figure vii, 4-eyes principle, NCB on behalf in contingency); Table Q *List of features available to Market Participants* (Wallet to Wallet Payment - Instruct: U2A ✓ / 2-eyes No / 4-eyes ✓); §3.6.2.c *Transactions for a Dedicated Cash Wallet query* (extraction filters and fields). • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions*; §4.4 *Transactions* (entry point); §4.4.2 *Wallet to wallet Payments* (field list and submission flow); §4.7 *Pending Validation (Four-eyes process)*; §4.7.2 *Approval of Transactions*; §4.7.5 *Reviewing a Pending validation item*.
Related profiles/privileges	• User profile: **Pilot Read Write** – for both initiator (User 1) and approver (User 2)

2.3 Test cases for Market Participants

PP-MP-01 – Fund own DCW from own RTGS DCA

Field	Description
Test Case ID	PP-MP-01
Test case name	Fund own DCW from own RTGS DCA
Relevant for (actor)	Market Participant (Dedicated Cash Wallet Owner)
Input mode	U2A (A2A also possible)
Mandatory	Yes
Detailed Description	This test case describes the steps required for a Market Participant, acting as Dedicated Cash Wallet (DCW) Owner, to fund one of its own Dedicated Cash Wallets by debiting its own T2 RTGS DCA. The request is initiated in U2A and submitted to the four-eyes validation process. Upon approval, the Pilot solution automatically triggers the payment instruction in T2 RTGS and, once the cash leg is settled, the issuance

EUROPEAN CENTRAL BANK
EUROSYSTEM

	request crediting the DCW with the corresponding amount of cash tokens. **U2A** *Step 1 – Initiation of the Funding Request (User 1 – Pilot Read Write):* 1. User 1 authenticates to the Pilot solution GUI using its credentials and the double authentication factor. 2. Select the **Funding & Defunding** tab from the main sidebar menu. 3. Click on the **New Request** button. 4. Select **Funding** as the request type. 5. Fill in the following fields: • **Wallet ID to be credited**: select the own DCW to be funded from the drop-down list. • **Amount**: enter the amount of the funding (positive value, EUR, format compliant with FR-AMT-001). • **Currency**: EUR (set by default). 6. Upload the **certificate** and the corresponding **private key** in the Signature details section. 7. Click on the **Generate** button to introduce a valid digital signature. 8. Click on the **Submit** button to send the Funding Request to four-eyes validation. The request status moves to *PENDING APPROVAL*. *Step 2 – Approval of the Funding Request (User 2 – Pilot Read Write, different from User 1):* 9. User 2 authenticates to the Pilot solution GUI. 10. Select the **Pending Validations** tab, then the **Funding & Defunding** sub-tab. 11. Locate the funding request in *PENDING APPROVAL* status and click on the **Approve** button. 12. Review the funding request details (Wallet ID, Amount, Currency). The information is read-only. 13. Upload User 2's **certificate** and **private key**. 14. Click on the **Generate** button. 15. Click on the **Approve for creation** button. A confirmation pop-up appears confirming the validation. *Step 3 – Verification of automatic processing:* 16. Navigate back to the **Funding & Defunding** tab and verify that the request status evolves through the following statuses: *ACCEPTED → PAYMENT_SUBMITTED → PAYMENT_SETTLED → ISSUANCE_CREATED → SETTLED*. 17. Navigate to the **Dedicated Cash Wallets** tab and verify that the balance of the credited DCW has been increased by the funded amount.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Preconditions/Details	• The Market Participant is duly set up in the Pilot solution with the role *DCW Owner*. • The Market Participant holds an active T2 RTGS DCA opened in T2 RTGS. • A valid **T2 account reference** linking the Market Participant to its T2 RTGS DCA is declared in the Pilot solution by the responsible NCB. • An active **Dedicated Cash Wallet** owned by the Market Participant is set up in the Pilot solution and linked to the above T2 account reference. • The Market Participant is **not blocked** (and neither is the DCW). • A **direct debit mandate** is in place in T2 RTGS allowing the Pilot solution to debit the Market Participant's T2 RTGS DCA. • Sufficient liquidity is available on the T2 RTGS DCA to cover the funding amount. • Two distinct users with the *Pilot Read Write* profile are available to perform the four-eyes principle. • Both users hold a valid certificate and the corresponding private key for signature purposes (non-expired). • The test is executed during the T2 RTGS settlement window.
Expected results	**U2A** • The Funding Request is successfully created and approved through the four-eyes process. • The payment instruction is automatically generated and successfully settled in T2 RTGS, debiting the Market Participant's T2 RTGS DCA and crediting (via the Technical Interim Account) the ECB's Token Issuance Account. • The issuance request is automatically generated and successfully settled in the ESY DLT, debiting the ECB's Token Issuance Wallet and crediting the Market Participant's DCW with the requested amount of cash tokens. • The final status of the Funding Request is *SETTLED*. • The balance of the credited DCW is increased by the funded amount; the T2 RTGS DCA is debited by the same amount.
Test evidence	**U2A** • Screenshot of the Funding Request submission screen (User 1) showing the entered Wallet ID, Amount and Currency. • Screenshot of the *PENDING APPROVAL* item in the Pending Validations – Funding & Defunding sub-tab. • Screenshot of the confirmation pop-up after approval by User 2. • Screenshot of the Funding & Defunding tab showing the request in *SETTLED* status, with Initiator and Approver clearly identifiable. • Screenshot of the credited DCW details (Dedicated Cash Wallets tab) showing the updated balance. • Extract (Excel) from the Funding & Defunding extraction sub-

	tab, filtered on the Request ID, evidencing the full lifecycle timestamps (Creation, Acceptance, Payment Submission, Payment Settlement, Issuance Creation, Issuance Settlement, Settlement).
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §3.5.1 *Funding Requests*; fig. iv *Funding request workflow in U2A mode*; §3.4.4 *Dedicated Cash Wallets*; §3.5.7.a *Unhappy Paths & Error codes – Funding Requests*. • PONTES Pilot – User Handbook v1.0: §4.5 *Funding & Defunding*; §4.7.3 *Approval of Funding & Defunding*; §3.2.6 *Funding and Defunding* (menu structure); §4.1.2 *Connect with User credentials*; §4.1.3 *Double Authentication Factor*.
Related profiles/privileges	• User profile: **Pilot Read Write** – for both initiator (User 1) and approver (User 2)

PP-MP-02 – Perform wallet-to-wallet cash token payment from own DCW to another Participant's DCW

Field	Description
Test Case ID	PP-MP-02
Test case name	Perform wallet-to-wallet cash token payment from own DCW to another Participant's DCW
Relevant for (actor)	Market Participant – acting as **Dedicated Cash Wallet Owner** of the debited DCW. The Market Participant instructs the payment on its own behalf as part of normal operations.
Input mode	U2A (A2A also possible)
Mandatory	Yes (but the Market Participant can delegate the execution to a Market DLT Operator)
Detailed Description	This test case describes the steps required for a Market Participant (MP-A), acting as Dedicated Cash Wallet Owner, to instruct in U2A a wallet-to-wallet cash token Payment debiting one of its own DCWs and crediting a DCW owned by another Market Participant (MP-B). This is the standard "happy path" scenario for wallet-to-wallet cash token Payments, where the

EUROPEAN CENTRAL BANK
EUROSYSTEM

	DCW Owner self-instructs the payment as part of normal operations (no contingency, no instruct-on-behalf delegation). The MP-A's authority to do so derives from its **DCW Owner role** (SDD §2.2.3: *"This role allows: The instruction of all the transactions on owned DCWs (funding & defunding requests, wallet-to-wallet transfers, wallet-to-wallet payments, PFoDs, DvPs, PvPs)"*). The payment is submitted following the **4-eyes principle** applicable to wallet-to-wallet cash token Payments in U2A (per SDD Table Q: *Wallet to Wallet Payment - Instruct: U2A ✓/ 2-eyes No / 4-eyes ✓*). **U2A** *Step 1 – Authentication (MP-A User 1 – Pilot Read Write profile):* 1. The MP-A User 1 authenticates to the Pilot solution GUI using credentials and the double authentication factor. *Step 2 – Create the Payment request:* 2. Select the **Transactions** tab from the main sidebar menu. 3. Click on the **New transaction** button (top right of the screen). 4. Select **Payments** as the transaction type (per UHB §4.4.2 – note: this is the *Transactions → Payments* sub-menu, not the *Direct RTGS → Payments* sub-menu). 5. Fill in the following fields: • **Debited Dedicated Cash Wallet ID**: select from the drop-down list one of the MP-A's own DCWs (only owned DCWs are visible to the MP-A user, per UHB §4.4.2). The available balance of the selected DCW is displayed under the drop-down. • **Credited Dedicated Cash Wallet ID**: enter or select the DCW owned by the receiving Market Participant (MP-B). • **Credited Dedicated Cash Wallet Manager BIC**: enter the BIC of the manager of the credited DCW (i.e. the BIC of MP-B's responsible NCB). In some cases, this value can be auto-deduced from the Credited DCW ID. • **Amount**: enter the amount of the payment (must be ≤ Available Balance of the debited DCW). • **Currency**: automatically pre-filled (EUR – the only currency configured in the solution, per UHB §4.8.3). • **Today / Later**: select **Today** for immediate settlement (current Business Date is set by default), or **Later** with a specific Intended Settlement Date for a future-dated payment. • **Payment Reference**: optional 30-character free text for reconciliation / correlation purposes. 6. Click on the **Submit** button. The payment request is submitted to the 4-eyes validation flow and appears in the **Pending Validations – Transactions** sub-tab of MP-A with

EUROPEAN CENTRAL BANK
EUROSYSTEM

	status *PENDING APPROVAL*. *Step 3 – Approve the Payment (MP-A User 2 – Pilot Read Write profile, different from User 1):* 7. The MP-A User 2 authenticates to the Pilot solution GUI. 8. Select the **Pending Validations** tab, then the **Transactions** sub-tab. 9. Locate the payment in *PENDING APPROVAL* status and click on the **Approve** button (per UHB §4.7.2). The payment details are displayed in read-only mode. If a correction is needed, the User 2 can click on **To be reviewed** to send the item back to the initiator (UHB §4.7.5). 10. Click on the **Approve for creation** button (or equivalent confirmation button) to validate the payment. A pop-up confirms the approval. *Step 4 – Verification of settlement on MP-A side:* 11. As MP-A User, navigate back to the **Transactions** tab. 12. Locate the Payment created in Step 2 and verify that the transaction status has evolved to *Settled* (per SDD §3.6.2.c). 13. Click on the Transaction ID to open the transaction detail view and verify: • Debited Wallet ID corresponds to the MP-A's DCW selected in Step 2.5; • Credited Wallet ID corresponds to MP-B's DCW; • Amount and Currency are correct; • Initiator = MP-A User 1 (Username); • Approver = MP-A User 2 (Username); • Settlement Timestamp is populated and Settlement Business Date is the expected one. *Step 5 – Verification of balances:* 14. Navigate to the **Dedicated Cash Wallets** tab (MP-A view). 15. Verify that the *Available Balance* of the debited DCW (MP-A's wallet) has decreased by the amount of the payment. 16. (If executed in parallel on MP-B side) MP-B User logs in and verifies in its **Dedicated Cash Wallets** tab that the *Available Balance* of the credited DCW has increased by the amount of the payment, and that the transaction is visible in MP-B's **Transactions** tab.
Preconditions/Details	• MP-A is a duly onboarded Market Participant (per PP-NCB-01) with status *Active*. • MP-A holds at least one active DCW with **sufficient cash token balance** to cover the payment amount (e.g. previously funded via PP-E2E-001). • MP-B is a duly onboarded Market Participant (under the same or a different NCB) with at least one active DCW able to receive cash tokens. • Neither MP-A nor MP-B is **blocked** (their Blocking Status is *Unblocked*); both relevant DCWs are *Unblocked* and within

EUROPEAN CENTRAL BANK
EUROSYSTEM

	their *Valid From / Valid To* period. • Two distinct MP-A users with the **Pilot Read Write** profile are available to perform the 4-eyes principle (initiator and approver). • Both users hold valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window.
Expected results	• The Payment request is successfully created by MP-A User 1 and approved by MP-A User 2 through the 4-eyes process. • The payment is settled on the Pilot solution (per SDD §3.5.4.a workflow fig. vii): - the DCW of MP-A (debited) is decreased by the payment amount; - the DCW of MP-B (credited) is increased by the payment amount; - the payment status reaches *Settled*. - The transaction is recorded with: ○ Initiator = MP-A User 1; ○ Approver = MP-A User 2; ○ Debited Owner Entity ID = MP-A's BIC; ○ Credited Owner Entity ID = MP-B's BIC; ○ Settlement Timestamp / Settlement Business Date populated. - The transaction is visible: ○ in the **Transactions** tab of MP-A (as DCW Owner of the debited DCW); ○ in the **Transactions** tab of MP-B (as DCW Owner of the credited DCW); ○ in the **Transactions** tab of MP-A's responsible NCB (as DCW Manager of the debited DCW); ○ in the **Transactions** tab of MP-B's responsible NCB (as DCW Manager of the credited DCW). The transaction appears in the cash-token Transactions extraction (SDD §3.6.2.c) with type *Payment* and status *Settled*.
Test evidence	**U2A** • Screenshot of the **Transactions → New transaction → Payments** screen (MP-A User 1) showing the entered Debited DCW ID (MP-A's DCW), Credited DCW ID (MP-B's DCW), Credited DCW Manager BIC, Amount, Today/Later selection and (if used) Payment Reference. • Screenshot of the **Pending Validations – Transactions** sub-tab showing the payment in *PENDING APPROVAL* status.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• Screenshot of the approval pop-up displayed to MP-A User 2, followed by the confirmation pop-up after approval. • Screenshot of the **Transactions** tab (MP-A view) showing the payment in *Settled* status. • Screenshot of the **Transaction detail** view showing Debited Wallet ID, Credited Wallet ID, Amount, Currency, Initiator (MP-A User 1), Approver (MP-A User 2), Settlement Timestamp and Settlement Business Date. • Screenshot of the **Dedicated Cash Wallets** tab (MP-A view) showing MP-A's DCW with the decreased balance. • (If feasible) Screenshot from MP-B's session showing the credited DCW with the increased balance and the transaction visible in MP-B's Transactions tab. • Excel export of the cash-token Transactions extraction (per SDD §3.6.2.c), filtered on the Operation ID, evidencing the full transaction details.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.3 *Dedicated Cash Wallet Owner* (role allowing instruction of wallet-to-wallet payments on owned DCWs); §3.4.4 *Dedicated Cash Wallets* (data model); §3.5.4.a *Payment with cash tokens* (workflow figure vii, 4-eyes principle); Table Q *List of features available to Market Participants* (Wallet to Wallet Payment - Instruct: U2A ✓ / 2-eyes No / 4-eyes ✓); §3.6.2.c *Transactions for a Dedicated Cash Wallet query* (extraction filters and fields). • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions*; §4.4 *Transactions* (entry point); §4.4.2 *Wallet to wallet Payments* (field list and submission flow, incl. *Payment in the Future* paragraph); §4.7 *Pending Validation (Four-eyes process)*; §4.7.2 *Approval of Transactions*; §4.7.5 *Reviewing a Pending validation item*; §4.8.3 *Currencies List* (EUR-only).
Related profiles/privileges	• User profile: **Pilot Read Write** – for both initiator (MP-A User 1) and approver (MP-A User 2) If the execution is delegated to a Market DLT Operator (acting in A2A mode), the Market DLT Operator user needs the **External User** profile

PP-MP-03 – Perform Direct RTGS payment from own RTGS DCA to another Participant's RTGS DCA

Field	Description
Test Case ID	PP-MP-03

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test case name	Perform Direct RTGS payment from own RTGS DCA to another Participant's RTGS DCA (4 eyes mode)
Relevant for (actor)	Market Participant
Input mode	U2A (A2A also possible)
Mandatory	Yes (but the Market Participant can delegate the execution to a Market DLT Operator)
Detailed Description	This test case verifies that a Market Participant can instruct, validate (four-eyes) and settle a Direct RTGS Payment from its own Debited DCA to another Participant's Credited DCA, reaching the terminal status COMPLETED. **Steps (U2A):** 1) User 1 logs in with the Pilot Read Write profile. 2) Navigate to Direct RTGS → Payments (UHB §4.6.1). 3) Click New Direct RTGS Payment. 4) Fill in Payer BIC (Account BIC of own Debited DCA). 5) Fill in Receiver BIC (Account BIC of counterparty Credited DCA). 6) Fill in Amount (positive EUR value). 7) Fill in Correlation ID (End-to-End ID linking to the T2 RTGS payment). 8) Fill in Signature details (as per funding/defunding case). 9) Click Generate. 10) Click Submit for validation. 11) User 2 (distinct, Pilot Read Write) logs in. 12) Navigate to Pending Validations, verify details and Approve under the four-eyes process (UHB §4.7.5). 13) 13) Navigate to Monitoring → Direct RTGS Payments (UHB §4.11.3) and verify status COMPLETED.
Preconditions/Details	• Two distinct users of the Payer Market Participant, both with the Pilot Read Write profile and valid (non-expired) certificates/private keys (required for the Generate/signature step). • The Payer holds a valid, active T2 RTGS DCA (Debited DCA) referenced in the Pilot solution. • The Receiver Participant holds a valid, active T2 RTGS DCA (Credited DCA). • The Payer's DCA holds sufficient available balance. • The external T2 RTGS system is available to settle the underlying payment.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• The test is executed within an “open for all” business window.
Expected results	**U2A:** • The Direct RTGS Payment request is successfully created and submitted for validation. • The request appears in Pending Validations with status INITIALIZED. • A second, distinct user approves the request under the four-eyes process (UHB §4.7.5). • The payment progresses: INITIALIZED → ACCEPTED → PAYMENT_SUBMITTED → COMPLETED. • The Monitoring → Direct RTGS Payments screen displays the payment with final status COMPLETED. • The Payer’s Debited DCA is debited and the Receiver’s Credited DCA is credited. *Note (R-M04): U2A terminal status = COMPLETED; the A2A/REST equivalent returns SETTLED.*
Test evidence	**U2A:** 1) Screenshot of the completed New Direct RTGS Payment form (Payer BIC, Receiver BIC, Amount, Correlation ID, Signature). 2) Screenshot after Generate and Submit for validation. 3) Screenshot of the request in Pending Validations (status INITIALIZED). 4) Screenshot of User 2’s Approve action (four-eyes). 5) Screenshot of Monitoring → Direct RTGS Payments showing final status COMPLETED. 6) Evidence (e.g. from T2 RTGS GUI) of T2 RTGS DCA debit/credit (balance change) where available.
Relevant Documentation	UHB §4.6.1 (Direct RTGS Payments); UHB §4.7.5 (Approval of Direct RTGS Payments); UHB §4.11.3 (Monitoring Direct RTGS Payments); SDD – Payment with Direct RTGS (Service Description v0.4, p.40).
Related profiles/privileges	Pilot Read Write profile. Instruction rights: Dedicated Cash Wallet Owner, or Dedicated Cash Wallet Manager on behalf of its Market Participant (UHB §4.6.1). Four-eyes approval requires a second distinct user with the Pilot Read Write profile.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	If the execution is delegated to a Market DLT Operator (acting in A2A mode), the Market DLT Operator user needs the **External User** profile.

PP-MP-04 – Complete an XvP transaction with token settlement and retrieve execution key (A2A)

Field	Description
Test Case ID	PP-MP-04
Test case name	As two Market Participants, complete an XvP transaction with token settlement and retrieve execution key
Relevant for (actor)	Market Participants – two distinct MPs interacting via REST APIs: • **MP-S** acting as **Seller** (DCW Owner of the credited DCW, responsible for the asset leg on the Market DLT platform). • **MP-B** acting as **Buyer** (DCW Owner of the debited DCW, responsible for paying the cash leg). Both MPs self-instruct via A2A using their **External User** application accounts, or they can delegate the execution of this test case to a Market DLT Operator acting on their behalf.
Input mode	A2A
Mandatory	Yes (but the Market Participant can delegate the execution to a Market DLT Operator)
Detailed Description	This test case describes the end-to-end happy-path of an XvP (DvP or PvP) transaction with **Settlement in Cash Token**, instructed entirely via the Pilot solution REST APIs. The flow exercises the four XvP REST functionalities exposed by the Pilot solution for the cash-token settlement model (SDD §3.5.6.a–d): (i) XvP Initialisation by the Seller, (ii) XvP Initialisation Query by the Buyer, (iii) Payment by the Buyer, with execution key returned in the response, and (iv) Get Key & Payment Info by the Seller, confirming the *SETTLED* status of the cash leg. The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking of the asset leg, cooperative or forced execution of the HLC) is **outside the scope of the Pilot solution** (SDD §3.5.6.e) and therefore

EUROPEAN CENTRAL BANK
EUROSYSTEM

	outside the scope of this test case; the test verifies only that the Pilot solution returns the data needed by the Seller to initialise the HLC (execution hash, cancellation hash, timeout) and that the Buyer correctly receives the **Execution Key** upon successful payment. All XvP functionalities are exposed only in **A2A** mode (SDD §3.5.6.a–d) and follow the **one-step validation workflow** (no four-eyes principle on XvP – see OpenAPI v1.0 endpoint descriptions). Both MPs use the **External User** profile. The test case may be run as a **DvP** (`type = "DVP"`) or a **PvP** (`type = "PVP"`) – the functional flow is identical; the default execution is `DVP`. **A2A** *Step 0 – Authentication (both MPs) – METHOD: mTLS handshake + IAM token issuance – Code: 200:* 0.a MP-S's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (`MP-S-EU`) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). 0.b MP-B's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (`MP-B-EU`) and obtains its own JWT. *Step 1 – XvP Initialisation (MP-S as Seller) – METHOD: POST – Code: 200:* 1. MP-S's application sends an HTTP `POST` request to the cash-token XvP initialisation endpoint: `POST /igw/{ncb}/v1/xvps` Path parameter: `{ncb}` = country code / NCB identifier of MP-S's responsible NCB. Headers: `Authorization: Bearer <JWT of MP-S-EU>`, `Content-Type: application/json`. Request body (schema `XvPInitRequest`): `json` { "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLT-Operator-ID>", "cashWalletAlias": "<MP-S DCW alias to credit>" }, "buyer": { "bic": "<MP-B BIC11>" }, "amount": "10000.50", "currency": "EUR", "type": "DVP" } Note: `cashWalletAlias` is optional and only used when the cash leg must be credited to a non-default DCW of MP-S (SDD §3.5.6.a); for the default DCW it can be omitted. 2. The Pilot solution returns HTTP `200 OK` with payload (schema `XvPInitResponse`) containing: • `xvpTransactionId` (UUID) – to be reused in all subsequent calls;

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• executionHash (SHA-256, 64 hex chars); • cancellationHash (SHA-256, 64 hex chars); • timeout (UTC ISO-8601, e.g. 2026-05-12T15:30:00Z); • buyer, seller, amount, currency, type (echo of the request, normalised). 3. MP-S records xvpTransactionId, executionHash, cancellationHash and timeout. The two hashes are the values MP-S will use to set up the Hash Link Contract on the Market DLT platform (out of scope of this test case). *Step 2 – XvP Initialisation Query (MP-B as Buyer) – METHOD: GET – Code: 200:* 4. MP-B becomes aware of the new XvP via the agreed off-Pilot channel (e.g. notification on the Market DLT platform once the HLC is created – out of scope). MP-B knows the xvpTransactionId from this off-Pilot exchange. 5. MP-B's application sends an HTTP GET request to the cash-token XvP initialisation query endpoint: GET /igw/{ncb}/v1/xvps/{xvpTransactionId} Path parameters: {ncb} = NCB of MP-B; {xvpTransactionId} = the UUID from Step 1. Headers: Authorization: Bearer <JWT of MP-B-EU>. 6. The Pilot solution returns HTTP 200 OK with the same XvPInitResponse payload as in Step 1. 7. MP-B verifies that the data returned matches the terms previously agreed with MP-S (BICs of Seller and Buyer, amount, currency, type, executionHash, cancellationHash, timeout). *Step 3 – Payment (MP-B as Buyer, cash leg settlement) – METHOD: POST – Code: 200:* 8. MP-B's application sends an HTTP POST request to the cash-token Payment endpoint: POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment Path parameters: {ncb} = NCB of MP-B; {xvpTransactionId} from Step 1. Headers: Authorization: Bearer <JWT of MP-B-EU>, Content-Type: application/json. Request body (schema PaymentRequest): json { "buyer": { "bic": "<MP-B BIC11>", "marketDLTOperator": "<MDLT-Operator-ID>", "cashWalletAlias": "<MP-B DCW alias to debit>" }, "seller": { "bic": "<MP-S BIC11>" }, "amount": "10000.50", "currency": "EUR" } The amount and currency must match exactly the values returned in Step 1/2 (Pilot validates consistency with the XvP cash leg). 9. The Pilot solution synchronously executes the cash leg on ESY-DLT: it debits MP-B's DCW and credits MP-S's DCW (or

EUROPEAN CENTRAL BANK
EUROSYSTEM

	the DCW indicated by cashWalletAlias in Step 1) by the agreed amount, and returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id (UUID of the cash-token payment); • payment.status = "SETTLED"; • payment.reason (status reason text); • executionKey (64–128 hex chars) – **returned only to the Buyer and only when the Payment is SETTLED** (SDD §3.5.6.c). 10. MP-B records the returned executionKey. This is the secret MP-B will use (if needed) to force the unlock of the asset on the Market DLT platform (out of scope of this test case). *Step 4 – Get Key & Payment Info (MP-S as Seller, cooperative-execution check) – METHOD: GET – Code: 200:* 11. MP-S's application sends an HTTP GET request to the cash-token Get Key & Payment Info endpoint, requesting the CANCELLATION key (the only key value allowed for the Seller, per SDD §3.5.6.d and OpenAPI KeyType): GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION Path parameters: {ncb} = NCB of MP-S; {xvpTransactionId} from Step 1. Headers: Authorization: Bearer <JWT of MP-S-EU>. 12. The Pilot solution returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id, payment.status = "SETTLED", payment.reason; • executionKey = null (not returned to the Seller); • cancellationKey = null (returned only if status is UNSETTLED or BURNED, per SDD §3.5.6.d). 13. MP-S confirms from the response that the cash leg is SETTLED and consequently is entitled to release the asset to MP-B on the Market DLT platform (cooperative execution – out of scope of this test case). *Step 5 – (Optional) Buyer re-retrieves execution key – METHOD: GET – Code: 200:* 14. As an additional resilience check, MP-B's application may re-call the Get Key & Payment Info endpoint with key=EXECUTION: GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=EXECUTION 15. The Pilot solution returns HTTP 200 OK with the same executionKey as in Step 3 (the Pilot solution returns the entitled key without retention-period constraints, per SDD §3.5.6 *Availability of Execution and Cancellation Key*).

EUROPEAN CENTRAL BANK
EUROSYSTEM

Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant). • Each MP holds at least one active DCW; **MP-B's DCW** has been previously funded with cash tokens sufficient to cover the XvP amount (e.g. via PP-E2E-001). • Neither MP-S nor MP-B is **blocked**; both relevant DCWs are *Unblocked* and within their *Valid From / Valid To* period. • Each MP has at least one configured **External User** application user (SDD §2.4.5, Table E) with a valid X.509 certificate installed and a working mTLS connection to the Pilot solution IAM (SDD §6.3). • If execution is delegated to a market DLT Operator, a **Market DLT Operator** is registered in the Pilot solution and **whitelisted** for MP-S and/or MP-B (SDD §3.4.8 *Whitelisting*); its ID is known by both MPs and is referenced as `marketDLTOperator` in the Initialisation and Payment request payloads. • Off-Pilot channels: MP-S and MP-B have agreed bilaterally (out of scope of the Pilot solution) on the XvP economic terms (asset, amount, currency, type) and on how MP-B will become aware of the `xvpTransactionId` after Step 1. • The test is executed within an open Business window. The XvP timeout (system parameter, per SDD §3.5.6.a) is sufficient to complete Steps 1 to 5 in sequence (i.e. Step 3 occurs strictly before `timeout`).
Expected results	**A2A** • **Step 1 (Initialisation):** HTTP `200 OK`. Response body contains a new `xvpTransactionId` (UUID v4), valid `executionHash` and `cancellationHash` (64 lower-case hex chars each), `timeout` in UTC ISO-8601, and the echo of `buyer`, `seller`, `amount`, `currency`, `type`. The XvP is registered in the Pilot solution as a new instance with an associated cash leg in `PENDING` status. • **Step 2 (Initialisation Query):** HTTP `200 OK`. The response payload is consistent with the one returned in Step 1 (same `xvpTransactionId`, same hashes, same `timeout`, same economic terms). • **Step 3 (Payment):** HTTP `200 OK`. The Payment is settled synchronously on ESY-DLT: – MP-B's DCW is debited by `amount`; – MP-S's DCW (or the alias-designated DCW) is credited by `amount`; – the response contains `payment.status = "SETTLED"` and a non-null `executionKey` (64–128 lower-case hex chars). • **Step 4 (Get Key – Seller):** HTTP `200 OK`. Response contains `payment.status = "SETTLED"`, `executionKey = null`, `cancellationKey = null`. The Seller is correctly **not** entitled to receive any key when the cash leg is `SETTLED` (per SDD §3.5.6.d). • **Step 5 (Get Key – Buyer, optional):** HTTP `200 OK`. Response contains `payment.status = "SETTLED"` and a non-null

EUROPEAN CENTRAL BANK
EUROSYSTEM

	executionKey equal to the one returned in Step 3. • The XvP transaction is visible as a Payment-XvP of type *Payment* with status *Settled* in the **Transactions** tab of both MP-S and MP-B (U2A side-verification, optional). • No HTTP error codes (400, 401, 403, 404, 409, 500, 503) are returned on any of the calls. • Authorisation enforcement is correctly applied: only the Buyer receives the executionKey; only the Seller can request the CANCELLATION key; neither MP can read the XvP outside its own NCB scope.
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for both MP-S-EU and MP-B-EU (with sensitive tokens redacted as appropriate). • Copy of the **HTTP request and response** (headers + JSON body) for Step 1 – POST /igw/{ncb}/v1/xvps – evidencing the returned xvpTransactionId, executionHash, cancellationHash, timeout. • Copy of the HTTP request and response for Step 2 – GET /igw/{ncb}/v1/xvps/{xvpTransactionId} – evidencing the consistency of the returned data with Step 1. • Copy of the HTTP request and response for Step 3 – POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment – evidencing payment.status = "SETTLED" and the executionKey returned to the Buyer. • Copy of the HTTP request and response for Step 4 – GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION – evidencing payment.status = "SETTLED" and the absence of any returned key for the Seller. • (Optional) Copy of the HTTP request and response for Step 5 – GET …?key=EXECUTION – evidencing the same executionKey. • (Optional, U2A cross-check) Screenshot of the **Transactions** tab on MP-S and MP-B sides showing the corresponding Payment-XvP transaction in *Settled* status, and screenshots of the *Dedicated Cash Wallets* tab showing the updated balances. • Excel export of the cash-token Transactions extraction filtered on the xvpTransactionId, evidencing the full transaction record.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.2.2 *Communication channels & User profiles* (Table E); §3.5.6 *DvP & PvP with Cash Token settlement* – sub-sections §3.5.6.a *Initialisation*, §3.5.6.b *Initialisation Query*, §3.5.6.c *Payment*, §3.5.6.d *Get Key and Payment Info*, §3.5.6.e *Initialisation on Market DLT platform*; §3.7 *A2A 1-step/2-steps illustration*; §3.4.8 *Whitelisting* (Market DLT Operator – MP association); §6.3 *Preparing your integration – Connectivity Guide*. • PONTES Pilot – OpenAPI Document v1.0: tag *XvP – Cash*

	Token – operations createXvP (POST /igw/{ncb}/v1/xvps), getXvP (GET /igw/{ncb}/v1/xvps/{xvpTransactionId}), payment (POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment), getPaymentStatus (GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key={key}); schemas XvPInitRequest, XvPInitResponse, PaymentRequest, PaymentResponse, Participant, SimpleParticipant, KeyType, PaymentStatus, TransactionType. • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions* (U2A cross-check view); §4.11 *Monitoring – Transactions*.
Related profiles/privileges	User profile (both MPs): **External User** If the execution is delegated to a Market DLT Operator, the Market DLT Operator user needs the **External User** profile

PP-MP-05 – Complete an XvP transaction with direct RTGS settlement and retrieve execution key (A2A)

Field	Description
Test Case ID	PP-MP-05
Test case name	As two Market Participants, complete an XvP transaction with direct RTGS settlement and retrieve execution key
Relevant for (actor)	Market Participants – two distinct MPs interacting via REST APIs: • **MP-S** acting as **Seller** (T2 RTGS DCA holder of the credited DCA, responsible for the asset leg on the Market DLT platform). • **MP-B** acting as **Buyer** (T2 RTGS DCA holder of the debited DCA, responsible for paying the cash leg). Both MPs self-instruct via A2A using their **External User** application accounts, or they can delegate the execution of this test case to a Market DLT Operator acting on their behalf.
Input mode	A2A
Mandatory	Yes (but the Market Participant can delegate the execution to a Market DLT Operator)
Detailed Description	This test case describes the end-to-end happy-path of an XvP (DvP or PvP) transaction with **Direct Settlement in T2 RTGS**, instructed entirely via the Pilot solution REST APIs. The flow

EUROPEAN CENTRAL BANK
EUROSYSTEM

	exercises the four XvP REST functionalities exposed by the Pilot solution for the direct-RTGS settlement model (SDD §3.5.6 – *Direct Settlement* sub-sections): (i) XvP Initialisation by the Seller, (ii) XvP Initialisation Query by the Buyer, (iii) Payment by the Buyer (with NRO signature, triggering the T2 RTGS DCA leg via pacs.010 / pacs.009 / pacs.002), with execution key returned in the response, and (iv) Get Key & Payment Info by the Seller, confirming the *SETTLED* status of the cash leg. The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking of the asset leg, cooperative or forced execution of the HLC) is **outside the scope of the Pilot solution** (SDD §3.5.6.e) and therefore outside the scope of this test case; the test verifies only that the Pilot solution returns the data needed by the Seller to initialise the HLC (execution hash, cancellation hash, timeout) and that the Buyer correctly receives the **Execution Key** upon successful T2 RTGS settlement. Compared to the cash-token settlement model (PP-MP-03), the direct-RTGS settlement model differs in three key aspects: – **Settlement venue**: the cash leg is settled on the Seller's and Buyer's **T2 RTGS DCAs** (not on DCWs on ESY-DLT). The Pilot solution orchestrates the debit/credit via T2 RTGS using ISO 20022 messages (pacs.010 to debit the Buyer's DCA, pacs.009 to credit the Seller's DCA, with pacs.002 status reports). – **NRO**: the Payment request requires a digital signature (`signature` + `signerPEM` fields in `RTGSPaymentRequest`) for Non-Repudiation of Origin (SDD §3.3 and §3.5.6 *Direct Settlement* fig. i, step 1). –If execution is delegated to a market DLT Operator, a Market DLT Operator is registered in the Pilot solution and whitelisted for MP-S and/or MP-B (SDD §3.4.8 Whitelisting); its ID is known by both MPs and is referenced as marketDLTOperator in the Initialisation and Payment request payloads. All XvP functionalities are exposed only in **A2A** mode (SDD §3.5.6) and follow the **one-step validation workflow** (no four-eyes principle on XvP – see OpenAPI v1.0 endpoint descriptions for the *XvP – Direct RTGS* tag). Both MPs use the **External User** profile. The test case may be run as a **DvP** (`type = "DVP"`) or a **PvP** (`type = "PVP"`) – the functional flow is identical; the default execution is `DVP`. **A2A** *Step 0 – Authentication (both MPs) – METHOD: mTLS*

EUROPEAN CENTRAL BANK
EUROSYSTEM

	handshake + IAM token issuance – Code: 200: 0.a MP-S's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (MP-S-EU) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). 0.b MP-B's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (MP-B-EU) and obtains its own JWT. *Step 1 – XvP Initialisation (MP-S as Seller) – METHOD: POST – Code: 200:* 1. MP-S's application sends an HTTP POST request to the direct-RTGS XvP initialisation endpoint: POST /igw/{ncb}/v1/direct-rtgs/xvps Path parameter: {ncb} = country code / NCB identifier of MP-S's responsible NCB. Headers: Authorization: Bearer <JWT of MP-S-EU>, Content-Type: application/json. Request body (schema RTGSXvPInitRequest): json { "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLT-Operator-ID>" }, "buyer": { "bic": "<MP-B BIC11>" }, "amount": "10000.50", "currency": "EUR", "type": "DVP" } Note: the RTGSParticipant schema (used for the Seller) does **not** support a cashWalletAlias – the cash leg is settled on the Seller's **T2 RTGS DCA** (the one declared via the *T2 account reference* configured in the LRDM per PP-NCB-01), not on a DCW. 2. The Pilot solution returns HTTP 200 OK with payload (schema RTGSXvPInitResponse) containing: • xvpTransactionId (UUID) – to be reused in all subsequent calls; • executionHash (SHA-256, 64 hex chars); • cancellationHash (SHA-256, 64 hex chars); • timeout (UTC ISO-8601, e.g. 2026-05-12T15:30:00Z); • buyer, seller, amount, currency, type (echo of the request, normalised). 3. MP-S records xvpTransactionId, executionHash, cancellationHash and timeout. The two hashes are the values MP-S will use to set up the Hash Link Contract on the Market DLT platform (out of scope of this test case). *Step 2 – XvP Initialisation Query (MP-B as Buyer) – METHOD: GET – Code: 200:* 4. MP-B becomes aware of the new XvP via the agreed off-Pilot channel (e.g. notification on the Market DLT platform once the HLC is created – out of scope). MP-B knows the xvpTransactionId from this off-Pilot exchange.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	5. MP-B's application sends an HTTP `GET` request to the direct-RTGS XvP initialisation query endpoint: `GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}` Path parameters: `{ncb}` = NCB of MP-B; `{xvpTransactionId}` = the UUID from Step 1. Headers: `Authorization: Bearer <JWT of MP-B-EU>`. 6. The Pilot solution returns HTTP `200 OK` with the same `RTGSXvPInitResponse` payload as in Step 1. 7. MP-B verifies that the data returned matches the terms previously agreed with MP-S (BICs of Seller and Buyer, `amount`, `currency`, `type`, `executionHash`, `cancellationHash`, `timeout`). *Step 3 – Payment (MP-B as Buyer, T2 RTGS cash leg settlement) – METHOD: POST – Code: 200:* 8. MP-B's application prepares the **NRO digital signature** of the payment request payload using MP-B's signing certificate (SDD §3.3 *Non-Repudiation of Origin*). 9. MP-B's application sends an HTTP `POST` request to the direct-RTGS Payment endpoint: `POST /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment` Path parameters: `{ncb}` = NCB of MP-B; `{xvpTransactionId}` from Step 1. Headers: `Authorization: Bearer <JWT of MP-B-EU>`, `Content-Type: application/json`. Request body (schema `RTGSPaymentRequest`): `json` `{` `"buyer": { "bic": "<MP-B BIC11>", "marketDLTOperator": "<MDLT-Operator-ID>" },` `"seller": { "bic": "<MP-S BIC11>" },` `"amount": "10000.50",` `"currency": "EUR",` `"signature": ECDSA_sign(privateKey, SHA256("xvp" + stripDashes(xvpTransactionId) + amount + buyer.bic + seller.bic))` `"signerPEM": "<PEM-encoded X.509 certificate of the MP-B signer>"` `}` The `amount` and `currency` must match exactly the values returned in Step 1/2 (Pilot validates consistency with the XvP cash leg per SDD §3.5.6 *Direct Settlement*, step 2). 10. The Pilot solution performs authorisation, consistency and signature-validity checks, then forwards the Payment instruction to the Eurosystem DLT node of MP-B's NCB, which creates a Payment Instruction with status `PENDING` and submits it to the BBK DLT node. The BBK DLT node's Trigger Backend component then orchestrates the T2 RTGS settlement leg via ISO 20022 messages (per SDD §3.5.6 *Direct Settlement*, steps 4–7): – `pacs.010` to debit MP-B's T2 RTGS DCA; – `pacs.002` reporting the debit outcome; – on success, `pacs.009` to credit MP-S's T2 RTGS DCA; – `pacs.002` reporting the credit outcome.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	11. Upon receipt of the T2 RTGS settlement outcome, the BBK DLT node propagates the final payment status to MP-B's and MP-S's NCB nodes, and the Pilot solution returns HTTP 200 OK to MP-B with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id (UUID of the direct-RTGS payment); • payment.status = "SETTLED"; • payment.reason (status reason text); • executionKey (64–128 hex chars) – **returned only to the Buyer and only when the Payment is SETTLED** (SDD §3.5.6 *Direct Settlement*, step 10). 12. MP-B records the returned executionKey. This is the secret MP-B will use (if needed) to force the unlock of the asset on the Market DLT platform (out of scope of this test case). *Step 4 – Get Key & Payment Info (MP-S as Seller, cooperative-execution check) – METHOD: GET – Code: 200:* 13. MP-S's application sends an HTTP GET request to the direct-RTGS Get Key & Payment Info endpoint, requesting the CANCELLATION key (the only key value allowed for the Seller, per SDD §3.5.6.d and OpenAPI KeyType): GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION Path parameters: {ncb} = NCB of MP-S; {xvpTransactionId} from Step 1. Headers: Authorization: Bearer <JWT of MP-S-EU>. 14. The Pilot solution returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id, payment.status = "SETTLED", payment.reason; • executionKey = null (not returned to the Seller); • cancellationKey = null (returned only if status is UNSETTLED or BURNED, per SDD §3.5.6.d). 15. MP-S confirms from the response that the cash leg is SETTLED on T2 RTGS (i.e. MP-S's T2 RTGS DCA has been credited by amount) and consequently is entitled to release the asset to MP-B on the Market DLT platform (cooperative execution – out of scope of this test case). *Step 5 – (Optional) Buyer re-retrieves execution key – METHOD: GET – Code: 200:* 16. As an additional resilience check, MP-B's application may re-call the Get Key & Payment Info endpoint with key=EXECUTION: GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=EXECUTION 17. The Pilot solution returns HTTP 200 OK with the same executionKey as in Step 3 (the Pilot solution returns the entitled key without retention-period constraints, per SDD §3.5.6 *Availability of Execution and Cancellation Key*).

EUROPEAN CENTRAL BANK
EUROSYSTEM

Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant). • Each MP holds an active **T2 RTGS DCA** in T2 RTGS, with a valid *T2 account reference* declared in the Pilot solution by the responsible NCB (per PP-NCB-01). The MP-B's T2 RTGS DCA has sufficient available liquidity to cover the XvP amount. • A **direct debit mandate** is in place in T2 RTGS allowing the Pilot solution (via the BBK DLT node / Trigger Backend) to debit MP-B's T2 RTGS DCA via pacs.010 (per SDD §3.5.6 *Direct Settlement*, steps 4–7, and *Unhappy Paths* TP-DR-DD-001). • Neither MP-S nor MP-B is **blocked** in the Pilot solution; neither MP-S's nor MP-B's T2 RTGS DCA is blocked in T2 RTGS (per SDD *Unhappy Paths* TP-DCA-002 / TP-DR-PART-002). • Each MP has at least one configured **External User** application user (SDD §2.4.5, Table E) with a valid X.509 certificate installed and a working mTLS connection to the Pilot solution IAM (SDD §6.3). • Both External Users are entitled to the XvP functionalities (XvP – Direct RTGS API tag, *Authorised Profiles: EXTERNAL_USER*, per OpenAPI v1.0). • MP-B's application disposes of a **signing certificate and corresponding private key** to produce the NRO signature required by RTGSPaymentRequest (signature + signerPEM, SDD §3.3). The signer certificate is recognised by the Pilot solution. • If execution is delegated to a market DLT Operator, a **Market DLT Operator** is registered in the Pilot solution and **whitelisted** for MP-S and/or MP-B (SDD §3.4.8 *Whitelisting*; OpenAPI Field Rules); its ID is known by both MPs and is referenced as marketDLTOperator in the Initialisation and Payment request payloads. • Off-Pilot channels: MP-S and MP-B have agreed bilaterally (out of scope of the Pilot solution) on the XvP economic terms (asset, amount, currency, type) and on how MP-B will become aware of the xvpTransactionId after Step 1. • The test is executed within the **T2 RTGS settlement window** and within an open Business window. The XvP timeout (system parameter, per SDD §3.5.6.a) is sufficient to complete Steps 1 to 5 in sequence (i.e. Step 3 occurs strictly before timeout). • The relevant T2 RTGS message subscriptions are in place if MP-B / MP-S wish to receive the corresponding pacs.002 status reports for evidence purposes (optional).
Expected results	**A2A** • **Step 1 (Initialisation):** HTTP 200 OK. Response body contains a new xvpTransactionId (UUID v4), valid executionHash and cancellationHash (64 lower-case hex chars each), timeout in UTC ISO-8601, and the echo of buyer, seller, amount, currency,

EUROPEAN CENTRAL BANK
EUROSYSTEM

	type. The XvP is registered in the Pilot solution as a new instance with an associated direct-RTGS payment in Initialized status (per SDD §3.6.2 *Direct RTGS Payments* status list). • **Step 2 (Initialisation Query):** HTTP 200 OK. The response payload is consistent with the one returned in Step 1 (same xvpTransactionId, same hashes, same timeout, same economic terms). • **Step 3 (Payment):** HTTP 200 OK. The cash leg is settled in T2 RTGS: – MP-B's T2 RTGS DCA is debited by amount (pacs.010 + positive pacs.002); – MP-S's T2 RTGS DCA is credited by amount (pacs.009 + positive pacs.002); – the Pilot solution direct-RTGS payment status reaches Completed (per SDD §3.6.2 *Direct RTGS Payments* status list); – the response contains payment.status = "SETTLED" and a non-null executionKey (64–128 lower-case hex chars). • **Step 4 (Get Key – Seller):** HTTP 200 OK. Response contains payment.status = "SETTLED", executionKey = null, cancellationKey = null. The Seller is correctly **not** entitled to receive any key when the cash leg is SETTLED (per SDD §3.5.6.d). • **Step 5 (Get Key – Buyer, optional):** HTTP 200 OK. Response contains payment.status = "SETTLED" and a non-null executionKey equal to the one returned in Step 3. • The XvP transaction is recorded in the Pilot solution's Direct RTGS Payments extraction (SDD §3.6.2) with type *Direct RTGS Payment-XvP* and status *Completed*, with the populated Correlation ID linking to the corresponding T2 RTGS payment messages. • No HTTP error codes (400, 401, 403, 404, 409, 500, 503) are returned on any of the calls. • Authorisation enforcement is correctly applied: only the Buyer receives the executionKey; only the Seller can request the CANCELLATION key; neither MP can read the XvP outside its own NCB scope. • The NRO signature provided in Step 3 is successfully validated by the Pilot solution against the signerPEM certificate.
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for both MP-S-EU and MP-B-EU (with sensitive tokens redacted as appropriate). • Copy of the **HTTP request and response** (headers + JSON body) for Step 1 – POST /igw/{ncb}/v1/direct-rtgs/xvps – evidencing the returned xvpTransactionId, executionHash, cancellationHash, timeout. • Copy of the HTTP request and response for Step 2 – GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId} – evidencing the consistency of the returned data with Step 1. • Copy of the HTTP request and response for Step 3 – POST /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment –

EUROPEAN CENTRAL BANK
EUROSYSTEM

	evidencing the submitted `signature` and `signerPEM`, `payment.status = "SETTLED"`, and the `executionKey` returned to the Buyer. • Copy of the HTTP request and response for Step 4 – `GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION` – evidencing `payment.status = "SETTLED"` and the absence of any returned key for the Seller. • (Optional) Copy of the HTTP request and response for Step 5 – `GET …?key=EXECUTION` – evidencing the same `executionKey`. • (Optional) Copies of the T2 RTGS ISO 20022 messages exchanged during settlement: `pacs.010` (Buyer DCA debit), `pacs.002` (debit outcome), `pacs.009` (Seller DCA credit), `pacs.002` (credit outcome) – obtained via the respective message subscriptions if configured. • Excel export of the Direct T2 RTGS Payments extraction (POST `/dlt/{ncb}/api/octopus/tms/direct-rtgs/payments/extract`, per SDD §3.6.2) evidencing the record with type Direct RTGS Payment-XvP and status Completed. Note: this endpoint is a POST with a request body, which only supports filtering on source and status. The record is located via the Payment ID or the Correlation ID. • (Optional, U2A cross-check) Screenshot of the **Direct RTGS Payments** monitoring tab showing the corresponding `Payment-XvP` transaction in *Completed* status.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.2.2 *Communication channels & User profiles* (Table E); §3.3 *Non-Repudiation of Origin (NRO)*; §3.5.6 *DvP & PvP* – sub-sections §3.5.6.a *Initialisation*, §3.5.6.b *Initialisation Query*, §3.5.6.c *Payment* (including *Direct Settlement* fig. i – pacs.010 / pacs.002 / pacs.009 flow), §3.5.6.d *Get Key and Payment Info*, §3.5.6.e *Initialisation on Market DLT platform*; §3.5.7 *Unhappy Paths & Error codes* (TP-DCA-001/002, TP-DR-DD-001, TP-DR-PART-002, etc.); §3.6.2 *Reporting and data extraction – Direct RTGS Payments* (extraction endpoint, status list `Initialized / Accepted / Payment submitted / Completed / Failed / Cancelled`); §3.7 *A2A 1-step/2-steps illustration*; §3.4.8 *Whitelisting* (Market DLT Operator – MP association); §6.3 *Preparing your integration – Connectivity Guide*. • PONTES Pilot – OpenAPI Document v1.0: tag *XvP – Direct RTGS* – operations `RTGSCreateXvP` (POST `/igw/{ncb}/v1/direct-rtgs/xvps`), `RTGSGetXvP` (GET `/igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}`), `RTGSPayment` (POST `/igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment`), `RTGSGetPaymentStatus` (GET `/igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key={key}`); schemas `RTGSXvPInitRequest`, `RTGSXvPInitResponse`, `RTGSPaymentRequest`, `PaymentResponse`, `RTGSParticipant`, `SimpleParticipant`, `KeyType`, `PaymentStatus`, `TransactionType`.

	• T2 RTGS UDFS – §12.4 ISO 20022 messages (pacs.010 FinancialInstitutionDirectDebit, pacs.009 FinancialInstitutionCreditTransfer, pacs.002 PaymentStatusReport). • PONTES Pilot – User Handbook v1.0: §3.2.7 *Direct RTGS Payments*; §4.6 *Direct RTGS* (entry point); §4.6.1 *Direct RTGS Payments* (U2A view, for the optional cross-check); §4.11 *Monitoring – Direct RTGS Payments*.
Related profiles/privileges	User profile (both MPs): **External User** If the execution is delegated to a Market DLT Operator, the Market DLT Operator user needs the **External User** profile

PP-MP-06 – Fail to complete an XvP transaction within timeout and retrieve cancellation key (A2A, any settlement method)

Field	Description
Test Case ID	PP-MP-06
Test case name	As two Market Participants, fail to complete an XvP transaction within timeout and retrieve cancellation key
Relevant for (actor)	Market Participants – two distinct MPs interacting via REST APIs: • **MP-S** acting as **Seller** (DCW Owner of the would-have-been-credited DCW, responsible for reclaiming the locked asset on the Market DLT platform). • **MP-B** acting as **Buyer** (DCW Owner of the would-have-been-debited DCW; deliberately does **not** submit the Payment within the XvP timeout window for this test). Both MPs self-instruct via A2A using their **External User** application accounts, or they can delegate the execution of this test case to a Market DLT Operator acting on their behalf.
Input mode	A2A
Mandatory	Yes (but the Market Participant can delegate the execution to a Market DLT Operator)
Detailed Description	**The test case is settlement-model-agnostic** and may be executed against **either** of the two XvP settlement models (cash-token OR direct-RTGS) using the corresponding endpoint family. The functional flow is identical, and the choice

EUROPEAN CENTRAL BANK
EUROSYSTEM

	of settlement model is made at Initialisation (Step 2) – from that point on, all subsequent calls must use the matching endpoint family This test case describes the **unhappy-path** of an XvP (DvP or PvP) transaction in which the Buyer does **not** submit the Payment for the cash leg before the XvP timeout expires. In this scenario, the Pilot solution: (i) does not create any payment in the ESY-DLT for the duration of the timeout window, then (ii) once the timeout is reached, automatically creates a payment with status BURNED for the XvP cash leg (per SDD §3.5.6 – State Diagram of Payment status). At that point, the cancellation key becomes retrievable by the Seller via the *Get Key and Payment Info* functionality, allowing the Seller to reclaim the asset locked in the Hash Link Contract on the Market DLT platform (out of scope of this test case). The test case shares the same Initialisation and Initialisation-Query steps as successful XvP cases (PP-MP-03 or PP-MP-04, depending on the payment method chosen), but diverges by deliberately **omitting** the Buyer's Payment call. It exercises the Pilot solution's two key oracle behaviours: – the **timeout-driven transition** from "no payment exists" to a BURNED payment (SDD §3.5.6, fig. i *State Diagram of Payment status* and fig. ii *Relationship between Timeout and Payment*); – the **selective key disclosure**: the cancellation key is released **only** to the Seller, **only** when the payment is in status UNSETTLED or BURNED (SDD §3.5.6 fig. iii *Relationship between payment status and keys disclosure*; OpenAPI v1.0 PaymentResponse.cancellationKey description). The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking of the asset leg, forced cancellation of the HLC using the cancellation key) is **outside the scope of the Pilot solution** (SDD §3.5.6.e) and therefore outside the scope of this test case; the test verifies only that the Pilot solution correctly transitions the cash leg to BURNED after timeout and correctly releases the **Cancellation Key** to the Seller. The test case also verifies the **Buyer-side rejection** behaviour: a late Payment attempt by the Buyer (after timeout) must be rejected by the Pilot solution (this is verified in Step 3-bis below). The test case may be run as a **DvP** (type = "DVP") or a **PvP** (type = "PVP"); the functional flow is identical; the default execution is DVP. For conciseness, the detailed steps are described with references to the endpoints for a settlement in cash tokens –

EUROPEAN CENTRAL BANK
EUROSYSTEM

market participants using the direct RTGS settlement method instead should adapt the end-points accordingly.
A2A

Step 0 – Authentication (both MPs) – METHOD: mTLS handshake + IAM token issuance – Code: 200:
0.a MP-S's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (MP-S-EU) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3).
0.b MP-B's application authenticates to the Pilot solution IAM using the X.509 certificate of its External User (MP-B-EU) and obtains its own JWT.

Step 1 – XvP Initialisation (MP-S as Seller) – METHOD: POST – Code: 200:
1. MP-S's application sends an HTTP POST request to the cash-token XvP initialisation endpoint (identical to PP-MP-03 Step 1):
POST /igw/{ncb}/v1/xvps
Path parameter: {ncb} = country code / NCB identifier of MP-S's responsible NCB.
Headers: Authorization: Bearer <JWT of MP-S-EU>, Content-Type: application/json.
Request body (schema XvPInitRequest):
json
```
{
  "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLT-Operator-ID>", "cashWalletAlias": "<MP-S DCW alias to credit>" },
  "buyer": { "bic": "<MP-B BIC11>" },
  "amount": "10000.50",
  "currency": "EUR",
  "type": "DVP"
}
```
2. The Pilot solution returns HTTP 200 OK with payload (schema XvPInitResponse) containing:
• xvpTransactionId (UUID) – to be reused in all subsequent calls;
• executionHash, cancellationHash (SHA-256, 64 hex chars each);
• timeout (UTC ISO-8601) – **the critical value for this test case**; MP-S records it precisely.
• buyer, seller, amount, currency, type (echo).
3. MP-S records xvpTransactionId, cancellationHash and timeout. The cancellationHash is the value MP-S will use to set up the Hash Link Contract on the Market DLT platform (out of scope).

Step 2 – XvP Initialisation Query (MP-B as Buyer) – METHOD: GET – Code: 200:
4. MP-B becomes aware of the new XvP via the agreed off-Pilot channel (as in PP-MP-03 Step 2). MP-B knows the xvpTransactionId from this off-Pilot exchange.
5. MP-B's application sends an HTTP GET request to the cash-

EUROPEAN CENTRAL BANK
EUROSYSTEM

	token XvP initialisation query endpoint: `GET /igw/{ncb}/v1/xvps/{xvpTransactionId}` Headers: `Authorization: Bearer <JWT of MP-B-EU>`. 6. The Pilot solution returns HTTP `200 OK` with the same `XvPInitResponse` payload as in Step 1. 7. MP-B verifies the data and records the `timeout` value. *Step 3 – Deliberate inaction by the Buyer until timeout – no API call:* 8. **The Buyer deliberately does not submit the Payment**. The test waits until the wall-clock time strictly exceeds the `timeout` value returned in Step 1. 9. During the waiting period, MP-S may optionally poll the *Get Key and Payment Info* endpoint to confirm that no payment exists yet (see Step 3-ter below for the expected pre-timeout response). *Step 3-bis – (Optional verification) Late Payment attempt by the Buyer – METHOD: POST – Code: 4xx (rejection expected):* 10. **Optional**: after the `timeout` has passed, MP-B's application sends a late Payment request: `POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment` Same request body as in PP-MP-03 Step 3. 11. The Pilot solution **rejects** the late Payment request (per SDD §3.5.6.c: "*Then EII checks if it is possible to pay for the DvP, namely if a payment already exists for the cash leg of the DvP or if the timeout for submitting the payment request for the DvP has been already reached. If any of this check fails, the request is rejected. In such case, the Actor is not permitted to retry the payment of the DvP*"). The expected HTTP response is `4xx` (`409 Conflict` or `400 Bad Request`). The error response body (schema `ErrorResponse`) carries a business error code HL-DVP-007 identifying the timeout-expired condition. *Step 3-ter – (Optional verification) Pre-final-status query by the Seller – METHOD: GET – Code: 200 with no key:* 12. **Optional**: between Step 2 and the expiry of `timeout`, MP-S may call the Get Key & Payment Info endpoint to confirm that the cancellation key is not yet released: `GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION` 13. The Pilot solution returns HTTP `200 OK` (or `404 Not Found`, depending on whether a payment object exists). If `200 OK`, the response contains `payment.status` indicating a non-final state and both `executionKey` and `cancellationKey` set to `null`. This confirms the oracle behaviour: while it is still possible to pay for the XvP, no key is released (per SDD §3.5.6 fig. iii). *Step 4 – Get Key & Payment Info – Seller retrieves cancellation key – METHOD: GET – Code: 200:* 14. **After the XvP `timeout` has expired** (and once the Pilot

EUROPEAN CENTRAL BANK
EUROSYSTEM

	solution has had time to perform the timeout-driven creation of the BURNED payment), MP-S's application sends an HTTP GET request to the cash-token Get Key & Payment Info endpoint, requesting the CANCELLATION key: GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION Path parameters: {ncb} = NCB of MP-S; {xvpTransactionId} from Step 1. Headers: Authorization: Bearer <JWT of MP-S-EU>. 15. The Pilot solution returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id (UUID of the cash-token payment automatically created by the Pilot solution on timeout); • payment.status = "BURNED"; • payment.reason (status reason text identifying timeout expiry); • executionKey = null (not released, since payment is not SETTLED); • cancellationKey (64–128 hex chars) – **returned only to the Seller, only when payment is UNSETTLED or BURNED** (SDD §3.5.6.d and OpenAPI PaymentResponse.cancellationKey description). 16. MP-S records the returned cancellationKey. This is the secret MP-S will use to reclaim the asset locked in the HLC on the Market DLT platform (out of scope of this test case). *Step 5 – (Optional) Buyer attempts to retrieve execution key – METHOD: GET – Code: 200 with executionKey = null:* 17. As a negative cross-check, MP-B's application may re-call the Get Key & Payment Info endpoint with key=EXECUTION: GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=EXECUTION 18. The Pilot solution returns HTTP 200 OK with payment.status = "BURNED" and executionKey = null (the Buyer is correctly **not** entitled to the execution key when the cash leg is BURNED, since no settlement occurred).
Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant). • Each MP holds at least one active DCW. **MP-B's DCW funding is not strictly required** for this test (since no payment will be submitted), although in a realistic test campaign MP-B's DCW will typically be funded; the test specifically verifies the *time-based* failure path, not the *insufficient-funds* failure path. • Neither MP-S nor MP-B is **blocked**; both relevant DCWs are *Unblocked* and within their *Valid From / Valid To* period. • Each MP has at least one configured **External User** application user (SDD §2.4.5, Table E) with a valid X.509

EUROPEAN CENTRAL BANK
EUROSYSTEM

	certificate installed and a working mTLS connection to the Pilot solution IAM (SDD §6.3). • Both External Users are entitled to the XvP functionalities (XvP – Cash Token API tag, *Authorised Profiles: EXTERNAL_USER*, per OpenAPI v1.0). • If execution is delegated to a market DLT Operator, a **Market DLT Operator** is registered in the Pilot solution and **whitelisted** for MP-S and/or MP-B (SDD §3.4.8 *Whitelisting*; OpenAPI Field Rules); its ID is known by both MPs and is referenced as marketDLTOperator in the Initialisation and Payment request payloads. • Off-Pilot channels: MP-S and MP-B have agreed bilaterally (out of scope of the Pilot solution) on the XvP economic terms and on how MP-B will become aware of the xvpTransactionId. • **The XvP timeout (system parameter, per SDD §3.5.6.a) is known and short enough to be exercisable within a single test session**. • The test is executed within an open Business window. The XvP timeout must fall **inside** the same Business window.
Expected results	**A2A** • **Step 1 (Initialisation):** HTTP 200 OK. Response body contains a new xvpTransactionId (UUID v4), valid executionHash and cancellationHash, a timeout in UTC ISO-8601 falling within the current Business window, and the echo of the economic terms. The XvP is registered in the Pilot solution; **no payment exists yet** in the ESY-DLT for this XvP. • **Step 2 (Initialisation Query):** HTTP 200 OK. The response payload is consistent with the one returned in Step 1. • **Step 3 (Deliberate inaction):** No API call. The wall-clock time crosses the timeout returned in Step 1. The Pilot solution automatically creates a payment for the XvP cash leg with status BURNED (per SDD §3.5.6 *State Diagram*: "*In case the DvP / PvP is timed out and no payment has been created yet; the Pilot solution creates a payment for the DvP / PvP and stores it in the ESY-DLT with status BURNED*"). • **Step 3-bis (Optional late Payment):** HTTP 4xx (rejection). The Pilot solution rejects the late Payment with a business-error code identifying the timeout-expired condition; **no payment is created from this rejected request** (the only payment is the auto-BURNED one). • **Step 3-ter (Optional pre-final query):** HTTP 200 OK (or 404 Not Found if no payment yet exists in the ESY-DLT). If 200 OK, the response contains a non-final payment.status (or no payment object at all, depending on implementation) and executionKey = null, cancellationKey = null. • **Step 4 (Get Key – Seller, after timeout):** HTTP 200 OK. Response contains payment.status = "BURNED", payment.reason identifying the timeout, executionKey = null, and a **non-null cancellationKey** (64–128 lower-case hex chars). The cancellation key correctly hashes to the cancellationHash

EUROPEAN CENTRAL BANK
EUROSYSTEM

	returned in Step 1 (the Seller may verify this off-Pilot using a SHA-256 implementation, although this is not a Pilot-side verification). • **Step 5 (Get Key – Buyer, optional negative cross-check):** HTTP 200 OK. Response contains payment.status = "BURNED" and executionKey = null. The Buyer is correctly **not** entitled to receive any key when the cash leg is BURNED. • **No DCW movement**: neither MP-B's DCW nor MP-S's DCW shows any debit or credit related to this XvP (no cash leg was ever settled). • **No transaction visible as Settled**: the XvP appears in the cash-token Transactions extraction (SDD §3.6.2.c) with type *Payment-XvP* and status reflecting the failure. • Authorisation enforcement is correctly applied: only the Seller receives the cancellationKey; only the Buyer can request the EXECUTION key (and gets null since payment is BURNED).
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for both MP-S-EU and MP-B-EU (with sensitive tokens redacted as appropriate). • Copy of the **HTTP request and response** for Step 1 – POST /igw/{ncb}/v1/xvps – evidencing the returned xvpTransactionId, cancellationHash, and **timeout value** (critical for this test). • Copy of the HTTP request and response for Step 2 – GET /igw/{ncb}/v1/xvps/{xvpTransactionId} – evidencing consistency with Step 1. • **Wall-clock evidence** demonstrating that no Payment was submitted before timeout: a timestamped test-execution log showing the timeout value, the wall-clock time of Step 4, and the absence of any POST …/payment call between Steps 2 and 4 (e.g. proxy / API-gateway log extract). • (Optional) Copy of the HTTP request and response for Step 3-bis – the late POST …/payment – evidencing the 4xx rejection and the business-error code. • (Optional) Copy of the HTTP request and response for Step 3-ter – the pre-timeout GET …?key=CANCELLATION – evidencing the absence of a released key while payment is still possible. • Copy of the HTTP request and response for Step 4 – GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION – evidencing payment.status = "BURNED" and the cancellationKey returned to the Seller. • (Optional) Copy of the HTTP request and response for Step 5 – GET …?key=EXECUTION – evidencing executionKey = null. • Excel export of the cash-token Transactions extraction filtered on the xvpTransactionId, evidencing the auto-BURNED payment record (and the absence of any SETTLED movement). • (Optional, U2A cross-check) Screenshot of the **Transactions** tab on MP-S and MP-B sides showing the XvP cash leg in its

EUROPEAN CENTRAL BANK
EUROSYSTEM

	failed final status, and screenshots of the *Dedicated Cash Wallets* tab showing **unchanged** balances on both MPs.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.3 *Non-Repudiation of Origin (NRO)* (not exercised here, but referenced for completeness); §3.5.6 *DvP & PvP* – including the *State Diagram of Payment status* (fig. i), *Relationship between Timeout and Payment* (fig. ii), and *Relationship between payment status and keys disclosure* (fig. iii); §3.5.6.a *Initialisation*; §3.5.6.b *Initialisation Query*; §3.5.6.c *Payment* (including the timeout-expired rejection rule); §3.5.6.d *Get Key and Payment Info* (cancellation-key release condition); §3.5.6.e *Initialisation on Market DLT platform* (HLC cancellation, out of scope); §3.5.6 *Availability of Execution and Cancellation Key* (post-final-status retrieval guarantee); §3.5.7 *Unhappy Paths & Error codes*; §3.6.2 *Reporting and data extraction*. • PONTES Pilot – OpenAPI Document v1.0: tag *XvP – Cash Token* – operations `createXvP`, `getXvP`, `payment` (rejection path), `getPaymentStatus`; schemas `PaymentResponse` (note: `cancellationKey` is "*returned only to Seller; returned only if Payment is in status UNSETTLED or BURNED*"); enum `PaymentStatus` (`SETTLED` / `UNSETTLED` / `PENDING` / `BURNED`); enum `KeyType` (`EXECUTION` / `CANCELLATION`). • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions* (U2A cross-check); §4.11 *Monitoring – Transactions*. • Related test cases: PP-MP-03 (XvP cash-token happy path – same Initialisation and Query steps)
Related profiles/privileges	User profile (both MPs): **External User** If the execution is delegated to a Market DLT Operator, the Market DLT Operator user needs the **External User** profile

PP-MP-07 – Query balances and transactions history for owned Dedicated Cash Wallets

Field	Description
Test Case ID	PP-MP-07
Test case name	Query balances and transactions history for owned Dedicated Cash Wallets

EUROPEAN CENTRAL BANK
EUROSYSTEM

Relevant for (actor)	Market Participant – acting as **Dedicated Cash Wallet Owner** of its own DCWs
Input mode	U2A
Mandatory	Yes
Detailed Description	This test case describes the steps required for a Market Participant (MP) to query and verify, in the Pilot solution GUI, the balances and transaction history of the Dedicated Cash Wallets (DCWs) it owns. The MP's visibility is strictly limited to **its own** owned DCWs (per SDD §2.2.3 *"Visibility on balances and transactions history related to owned wallets"*); no managed-scope or supervisor-scope visibility applies, since Market Participants do not hold the *DCW Manager*, *National Cash Token Supervisor* or *Cash Token Supervisor* roles. The test exercises three complementary views in U2A: (a) the **Dashboard** (aggregated activity counters for the MP's owned DCWs); (b) the **Dedicated Cash Wallets** list with per-wallet balances and details; and (c) the **Transactions** tab with the transaction history covering debit and credit legs impacting the MP's owned DCWs, including detailed drill-down and Excel export. **U2A** *Step 1 – Authentication:* 1. The MP user authenticates to the Pilot solution GUI using credentials and the double authentication factor. The user must hold the **Pilot Read Write** profile (or, for read-only verification, the **Pilot Read Only** profile). *Step 2 – Dashboard query (aggregated view):* 2. Select the **Dashboard** tab from the main sidebar menu. 3. Verify that the Dashboard displays the counters relevant to the MP's own activity for the current Business Date (total amount and number of transactions). 4. For each counter, hover over the information icon (i) to view the additional context/description in the pop-up box. 5. Note that the Dashboard refreshes automatically as transactions complete throughout the business day. *Step 3 – Dedicated Cash Wallets balance query (list view):* 6. Select the **Dedicated Cash Wallets** tab from the main sidebar menu. 7. Verify that the displayed list contains **only the MP's own DCWs** (per UHB §4.3.2 *"Owned by the participant to every Dedicated Cash Wallet owner"*). No DCWs belonging to other

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Market Participants are visible. 8. Click on the **Columns** button and select the columns relevant for the query, at minimum: • **Wallet ID** • **Owner** (MP's own BIC) • **Manager** (the responsible NCB's BIC) • **Available Balance** • **Currency** • **Country Code** • **Blocking Status** • **Main Wallet Flag** • **Valid From / Valid To** • **Last modified** 9. Click on the **Apply** button to confirm the column selection. 10. Use the **Filters** to narrow the result set (e.g. filter by Country Code) and verify the filtered list. 11. Click on a specific **Wallet ID** to open the wallet detail view; then click on **Details** to access the complete wallet attributes, including the *T2 RTGS account reference*, *Initiator* / *Approver* of the DCW creation, *Validity dates*, *Main Flag*, *Power of Attorney* details if any (*PoA Given To*, *PoA Validity Dates*, *PoA Max Amount*) and *Token details*. 12. From the same detail view, click on the **Transactions** sub-section to visualise all settled transactions of the selected DCW (per UHB §4.3.2: *"By clicking on Transactions, the user can visualise all settled transactions history of the Dedicated Cash Wallet. Each debit/credit on the Dedicated Cash Wallet is registered with its date, time, the counterpart (accordingly to transactions' type privacy requirements), the amount of the transaction and the transaction ID."*). 13. (Optional export) Click on the **Export** button to download an Excel file containing the current list with the applied filters and selected columns (per SDD §3.6.2.b: *"If connected as a Market Participant, the file reports the balances of their owned wallets"*). *Step 4 – Transactions history query (list view):* 14. Select the **Transactions** tab from the main sidebar menu. 15. Click on the **Filters** button and apply the desired filters (per SDD §3.6.2.c): • **Transaction type**: *Transfer*, *Issuance*, *Redemption*, *Payment*, *Payment-XvP*, or *All* • **Transaction status**: *Accepted*, *Settled*, *Unsettled*, *Rejected*, *Pending settlement*, or *All* 16. Click on **Apply** to refresh the displayed list. 17. Click on the **Columns** button and select the columns relevant for the query, at minimum: • **Transaction ID** / **Request ID** • **Type** / **Status** • **Amount** / **Currency** • **Debited Wallet ID** / **Debited Owner Entity ID** / **Debited Manager Entity ID**

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• **Credited Wallet ID** / **Credited Owner Entity ID** / **Credited Manager Entity ID** • **Creation date** / **Acceptance Timestamp** / **Settlement Timestamp** / **Settlement Business Date** • **Instructing Entity ID** (where the MP is acting on behalf or is acted upon) • **Correlation ID** 18. Click on **Apply** to confirm the column selection. 19. Verify that the list displays **only transactions impacting the MP's own DCWs** (either as Debited or as Credited counterparty). Transactions between two third-party DCWs are **not** visible. 20. Click on a specific **Transaction ID** to open the transaction detail view and verify the lifecycle timeline (Creation, Validation, Settlement), the From/To Wallet IDs and Manager BICs, and (per privacy rules) the counterpart identification where applicable. 21. (Optional export) Click on the **Export** button to download an Excel file containing the current list with the applied filters and selected columns (per SDD §3.6.2.c: *"If connected as a Market Participant, the file reports the transactions related to their owned wallets"*). *Step 5 – Cross-check between balance and transaction history (consistency check):* 22. Pick one of the MP's own DCWs from the list of Step 3 and note its **Available Balance**. 23. Navigate back to the **Transactions** tab, apply filters on the selected wallet (using the appropriate filter on Debited/Credited Wallet ID) and on status *Settled*. 24. Verify that the sum of credits minus debits across the displayed settled transactions for that wallet reconciles with the **Available Balance** noted in Step 22, taking into account any pending or unsettled transactions.
Preconditions/Details	• The Market Participant is a duly onboarded entity (per PP-NCB-01) with status *Active* and the *DCW Owner* role. • The Market Participant holds at least one active DCW that has been previously funded (e.g. via PP-E2E-001) and has been involved in at least one cash-token transaction (e.g. via PP-MP-02), so that the Dashboard counters and Transactions list contain non-empty data. • The MP user holds the **Pilot Read Write** profile (or **Pilot Read Only** for read-only verification). • The user holds valid (non-expired) credentials and double-authentication factor. • The test is executed within an open Business window.
Expected results	**U2A** • **Dashboard**: the counters are displayed for the MP scope

EUROPEAN CENTRAL BANK
EUROSYSTEM

	and refresh in real time as the MP's owned DCWs receive activity. • **Dedicated Cash Wallets tab**: the list correctly shows **only the MP's own DCWs**; **no DCWs belonging to other Market Participants are visible**. For each owned wallet, the *Available Balance*, *Manager BIC* (responsible NCB), *Country Code*, *Blocking Status*, *Main Wallet Flag*, *Valid From/To* and *Last modified* are correctly displayed. The detail view shows all wallet attributes including PoA details (if configured) and Token details. The wallet's *Transactions* sub-section shows all settled debit/credit lines with date, time, counterpart (per privacy rules), amount and transaction ID. • **Transactions tab**: the list correctly shows all transactions impacting the MP's owned DCWs (debit leg or credit leg). Filters on Type and Status correctly reduce the displayed set. Detail view shows the transaction lifecycle, From/To wallets and managers, and (per privacy rules) the counterpart identification. **Transactions between two third-party DCWs are not visible.** • **Excel exports**: per SDD §3.6.2.b and §3.6.2.c, the downloaded files report **only the MP's owned-wallets data**, with all applicable fields populated. • **Cross-check**: the *Available Balance* of an owned wallet reconciles with the net of credits and debits of its settled transactions. • **Data scope enforcement**: at no point during the test does the MP see balances, transactions, or wallet details belonging to other Market Participants or to NCBs (except the BIC of its own DCW Manager, which is its responsible NCB).
Test evidence	**U2A** • Screenshot of the **Dashboard** tab showing the MP-scope counters. • Screenshot of the **Dedicated Cash Wallets** list with selected columns and applied filters, showing only the MP's own DCWs. • Screenshot of the **Dedicated Cash Wallet detail / Details** view for one of the MP's wallets, with all attributes visible. • Screenshot of the **Transactions sub-section** within the wallet detail view, showing the wallet's settled debit/credit lines (per UHB §4.3.2). • Excel export of the Dedicated Cash Wallets list (per SDD §3.6.2.b) opened locally, showing **only the MP's owned wallets**. • Screenshot of the **Transactions** tab list with selected columns and applied filters. • Screenshot of the **Transaction detail** view (after click on a Transaction ID) showing the lifecycle timeline (Creation, Validation, Settlement). • Excel export of the Transactions list (per SDD §3.6.2.c) opened locally, showing **only transactions impacting the**

	MP's owned wallets. • Screenshot or note evidencing the reconciliation between the wallet *Available Balance* and the sum of settled credit/debit legs for that wallet.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.3 *Dedicated Cash Wallet Owner* (visibility on balances and transactions history related to owned wallets); §3.4.4 *Dedicated Cash Wallets* (data model); §3.6.1 *Queries* (Table S — Dedicated Cash Wallet reference data / balance / Statement of Wallet report / All transactions queries all available in U2A and A2A); §3.6.2 *Reporting and data extraction* — sub-sections §3.6.2.a *Dedicated Cash Wallets reference data query*, §3.6.2.b *Dedicated Cash Wallets balance query*, §3.6.2.c *Transactions for a Dedicated Cash Wallet query* (MP scope: own wallets only). • PONTES Pilot – User Handbook v1.0: §3.2.1 *Dashboard* (MPs as DCW Owners); §3.2.3 *Dedicated Cash Wallets* (MPs as DCW Owners); §3.2.5 *Transactions*; §4.2 *Dashboard Screen*; §4.3.2 *Dedicated Cash Wallets view* (Owned by the participant to every Dedicated Cash Wallet owner; Transactions sub-section); §4.11 *Monitoring*.
Related profiles/privileges	• User profiles: – **Pilot Read Write** – preferred profile for this test case – **Pilot Read Only** – equally valid for the read-only nature of this test case

2.4 Test cases for Market DLT Operators

PP-DLTO-01 – Perform wallet-to-wallet cash token payment on behalf of a Market Participant to another Participant's DCW (A2A)

Field	Description
Test Case ID	PP-DLTO-01
Test case name	As a Market DLT Operator, perform token payment from DCW on behalf of a Market Participant to the DCW of another Market Participant
Relevant for (actor)	**Market DLT Operator (MDLTO)** acting on behalf of a Market Participant – the MDLTO is configured in the Pilot solution as an external participant (SDD §2.1.5), holds no DCW of its own,

EUROPEAN CENTRAL BANK
EUROSYSTEM

	but is **whitelisted** to instruct payments on behalf of one Market Participant whose DCW will be debited. Two MPs are involved as wallet holders: • **MP-A** – the Market Participant on whose behalf the MDLTO acts; owner of the **debited DCW**. • **MP-B** – the receiving Market Participant; owner of the **credited DCW**, passive role in this test.
Input mode	A2A
Mandatory	Yes
Detailed Description	This test case describes the steps required for a Market DLT Operator (MDLTO), acting on behalf of a Market Participant (MP-A), to instruct in A2A a **wallet-to-wallet cash token Payment** debiting a DCW owned by MP-A and crediting a DCW owned by another Market Participant (MP-B). The MDLTO's authority to act on behalf of MP-A derives from the **Whitelisting** relationship configured in the LRDM of MP-A's responsible NCB (per SDD §3.4.7 *Whitelists* and PP-NCB-02). Unlike MP-on-MP instruct-on-behalf (which requires a Power of Attorney – PoA – configured on the DCW per SDD §3.4.8.a), **no PoA is required** for the MDLTO route; the Whitelisting alone is sufficient. Per SDD Table R (*List of features available to Market DLT Operators*), the MDLTO's *Wallet to Wallet Payment – Instruct* functionality is available **only in A2A** and **only in 1-step mode** (no 4-eyes approval). This is in contrast with the equivalent MP-self-instructed flow which, in U2A, requires the 4-eyes principle (cf. PP-MP-02). The 1-step A2A path is also available to MPs themselves via the same endpoint, but the Whitelisted MDLT Operator route is the one exercised here. The endpoint used is the cash-token 1-step wallet-to-wallet payment endpoint (per SDD §3.5.4.a and OpenAPI v1.0 `POST /dlt/{ncb}/api/bridge/payments`, tag *Payment – Cash Token – 1 Step*). The Pilot solution synchronously settles the cash leg on ESY-DLT and returns a confirmation. **A2A** *Step 0 – Authentication (MDLTO) – METHOD: mTLS handshake + IAM token issuance – Code: 200:* 0. The MDLTO's application authenticates to the Pilot solution IAM using the X.509 certificate of its A2A External User (`MDLTO-EU`) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). Market DLT Operators have access only via A2A;

EUROPEAN CENTRAL BANK
EUROSYSTEM

	their A2A user must have the **External User** profile (SDD §2.4.5 and Table E). *Step 1 – (Optional) Verify whitelisting before instructing – METHOD: GET – Code: 200:* 1. As a defensive pre-check, the MDLTO's application may verify that it is currently authorised to instruct on behalf of MP-A. It sends an HTTP GET request to the whitelist-verification endpoint: GET /dlt/{ncb}/api/bridge/whitelist/verify?participantBIC=<MP-A BIC11>&MDLTOperatorID=<MDLTO-ID> Path parameter: {ncb} = NCB identifier of MP-A's responsible NCB. Headers: Authorization: Bearer <JWT of MDLTO-EU>. 2. The Pilot solution returns HTTP 200 OK with payload (schema bridge.MDLTOperatorWhitelistCheckResponse) containing: • isAuthorized = true (the MDLTO is whitelisted for MP-A and the whitelisting is within its validity period); • reason (free-text confirmation). *Step 2 – Submit the wallet-to-wallet cash token Payment on behalf of MP-A – METHOD: POST – Code: 200:* 3. The MDLTO's application sends an HTTP POST request to the cash-token 1-step Payment endpoint: POST /dlt/{ncb}/api/bridge/payments Path parameter: {ncb} = NCB identifier of MP-A's responsible NCB (the NCB that manages the debited DCW). Headers: Authorization: Bearer <JWT of MDLTO-EU>, Content-Type: application/json. Request body (schema bridge.PaymentRequest): json { "paymentID": "payment_<UUID generated by MDLTO>", "debitedCashWalletAlias": "<MP-A DCW alias>", "debitedCashWalletManagerID": "<BIC of MP-A's responsible NCB>", "creditedCashWalletAlias": "<MP-B DCW alias>", "creditedCashWalletManagerID": "<BIC of MP-B's responsible NCB>", "amount": "100.50", "currency": "EUR" } Field notes: • paymentID is generated by the MDLTO (client-side); it serves as the idempotency / correlation identifier. • debitedCashWalletAlias identifies the DCW of MP-A to be debited (MDLTO must know this alias from off-Pilot agreement with MP-A). • debitedCashWalletManagerID and creditedCashWalletManagerID are the BICs of the NCBs managing the respective DCWs; if MP-A and MP-B are under

EUROPEAN CENTRAL BANK
EUROSYSTEM

	the same NCB they will be identical, otherwise different (cross-NCB variant). • amount must be ≤ Available Balance of the debited DCW and must respect the format constraint *"at most 15 digits before decimal and 2 digits after decimal"* (per OpenAPI Field Rules). • currency = EUR (the only currency configured in the Pilot solution, per UHB §4.8.3). 4. The Pilot solution performs authorisation, whitelisting and consistency checks (incl. that MDLTO-EU is whitelisted for the BIC owning debitedCashWalletAlias), then synchronously settles the cash leg on ESY-DLT (per SDD §3.5.4.a): debit MP-A's DCW, credit MP-B's DCW, propagate status to both NCB nodes. 5. The Pilot solution returns HTTP 200 OK with payload (plain string per the OpenAPI v1.0 declared response): "Cash Token Payment Settled Succesfully" *Step 3 – Post-settlement verification – METHOD: U2A or A2A query – Code: 200:* 6. **U2A cross-check on MP-A side**: an MP-A user (Pilot Read Write or Pilot Read Only) authenticates to the Pilot solution GUI and navigates to the **Transactions** tab. The user verifies that the new transaction is visible with: • Transaction type *Payment*; • Status *Settled*; • Debited Wallet ID = <MP-A DCW alias>; • Credited Wallet ID = <MP-B DCW alias>; • Amount and Currency as instructed; • **Instructing Entity ID = <MDLTO-ID>** – this is the key field demonstrating the on-behalf nature of the instruction (per SDD §3.6.2.c, the Instructing Entity ID is the ID of the participant that initiated the transaction, here the MDLTO and not MP-A). 7. **U2A cross-check on DCW balances**: the MP-A user navigates to the **Dedicated Cash Wallets** tab and verifies that the *Available Balance* of MP-A's DCW has decreased by the payment amount. Similarly (optionally) on MP-B side, the MP-B user verifies that its DCW balance has increased by the same amount. 8. **A2A alternative**: the MDLTO may retrieve the same information via the cash-token Transactions extraction (per SDD §3.6.2.c – endpoint and authorisation rules in OpenAPI v1.0), filtered on the paymentID generated in Step 2.
Preconditions/Details	• **MP-A** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant). • **MDLTO** is a Market DLT Operator duly configured in the Pilot solution as an external participant (per SDD §2.1.5 / §3.4.6, set up via PP-NCB-02 by MP-A's responsible NCB) with at least

EUROPEAN CENTRAL BANK
EUROSYSTEM

	one A2A user with the **External User** profile (SDD §2.4.5, Table E). • A valid **Whitelisting** of MDLTO for MP-A is configured in the LRDM of MP-A's responsible NCB (per SDD §3.4.7 and PP-NCB-02), with VALID FROM ≤ current Business Date ≤ VALID TO (inclusive). The Whitelisting must reference the correct MARKET DLT PLATFORM ID and PARTICIPANT BIC (MP-A's BIC). • MP-A holds at least one active DCW with **sufficient cash token balance** to cover the payment amount (e.g. previously funded via PP-E2E-001). • MP-B holds at least one active DCW able to receive cash tokens. • Neither MP-A, MP-B nor MDLTO is **blocked** (per SDD §3.4.9 / §3.4.10): MP-A's and MP-B's *Blocking Status* are *Unblocked*; MDLTO's *Blocking Status* is *Unblocked* (per PP-NCB-06 / PP-NCB-07 semantics). Both relevant DCWs are *Unblocked* and within their *Valid From / Valid To* period. • The MDLTO disposes of a valid (non-expired) X.509 certificate installed for mTLS authentication; the Common Name matches the username declared at user creation in the Pilot solution (per SDD §6.3.5). • **No PoA configuration is required** on MP-A's DCW for this scenario – the *PoA Given To* attribute of the debited DCW is **not** set (or, if set, refers to a different PoA Holder and is irrelevant here). • The test is executed within an open Business window (ESY-DLT Business window – may be verified before instructing via GET /dlt/{ncb}/api/bridge/current-business-window, optional).
Expected results	**A2A** • **Step 0 (Authentication):** mTLS handshake succeeds and the Pilot IAM issues a valid JWT to MDLTO-EU. • **Step 1 (Optional whitelist verification):** HTTP 200 OK. Response body contains isAuthorized = true and a confirmation reason. • **Step 2 (Payment instruction):** HTTP 200 OK. Response body is the success confirmation string ("Cash Token Payment Settled Succesfully"). The cash leg is settled synchronously on ESY-DLT: – MP-A's DCW (debitedCashWalletAlias) is debited by amount; – MP-B's DCW (creditedCashWalletAlias) is credited by amount; – the status of the payment is updated on both the debited and credited NCB nodes (per SDD §3.5.4.a workflow). • **Step 3 (Post-settlement verification):** The transaction is recorded with: – Type = *Payment*, Status = *Settled*; – Debited Owner Entity ID = MP-A's BIC; – Credited Owner Entity ID = MP-B's BIC; – **Instructing Entity ID = <MDLTO-ID>** (the key on-behalf

EUROPEAN CENTRAL BANK
EUROSYSTEM

	indicator); – Initiator = MDLTO-EU (the A2A application user); Approver = N/A (1-step workflow, no separation of duties applicable); – Settlement Timestamp / Settlement Business Date populated. • The transaction is visible: – in the **Transactions** tab of MP-A (as DCW Owner of the debited DCW); – in the **Transactions** tab of MP-B (as DCW Owner of the credited DCW); – in the **Transactions** tab of MP-A's responsible NCB (as DCW Manager of the debited DCW); – in the **Transactions** tab of MP-B's responsible NCB (as DCW Manager of the credited DCW). • The MDLTO has visibility on the transaction it instructed via the A2A extraction (per its External User scope). • No HTTP error codes (400, 401, 500) are returned on any of the calls. • Authorisation enforcement is correctly applied: the Pilot solution accepts the instruction only because MDLTO is whitelisted for MP-A; an equivalent call with debitedCashWalletAlias belonging to a non-whitelisted MP would be rejected (out of scope of this happy-path test case but a candidate for a sibling negative test).
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for MDLTO-EU (with sensitive tokens redacted as appropriate). • (Optional) Copy of the **HTTP request and response** for Step 1 – GET /dlt/{ncb}/api/bridge/whitelist/verify?... – evidencing isAuthorized = true. • Copy of the **HTTP request and response** (headers + JSON body) for Step 2 – POST /dlt/{ncb}/api/bridge/payments – evidencing the submitted paymentID, debitedCashWalletAlias, creditedCashWalletAlias, amount, currency, and the success-confirmation response. • (U2A cross-check) Screenshot of the **Transactions** tab on MP-A side showing the payment in *Settled* status, with the **Instructing Entity ID** column populated with the MDLTO ID. • (U2A cross-check) Screenshot of the **Transaction detail** view showing all relevant fields, including the on-behalf attribution. • (U2A cross-check) Screenshot of the **Dedicated Cash Wallets** tab on MP-A side showing the decreased *Available Balance*. • (Optional, U2A cross-check) Screenshot from MP-B's session showing the credited DCW with the increased balance and the transaction visible in MP-B's Transactions tab. • Excel export of the cash-token Transactions extraction (per

	SDD §3.6.2.c), filtered on the paymentID or on the Instructing Entity ID = MDLTO-ID, evidencing the full transaction details.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.1.5 *Market DLT Operators* (definition and access model); §2.4.5 *External Users* (A2A-only profile); §3.2.1 *Authentication & Authorisation*; §3.4.6 *Market DLT Operators*; §3.4.7 *Whitelists* (data model and configuration endpoint, Table M); §3.5.4.a *Payment with cash tokens* (workflow fig. vii) – note the explicit *Actors* line on the 1-step endpoint: "Market Participant, NCB, Market DLT Operator on behalf of Market Participant"; Table R *List of features available to market DLT Operators* (entry: *Wallet to Wallet Payment – Instruct: A2A 1-step only*); §3.6.2.c *Transactions for a Dedicated Cash Wallet query* (Instructing Entity ID field); §3.7 *A2A 1-step/2-steps illustration*; §6.3 *Preparing your integration – Connectivity Guide*. • PONTES Pilot – OpenAPI Document v1.0: tag *Payment – Cash Token – 1 Step* – operation POST /dlt/{ncb}/api/bridge/payments; tag *Market DLT Operator – Queries* – operation GET /dlt/{ncb}/api/bridge/whitelist/verify; schemas bridge.PaymentRequest, bridge.MDLTOperatorWhitelistCheckResponse, bridgecommon.CommonError. • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions* (U2A cross-check view); §4.11 *Monitoring – Transactions*. • Related test cases: PP-NCB-02 (configures the MDLTO and its Whitelisting – mandatory prerequisite); PP-MP-02 (the MP-self-instructed U2A counterpart of this test case); PP-NCB-08 (NCB-on-behalf in contingency, no PoA – administrative counterpart of this MDLTO-on-behalf scenario).
Related profiles/privileges	• User profile: **External User** – the only profile authorised for the MDLTO A2A user (SDD §2.4.5, Table E); For the optional U2A cross-check, MP-A / MP-B users hold **Pilot Read Write** or **Pilot Read Only**.

PP-DLTO-02 – Complete an XvP transaction with token settlement on behalf of two Market Participants (A2A)

Field	Description
Test Case ID	PP-DLTO-02

EUROPEAN CENTRAL BANK
EUROSYSTEM

Test case name	As a Market DLT Operator acting on behalf of two Market Participants, complete an XvP transaction with token settlement
Relevant for (actor)	**Market DLT Operator (MDLTO)** acting on behalf of both Market Participants involved in the XvP – sole active actor in this test case. The MDLTO is configured in the Pilot solution as an external participant (SDD §2.1.5), holds no DCW of its own, and is **whitelisted for both MPs**: • whitelisted for **MP-S** (the Seller, owner of the credited DCW) – enables the Initialisation call on behalf of MP-S. • whitelisted for **MP-B** (the Buyer, owner of the debited DCW) – enables the Payment call on behalf of MP-B. **MP-S** and **MP-B** are purely passive in this test case – neither MP authenticates to the Pilot solution.
Input mode	A2A
Mandatory	Yes
Detailed Description	This test case describes the end-to-end happy-path of an XvP (DvP or PvP) transaction with **Settlement in Cash Token**, instructed entirely by a single Market DLT Operator acting on behalf of **both** Market Participants involved (the Seller MP-S and the Buyer MP-B). The functional flow is the same as PP-MP-03 (XvP cash-token A2A happy path) but is **simplified by two interactions** because the MDLTO already possesses all the data needed at every step (SDD §3.5.6 fig. xii–xvi: *"Having a third party acting on behalf of the Seller and/or the Buyer, reduces the number of interactions with the Pilot solution"*): – **No Initialisation Query** (no `GET /igw/{ncb}/v1/xvps/{xvpTransactionId}` call): when the MDLTO instructs the Initialisation on behalf of the Seller, it receives the full `XvPInitResponse` payload directly in the response. Since the MDLTO is also the Buyer-side instructing actor, it does **not** need to verify the DvP Initialisation data with a separate query call – it already trusts the data it itself just received (SDD §3.5.6.b – *"the Market DLT Operator calls the Initialisation functionality on behalf of the Seller, but it does not call the Initialisation Query functionality on behalf of the Buyer because it does not need to verify DvP Initialisation data"*). – **No Seller-side Get Key & Payment Info** (no `GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION` call on the Seller side): in the *Cooperative Execution* outcome, the MDLTO already knows the Payment status from the response to its own Payment call (the same MDLTO instructed the Payment), so the cooperative-execution check used by an MP-Seller in PP-MP-03 Step 4 is redundant (SDD

EUROPEAN CENTRAL BANK
EUROSYSTEM

	§3.5.6 fig. xv – *"In the Cooperative Execution, the Market DLT Operator does not need to check the status of the Payment on behalf of the Seller because it already knows it from the last step of the Payment flow"*). The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking of the asset leg, cooperative or forced execution of the HLC) is **outside the scope of the Pilot solution** (SDD §3.5.6.e). In the MDLTO-acting-for-both scenario, the *Cooperative Execution* is the natural outcome: the MDLTO, directly operating on the Market DLT platform, executes the transfer of the asset to the Buyer on the Market DLT platform without involving either MP. *Forced Execution* using the executionKey is also available but irrelevant in the happy path (the MDLTO does not need to force a transfer to itself or to the Buyer when it controls the asset leg). The endpoints exercised are the same as PP-MP-03 (cash-token *XvP – Cash Token* tag in the OpenAPI v1.0). Per the OpenAPI v1.0 Field Rules: – on POST /igw/{ncb}/v1/xvps: marketDLTOperator must be whitelisted for the **Seller** Entity (here, MP-S); – on POST /igw/{ncb}/v1/xvps/{id}/payment: marketDLTOperator must be whitelisted for the **Buyer** Entity (here, MP-B). A single MDLTO acting for both must therefore hold **two distinct Whitelisting configurations** in the Pilot solution (one against MP-S, one against MP-B). Both whitelistings are typically configured by each MP's responsible NCB during onboarding (PP-NCB-02). The Initialisation, Payment and Get Key endpoints all follow the **one-step validation workflow** (no 4-eyes principle on XvP), and the MDLTO's A2A user uses the **External User** profile (the only profile authorised for the MDLTO – SDD §2.4.5 and Table E). The test case may be run as a **DvP** (type = "DVP") or a **PvP** (type = "PVP"); the functional flow is identical; the default execution is DVP. **A2A** *Step 0 – Authentication (MDLTO) – METHOD: mTLS handshake + IAM token issuance – Code: 200:* 0. The MDLTO's application authenticates to the Pilot solution IAM using the X.509 certificate of its A2A External User (MDLTO-EU) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). A **single authentication** is sufficient for the entire flow – the same JWT is reused for both the Seller-side Initialisation call and the Buyer-side Payment call, because both are made by the same MDLTO actor.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 1 – (Optional) Verify whitelisting for both MPs – METHOD: GET ×2 – Code: 200: 1.a As a defensive pre-check, the MDLTO's application may verify that it is currently authorised to instruct on behalf of MP-S: GET /dlt/{ncb-of-MP-S}/api/bridge/whitelist/verify?participantBIC=<MP-S BIC11>&MDLTOperatorID=<MDLTO-ID> 1.b And likewise on behalf of MP-B: GET /dlt/{ncb-of-MP-B}/api/bridge/whitelist/verify?participantBIC=<MP-B BIC11>&MDLTOperatorID=<MDLTO-ID> 2. Each call returns HTTP 200 OK with isAuthorized = true. *Step 2 – XvP Initialisation on behalf of MP-S – METHOD: POST – Code: 200:* 3. The MDLTO's application sends an HTTP POST request to the cash-token XvP initialisation endpoint, instructing on behalf of the Seller (MP-S): POST /igw/{ncb}/v1/xvps Path parameter: {ncb} = NCB identifier of **MP-S's responsible NCB** (the Initialisation routes through the Seller's NCB node, since the request is on behalf of the Seller). Headers: Authorization: Bearer <JWT of MDLTO-EU>, Content-Type: application/json. Request body (schema XvPInitRequest): json { "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLTO-ID>", "cashWalletAlias": "<MP-S DCW alias to credit>" }, "buyer": { "bic": "<MP-B BIC11>" }, "amount": "10000.50", "currency": "EUR", "type": "DVP" } Field notes: • seller.marketDLTOperator must contain the MDLTO ID and must match a Whitelisting record with PARTICIPANT BIC = <MP-S BIC11> (per OpenAPI Field Rules: *"marketDLTOperator shall be whitelisted for Seller Entity"*). • buyer.bic is sufficient at Initialisation – the Buyer-side Whitelisting is **not** verified here; it is verified in Step 3 at Payment time. • cashWalletAlias is optional and identifies the specific DCW of MP-S to credit (if omitted, the default DCW of MP-S is used). 4. The Pilot solution performs authorisation, whitelisting (for MP-S) and consistency checks, creates the new XvP and returns HTTP 200 OK with payload (schema XvPInitResponse): • xvpTransactionId (UUID) – to be reused in Step 3; • executionHash, cancellationHash (SHA-256, 64 hex chars each); • timeout (UTC ISO-8601);

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• buyer, seller, amount, currency, type (echo). 5. The MDLTO records xvpTransactionId, executionHash, cancellationHash and timeout. The two hashes are the values the MDLTO will use to set up the Hash Link Contract on the Market DLT platform (out of scope of this test case). *Step 3 – (Skipped) Initialisation Query – METHOD: N/A – Code: N/A:* 6. **No call is made**. The MDLTO does not need to call GET /igw/{ncb}/v1/xvps/{xvpTransactionId} on behalf of the Buyer (per SDD §3.5.6.b and fig. xiii) because the MDLTO already received the full XvPInitResponse in Step 2 and does not need to "verify" data it itself just produced. This is the **first of the two simplifications** of the MDLTO-acting-for-both flow vs. PP-MP-03. *Step 4 – Payment on behalf of MP-B – METHOD: POST – Code: 200:* 7. The MDLTO's application sends an HTTP POST request to the cash-token Payment endpoint, instructing on behalf of the Buyer (MP-B): POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment Path parameters: {ncb} = NCB identifier of **MP-B's responsible NCB** (the Payment routes through the Buyer's NCB node); {xvpTransactionId} from Step 2. Headers: Authorization: Bearer <JWT of MDLTO-EU>, Content-Type: application/json. Request body (schema PaymentRequest): json { "buyer": { "bic": "<MP-B BIC11>", "marketDLTOperator": "<MDLTO-ID>", "cashWalletAlias": "<MP-B DCW alias to debit>" }, "seller": { "bic": "<MP-S BIC11>" }, "amount": "10000.50", "currency": "EUR" } Field notes: • buyer.marketDLTOperator must contain the MDLTO ID and must match a Whitelisting record with PARTICIPANT BIC = <MP-B BIC11> (per OpenAPI Field Rules: *"marketDLTOperator shall be whitelisted for Buyer Entity"*). This is a **different** Whitelisting record from the one verified in Step 2. • cashWalletAlias identifies the DCW of MP-B to debit (if omitted, the default DCW of MP-B is used). • amount and currency must match exactly the values returned in Step 2. 8. The Pilot solution performs authorisation, whitelisting (for MP-B) and consistency checks, synchronously settles the cash leg on ESY-DLT (debit MP-B's DCW, credit MP-S's DCW or the alias-designated DCW) and returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo);

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• payment.id (UUID of the cash-token payment); • payment.status = "SETTLED"; • payment.reason; • executionKey (64–128 hex chars) – **returned to the MDLTO acting on behalf of the Buyer**, since the MDLTO is the entitled actor on the Buyer side (SDD §3.5.6 fig. xiv: *"The Market DLT Operator, on behalf of the Buyer, receives the execution secret (and not the cancellation secret)"*). 9. The MDLTO records the returned executionKey. *Step 5 – (Skipped) Get Key & Payment Info on Seller side – METHOD: N/A – Code: N/A:* 10. **No call is made**. In the *Cooperative Execution* outcome, the MDLTO does not need to call GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION on behalf of the Seller (per SDD §3.5.6 fig. xv) because the MDLTO already knows from the response to Step 4 that the cash leg is SETTLED. The MDLTO then executes the asset transfer on the Market DLT platform directly (out of scope of this test case). This is the **second of the two simplifications** of the MDLTO-acting-for-both flow vs. PP-MP-03. *Step 6 – (Optional) MDLTO re-retrieves execution key – METHOD: GET – Code: 200:* 11. As an additional resilience check, the MDLTO may re-call the Get Key & Payment Info endpoint on the Buyer side with key=EXECUTION (still on behalf of MP-B): GET /igw/{ncb-of-MP-B}/v1/xvps/{xvpTransactionId}/payment?key=EXECUTION Headers: Authorization: Bearer <JWT of MDLTO-EU>. 12. The Pilot solution returns HTTP 200 OK with the same executionKey as in Step 4 (per SDD §3.5.6 *Availability of Execution and Cancellation Key*).
Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant, but requires additional steps not described here, e.g. MDLTO must use two different users and certificates). • **MDLTO** is duly configured in the Pilot solution as an external participant (per SDD §2.1.5 / §3.4.6) with at least one A2A user with the **External User** profile (SDD §2.4.5, Table E). • **Two distinct Whitelisting records** are configured for the MDLTO in the LRDM, both within their VALID FROM / VALID TO validity period (per PP-NCB-02 and SDD §3.4.7): – one with PARTICIPANT BIC = <MP-S BIC11>, configured by MP-S's responsible NCB – enables the Initialisation on behalf of MP-S; – one with PARTICIPANT BIC = <MP-B BIC11>, configured by MP-B's responsible NCB – enables the Payment on behalf of MP-B.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• Both Whitelistings reference the same MARKET DLT OPERATOR ID = <MDLTO-ID> and the same MARKET DLT PLATFORM ID (the Market DLT platform on which the asset is locked in the HLC). • MP-S holds at least one active DCW to receive cash tokens. • MP-B holds at least one active DCW with **sufficient cash token balance** to cover the XvP amount (e.g. previously funded via PP-E2E-001). • Neither MP-S, MP-B nor MDLTO is **blocked** in the Pilot solution; both relevant DCWs are *Unblocked* and within their *Valid From / Valid To* period. • The MDLTO disposes of a valid (non-expired) X.509 certificate installed for mTLS authentication. • **No PoA configuration is required** on either MP-S's or MP-B's DCW for this scenario – the MDLTO's authority derives from the Whitelistings, not from PoA. • The test is executed within an open Business window. The XvP timeout (system parameter, per SDD §3.5.6.a) is sufficient to complete Step 2 (Initialisation) and Step 4 (Payment) in sequence (i.e. the Payment occurs strictly before timeout).
Expected results	**A2A** • **Step 0 (Authentication):** mTLS handshake succeeds and the Pilot IAM issues a single valid JWT to MDLTO-EU reusable across all subsequent calls. • **Step 1 (Optional whitelist verification, ×2):** Both calls return HTTP 200 OK with isAuthorized = true. • **Step 2 (Initialisation on behalf of MP-S):** HTTP 200 OK. Response body contains a new xvpTransactionId (UUID v4), valid executionHash and cancellationHash, timeout in UTC ISO-8601, and the echo of buyer, seller, amount, currency, type. The XvP is registered in the Pilot solution; the Instructing Entity ID recorded on this Initialisation is **<MDLTO-ID>** (not MP-S's BIC), demonstrating the on-behalf nature of the instruction. • **Step 3 (Skipped Initialisation Query):** No HTTP call; no entry in the API gateway log between Step 2 and Step 4. • **Step 4 (Payment on behalf of MP-B):** HTTP 200 OK. The cash leg is settled synchronously on ESY-DLT: – MP-B's DCW (cashWalletAlias from Payment request, or default DCW of MP-B) is debited by amount; – MP-S's DCW (cashWalletAlias from Initialisation, or default DCW of MP-S) is credited by amount; – the response contains payment.status = "SETTLED" and a non-null executionKey (64–128 lower-case hex chars). – the Instructing Entity ID recorded on this Payment is **<MDLTO-ID>** (not MP-B's BIC). • **Step 5 (Skipped Seller-side Get Key):** No HTTP call. • **Step 6 (Optional re-retrieval):** HTTP 200 OK with the same executionKey and payment.status = "SETTLED". • The XvP transaction is visible as a Payment-XvP of type *Payment* with status *Settled* in the **Transactions** tab of **both**

EUROPEAN CENTRAL BANK
EUROSYSTEM

	MP-S and MP-B (each MP sees the transaction in which it participates, even though it was the MDLTO that instructed it). The Instructing Entity ID column is populated with <MDLTO-ID> on **both** sides. • **Total HTTP calls in the mandatory path: 2** (one Initialisation + one Payment) – versus 4 in PP-MP-03 (one Initialisation + one Query + one Payment + one Seller-side Get Key). The two-call savings vs. PP-MP-03 is the measurable benefit of the MDLTO-acting-for-both flow. • Authorisation enforcement is correctly applied: the Pilot solution validates the MP-S Whitelisting on Step 2 and the MP-B Whitelisting on Step 4; an equivalent flow where the MDLTO is whitelisted for only one of the two MPs is rejected with 401/403 on the call to the un-whitelisted MP's side (out of scope of this happy-path test case but a candidate for a sibling negative test). • No HTTP error codes (400, 401, 403, 404, 409, 500, 503) are returned on any of the calls.
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for MDLTO-EU (single authentication, reused throughout). • (Optional) Copies of the HTTP request and response for Step 1 – both GET /dlt/{ncb}/api/bridge/whitelist/verify calls (one per MP) – evidencing isAuthorized = true for both MP-S and MP-B. • Copy of the **HTTP request and response** (headers + JSON body) for Step 2 – POST /igw/{ncb-of-MP-S}/v1/xvps – evidencing the returned xvpTransactionId, executionHash, cancellationHash, timeout. • Copy of the HTTP request and response for Step 4 – POST /igw/{ncb-of-MP-B}/v1/xvps/{xvpTransactionId}/payment – evidencing payment.status = "SETTLED" and the executionKey returned to the MDLTO acting on behalf of the Buyer. • (Optional) Copy of the HTTP request and response for Step 6 – GET …?key=EXECUTION – evidencing the same executionKey. • **API gateway / proxy log** evidencing the **absence** of calls to GET /igw/{ncb}/v1/xvps/{xvpTransactionId} (Initialisation Query) and to GET /igw/{ncb-of-MP-S}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION (Seller-side Get Key) between Step 2 and Step 6 – this is the positive evidence of the two-call reduction vs. PP-MP-03. • (U2A cross-check) Screenshots of the **Transactions** tab on **both** MP-S and MP-B sides showing the corresponding Payment-XvP transaction in *Settled* status, with the **Instructing Entity ID = <MDLTO-ID>** column populated on both sides. • (U2A cross-check) Screenshots of the **Dedicated Cash Wallets** tab on both MPs showing the updated balances (MP-B's DCW decreased by amount, MP-S's DCW increased by amount). • Excel export of the cash-token Transactions extraction filtered

EUROPEAN CENTRAL BANK
EUROSYSTEM

	on the xvpTransactionId, evidencing the full XvP transaction record with both legs and the on-behalf attribution.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.1.5 *Market DLT Operators*; §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.4.6 *Market DLT Operators*; §3.4.7 *Whitelists*; §3.5.6 *DvP & PvP with Cash Token settlement* – particularly §3.5.6.a *Initialisation* (note the *Actors* line: "Market Participant (Seller), Market DLT Operator instructing on behalf of the Seller"), §3.5.6.b *Initialisation Query* (skipped in this flow), §3.5.6.c *Payment* (Actors: "Market Participant (Buyer), Market DLT Operator instructing on behalf of the Buyer"), §3.5.6.d *Get Key and Payment Info*, §3.5.6.e *Initialisation on Market DLT platform*, and especially §3.5.6 *Key Management flows for Market DLT Operator* (fig. xii–xvi: *XvP initialisation high level workflow in A2A – via Market DLT Operator*, *Payment settled situation high level workflow in A2A – via Market DLT Operator*, *Cooperative execution high level workflow in A2A – via Market DLT Operator*, *Forced execution high level workflow in A2A – via Market DLT Operator*); §3.6.2.c *Transactions for a Dedicated Cash Wallet query*; Table R *List of features available to market DLT Operators* (entries: *DvP and PvP transaction – Initialisation*, *Payment on Eurosystem DLT*, *Get Key and payment info*); §3.7 *A2A 1-step/2-steps illustration*; §6.3 *Preparing your integration – Connectivity Guide*. • PONTES Pilot – OpenAPI Document v1.0: tag *XvP – Cash Token* – operations createXvP (POST /igw/{ncb}/v1/xvps), payment (POST /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment), getPaymentStatus (GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key={key}); schemas XvPInitRequest, XvPInitResponse, PaymentRequest, PaymentResponse, Participant, SimpleParticipant, KeyType, PaymentStatus, TransactionType. Note the per-endpoint Field Rules: *"marketDLTOperator shall be whitelisted for Seller Entity"* (Initialisation) and *"marketDLTOperator shall be whitelisted for Buyer Entity"* (Payment). • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions* (U2A cross-check); §4.11 *Monitoring – Transactions*. • Related test cases: PP-NCB-02 (configures the MDLTO and its **two** Whitelistings – mandatory prerequisite); PP-DLTO-01 (MDLTO instructing wallet-to-wallet payment – simpler same-actor flow); PP-MP-03 (the MP-self-instructed XvP cash-token A2A counterpart of this test case – useful for direct comparison of the two-call savings).
Related profiles/privileges	• User profile (MDLTO): **External User** – the only profile authorised for the MDLTO A2A user (SDD §2.4.5, Table E); for the optional U2A cross-check, MP-S / MP-B users hold **Pilot Read Write** or **Pilot Read Only**.

EUROPEAN CENTRAL BANK
EUROSYSTEM

PP-DLTO-03 – Complete an XvP transaction with direct RTGS settlement on behalf of two Market Participants and retrieve execution key (A2A)

Field	Description
Test Case ID	PP-DLTO-03
Test case name	As a Market DLT Operator acting on behalf of two Market Participants, complete an XvP transaction with direct RTGS settlement and retrieve the execution key
Relevant for (actor)	**Market DLT Operator (MDLTO)** acting on behalf of both Market Participants involved in the XvP – sole active actor in this test case. The MDLTO is configured in the Pilot solution as an external participant (SDD §2.1.5), and is **whitelisted for both MPs**: • whitelisted for **MP-S** (the Seller, T2 RTGS DCA holder of the credited DCA) – enables the Initialisation call on behalf of MP-S. • whitelisted for **MP-B** (the Buyer, T2 RTGS DCA holder of the debited DCA) – enables the Payment call on behalf of MP-B. **MP-S** and **MP-B** are purely passive in this test case – neither MP authenticates to the Pilot solution.
Input mode	A2A
Mandatory	Yes
Detailed Description	This test case describes the end-to-end happy-path of an XvP (DvP or PvP) transaction with **Direct Settlement in T2 RTGS**, instructed entirely by a single Market DLT Operator acting on behalf of **both** Market Participants involved (Seller MP-S and Buyer MP-B). It is the **direct RTGS counterpart of PP-DLTO-02** (XvP cash-token MDLTO-acting-for-both) and the **MDLTO-acting-for-both counterpart of PP-MP-04** (XvP direct-RTGS MP-self-instructed). As in PP-DLTO-02, the functional flow is **simplified by two interactions** vs. PP-MP-04 because the MDLTO already possesses all the data needed at every step (SDD §3.5.6 fig. xii–xvi: *"Having a third party acting on behalf of the Seller and/or the Buyer, reduces the number of interactions with the Pilot solution"*): – **No Initialisation Query** (no `GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}` call): the MDLTO does not need to

EUROPEAN CENTRAL BANK
EUROSYSTEM

	verify DvP Initialisation data it itself just produced. – **No Seller-side Get Key & Payment Info** (no `GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION` call on the Seller side): the MDLTO already knows the Payment status from the response to its own Payment call. Compared to PP-DLTO-02 (cash-token), PP-DLTO-03 (direct-RTGS) inherits three settlement-specific differences from PP-MP-04: – **Settlement venue**: the cash leg is settled on the Seller's and Buyer's **T2 RTGS DCAs** (not on DCWs on ESY-DLT). The Pilot solution orchestrates the debit/credit via T2 RTGS using ISO 20022 messages (pacs.010 to debit MP-B's DCA, pacs.009 to credit MP-S's DCA, with pacs.002 status reports). – **NRO**: the Payment request requires a digital signature (`signature` + `signerPEM` fields in `RTGSPaymentRequest`) for Non-Repudiation of Origin (SDD §3.3). Per the OpenAPI v1.0 Field Rules: *"signature shall contain signature of Payload containing only amount, xvpTransactionId, buyer bic and seller bic"*. The identity of the signer (MDLTO's certificate vs. MP-B's certificate vs. a delegated certificate) is a key point in the MDLTO-on-behalf RTGS variant. – **No `cashWalletAlias`** in the request payload: the `RTGSParticipant` schema does not support a wallet alias – the cash leg is settled on the T2 RTGS DCA configured for the MP in the LRDM, not on an ESY-DLT DCW. The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking of the asset leg, cooperative or forced execution of the HLC) is **outside the scope of the Pilot solution** (SDD §3.5.6.e). In the MDLTO-acting-for-both scenario, the *Cooperative Execution* is the natural outcome: the MDLTO, directly operating on the Market DLT platform, executes the transfer of the asset to MP-B on the Market DLT platform without involving either MP. The endpoints exercised are the *XvP – Direct RTGS* family (same as PP-MP-04). Per the OpenAPI v1.0 Field Rules: – on `POST /igw/{ncb}/v1/direct-rtgs/xvps`: `marketDLTOperator` must be whitelisted for the **Seller** Entity (here, MP-S); – on `POST /igw/{ncb}/v1/direct-rtgs/xvps/{id}/payment`: `marketDLTOperator` must be whitelisted for the **Buyer** Entity (here, MP-B). A single MDLTO acting for both must therefore hold **two distinct Whitelisting configurations** in the Pilot solution (one against MP-S, one against MP-B), as in PP-DLTO-02. All XvP – Direct RTGS endpoints declare *Authorized Actors: DedicatedCashWalletOwner; Whitelisted Market DLT Operator* and follow the **one-step validation workflow**. The MDLTO's A2A user uses the **External User** profile (SDD §2.4.5 and Table E).

EUROPEAN CENTRAL BANK
EUROSYSTEM

The test case may be run as a **DvP** (type = “DVP”) or a **PvP** (type = “PVP”); the functional flow is identical; the default execution is DVP.

A2A

Step 0 – Authentication (MDLTO) – METHOD: mTLS handshake + IAM token issuance – Code: 200:
0. The MDLTO’s application authenticates to the Pilot solution IAM using the X.509 certificate of its A2A External User (MDLTO-EU) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). A **single authentication** is sufficient for the entire flow (same JWT reused across the Seller-side Initialisation call and the Buyer-side Payment call).

Step 1 – (Optional) Verify whitelisting for both MPs – METHOD: GET ×2 – Code: 200:
1.a GET /dlt/{ncb-of-MP-S}/api/bridge/whitelist/verify?participantBIC=<MP-S BIC11>&MDLTOperatorID=<MDLTO-ID> → isAuthorized = true.
1.b GET /dlt/{ncb-of-MP-B}/api/bridge/whitelist/verify?participantBIC=<MP-B BIC11>&MDLTOperatorID=<MDLTO-ID> → isAuthorized = true.

Step 2 – XvP Initialisation on behalf of MP-S – METHOD: POST – Code: 200:
2. The MDLTO’s application sends an HTTP POST request to the direct-RTGS XvP initialisation endpoint, instructing on behalf of the Seller (MP-S):
POST /igw/{ncb}/v1/direct-rtgs/xvps
Path parameter: {ncb} = NCB identifier of **MP-S’s responsible NCB**.
Headers: Authorization: Bearer <JWT of MDLTO-EU>, Content-Type: application/json.
Request body (schema RTGSXvPInitRequest):
json
{
“seller”: { “bic”: “<MP-S BIC11>”, “marketDLTOperator”: “<MDLTO-ID>” },
“buyer”: { “bic”: “<MP-B BIC11>” },
“amount”: “10000.50”,
“currency”: “EUR”,
“type”: “DVP”
}
Field notes:
• seller.marketDLTOperator must match a Whitelisting record with PARTICIPANT BIC = <MP-S BIC11> (per OpenAPI Field Rules: *“marketDLTOperator shall be whitelisted for Seller Entity”*).
• The RTGSParticipant schema does **not** support cashWalletAlias – the cash leg is settled on MP-S’s T2 RTGS

EUROPEAN CENTRAL BANK
EUROSYSTEM

	DCA declared via the *T2 account reference* configured in the LRDM by MP-S's NCB. 3. The Pilot solution performs authorisation, whitelisting (for MP-S) and consistency checks, creates the new XvP and returns HTTP 200 OK with payload (schema RTGSXvPInitResponse): xvpTransactionId, executionHash, cancellationHash, timeout, and the echo of the economic terms. 4. The MDLTO records xvpTransactionId, executionHash, cancellationHash and timeout. *Step 3 – (Skipped) Initialisation Query – METHOD: N/A – Code: N/A:* 5. **No call is made**. As in PP-DLTO-02 Step 3, the MDLTO does not need to call GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId} because it already received the full RTGSXvPInitResponse in Step 2. First simplification vs. PP-MP-04. *Step 4 – Payment on behalf of MP-B – METHOD: POST – Code: 200:* 6. The MDLTO's application prepares the **NRO digital signature** of the Payment request payload, signing the canonical fields amount, xvpTransactionId, buyer bic, seller bic (per OpenAPI v1.0 Field Rules). The instruction must be signed with the MDLTO's certificate identity. 7. The MDLTO's application sends an HTTP POST request to the direct-RTGS Payment endpoint, instructing on behalf of the Buyer (MP-B): POST /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment Path parameters: {ncb} = NCB identifier of **MP-B's responsible NCB**; {xvpTransactionId} from Step 2. Headers: Authorization: Bearer <JWT of MDLTO-EU>, Content-Type: application/json. Request body (schema RTGSPaymentRequest): json { "buyer": { "bic": "<MP-B BIC11>", "marketDLTOperator": "<MDLTO-ID>" }, "seller": { "bic": "<MP-S BIC11>" }, "amount": "10000.50", "currency": "EUR", "signature": "<base64-encoded signature over (amount + xvpTransactionId + MP-B BIC + MP-S BIC)>", "signerPEM": "<PEM-encoded X.509 certificate of the signer>" } Field notes: • buyer.marketDLTOperator must match a Whitelisting record with PARTICIPANT BIC = <MP-B BIC11> (a **different** Whitelisting record from the one verified in Step 2). • amount, currency must match exactly the values returned in Step 2. • The NRO signature must validate against signerPEM, with

EUROPEAN CENTRAL BANK
EUROSYSTEM

	the signed payload restricted to the four canonical fields. 8. The Pilot solution performs authorisation, whitelisting (for MP-B), consistency and signature-validity checks, then forwards the Payment instruction to MP-B's NCB's Eurosystem DLT node, which creates a Payment Instruction with status `Initialized` and submits it to the BBK DLT node. The BBK DLT node's Trigger Backend orchestrates the T2 RTGS settlement leg (per SDD §3.5.6 *Direct Settlement*, steps 4–7): – `pacs.010` to debit MP-B's T2 RTGS DCA; – positive `pacs.002` reporting the debit; – `pacs.009` to credit MP-S's T2 RTGS DCA; – positive `pacs.002` reporting the credit. 9. Upon completion of the T2 RTGS settlement leg, the BBK DLT node propagates the final payment status to MP-B's and MP-S's NCB nodes, and the Pilot solution returns HTTP `200 OK` to the MDLTO with payload (schema `PaymentResponse`): • `xvpTransactionId` (echo); • `payment.id`, `payment.status = "SETTLED"`, `payment.reason`; • `executionKey` (64–128 hex chars) – **returned to the MDLTO acting on behalf of the Buyer** (per SDD §3.5.6 fig. xiv). 10. The MDLTO records the returned `executionKey`. *Step 5 – (Skipped) Get Key & Payment Info on Seller side – METHOD: N/A – Code: N/A:* 11. **No call is made**. Second simplification vs. PP-MP-04: the MDLTO already knows the Payment status from Step 4 and proceeds directly with *Cooperative Execution* on the Market DLT platform (out of scope of this test case). *Step 6 – (Optional) MDLTO re-retrieves execution key – METHOD: GET – Code: 200:* 12. As an additional resilience check, the MDLTO may re-call the Get Key & Payment Info endpoint on the Buyer side with `key=EXECUTION` (still on behalf of MP-B): `GET /igw/{ncb-of-MP-B}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=EXECUTION` 13. The Pilot solution returns HTTP `200 OK` with the same `executionKey` as in Step 4 (per SDD §3.5.6 *Availability of Execution and Cancellation Key*).
Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant, but requires additional steps not described here, e.g. MDLTO must use two different users and certificates). • Each MP holds an active **T2 RTGS DCA** in T2 RTGS, with a valid *T2 account reference* declared in the Pilot solution by the responsible NCB (per PP-NCB-01). MP-B's T2 RTGS DCA has sufficient available liquidity to cover the XvP amount. • A **direct debit mandate** is in place in T2 RTGS allowing the

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Pilot solution (via the BBK DLT node / Trigger Backend) to debit MP-B's T2 RTGS DCA via pacs.010 (per SDD §3.5.6 *Direct Settlement* and *Unhappy Paths* TP-DR-DD-001). • **MDLTO** is duly configured in the Pilot solution as an external participant (per SDD §2.1.5 / §3.4.6) with at least one A2A user with the **External User** profile. • **Two distinct Whitelisting records** are configured for the MDLTO in the LRDM, both within their VALID FROM / VALID TO validity period: – one with PARTICIPANT BIC = <MP-S BIC11> – enables the Initialisation on behalf of MP-S; – one with PARTICIPANT BIC = <MP-B BIC11> – enables the Payment on behalf of MP-B. • Both Whitelistings reference the same MARKET DLT OPERATOR ID = <MDLTO-ID> and the same MARKET DLT PLATFORM ID. • Neither MP-S, MP-B nor MDLTO is **blocked** in the Pilot solution; neither MP-S's nor MP-B's T2 RTGS DCA is blocked in T2 RTGS (per SDD *Unhappy Paths* TP-DCA-002 / TP-DR-PART-002). • The MDLTO disposes of a valid (non-expired) X.509 certificate installed for mTLS authentication. • The MDLTO (or, alternatively, MP-B) disposes of a valid **signing certificate and corresponding private key** to produce the NRO signature required by RTGSPaymentRequest (signature + signerPEM). The signer certificate is recognised by the Pilot solution. • **No PoA configuration is required** on either MP-S's or MP-B's side – the MDLTO's authority derives from the Whitelistings, not from PoA. • The test is executed within the **T2 RTGS settlement window** and within an open Business window. The XvP timeout is sufficient to complete Step 2 and Step 4 in sequence. • Relevant T2 RTGS message subscriptions are in place if MP-B / MP-S wish to receive the corresponding pacs.002 status reports for evidence purposes (optional).
Expected results	**A2A** • **Step 0 (Authentication):** mTLS handshake succeeds and the Pilot IAM issues a single valid JWT to MDLTO-EU reusable across all subsequent calls. • **Step 1 (Optional whitelist verification, ×2):** Both calls return HTTP 200 OK with isAuthorized = true. • **Step 2 (Initialisation on behalf of MP-S):** HTTP 200 OK. Response body contains a new xvpTransactionId (UUID v4), valid executionHash and cancellationHash, timeout in UTC ISO-8601, and the echo of the economic terms. The XvP is registered in the Pilot solution with an associated direct-RTGS payment in Initialized status (per SDD §3.6.2 *Direct RTGS Payments* status list). The Instructing Entity ID recorded on this Initialisation is **<MDLTO-ID>**.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• **Step 3 (Skipped Initialisation Query):** No HTTP call; no entry in the API gateway log between Step 2 and Step 4. • **Step 4 (Payment on behalf of MP-B):** HTTP 200 OK. The cash leg is settled in T2 RTGS: – MP-B's T2 RTGS DCA is debited by amount (pacs.010 + positive pacs.002); – MP-S's T2 RTGS DCA is credited by amount (pacs.009 + positive pacs.002); – the Pilot solution direct-RTGS payment status reaches Completed (per SDD §3.6.2 *Direct RTGS Payments* status list); – the response contains payment.status = "SETTLED" and a non-null executionKey (64–128 lower-case hex chars). – the Instructing Entity ID recorded on this Payment is **<MDLTO-ID>**. • **Step 5 (Skipped Seller-side Get Key):** No HTTP call. • **Step 6 (Optional re-retrieval):** HTTP 200 OK with the same executionKey and payment.status = "SETTLED". • The XvP transaction is recorded in the Pilot solution's Direct RTGS Payments extraction (SDD §3.6.2) with type *Direct RTGS Payment-XvP* and status *Completed*, with the populated Correlation ID linking to the corresponding T2 RTGS payment messages, and the Instructing Entity ID column populated with <MDLTO-ID> on both MP-S's and MP-B's views. • **Total HTTP calls in the mandatory path: 2** (one Initialisation + one Payment) – versus 4 in PP-MP-04 (one Initialisation + one Query + one Payment + one Seller-side Get Key). Identical savings to PP-DLTO-02 (cash-token), confirming the symmetry between the two settlement models. • Authorisation enforcement is correctly applied: MP-S Whitelisting validated on Step 2; MP-B Whitelisting validated on Step 4; NRO signature validated against signerPEM on Step 4. • No HTTP error codes (400, 401, 403, 404, 409, 500, 503) are returned on any of the calls.
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for MDLTO-EU (single authentication, reused throughout). • (Optional) Copies of the HTTP request and response for Step 1 – both GET /dlt/{ncb}/api/bridge/whitelist/verify calls – evidencing isAuthorized = true for both MP-S and MP-B. • Copy of the **HTTP request and response** (headers + JSON body) for Step 2 – POST /igw/{ncb-of-MP-S}/v1/direct-rtgs/xvps – evidencing the returned xvpTransactionId, executionHash, cancellationHash, timeout. • Copy of the HTTP request and response for Step 4 – POST /igw/{ncb-of-MP-B}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment – evidencing the submitted signature and signerPEM, payment.status = "SETTLED", and the executionKey returned to the MDLTO.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• (Optional) Copy of the HTTP request and response for Step 6 – GET …?key=EXECUTION – evidencing the same executionKey. • **API gateway / proxy log** evidencing the **absence** of calls to GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId} (Initialisation Query) and to GET /igw/{ncb-of-MP-S}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION (Seller-side Get Key) between Step 2 and Step 6 – positive evidence of the two-call reduction vs. PP-MP-04. • (Optional) Copies of the T2 RTGS ISO 20022 messages exchanged during settlement: pacs.010 (Buyer DCA debit), pacs.002 (debit outcome), pacs.009 (Seller DCA credit), pacs.002 (credit outcome) – obtained via the respective message subscriptions if configured. • Excel export of the Direct T2 RTGS Payments extraction (per SDD §3.6.2) filtered on the xvpTransactionId / Correlation ID, evidencing the full record with type *Direct RTGS Payment-XvP*, status *Completed*, and Instructing Entity ID = <MDLTO-ID>. • (Optional, U2A cross-check) Screenshots of the **Direct RTGS Payments** monitoring tab showing the corresponding Payment-XvP transaction in *Completed* status, with the Instructing Entity ID column populated with the MDLTO ID.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.1.5 *Market DLT Operators*; §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.3 *Non-Repudiation of Origin (NRO)*; §3.4.6 *Market DLT Operators*; §3.4.7 *Whitelists*; §3.5.6 *DvP & PvP* – particularly §3.5.6.a *Initialisation* (note: *"Actors: Market Participant (Seller), Market DLT Operator instructing on behalf of the Seller"* for the direct-RTGS endpoint), §3.5.6.b *Initialisation Query* (skipped in this flow), §3.5.6.c *Payment* (Actors: *"Market Participant (Buyer), Market DLT Operator instructing on behalf of the Buyer"* for the direct-RTGS endpoint; including *Direct Settlement* fig. i – pacs.010 / pacs.002 / pacs.009 flow), §3.5.6.d *Get Key and Payment Info*, §3.5.6.e *Initialisation on Market DLT platform*, and especially the *Key Management flows for Market DLT Operator* (fig. xii–xvi); §3.5.7 *Unhappy Paths & Error codes* (TP-DCA-001/002, TP-DR-DD-001, TP-DR-PART-002, etc.); §3.6.2 *Reporting and data extraction – Direct RTGS Payments*; Table R *List of features available to market DLT Operators* (entries: *DvP and PvP transaction (direct settlement in T2 RTGS) – Initialisation*, *Payment in T2 RTGS*, *Get Key and payment info*); §3.7 *A2A 1-step/2-steps illustration*; §6.3 *Preparing your integration – Connectivity Guide*. • PONTES Pilot – OpenAPI Document v1.0: tag *XvP – Direct RTGS* – operations RTGSCreateXvP (POST /igw/{ncb}/v1/direct-rtgs/xvps), RTGSPayment (POST /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment), RTGSGetPaymentStatus (GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key={key}); schemas RTGSXvPInitRequest, RTGSXvPInitResponse,

	RTGSPaymentRequest, PaymentResponse, RTGSParticipant, SimpleParticipant, KeyType, PaymentStatus, TransactionType. Note the per-endpoint Field Rules: *"marketDLTOperator shall be whitelisted for Seller Entity"* (Initialisation) and *"marketDLTOperator shall be whitelisted for Buyer Entity"* + *"signature shall contain signature of Payload containing only amount, xvpTransactionId, buyer bic and seller bic"* (Payment). • T2 RTGS UDFS – §12.4 ISO 20022 messages (pacs.010 FinancialInstitutionDirectDebit, pacs.009 FinancialInstitutionCreditTransfer, pacs.002 PaymentStatusReport). • PONTES Pilot – User Handbook v1.0: §3.2.7 *Direct RTGS Payments*; §4.6.1 *Direct RTGS Payments*; §4.11 *Monitoring – Direct RTGS Payments*. • Related test cases: PP-NCB-02 (configures the MDLTO and its **two** Whitelistings – mandatory prerequisite); PP-DLTO-02 (the cash-token counterpart of this test case – useful for direct comparison); PP-MP-04 (the MP-self-instructed direct-RTGS XvP counterpart – useful for the two-call savings demonstration).
Related profiles/privileges	• User profile (MDLTO): **External User** – the only profile authorised for the MDLTO A2A user.

PP-DLTO-04 – Fail to complete an XvP transaction within timeout and retrieve cancellation key on behalf of two Market Participants (A2A, any settlement method)

Field	Description
Test Case ID	PP-DLTO-04
Test case name	As a Market DLT Operator acting on behalf of two Market Participants, fail to complete an XvP transaction (any settlement method) within timeout and retrieve cancellation key
Relevant for (actor)	**Market DLT Operator (MDLTO)** acting on behalf of both Market Participants involved in the XvP – sole active actor in this test case. The MDLTO is configured in the Pilot solution as an external participant (SDD §2.1.5), and is **whitelisted for both MPs**: • whitelisted for **MP-S** (the Seller) – enables the Initialisation

EUROPEAN CENTRAL BANK
EUROSYSTEM

	call on behalf of MP-S **and** the cancellation-key retrieval call after timeout (per OpenAPI v1.0: *"Authorized Actors on Get Payment Status: DedicatedCashWalletOwner; Whitelisted Market DLT Operator"*); • whitelisted for **MP-B** (the Buyer) – would enable the Payment call on behalf of MP-B, but deliberately **not** exercised in this test (no Payment is submitted before timeout). **MP-S** and **MP-B** are purely passive in this test case – neither MP authenticates to the Pilot solution.
Input mode	A2A
Mandatory	Yes
Detailed Description	This test case describes the **unhappy-path** of an XvP (DvP or PvP) transaction, instructed by a single Market DLT Operator on behalf of **both** the Seller (MP-S) and the Buyer (MP-B), in which the MDLTO **does not submit the Payment** for the cash leg before the XvP timeout expires. The test verifies that the MDLTO can subsequently retrieve the **cancellation key** on behalf of MP-S (the entitled side after timeout) and thus reclaim the asset locked in the Hash Link Contract on the Market DLT platform (out of scope of this test case). **The test case is settlement-model-agnostic** and may be executed against **either** of the two XvP settlement models (cash-token OR direct-RTGS) using the corresponding endpoint family. The functional flow is identical, and the choice of settlement model is made at Initialisation (Step 2) – from that point on, all subsequent calls must use the matching endpoint family (per SDD §3.5.6: *"From this step on, the settlement mode is fixed for all the lifecycle of the DvP / PvP, and the endpoints of any other functionality must be called accordingly"*). The endpoint families are: – **Cash Token settlement** (counterpart of PP-MP-05 / PP-DLTO-02): `POST /igw/{ncb}/v1/xvps` (Initialisation) `GET /igw/{ncb}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION` (Get Key) – **Direct RTGS settlement** (counterpart of PP-DLTO-03): `POST /igw/{ncb}/v1/direct-rtgs/xvps` (Initialisation) `GET /igw/{ncb}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION` (Get Key) For brevity, the steps below describe the cash-token flow with explicit parenthetical references to the direct-RTGS endpoint substitutions. The test execution plan should specify which settlement model is being exercised (or, ideally, run the test

EUROPEAN CENTRAL BANK
EUROSYSTEM

	twice – one execution per model). This test case combines two prior test cases: it inherits the **timeout / cancellation oracle behaviour** from PP-MP-05 (auto-creation of a BURNED payment on timeout, selective key disclosure to the Seller side only), and the **MDLTO-on-behalf simplification** from PP-DLTO-02 / PP-DLTO-03 (single actor, single authentication, dual Whitelisting). A subtle and important authorisation symmetry is exercised: – the MDLTO uses its **MP-S Whitelisting** for the Initialisation (Step 2) – here the MDLTO acts on the Seller side; – the MDLTO would have used its **MP-B Whitelisting** for the Payment (deliberately skipped in this test); – the MDLTO uses its **MP-S Whitelisting again** for the cancellation-key retrieval (Step 4) – here the MDLTO is back on the Seller side, since cancellation key is released only to the Seller (per SDD §3.5.6.d). This is consistent with the OpenAPI authorisation rules on the Get-Payment-Status endpoints, which list *DedicatedCashWalletOwner* (i.e. the Seller-side MP) and *Whitelisted Market DLT Operator* as authorised actors. The MDLTO whitelisted for MP-S is therefore the entitled actor on the cancellation-key side. The functional perimeter is the **cash leg only**. The Hash Link Contract on the Market DLT platform (locking, reclaiming the asset using the cancellation key) is **outside the scope of the Pilot solution** (SDD §3.5.6.e). The test case may be run as a **DvP** (type = “DVP”) or a **PvP** (type = “PVP”); the default execution is DVP. **A2A** *Step 0 – Authentication (MDLTO) – METHOD: mTLS handshake + IAM token issuance – Code: 200:* 0. The MDLTO’s application authenticates to the Pilot solution IAM using the X.509 certificate of its A2A External User (MDLTO-EU) over mTLS and obtains a JWT (per SDD §3.2.1 and §6.3). A **single authentication** is sufficient for the entire flow. *Step 1 – (Optional) Verify whitelisting for MP-S – METHOD: GET – Code: 200:* 1. As a defensive pre-check, the MDLTO’s application may verify that it is currently authorised to instruct on behalf of MP-S: GET /dlt/{ncb-of-MP-S}/api/bridge/whitelist/verify?participantBIC=<MP-S BIC11>&MDLTOperatorID=<MDLTO-ID> Returns HTTP 200 OK with isAuthorized = true.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Note: the MP-B Whitelisting verification is **not** strictly required for this test, since no Payment call is made; however the precondition (MP-B Whitelisting in place) should still be satisfied to make the test scenario realistic (the MDLTO would not knowingly initialise an XvP it cannot pay for). *Step 2 – XvP Initialisation on behalf of MP-S – METHOD: POST – Code: 200:* 2. The MDLTO's application sends an HTTP `POST` request to the XvP initialisation endpoint, **choosing the settlement model**: **Cash-token model:** `POST /igw/{ncb-of-MP-S}/v1/xvps` Request body (schema `XvPInitRequest`): `json` { "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLTO-ID>", "cashWalletAlias": "<MP-S DCW alias>" }, "buyer": { "bic": "<MP-B BIC11>" }, "amount": "10000.50", "currency": "EUR", "type": "DVP" } **OR direct-RTGS model:** `POST /igw/{ncb-of-MP-S}/v1/direct-rtgs/xvps` Request body (schema `RTGSXvPInitRequest`): `json` { "seller": { "bic": "<MP-S BIC11>", "marketDLTOperator": "<MDLTO-ID>" }, "buyer": { "bic": "<MP-B BIC11>" }, "amount": "10000.50", "currency": "EUR", "type": "DVP" } Headers in either case: `Authorization: Bearer <JWT of MDLTO-EU>`, `Content-Type: application/json`. Field rule: `seller.marketDLTOperator` must match the MP-S Whitelisting record. 3. The Pilot solution returns HTTP `200 OK` with the corresponding `XvPInitResponse` or `RTGSXvPInitResponse`: `xvpTransactionId`, `executionHash`, `cancellationHash`, `timeout`, and the echo of the economic terms. 4. The MDLTO records `xvpTransactionId`, `cancellationHash` and `timeout`. The **timeout** value is the critical data point for this test case – it determines when the MDLTO must wait until before retrieving the cancellation key. The `cancellationHash` is the value the MDLTO will use to set up the Hash Link Contract on the Market DLT platform (out of scope). *Step 3 – Deliberate inaction – no API call:* 5. **The MDLTO deliberately does not submit the Payment**. The test waits until the wall-clock time strictly exceeds the `timeout` value returned in Step 2. 6. During the waiting period, no calls are made on the XvP transaction.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	Step 3-bis – (Optional verification) Late Payment attempt by the MDLTO – METHOD: POST – Code: 4xx (rejection expected): 7. **Optional**: after the timeout has passed, the MDLTO may send a late Payment request as a negative verification: **Cash-token:** POST /igw/{ncb-of-MP-B}/v1/xvps/{xvpTransactionId}/payment with PaymentRequest body. **Direct-RTGS:** POST /igw/{ncb-of-MP-B}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment with RTGSPaymentRequest body (including signature and signerPEM). 8. The Pilot solution **rejects** the late Payment request with HTTP 4xx (most likely 409 Conflict per SDD §3.5.6.c: *"if the timeout for submitting the payment request for the DvP has been already reached … the request is rejected"*). The error response body carries a business error code identifying the timeout-expired condition. *Step 4 – Get Key & Payment Info – MDLTO retrieves cancellation key on behalf of MP-S – METHOD: GET – Code: 200:* 9. **After the XvP timeout has expired**, the MDLTO's application sends an HTTP GET request to the Get Key & Payment Info endpoint, on behalf of the Seller (MP-S), requesting the CANCELLATION key: **Cash-token:** GET /igw/{ncb-of-MP-S}/v1/xvps/{xvpTransactionId}/payment?key=CANCELLATION **Direct-RTGS:** GET /igw/{ncb-of-MP-S}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=CANCELLATION Headers: Authorization: Bearer <JWT of MDLTO-EU>. Important: the {ncb} is **MP-S's NCB** (since this call is on behalf of the Seller side). The MDLTO's authority here derives from its **MP-S Whitelisting**, not its MP-B one (per OpenAPI v1.0 *"Authorized Actors: DedicatedCashWalletOwner; Whitelisted Market DLT Operator"* on the Get-Payment-Status endpoint). 10. The Pilot solution returns HTTP 200 OK with payload (schema PaymentResponse): • xvpTransactionId (echo); • payment.id (UUID of the cash-token or direct-RTGS payment automatically created by the Pilot solution on timeout); • payment.status = "BURNED" (cash-token) or equivalent status; • payment.reason (status reason text identifying timeout expiry); • executionKey = null (not released, since no settlement occurred); • cancellationKey (64–128 hex chars) – **returned to the MDLTO acting on behalf of the Seller**, since the MDLTO whitelisted for MP-S is the entitled actor (per SDD §3.5.6.d and the OpenAPI authorisation rule).

EUROPEAN CENTRAL BANK
EUROSYSTEM

	11. The MDLTO records the returned cancellationKey. This is the secret the MDLTO will use to reclaim the asset locked in the HLC on the Market DLT platform (out of scope of this test case). The MDLTO may verify off-Pilot that SHA-256(cancellationKey) == cancellationHash returned in Step 2. *Step 5 – (Optional) MDLTO attempts to retrieve execution key on behalf of MP-B – METHOD: GET – Code: 200 with executionKey = null:* 12. As a negative cross-check, the MDLTO may also call the Get Key endpoint on behalf of MP-B (using its MP-B Whitelisting) with key=EXECUTION: **Cash-token:** GET /igw/{ncb-of-MP-B}/v1/xvps/{xvpTransactionId}/payment?key=EXECUTION **Direct-RTGS:** GET /igw/{ncb-of-MP-B}/v1/direct-rtgs/xvps/{xvpTransactionId}/payment?key=EXECUTION 13. The Pilot solution returns HTTP 200 OK with payment.status = "BURNED" and executionKey = null (the Buyer side is correctly **not** entitled to receive any key when the cash leg is BURNED).
Preconditions/Details	• **MP-S** and **MP-B** are duly onboarded Market Participants (per PP-NCB-01) with status *Active*, each under its responsible NCB (same or different NCBs – cross-NCB is a valid variant, but requires additional steps not described here, e.g. MDLTO must use two different users and certificates). • For the **cash-token** execution: each MP holds at least one active DCW. **MP-B's DCW funding is not required** for this test (no payment is submitted); the test specifically targets the *time-based* failure (BURNED), not the *insufficient-funds* failure (UNSETTLED). • For the **direct-RTGS** execution: each MP holds an active T2 RTGS DCA with a valid *T2 account reference* in the LRDM; the direct debit mandate on MP-B's DCA may be in place or absent (irrelevant since no debit is attempted). • **MDLTO** is duly configured in the Pilot solution as an external participant (per SDD §2.1.5 / §3.4.6) with at least one A2A user with the **External User** profile. • **Two distinct Whitelisting records** are configured for the MDLTO in the LRDM, both within their VALID FROM / VALID TO validity period: – one with PARTICIPANT BIC = <MP-S BIC11> – **strictly required** for both the Initialisation (Step 2) and the cancellation-key retrieval (Step 4); – one with PARTICIPANT BIC = <MP-B BIC11> – **required for realism** (a coherent test scenario where the MDLTO theoretically could have paid but chose not to); strictly needed only for the optional Step 3-bis (late Payment rejection check) and Step 5 (negative cross-check on the Buyer side). • Both Whitelistings reference the same MARKET DLT OPERATOR ID = <MDLTO-ID> and the same MARKET DLT PLATFORM ID.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• Neither MP-S, MP-B nor MDLTO is **blocked** in the Pilot solution. • The MDLTO disposes of a valid (non-expired) X.509 certificate installed for mTLS authentication. • **No PoA configuration is required** – the MDLTO's authority derives from the Whitelistings. • **The XvP timeout (system parameter, per SDD §3.5.6.a) is known and short enough to be exercisable within a single test session**. The test execution plan should reserve a wall-clock duration of at least the XvP timeout value plus a safety margin. • The test is executed within an open Business window, with the XvP timeout falling inside the same Business window (otherwise a cross-window variant is needed). • **Settlement model selection**: the test execution plan must declare which model (cash-token or direct-RTGS) is being exercised in each test run. Ideally, **both runs** should be performed to cover both endpoint families.
Expected results	**A2A** • **Step 0 (Authentication):** mTLS handshake succeeds and the Pilot IAM issues a single valid JWT to MDLTO-EU. • **Step 1 (Optional whitelist verification):** HTTP 200 OK with isAuthorized = true for the MP-S Whitelisting. • **Step 2 (Initialisation on behalf of MP-S):** HTTP 200 OK. Response body contains a new xvpTransactionId, valid executionHash and cancellationHash, a timeout in UTC ISO-8601, and the echo of the economic terms. **No payment exists yet** in the corresponding ledger (ESY-DLT for cash-token; no T2 RTGS instruction submitted yet for direct-RTGS). Instructing Entity ID on the XvP record is **<MDLTO-ID>**. • **Step 3 (Deliberate inaction):** No API call. The wall-clock time crosses the timeout returned in Step 2. The Pilot solution automatically creates a payment for the XvP cash leg with status BURNED (per SDD §3.5.6 *State Diagram*). • **Step 3-bis (Optional late Payment):** HTTP 4xx (rejection). The Pilot solution rejects the late Payment with a business-error code HL-DVP-007 identifying the timeout-expired condition ; **no payment is created from this rejected request** (the only payment is the auto-BURNED one). • **Step 4 (Get Key – MDLTO on behalf of MP-S, after timeout):** HTTP 200 OK. Response contains payment.status = "BURNED" (API-level), payment.reason identifying the timeout, executionKey = null, and a **non-null cancellationKey** (64–128 lower-case hex chars). The cancellation key correctly hashes to the cancellationHash returned in Step 2 (off-Pilot verification). The Instructing Entity ID on the auto-BURNED payment record is the Pilot solution itself. • **Step 5 (Get Key – MDLTO on behalf of MP-B, optional negative cross-check):** HTTP 200 OK. Response contains payment.status = "BURNED" and executionKey = null.

EUROPEAN CENTRAL BANK
EUROSYSTEM

	• **No DCW / DCA movement**: – Cash-token: neither MP-B's DCW nor MP-S's DCW shows any debit or credit related to this XvP. – Direct-RTGS: neither MP-B's nor MP-S's T2 RTGS DCA shows any pacs.010 / pacs.009 settlement; no T2 RTGS message is generated by the Pilot solution for this XvP (settlement leg was never triggered). • **No transaction visible as Settled / Completed**: the XvP appears in the relevant Transactions / Direct RTGS Payments extraction with status reflecting the failure. • Authorisation enforcement is correctly applied: the MDLTO (whitelisted for MP-S) receives the cancellationKey on Step 4; the same MDLTO (whitelisted for MP-B) does not receive the executionKey on Step 5; a hypothetical MDLTO whitelisted only for one side would be rejected on the other side (out of scope of this happy-unhappy path test case but a candidate for a sibling negative test).
Test evidence	**A2A** • Copy of the **JWT issuance** exchange (request and response) with the Pilot IAM for MDLTO-EU. • (Optional) Copy of the HTTP request and response for Step 1 – GET /dlt/{ncb}/api/bridge/whitelist/verify?... – evidencing isAuthorized = true for MP-S. • Copy of the **HTTP request and response** for Step 2 – POST /igw/{ncb}/v1/xvps (cash-token) or POST /igw/{ncb}/v1/direct-rtgs/xvps (direct-RTGS) – evidencing the returned xvpTransactionId, cancellationHash, and **timeout value** (critical for this test). • **Wall-clock evidence** demonstrating that no Payment was submitted before timeout: a timestamped test-execution log showing the timeout value, the wall-clock time of Step 4, and the absence of any POST …/payment call between Step 2 and Step 4 (e.g. proxy / API-gateway log extract). • (Optional) Copy of the HTTP request and response for Step 3-bis – the late POST …/payment – evidencing the 4xx rejection and the business-error code. • Copy of the HTTP request and response for Step 4 – GET …/payment?key=CANCELLATION – evidencing payment.status = "BURNED" and the cancellationKey returned to the MDLTO. • (Optional) Copy of the HTTP request and response for Step 5 – GET …/payment?key=EXECUTION on the Buyer side – evidencing executionKey = null. • Excel export of the relevant Transactions / Direct RTGS Payments extraction filtered on the xvpTransactionId, evidencing the auto-BURNED payment record and the absence of any cash movement. • (Optional, U2A cross-check) – Cash-token: screenshot of the **Transactions** tab on MP-S and MP-B sides showing the XvP cash leg in its failed final status; screenshots of the *Dedicated Cash Wallets* tab showing

EUROPEAN CENTRAL BANK
EUROSYSTEM

	unchanged balances on both MPs. – Direct-RTGS: screenshot of the *Direct RTGS Payments* monitoring tab showing the failed Direct RTGS Payment-XvP record.
Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.1.5 *Market DLT Operators*; §2.4.5 *External Users*; §3.2.1 *Authentication & Authorisation*; §3.3 *Non-Repudiation of Origin (NRO)* (not exercised here, but referenced for completeness); §3.4.6 *Market DLT Operators*; §3.4.7 *Whitelists*; §3.5.6 *DvP & PvP* – including the *State Diagram of Payment status* (fig. i), *Relationship between Timeout and Payment* (fig. ii), *Relationship between payment status and keys disclosure* (fig. iii); §3.5.6.a *Initialisation*; §3.5.6.c *Payment* (timeout-expired rejection rule); §3.5.6.d *Get Key and Payment Info* (cancellation-key release condition: *"returned only if Payment status is UNSETTLED or BURNED and the Actor is the Seller"*); §3.5.6.e *Initialisation on Market DLT platform* (HLC cancellation, out of scope); §3.5.6 *Key Management flows for Market DLT Operator* (fig. xii–xvi); §3.5.6 *Availability of Execution and Cancellation Key*; §3.5.7 *Unhappy Paths & Error codes*; §3.6.2 *Reporting and data extraction*; Table R *List of features available to market DLT Operators*. • PONTES Pilot – OpenAPI Document v1.0: tags *XvP – Cash Token* and *XvP – Direct RTGS* – operations `createXvP` / `RTGSCreateXvP` (Initialisation), `payment` / `RTGSPayment` (rejected late-Payment path), `getPaymentStatus` / `RTGSGetPaymentStatus` (cancellation-key retrieval). The Authorisation rules on the Get-Payment-Status endpoints (*"Authorized Actors: DedicatedCashWalletOwner; Whitelisted Market DLT Operator"*) explicitly enable the MDLTO-on-behalf-of-Seller route for cancellation-key retrieval; schemas `PaymentResponse` (note: `cancellationKey` is *"returned only to Seller; returned only if Payment is in status UNSETTLED or BURNED"*); enum `PaymentStatus` (`SETTLED` / `UNSETTLED` / `PENDING` / `BURNED`); enum `KeyType` (`EXECUTION` / `CANCELLATION`). • PONTES Pilot – User Handbook v1.0: §3.2.5 *Transactions* (U2A cross-check, cash-token); §3.2.7 *Direct RTGS Payments* (U2A cross-check, direct-RTGS); §4.11 *Monitoring*. • Related test cases: PP-MP-05 (the MP-self-instructed cash-token counterpart – the present test case is its MDLTO-on-behalf-of-both extension); PP-DLTO-02 / PP-DLTO-03 (the happy-path counterparts for cash-token / direct-RTGS); PP-NCB-02 (configures the MDLTO and its two Whitelistings – mandatory prerequisite).
Related profiles/privileges	• User profile (MDLTO): **External User** – the only profile authorised for the MDLTO A2A user.

EUROPEAN CENTRAL BANK
EUROSYSTEM

PP-DLTO-05 – Perform direct RTGS payment on behalf of a Market Participant to another Market Participant in another NCB node (A2A)

Field	Description
Test Case ID	PP-DLTO-05
Test case name	Perform direct RTGS payment on behalf of a Market Participant to another Market Participant
Relevant for (actor)	**Market DLT Operator (MDLTO)**, acting on behalf of a debited Market Participant (MP-A) and instructing a direct RTGS payment to a credited Market Participant (MP-B)
Input mode	A2A
Mandatory	Yes
Detailed Description	This test case verifies the ability of a Market DLT Operator (MDLTO) to instruct a **Direct RTGS Payment** on behalf of a whitelisted Market Participant (**MP-A**) towards another Market Participant (**MP-B**). The payment is settled through **T2 RTGS** rather than through cash tokens. The scenario validates, MDLTO whitelisting controls, direct settlement processing, transaction visibility and reporting. **A2A** **Step 1 – Authentication (MDLTO External User)** 1. The MDLTO application authenticates to the Pilot IAM using its X.509 certificate over mTLS. 2. The Pilot IAM issues a valid JWT access token. **Step 2 – Verify participant eligibility** 3. Verify that MP-A is whitelisted for the MDLTO under NCB-A. 4. Verify that MP-A owns a valid T2 RTGS Account Reference. 5. Verify that MP-B owns a valid T2 RTGS Account Reference. 6. Verify that neither Market Participant nor MDLTO is blocked. **Step 3 – Submit Direct RTGS Payment** 7. The MDLTO application sends an HTTP POST request to the Direct RTGS Payment endpoint: POST /dlt/{ncb}/api/bridge/direct-rtgs/payments

EUROPEAN CENTRAL BANK
EUROSYSTEM

	where {ncb} corresponds to the NCB of MP-A (payer Market Participant). 8. Headers include: • Authorization: Bearer <JWT> • Content-Type: application/json 9. The request payload contains the following fields: • **id**: unique payment identifier (UUID) • **correlationId**: unique business correlation identifier used for end-to-end tracking and reconciliation • **amount**: payment amount (e.g. EUR 10,000.00) • **currency**: EUR • **payerBank**: BIC11 of MP-A • **receiverBank**: BIC11 of MP-B • **signature**: Base64-encoded ECDSA signature • **signerPEM**: PEM-encoded public certificate used to verify the signature 10. The MDLTO application generates the signature using the private key associated with its External User certificate. The signature is calculated over the SHA-256 hash of the concatenation: id + amount + payerBank + receiverBank 11. The MDLTO identity is derived from the authenticated External User session (mTLS certificate and JWT token); therefore no separate Market DLT Operator identifier is included in the request payload. 12. The Pilot solution validates: • MDLTO authentication and authorisation as External User. • Valid whitelist relationship between the MDLTO and MP-A. • Signature integrity and signer certificate validity. • Existence of valid RTGS account references for payer and receiver. • Participant, wallet and MDLTO status. • Business date and settlement-window rules. **Step 4 – RTGS Settlement Processing** 13. The Pilot solution generates the corresponding T2 RTGS settlement instructions. 14. The cash leg is executed through T2 RTGS between the RTGS DCA linked to MP-A and the RTGS DCA linked to MP-

	B. 15. The payment progresses through the statuses: **INITIALIZED → ACCEPTED → PAYMENT_SUBMITTED → COMPLETED** (per SDD §3.6.2 Direct RTGS Payments status list: Initialized / Accepted / Payment submitted / Completed / Failed / Cancelled). Note: the synchronous response of the bridge endpoint returns "status": "SETTLED" at submission time, while the payment instruction reaches its final status COMPLETED a few seconds later. **Step 5 – Verify settlement** 16. Query the Direct RTGS Payment using the returned Payment ID or Correlation ID. 17. Verify the following attributes: • Status = SETTLED. • Payer Bank = MP-A BIC. • Receiver Bank = MP-B BIC. • Settlement Timestamp populated. • Correlation ID populated. • Cross-NCB routing information correctly populated. **Step 6 – Cross-participant visibility verification** 18. Verify that MP-A can view the transaction as payer. 19. Verify that MP-B can view the transaction as receiver. 20. Verify that reporting and extraction services reflect the settled payment.
Preconditions/Details	• MP-A has been onboarded and configured with a valid T2 RTGS Account Reference. No Dedicated Cash Wallet is required: the cash leg is settled directly on the T2 RTGS DCAs, not on DCWs. • MP-B has been onboarded and configured with a valid T2 RTGS Account Reference. No Dedicated Cash Wallet is required. • The MDLTO is configured and approved in the Pilot solution (PP-NCB-02). • MP-A is whitelisted for the MDLTO. • All participants and the MDLTO are in Unblocked status. • Sufficient liquidity exists on MP-A's RTGS DCA to cover the payment amount. • A valid Direct Debit Mandate and RTGS linkage exist for MP-A. • The MDLTO External User holds a valid certificate and JWT

EUROPEAN CENTRAL BANK
EUROSYSTEM

	can be obtained successfully. • The test is executed during the T2 RTGS settlement window.
Expected results	**A2A** • The MDLTO successfully authenticates using mTLS and JWT. • The Direct RTGS Payment is accepted by the Pilot solution. • The signature and signer certificate are successfully validated. • T2 RTGS settlement is successfully completed. • Payment status reaches **COMPLETED** . • MP-A's RTGS DCA is debited by the payment amount. • MP-B's RTGS DCA is credited by the payment amount. • The transaction is recorded with the authenticated Market DLT Operator as the **Instructing Entity**, derived from the External User identity used during mTLS/JWT authentication. • Settlement Timestamp and Correlation ID are populated. • The transaction is visible to MP-A, MP-B, and involved NCB(s) according to their respective scopes. • The payment appears in the Direct RTGS Payments extraction and monitoring views.
Test evidence	**A2A** • Evidence of successful mTLS authentication and JWT issuance. • Copy of the Direct RTGS Payment request payload, including id, amount, payerBank, receiverBank, signature and signerPEM. • Copy of the successful API response containing Payment ID and/or Correlation ID. • Evidence that the payment progressed through INITIALIZED, ACCEPTED, PAYMENT_SUBMITTED and COMPLETED statuses. • Evidence of Settlement Timestamp and Correlation ID. • Screenshot or extract from NCB-A monitoring/reporting view showing the transaction. • Screenshot or extract from MP-A transaction view showing the payment. • Screenshot or extract from MP-B transaction view showing the payment. • Extract from the Direct RTGS Payments reporting/export facility filtered on the relevant Payment ID or Correlation ID.

EUROPEAN CENTRAL BANK
EUROSYSTEM

Relevant Documentation	• PONTES Pilot – Service Description v1.0: §2.2.7 External entity instructing on behalf; §3.3 Direct Settlement; §3.4.3 T2 Account References; §3.5.4 Direct T2 RTGS Payment instructed by Market DLT Operator on behalf of Market Participant; §3.6.2 Reporting and data extraction. • PONTES Pilot – OpenAPI Specification v1.0: Direct RTGS Payment endpoint /dlt/{ncb}/api/bridge/direct-rtgs/payments, request schema and signature validation requirements. • PONTES Pilot – User Handbook v1.0: Transactions, Monitoring and Reporting sections.
Related profiles/privileges	• **External User** profile for the MDLTO A2A user. • Valid whitelist association between MDLTO and MP-A.

APPENDIX – Signing Rules:

Signing Rules and Step-by-Step Example (please refer to the OpenAPI specifications for further information):

Signing Rules:

- The fields shall be concatenated as a string in the following order:

  1. "xvpTransactionId", once transformed as described below

  2. "amount"

  3. "buyer bic"

  4. "seller bic"

- No separator, delimiter, or any additional character shall be inserted between concatenated values

- The SHA256 hash shall be computed on the resulting concatenated string

- The resulting hash shall then be signed using the user private key

- The generated signature shall be provided in the "signature" field of the request payload

EUROPEAN CENTRAL BANK
EUROSYSTEM

Step-by-Step Example:

Based on the following payload values:

- "xvpTransactionId": "517ae232-29e7-4efb-8743-0177bbe6d577"
- "amount": "10000.50"
- "buyer": {"bic": "BEUMFXB1XXX"}
- "seller": {"bic": "BEUMFXB1XX2"}

Step 1 - Concatenate the fields:

The "xvpTransactionId" needs to be transformed before concatenation by removing "-" characters and adding "xvp" prefix:

517ae232-29e7-4efb-8743-0177bbe6d577 = xvp517ae23229e74efb87430177bbe6d577

The concatenated string to be hashed is:

xvp517ae23229e74efb87430177bbe6d57710000.50BEUMFXB1XXXBEUMFXB1XX2

Step 2 - Generate the SHA256 hash:

The generated hash to be signed is:

a7e717ea9fae0d970d60b47397858084e7659dc29a426b53eb04ef35c40a399c

EUROPEAN CENTRAL BANK
EUROSYSTEM

Step 3 - Sign the hash:

The SHA256 hash shall then be signed using the relevant user private key and applying the SHA256withECDSA algorithm.

The signing operation can be performed using different technical means, including command lines (e.g. OpenSSL), application code (e.g. NOdeJS), or scripts (e.g. bash).

The generated signature is:

MEYCIQDstSFJUr9CpSXFVCj0aSCMpsIH3IfGwKysE5jIZUTWlgIhAJY5Tsglo6kaCbAHMNY
szfO80rCqrLdLPzkwzMsCVSg1

Step 4 - Populate the payload:

The generated signature shall be inserted into the "signature" of the request payload:

"signature":
"MEYCIQDstSFJUr9CpSXFVCj0aSCMpsIH3IfGwKysE5jIZUTWlgIhAJY5Tsglo6kaCbAHMN
YszfO80rCqrLdLPzkwzMsCVSg1"

Response Semantics

- Returns confirmation that the new Direct RTGS XvP Payment has been settled and the corresponding "executionKey"