fairsfair Initial Body of Work

Reference Implementation

Edition v1.0 · Published

Series	Reference
Source	fairsfair Reference Implementation
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 status

R03 records the reference implementation of the Reference API. It is a small, inspectable implementation and evidence package for the behaviour established by FF-R02 and assessed through FF-R04. It does not add a separate normative layer.

The release coordinates formal operational facts: the proposed TransactionProcess, Authorisation, state of play, Formal Expressions, Actions, participant notices and the Reference lifecycle. It does not receive, transfer, inspect or store a Material Expression.

2. What the implementation demonstrates

Boundary	Implemented behaviour
Interface	All seven Reference API operations, their accepted responses and their refusal boundary.
Authority	Authentication and AccessTokens are limited to the recorded operational scope. CommitInput and CommitOutput authority ends with the applicable TransactionActivity period; retrieval authority ends with the relevant Formal Expression availability period.
Transaction	A supplied four-activity proposed order, state of play, mandatory Input, Output, progression, finality, default time scope and recorded extended Activity and Formal Expression periods.
Accountability	Timestamped, immutable CommitInput and CommitOutput Actions in the AuditTrail.
Expression boundary	Formal Expressions and SourceLocations are coordinated; Material Expressions remain at their sources.
Informing and refusal	Applicable participants receive formal events; a rejected operation constitutes no protected operational fact.

3. The four TransactionActivities in the proposed process

TransactionInitiation receives the proposed TransactionProcess and records it as supplied. In this reference implementation, the process contains each of the four TransactionActivities once. Delivery Identification precedes Delivery Fulfilment; Payment Identification precedes Payment Fulfilment. The implementation preserves the proposed order but does not assess whether the Agreement is substantively executable.

Position in proposed order	TransactionActivity	When it is complete
1	Delivery Identification	Its authorised Input is committed; its Output is committed; then the next proposed activity becomes the state of play.
2	Delivery Fulfilment	Its authorised Input is committed; its Output is committed; then the next proposed activity becomes the state of play.
3	Payment Identification	Its authorised Input is committed; its Output is committed; then the next proposed activity becomes the state of play.
4	Payment Fulfilment	Its authorised Input is committed; its Output is committed; no TransactionActivity remains the state of play.

The positions shown are a complete reference run. The TransactionProcess records the order supplied for the Transaction; no activity is invented and TransactionInitiation itself is not a TransactionActivity or an Action.

4. The seven Reference API operations

Operation	Reference implementation behaviour
HealthCheck	Returns availability without authentication and does not change a Transaction.
TransactionInitiation	Constitutes one Transaction under its TransactionID, records the proposed process and Authorisation Register, makes its first TransactionActivity the state of play and informs its participants.
IssueAccessToken	Authenticates the requester and issues an AccessToken within recorded authority, purpose and duration. CommitInput and CommitOutput authority ends with the applicable TransactionActivity period; retrieval authority ends with the relevant Formal Expression availability period.
GetOngoingTransactions	Lists ongoing Transactions only where the authenticated requester is authorised to participate or act.
RetrieveTransaction	Returns the authorised operational representation: TransactionProcess, state of play and visible Formal Expressions, Actions and Authorisation, never a Material Expression.
CommitInput	Constitutes the Input Formal Expression and immutable Action for the state-of-play TransactionActivity, including a SourceLocation only where a Material Expression exists.
CommitOutput	Constitutes the Output Formal Expression and immutable Action, completes the TransactionActivity and progresses the proposed process or concludes it.

5. One complete four-activity Transaction

The run below shows the observable formal state. It is intentionally concerned with coordination and proof, not with the Material Expressions or the substantive work performed by participants.

Moment	State of play	Constituted record and informing
Initiation	First proposed activity	One TransactionID and the preserved proposed TransactionProcess. No Action is constituted. Participants recorded for the first activity are informed.
Each CommitInput	Current activity	One Input Formal Expression and one immutable, timestamped CommitInput Action. Applicable Input participants receive the formal event.
CommitOutput of activities 1–3	Next proposed activity	One Output Formal Expression and one immutable, timestamped CommitOutput Action. The completed activity is executed; the next activity becomes state of play and its participants are informed.
CommitOutput of activity 4	None	One Output Formal Expression and one immutable, timestamped CommitOutput Action. The fourth activity is executed; no completion activity or separate completion state is constituted.

Every TransactionActivity requires at least one CommitInput before CommitOutput can be accepted. A materially empty Input still requires its formal CommitInput. Independently governed sources commit their Input contributions separately.

6. Formal expression, source and federation

Object	The implementation retains or communicates	The implementation does not do
Formal Expression	The TransactionID, TransactionActivity, InformationContext and, where applicable, a SourceLocation.	Receive, transfer, inspect or store its Material Expression.
SourceLocation	Where an authorised participant may directly retrieve a Material Expression from its source.	Turn the source into a Reference API data store.
Material Expression	Nothing beyond the content-free formal fact that it is available.	Carry or reproduce the information through the Reference API.
Federated retrieval	The formal boundary through which an authorised Entity or Data Services Provider can retrieve at the source.	Prescribe how the source provides authority or preserves its content.



7. Authority, recovery and informing

The Authorisation Register relates the recorded TransactionActivity and InformationContext to Data Role, authorised Actions, Principal, any delegated agency, purpose and duration. The reference implementation uses that record to issue and validate operational scope. It does not determine the source of authority, Constitutional Capability, or an Entity's substantive ability to perform its work.

A participant can recover an authorised Formal Expression through RetrieveTransaction if a notice is lost. Notices identify the TransactionID and TransactionActivity and convey only the formal event within the recipient's authorised operational scope. They never carry a Material Expression.

8. Refusal, retry and lifecycle

Unless an applicable Agreement states otherwise, a 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 shall record the applicable period for each such Activity and the corresponding availability period for its relevant Formal Expressions.

Condition	Reference effect
Authentication, structure or authority does not satisfy the applicable operation	The operation is rejected. No TransactionID, Formal Expression, Action, state change or participant notice is constituted.
A request targets an activity other than the state of play	The operation is rejected; the current state and AuditTrail remain unchanged.
CommitOutput is attempted before a CommitInput Action	The operation is rejected; the activity remains the state of play.
The same accepted TransactionInitiation is retried	The prior TransactionID is returned. No duplicate Transaction or participant notice is constituted.
Fourth CommitOutput is accepted	The Transaction ends execution without a separate completion state. The formal operational record remains available for authorised retrieval for its applicable availability period and is then disposed under the Reference lifecycle.

9. Evidence and release boundary

The sealed implementation package contains the executable contract, the source implementation, examples, the R02/R04 coverage review, a complete four-activity R04 fixture and a refusal-and-retry fixture. Its published conformance record reports 33 tests passed, with all 52 R04 requirements covered by the recorded evidence mapping.

This result is evidence for the exact sealed package. It does not certify legal compliance, participant capability, substantive authority, source availability, additional security assurance or general fitness for purpose.

10. What R03 is—and is not

R03 is	R03 is not
A concrete Reference Implementation and inspectable evidence package for FF-R02 and FF-R04.	A replacement for the authoritative Reference API or the conformance requirements.
A proof of coordinated formal behaviour across all four TransactionActivities.	A processor, store or intermediary of a Material Expression.

A bounded implementation of the Initial Body of Work.	A finding on the validity of an Agreement, authority, capability or substantive outcome.

Publication note. R03 is read with FF-F01 Foundational Series, FF-R02 Reference API, FF-R04 Reference Conformance and the applicable Licence and governance publications.