fairsfair Initial Body of Work

Reference API

Edition v1.0 · Published

Series	Reference
Source	fairsfair Reference API
Technical contract	FF-R02 Reference API OpenAPI v1.0
Licence	FF-L02 — fairsfair Licence Agreement
Owner	FRCB B.V.
Custodian	fairsfair foundation

Copyright © 2026 FRCB B.V.

Maintained under the custodianship of fairsfair foundation

fairsfair.org · info@fairsfair.org

1. Purpose and scope

The Reference API defines the operations through which the operational representation of an Agreement is brought into execution as a fairsfair Transaction, coordinated through its TransactionProcess and made available for authorised inspection.

TransactionInitiation establishes the Transaction under its TransactionID, organises the TransactionActivities in the proposed order and makes the first TransactionActivity the state of play. Participants obtain AccessTokens within the scope recorded in the Authorisation Register. As the Transaction progresses, CommitInput and CommitOutput constitute Formal Expressions and Actions. GetOngoingTransactions and RetrieveTransaction provide the applicable representations of the Transaction.

The Reference API retains the operational record constituted through that progression, including the TransactionProcess, Authorisation Register, Formal Expressions and Actions. Formal Expressions communicate the availability and SourceLocations of Input and Output for authorised access. The Reference API does not receive, transfer or store their Material Expressions.

The Reference API does not create or assess the Agreement, determine or verify rights, obligations, agency or Constitutional Capabilities, or determine whether the proposed TransactionProcess and supplied Authorisation correctly reflect their source. It does not process Input or Output or prescribe how participating systems are built.

2. Operation map

The Reference API consists of seven operations:

1. **HealthCheck** confirms that the service is available.
2. **TransactionInitiation** brings the operational representation of an Agreement into execution as a Transaction under its TransactionID.
3. **IssueAccessToken** authenticates the requester and issues an AccessToken limited to its authorised operational scope within a Transaction.
4. **GetOngoingTransactions** lists the ongoing Transactions in which an authenticated requester is authorised to participate or act.
5. **RetrieveTransaction** returns the authorised representation of a specific Transaction.
6. **CommitInput** constitutes the Formal Expression of the Input for a TransactionActivity. Where the Input is not empty, it communicates the SourceLocation of its Material Expression.
7. **CommitOutput** constitutes the Formal Expression of the Output, completes the TransactionActivity and progresses the TransactionProcess.

These operations coordinate and account for the Transaction without receiving, transferring or storing its Material Expressions.

3. HealthCheck

Purpose

HealthCheck confirms that the participating service is available and able to respond.

Precondition

None.

Request

A health-check request is submitted.

Response

The service returns its availability status and may include non-sensitive implementation-specific diagnostic information.

Postcondition

Nothing changes in any Transaction.

4. TransactionInitiation

Purpose

TransactionInitiation brings the operational representation of an Agreement into execution as a Transaction. It is not a TransactionActivity.

Precondition

The Agreement has established the rights, obligations and agency from which the Transaction arises.

Request

The requester provides the proposed TransactionProcess and the Authorisation Register.

The TransactionProcess organises the TransactionActivities in the proposed order. The Authorisation Register reflects the supplied rights, obligations and agency by relating the applicable Constitutional Roles, Data Roles and authorised Actions to those TransactionActivities.

A SourceID is included where the Transaction originates from an identifiable Agreement.

Response

The service returns the TransactionID under which the Transaction and its operational record are constituted.

Postcondition

The TransactionProcess is in execution. The first TransactionActivity becomes the state of play, and its participants are informed.

5. IssueAccessToken

Purpose

IssueAccessToken issues an AccessToken establishing which authorised Actions the authenticated requester may perform within the operational scope of a TransactionActivity.

Precondition

The requester satisfies the applicable authentication requirements and is recorded in the Authorisation Register as participating directly or acting under delegated authority. CommitInput and CommitOutput authority ends with the applicable TransactionActivity period; retrieval authority ends with the relevant Formal Expression availability period.

Request

The requester identifies the Transaction and the operational scope for which access is requested.

A legal entity is represented by its EntityID. A natural person’s participation is related under the TransactionID without requiring a persistent fairsfair identifier.

Response

The service relates the request to the Authorisation Register and returns an AccessToken limited to the recorded Actions, TransactionActivity, purpose and duration. Its duration does not extend beyond the

applicable TransactionActivity period for CommitInput or CommitOutput, or the relevant Formal Expression availability period for retrieval.

Postcondition

The requester can use the AccessToken within that operational scope. Nothing changes in the Transaction.

6. GetOngoingTransactions

Purpose

GetOngoingTransactions lists the ongoing Transactions in which an authenticated requester is authorised to participate or act.

Precondition

The requester satisfies the applicable authentication requirements and is represented by its EntityID.

Request

The requester asks for the ongoing Transactions within its authorised participation or operational scope.

Response

The service checks each ongoing Transaction’s Authorisation Register and returns only the Transactions in which the requester is authorised to participate or act.

For each returned Transaction, the representation includes the TransactionID, complete TransactionProcess, state of play and SourceID where supplied.

Postcondition

Nothing changes in any Transaction.

7. RetrieveTransaction

Purpose

RetrieveTransaction returns the authorised representation of a specific Transaction.

Precondition

The requester satisfies the applicable authentication requirements and presents a valid AccessToken for the Transaction.

Request

The requester identifies the Transaction by its TransactionID.

Response

The service returns the TransactionID, complete TransactionProcess, state of play and SourceID where supplied.

It also returns the Formal Expressions, Actions and Authorisation available within the requester’s authorised operational scope. It does not return a Material Expression.

Postcondition

Nothing changes in the Transaction.

8. CommitInput

Purpose

CommitInput constitutes the Formal Expression of an Input contribution for a TransactionActivity.

Precondition

The TransactionActivity is the state of play. The requester is authenticated, presents a valid AccessToken and is authorised to commit Input from the applicable source.

Request

The requester provides the TransactionID and TransactionActivity. Where a Material Expression exists, it provides its SourceLocation.

Several independently governed sources perform CommitInput separately. Several datasets organised at one SourceLocation are committed together.

Response

The service confirms that the Formal Expression and Action have been constituted and informs the authorised participants.

Postcondition

The Input contribution is formally committed. Where a SourceLocation is provided, its Material Expression is available for authorised access. The TransactionActivity remains the state of play.

9. CommitOutput

Purpose

CommitOutput communicates that Output is available and completes the TransactionActivity.

Precondition

The TransactionActivity is the state of play. At least one CommitInput Action has been constituted for that TransactionActivity. The requester is authenticated, presents a valid AccessToken and is authorised to commit Output.

Request

The requester provides the TransactionID, TransactionActivity and SourceLocation of the Output’s Material Expression.

Response

The service confirms that the Formal Expression and Action have been constituted and informs the authorised participants.

Postcondition

The TransactionActivity is complete. Its Output is available for authorised access, and access to its Input for performing that activity ends.

The next TransactionActivity—if any—becomes the state of play. Where the completed activity is final, no TransactionActivity remains the state of play.

Part II — API Objects

10. AccessToken

Purpose

An AccessToken proves the operational scope within which the authenticated requester may act through the Reference API.

Constitution

It identifies the Transaction, the applicable TransactionActivity, whether the Authorisation concerns Input or Output, the authorised Actions, the purpose and duration of access, and any delegated agency under which those Actions may be performed. Its duration cannot exceed the applicable TransactionActivity period for a commitment or the relevant Formal Expression availability period for retrieval.

Use

The requester presents the AccessToken when performing a protected API operation. The receiving service validates the requested Action and TransactionActivity against its authorised scope.

Effect

The receiving service can determine whether the requested Action may be performed. Where a Delegated Agent acts, the Principal's Constitutional Role, rights, obligations and responsibility remain with the Principal.

11. Transaction

Purpose

The Transaction is the operational representation of an Agreement brought into execution under its TransactionID.

Constitution

It contains the TransactionProcess, Authorisation Register and state of play. As the Transaction is performed, its constituted Formal Expressions and Actions form part of its operational record. It may include the SourceID identifying the originating Agreement.

It records when the Transaction was initiated, the default Transaction Process time scope and the availability of its Formal Expressions. Where an Activity is planned outside the default scope, it records that Activity's applicable period and the corresponding Formal Expression availability period.

Use

Its authorised representation is returned by GetOngoingTransactions and RetrieveTransaction, limited to the requester's operational scope.

Effect

The Transaction provides the dossier through which its execution can be coordinated, recovered and accounted for without receiving or storing its Material Expressions.

12. TransactionID

Purpose

The TransactionID identifies the Transaction across all participating systems.

Constitution

It is unique and remains unchanged throughout the TransactionProcess.

Use

It relates the Transaction, its TransactionActivities, their Input and Output, SourceLocations, Formal Expressions and constituted Actions.

For a natural person, it also provides the Transaction context under which participation is related without creating a persistent fairsfair identifier for that person.

Effect

Participating systems can recognise that their authorised representations, information and Actions belong to the same Transaction.

13. TransactionProcess

Purpose

The TransactionProcess arranges the TransactionActivities in the proposed sequence through which the Transaction is performed.

Constitution

It consists of the TransactionActivities in the proposed order. Unless an applicable Agreement states otherwise, the Transaction Process has a time scope of 24 hours from TransactionInitiation. Its Formal Expressions remain available for authorised retrieval for a further 48 hours.

Use

It places each TransactionActivity and its operational scope in the context of the Transaction. Where an Activity is planned outside the default Transaction Process time scope, its applicable period and the corresponding availability period for its relevant Formal Expressions are recorded. That context contributes to determining the Actions required within the TransactionActivity and establishes which TransactionActivity becomes the state of play after the preceding TransactionActivity is executed.

Effect

All participating systems follow the same agreed sequence while performing the Actions required within each TransactionActivity.

14. TransactionActivity

Purpose

A TransactionActivity is the state of play in the TransactionProcess while it is being performed.

Constitution

Its name establishes its operational scope. Where it is planned outside the default Transaction Process time scope, its applicable period establishes when it may be performed and the corresponding Formal Expression availability period is recorded. Within that scope, it is constituted through the Actions required to transform its Input into Output in the context of the Agreement and its position in the TransactionProcess.

Use

It establishes what is being performed and provides the operational scope against which participation, authorised Actions and the use of Input and Output are validated.

Effect

Once the TransactionActivity is executed, its Output establishes what it achieved. The next TransactionActivity—if any—becomes the state of play, and all its participants are informed.

15. Input

Purpose

Input is the information minimally required to perform the Actions of a TransactionActivity. Where no information is minimally required, Input is empty.

Constitution

Where information is required, it is organised for the TransactionActivity at its SourceLocation. Where Material Expressions are made available from independently governed sources, each remains organised at its own SourceLocation.

Each source formally commits its Input contribution through CommitInput.

Use

The Formal Expression relates Input to the TransactionActivity under the TransactionID and, where Input is available for retrieval, identifies its SourceLocation. Authorised participants are informed that Input has been committed.

Effect

The Actions of the TransactionActivity may be performed after Input has been formally committed, whether Input contains information or is empty.

16. Output

Purpose

Output is the information produced by the Actions of a TransactionActivity.

Constitution

It establishes what the TransactionActivity achieved and contributes to the state of play of the Transaction.

Use

The Formal Expression relates Output to the TransactionActivity under the TransactionID and identifies the SourceLocation at which it is available. Output may serve as Input for a subsequent TransactionActivity where required by the TransactionProcess.

Effect

When Output becomes available, the TransactionActivity is complete. Output remains available at its SourceLocation for authorised retrieval and for as long as required by the Agreement or an applicable obligation.

17. Formal Expression

Purpose

A Formal Expression communicates that Input or Output is available for a TransactionActivity.

Constitution

It relates Input or Output to the TransactionActivity under the TransactionID and identifies its SourceLocation. Where Input is empty, no SourceLocation is required. Where an availability period applies, it records the period during which the Formal Expression remains available for authorised retrieval.

A TransactionActivity may have several Input Formal Expressions where Material Expressions are committed from independently governed sources. Each Formal Expression identifies the SourceLocation of one Material Expression.

Once constituted, the Formal Expression forms part of the Transaction's operational record and carries the applicable availability period where one is recorded.

Use

It provides a content-free notification that Input or Output has been committed and, where a SourceLocation is present, where its Material Expression is available for authorised access. Its availability period concerns only retrieval of the Formal Expression; it does not determine the availability of the Material Expression at its source.

It is returned through RetrieveTransaction where it falls within the requester's authorised operational scope and the applicable Formal Expression availability period.

Effect

Authorised participants can receive and recover the Formal Expression without fairsfair receiving or storing its Material Expression.

18. Material Expression

Purpose

A Material Expression is the information made available as Input or Output.

Constitution

It is the information organised at the SourceLocation identified by its Formal Expression.

Use

It may be read directly or retrieved and processed through federation in accordance with the applicable Data Role and Authorisation.

Effect

Input or Output can be used without passing through the Reference API.

19. SourceLocation

Purpose

A SourceLocation identifies where the Material Expression is available.

Constitution

It is an address or reference that can be resolved for authorised access.

Several SourceLocations may contribute Input to the same TransactionActivity. Each is identified by its own Formal Expression.

Use

The SourceLocation forms part of the Formal Expression together with the TransactionID and TransactionActivity. It enables the Material Expression to be read directly or retrieved and processed through federation in accordance with the applicable Data Role and Authorisation.

Where Input is empty, no SourceLocation is required.

Effect

The Material Expression remains available outside the Reference API at the location identified by its Formal Expression.

20. Authorisation Register

Purpose

The Authorisation Register records the authority against which participation and the performance of Actions within a Transaction are validated.

Constitution

It relates each TransactionActivity to the supplied Authorisation concerning Input or Output. For each Authorisation, it records the applicable Data Role, authorised Actions, Principal's Constitutional Role and any delegated authority.

It records who may perform those Actions. A participating legal entity is represented by its EntityID. A natural person's participation is related to the Transaction under its TransactionID.

Use

The Authorisation Register is used throughout the TransactionProcess to validate participation, access to Input or Output, and the authority to perform an Action within the relevant TransactionActivity.

Effect

Only Actions and access falling within the recorded Authorisation can be accepted. Where a Delegated Agent acts, the Principal's Constitutional Role, rights, obligations and responsibility remain with the Principal.

21. SourceID

Purpose

A SourceID identifies the Agreement from which the Transaction arises, where that Agreement can be identified.

Constitution

It is a reference to the Agreement held outside the Reference API.

Use

It may be provided during TransactionInitiation to preserve the link between the Transaction and its originating Agreement.

Effect

The Transaction can be traced to its constitutional source in accordance with the applicable Authorisation without bringing the Agreement into the Transaction.

Part III — Execution and Accountability

22. Action

Purpose

An Action records an operation accepted through the Reference API concerning the progression of a TransactionActivity.

Constitution

It records the operation performed, the Transaction and TransactionActivity within which it occurred, the applicable role and authority, and when it was accepted.

Where a legal entity performs the Action, it is represented by its EntityID. Where a natural person performs the Action, that participation is related under the TransactionID and the applicable role. Where a Delegated Agent acts, the Action also records the agency under which it acted.

Use

The Action is related to the Transaction and TransactionActivity under the TransactionID and is validated against the Authorisation Register.

Effect

Once constituted, the Action is immutable and forms part of the AuditTrail.

23. AuditTrail

Purpose

The AuditTrail provides the record through which the defining operational facts of the Transaction can be verified.

Constitution

It is constituted by the CommitInput and CommitOutput Actions accepted throughout the TransactionProcess.

Use

It establishes what operation was performed, within which TransactionActivity, under which role and authority, by whom and when. Where a Delegated Agent acted, it also establishes the agency under which the Action was performed.

Effect

The commitment of Input and Output and the resulting progression of the Transaction can be verified without recording or assessing how participants transform Input into Output.

24. Retention and Disposal

Rule

Unless an applicable Agreement states otherwise, the Transaction Process has a time scope of 24 hours from TransactionInitiation. Its Formal Expressions remain available for authorised retrieval for a further 48 hours.

Where one or more TransactionActivities are planned beyond that default period, the TransactionProcess records the applicable period for each such Activity and the corresponding availability period for its relevant Formal Expressions.

Application

During an applicable TransactionActivity period, the permitted Reference Actions remain available according to the state of play and Authorisation. CommitInput and CommitOutput authority ends with that applicable period.

The Transaction representation returns the recorded TransactionProcess, including any applicable Activity period and corresponding Formal Expression availability period.

After the default Transaction Process time scope has ended, or after an Activity's applicable period has ended, no further CommitInput or CommitOutput Action can be constituted for that Activity. IssueAccessToken remains available only for authorised retrieval until the relevant Formal Expression availability period ends.

A completed or expired Transaction is no longer returned by GetOngoingTransactions. Authorised retrieval remains available through RetrieveTransaction only for the relevant Formal Expressions and only while their availability period remains in force.

Participants are responsible for retrieving and preserving the records they require. Disposal is not an operation available to a participant and does not concern Material Expressions held at their SourceLocations.

Where an extended Activity or Formal Expression availability period is required, it is recorded in the TransactionProcess. It is not implied by the continued existence of a Material Expression at its SourceLocation.

Effect

The default Reference lifecycle is a 24-hour Transaction Process time scope followed by a further 48 hours of Formal Expression availability. Any extension is explicit in the recorded applicable Activity period and corresponding Formal Expression availability period.

25. TransactionActivity Execution

Purpose

TransactionActivity Execution performs the Actions required within the operational scope of the TransactionActivity and progresses the TransactionProcess.

Constitution

The TransactionActivity is executed through the authorised Actions that transform its Input into Output.

Use

The Authorisation Register is used to validate each Action and any access to Input or Output within the operational scope of the TransactionActivity.

Effect

When Output becomes available, the TransactionActivity is complete. Access to its Input for the performance of that TransactionActivity ends. Its Output contributes to the state of play and remains available at its SourceLocation in accordance with the applicable Authorisation.

The next TransactionActivity—if any—becomes the state of play, and all its participants are informed.

Part IV — TransactionActivity Types

26. Delivery Identification

Purpose

Delivery Identification establishes the Party entitled to receive the agreed Delivery.

Constitution

It is executed through the authorised Actions required within its operational scope to transform Input into Output establishing that entitlement.

Use

Its Output provides the authorised basis for fulfilling the Delivery Obligation.

Effect

The entitlement to receive the agreed Delivery is established. The TransactionActivity is complete, and access to its Input for performing that TransactionActivity ends.

27. Delivery Fulfilment

Purpose

Delivery Fulfilment performs the agreed Delivery for the Party entitled to receive it.

Constitution

It is executed through the authorised Actions required within its operational scope to transform Input into Output establishing that the agreed Delivery was performed.

Use

Its Output establishes the fact that fulfilment of the Delivery Obligation took place.

Effect

The Delivery Obligation is fulfilled. The TransactionActivity is complete, and access to its Input for performing that TransactionActivity ends.

28. Payment Identification

Purpose

Payment Identification establishes the Party obligated to perform the agreed Payment.

Constitution

It is executed through the authorised Actions required within its operational scope to transform Input into Output establishing the Payment Obligation.

Where required by the agreed payment arrangement and the position of Payment Identification within the TransactionProcess, those Actions may include preauthorisation or reservation of the required amount.

Use

Its Output provides the authorised basis for performing the agreed Payment.

Effect

The obligation to perform the agreed Payment is established. The TransactionActivity is complete, and access to its Input for performing that TransactionActivity ends.

29. Payment Fulfilment

Purpose

Payment Fulfilment performs the agreed Payment for the Party entitled to receive it.

Constitution

It is executed through the authorised Actions required within its operational scope to transform Input into Output establishing that the agreed Payment was performed.

Use

Its Output establishes the fact that the fulfilment of the Payment Obligation took place.

Effect

The Payment Obligation is fulfilled. The TransactionActivity is complete, and access to its Input for performing that TransactionActivity ends.

Part V — Reference API Rules

30. Separation of Expression and Content

Rule

The Reference API communicates through the Formal Expression that Input or Output is available and identifies its SourceLocation. It does not receive, transfer or store the Material Expression.

Where Input is empty, the Formal Expression contains no SourceLocation.

Application

The Formal Expression passes through fairsfair. The Material Expression remains at its SourceLocation and may be read directly or retrieved and processed through federation in accordance with the applicable Data Role and Authorisation.

Effect

Fairsfair can coordinate and account for the Transaction without receiving or knowing its substantive content.

31. Authorised Access

Rule

Access to Input and Output is governed by the Authorisation Register.

Application

Input or Output may be held and organised, read directly, or retrieved and processed only where the applicable Data Role, authorised Actions, purpose and duration permit that use within the relevant TransactionActivity.

A Principal may perform the authorised Actions directly. Where a Delegated Agent acts, it may perform only the Actions authorised on behalf of the Principal.

Effect

Access remains limited to the operational scope and Authorisation established for the TransactionActivity. Delegation does not transfer the Principal’s Constitutional Role, rights, obligations or responsibility.

32. Fairsfair Is Not a Participant

Rule

Fairsfair does not participate in any Transaction.

Application

Fairsfair coordinates authorised Actions and communicates Formal Expressions. It does not act as a Principal, Delegated Agent or Party and holds no rights or obligations arising from the Agreement or Transaction.

Effect

Accountability remains with the participating Principals and any Delegated Agents performing Actions on their behalf. Fairsfair enables their cooperation without becoming part of it.

33. Duration of Access to Input

Rule

Authorisation to perform CommitInput or CommitOutput for a TransactionActivity exists only while that TransactionActivity is the state of play and within its applicable period.

Application

Input may be read directly, retrieved or processed only through the applicable Data Role and for the Actions, purpose and duration authorised for that TransactionActivity. A retrieval-only authority may continue only for the relevant Formal Expression availability period.

When the TransactionActivity is complete or its applicable period ends, authority to commit for that Activity ends. The continued existence or availability of the information at its SourceLocation is not determined by the Reference API.

Effect

Input is accessible through the Transaction only when required for the authorised performance of the relevant TransactionActivity.

34. Retrieval of Output

Rule

Output may be retrieved and processed by a Data Services Provider appointed by the Principal for that purpose, within the applicable Authorisation and the relevant Formal Expression availability period.

Application

The Data Services Provider retrieves the Material Expression from its SourceLocation under the applicable Authorisation. Where it acts on behalf of the Principal, it does so as a Delegated Agent and only for the Actions, purpose and duration established. The Formal Expression may be recovered through RetrieveTransaction only during its recorded availability period.

Output is not retrieved through the Reference API.

Effect

The Principal determines who may retrieve and process its Output. The Material Expression remains outside the Reference API, while its retrieval and processing remain governed by the Authorisation Register.

35. Holding Output

Rule

Output is held and maintained by the Data Holder responsible for its preservation at the applicable SourceLocation.

Application

The Principal determines the authorised arrangement under which the Material Expression is held and maintained. The Data Holder may also perform authorised Actions in another Data Role where that role has been established separately.

Where the Data Holder acts on behalf of the Principal, it does so as a Delegated Agent within the established purpose and duration.

Effect

Output remains available at its SourceLocation for as long as required by the Agreement or an applicable obligation. The Principal’s Constitutional Role, rights, obligations and responsibility remain with the Principal.

36. Transaction Accountability

Rule

Each Action is recorded without recording the Material Expression of the Input or Output.

Application

The Action establishes what operation was performed, within which TransactionActivity, under which role and authority, by whom and when. Where a Delegated Agent acted, it also establishes the agency under which the Action was performed.

The constituted Actions form the AuditTrail.

Effect

The commitment of Input and Output and the resulting progression of the Transaction can be verified without fairsfair receiving or knowing the Material Expressions.

37. Constitutional Privacy

Rule

A natural person participates in a Transaction without requiring a persistent fairsfair identifier.

Application

The natural person’s participation is related under the TransactionID and distinguished through the applicable Constitutional Role. Data Roles and Authorisation determine what may be done with Input or Output within each TransactionActivity.

Fairsfair communicates the Formal Expression but does not receive the Material Expression. Identification or evidence of an attribute is made available as Input only where minimally required for the relevant TransactionActivity.

Effect

The Transaction remains accountable without enabling fairsfair to identify, recognise or profile the natural person across Transactions.

38. Agreement Remains Outside the Reference API

Rule

The Agreement constitutes the Transaction and establishes the rights and reciprocal obligations from which it arises. Where authority is delegated, it also establishes that agency. The Agreement does not form part of the Reference API.

Application

Where the Agreement can be identified, a SourceID may refer to it at its source. The Reference API uses the TransactionProcess and Authorisation Register required to coordinate and account for the Transaction’s execution.

Effect

The Transaction can be executed without the Agreement being received, transferred or stored by fairsfair.

39. Immutability of Actions

Rule

Once constituted, an Action cannot be changed or removed.

Application

An Action remains as constituted. Any subsequent operation that is separately authorised and accepted constitutes its own Action under the same TransactionID and the applicable TransactionActivity. It does not alter or replace an earlier Action.

Effect

What occurred remains traceable. Subsequent accepted operations can be established without rewriting the Transaction’s execution.

40. Timestamping of Actions

Rule

Each Action is timestamped when it is constituted.

Application

The timestamp forms part of the Action related to the Transaction and TransactionActivity under the TransactionID. It records when the operation was accepted and constituted, not when an event evidenced by its Input or Output took place.

Effect

The AuditTrail establishes the sequence in which Actions were constituted without fairsfair receiving or accessing the Material Expressions of the Transaction.

41. Constitution of Actions

Rule

An Action is constituted only when the requested operation has been authenticated where required, validated against the applicable Authorisation and accepted.

Application

A request that is rejected does not constitute an Action and does not change the Transaction or its AuditTrail.

Effect

The AuditTrail contains only accepted Actions performed within the recorded Authorisation.

42. Idempotency of Operations

Rule

Repeating a request for an operation already accepted must not constitute a second Action.

For CommitInput, the same requester may constitute separate Actions for different SourceLocations. A SourceLocation already committed as Input for the TransactionActivity cannot constitute another Action. A repeated commitment of that SourceLocation is treated as a retry only where its idempotency reference identifies the previously accepted request.

A SourceLocation-free CommitInput constitutes empty Input and cannot be combined with a sourced Input commitment for the same TransactionActivity.

Application

A repeated request is recognised through its idempotency reference and returns the result of the previously constituted Action. A separately authorised performance of the same operation constitutes a new Action.

Effect

Retries or interrupted communication do not duplicate an Action or alter the Transaction’s state of play.

43. Rejection of Operations

Rule

A requested operation is rejected where required authentication fails, it falls outside the applicable Authorisation, it does not meet the operation’s requirements, or it is incompatible with the Transaction’s state of play.

Application

The response identifies the reason for rejection to the extent permitted by the applicable security rules. No Action or Formal Expression is constituted.

Effect

The Transaction, its state of play and its AuditTrail remain unchanged.

44. Ordering of Actions

Rule

Actions within a TransactionActivity are ordered by their constitution in the AuditTrail.

Application

Each Action is related to the Transaction and TransactionActivity under the TransactionID and records when it was constituted. Where an Action depends on an earlier Action, the requested operation is rejected until that required Action has been constituted.

Effect

The accepted sequence of Actions and the satisfaction of their dependencies can be determined without accessing the Material Expressions of the Transaction.

45. Mandatory CommitInput

Rule

Every TransactionActivity requires at least one CommitInput Action before CommitOutput can be accepted.

Application

Each independently governed source commits its own Input contribution. Where a Material Expression exists, its Formal Expression identifies its SourceLocation.

Where Input is empty, CommitInput remains required without a SourceLocation.

Effect

Every TransactionActivity follows the same formal sequence whether Input is made available from one source, several sources or is empty.

46. Progression of the TransactionProcess

Rule

Acceptance of CommitOutput completes the TransactionActivity and progresses the TransactionProcess.

Application

CommitOutput is accepted only after at least one CommitInput Action has been constituted.

Its acceptance constitutes the Formal Expression of the Output. The next TransactionActivity—if any—becomes the state of play, and its participants are informed.

Effect

Authorisation to access the completed TransactionActivity's Input for its performance ends. Its Output becomes available at its SourceLocation for authorised reading, retrieval or processing.

47. Fourth and final TransactionActivity

Rule

Acceptance of CommitOutput completes the fourth and final TransactionActivity without constituting a separate completion state.

Application

The Formal Expression of the final Output is constituted, and the participants in the fourth and final TransactionActivity are informed. No subsequent TransactionActivity becomes the state of play.

Effect

No TransactionActivity remains the state of play. The constituted Actions remain in the AuditTrail, and the fourth and final TransactionActivity Output remains available at its SourceLocation in accordance with the applicable Authorisation and preservation requirements.

48. Participant Notification

Rule

Participants are informed when a TransactionActivity becomes the state of play, when its Input is committed and when the TransactionActivity is completed.

Application

Participants recorded for the applicable TransactionActivity are informed of its change in state.

Where Input or Output is committed, only participants whose Authorisation concerns that Input or Output are informed. The notification identifies the TransactionID and TransactionActivity and includes the Formal Expression available within the participant's authorised operational scope.

The notification does not include a Material Expression.

Effect

Each participant knows when it may perform its authorised Actions and can retrieve the applicable Material Expression from its SourceLocation. Loss of the notification does not prevent recovery of the Formal Expression through RetrieveTransaction.

49. Availability of Output

Rule

Output remains available at its SourceLocation for as long as required by the Agreement or an applicable obligation.

Application

The Data Holder is responsible for preserving the Material Expression at the applicable SourceLocation. Access to it is governed by the Authorisation Register and may occur through direct reading or authorised retrieval and processing in accordance with the applicable Data Role.

Effect

Fairsfair communicates the availability and SourceLocation of the Output through its Formal Expression. It does not hold the Material Expression or determine its preservation period.

50. Retrieval Through Federation

Rule

A Material Expression may be retrieved from its SourceLocation through federation by an authorised Data Services Provider.

Application

The source validates the retrieval against the applicable Authorisation. The Reference API communicates the Formal Expression but does not receive, transfer or intermediate the Material Expression.

Effect

The Material Expression passes only through the authorised operational arrangement established for its retrieval and processing. It does not pass through the Reference API.

Release history

v1.0 — Published Reference API release.
Aligned the human-readable specification with the frozen OpenAPI technical contract. It establishes the default 24-hour Transaction Process time scope, a further 48 hours of Formal Expression availability for authorised retrieval, and the explicit recording of extended Activity and Formal Expression availability periods. Commit authority ends with the applicable Activity period; retrieval authority ends with the relevant Formal Expression availability period.

The Reference API is issued as the v1.0 operational specification and is aligned with its OpenAPI technical contract.

v0.7 — Release-gate alignment following R02–R03 coverage review.
Clarified that an AccessToken applies to one TransactionActivity, that the TransactionProcess arranges the proposed sequence without assessing its executability, and that subsequent accepted operations do not alter or replace constituted Actions. Corrected minor publication formatting. The seven-operation Reference API and its established operational boundaries remain unchanged.

v0.6 — Source contribution and Reference lifecycle alignment.
Established separately committed Input contributions from independently governed SourceLocations and distinguished Authorisation concerning Input or Output. Fixed the Reference baseline at no more than 24 hours for execution followed by 48 hours of retention, limited retention-period access to authorised retrieval, and clarified mandatory CommitInput, the fourth and final TransactionActivity and context-specific participant notification. The seven-operation Reference API and its established operational boundaries remain unchanged.

v0.5 — Transaction record and lifecycle alignment.
Established the Transaction's progressively constituted operational record, including retrievable Formal Expressions and Actions. Aligned TransactionInitiation with the proposed TransactionProcess, made the complete TransactionProcess and SourceID available to participants, introduced Retention and Disposal, and connected participant notification with recovery through RetrieveTransaction. The seven-operation Reference API and the boundary excluding Material Expressions are preserved.

v0.4 — Alignment with the Foundational Series and R01.
Aligned the Reference API with the settled operational model, including what-before-who ordering; direct and delegated performance; natural-person participation under the TransactionID; Constitutional Roles, Data Roles and Authorisation; Input, Output, Formal Expression and Material Expression; activity-specific access; Action constitution and accountability; and progression of the TransactionProcess. Removed Equivalent Material Expression and the generic Optional Services provision. Corresponding schema repairs are identified where required. The seven-operation Reference API and its content-free boundary are preserved.

V0.3 — Controlled alignment following R03 implementation review.
GetOngoingTransactions authenticates the discovering Entity directly and filters results through each Transaction's Authorisation Register. Transaction-specific AccessTokens remain required for RetrieveTransaction, CommitInput and CommitOutput.