TCBON Account Services

Developer & Brand Standard

Applies to: **accountservice.thecountrybankofneedham.com**
Audience: Engineering, Compliance, Authorized Integrators, Examiners
Document Status: Controlled / Examiner-Facing
Version: v1.0
Effective Date: January 29, 2026

Classification	Internal / Examiner-Facing
Change Control	Amendments only by supersession record; no in-place edits
Primary Purpose	Define canonical identity, governance, API behavior, and brand usage for TCBON Account Services
Record Principle	Every authorization, lifecycle transition, and artifact is immutable, hashed, and verifiable

This standard is intended to produce consistent, auditable, regulator-safe implementations across WordPress plugins, Node/FastAPI services, and any authorized partner integration. All implementations SHALL follow the normative language in this document.

Table of Contents

1. Authority & Scope

1.1 Purpose. This document defines mandatory implementation and brand rules for TCBON Account Services, including record identity, lifecycle governance, cryptographic controls, artifact sealing, verification, and developer-facing attribution.

1.2 Scope. The standard applies to systems and workflows surfaced or supported by accountservice.thecountrybankofneedham.com, including: ACH lifecycle and release, wire desk intake and release, issuer-grade card lifecycle actions, lease and equipment-related records, and asset-backed credit enhancement records.

1.3 Controlled vocabulary. Implementations SHALL use the canonical identifiers and status values defined herein. Implementations SHALL NOT introduce alternate meaning for defined fields.

1.4 Normative terms. The terms SHALL, SHALL NOT, SHOULD, and MAY are used as normative requirements.

2. Canonical Identity & Record Model

2.1 Governance anchor (meeting_id). A meeting_id represents the governance context for one or more actions, including approvals, attestations, and dual-control release authorization.

2.2 Immutable primary record (record_id). A record_id uniquely identifies an immutable record describing an instruction, authorization, lifecycle transition, or evidence artifact. Once created, a record SHALL NOT be edited in place.

2.3 Program umbrella (plan_id). A plan_id MAY be used to group recurring or programmatic actions (e.g., payroll schedules, standing reserve policies, recurring credit enhancement triggers).

2.4 Supersession lineage (chain_id). chain_id identifies the immutable lineage of a logical record over time. When an item is replaced, a new record is created and linked by supersession; chain_id remains stable across the lineage.

2.5 Required metadata. Each record SHALL include: created_at, created_by, status, effective_at, expires_at (if applicable), revoked_at and revocation_reason (if applicable), supersedes_record_id (if applicable), and a canonical payload hash (SHA-256).

2.6 Check-digit enforcement. All externally referenced identifiers (meeting_id, record_id, plan_id, chain_id, key_id) SHALL include the Fibonacci check-digit suffix per the TCBON Fibonacci Check Digit System.

3. Lifecycle Governance

3.1 State machine enforcement. Implementations SHALL enforce allowed transitions and SHALL reject invalid transitions. Terminal states SHALL prevent subsequent transitions except through a new superseding record.

3.2 Effective vs expired. effective_at defines when a record becomes operative. expires_at defines when the record is no longer valid for new actions. Expiration does not delete history; it prevents new reliance on the record for authorizations and releases.

3.3 Revocation. Revocation terminates reliance on a record prior to expiry. A revocation SHALL be recorded as an immutable event linked to the revoked record, including revoked_by, revoked_at, and revocation_reason.

3.4 Supersession. Corrections and changes SHALL be performed by supersession. A superseding record SHALL reference the prior record via supersedes_record_id and SHALL share the same chain_id. The prior record SHALL be marked SUPERSEDED and remain verifiable.

4. Cryptographic & Token Controls

4.1 Key classes. TCBON implementations SHALL support three key classes: (a) Root policy keys for trust anchoring, (b) Service signing keys for pack and receipt signatures, and (c) Ephemeral keys for meeting-scoped or record-scoped actions.

4.2 Key identifiers. key_id SHALL be stable, unique, and check-digit protected. Rotations SHALL create a new key_id; keys SHALL NOT be overwritten in place.

4.3 One-time authorization tokens. One-time tokens SHALL be bound to (a) a record_id, (b) an intended audience (endpoint/service), and (c) a nonce. Tokens SHALL include issued_at and expires_at, SHALL enforce single redemption by setting redeemed_at exactly once, and SHALL be invalidated on record revocation or supersession where applicable.

4.4 Non-replay and deterministic denial logging. Replay attempts SHALL be rejected deterministically and SHALL generate an immutable denial event.

5. Asset-Backed Credit Enhancement (ABSCE)

5.1 Record family. Asset-backed credit enhancement SHALL be represented as its own record family (ABSCE) to ensure audit clarity and accurate investor and examiner interpretation.

5.2 Enhancement techniques. ABSCE records SHALL explicitly declare enhancement_type (overcollateralization, reserve, insurance, guarantee), coverage amounts and/or percentages, counterparty identifiers, and trigger conditions that modify coverage.

5.3 Result semantics. Where the system computes investor-facing results (e.g., expected pricing impact, target uplift), those fields SHALL be labeled as model outputs and SHALL reference supporting assumptions stored as sealed attachments.

6. API Design & Integration Rules

6.1 Naming conventions. JSON fields SHALL use snake_case. Endpoint paths SHALL use kebab-case. Identifiers SHALL follow the canonical ID formats defined in Appendix A.

6.2 Idempotency. Create and state-transition endpoints SHOULD support idempotency keys. Where supported, replays with identical payload fingerprints SHALL be treated as no-ops and return the existing record.

6.3 Error semantics. Errors SHALL be deterministic, machine-readable, and stable. Each error response SHALL include code, message, and, where applicable, the relevant record_id and policy snapshot reference.

6.4 Versioning. The API version is part of the URL (/v1). Breaking changes SHALL result in a new major version. Deprecations SHALL include explicit dates and a migration guide.

7. Immutable Packs & PDFs

7.1 Pack structure. Sealed packs SHALL include a canonical record payload, manifest.json, file hashes, optional signature artifacts, attachments, and a machine-readable event log (e.g., NDJSON).

7.2 Manifest requirements. manifest.json SHALL include record_id, chain_id, meeting_id, sealed_at, key_id, and a file list with path, sha256, size, MIME type, and purpose.

7.3 Verification. Verification endpoints SHALL validate hashes and signature metadata and SHALL display effective/expiration/revocation status without alteration of historical artifacts.

7.4 Examiner view. Examiner views MAY include expanded metadata (approvals, role attestations, policy snapshot ids) that are withheld from public verification views.

8. Capability Registry & Role Enforcement

8.1 Single canonical registry. All WordPress plugins and all service implementations SHALL use a single canonical capability namespace. Capabilities SHALL NOT be renamed or reinterpreted locally.

8.2 Role mapping. Roles are defined as bundles of capabilities. Role names and capability strings SHALL match exactly across PHP and Node/FastAPI implementations.

8.3 Dual control. High-risk actions (including ACH and wire release, certain card issuance actions, and ABSCE approvals) SHALL require two distinct approvers with distinct roles (Treasury Approver and Compliance Approver).

8.4 Enforcement point. Enforcement MUST occur at the API boundary (permission callbacks / middleware) and MUST be recorded as immutable events for both approvals and denials.

9. Brand & Attribution Standard

9.1 Tone and voice. Developer documentation and UI copy SHALL be factual, regulator-safe, and non-hyperbolic. Claims of endorsement SHALL NOT be made without explicit contractual authority.

9.2 Logo rules. Logos SHALL NOT be modified (no stretching, recoloring, or aspect ratio changes). Logos SHALL be placed on backgrounds that preserve legibility. Minimum clear-space rules SHALL be respected.

9.3 Attribution. Integrations that surface the API SHALL include the required attribution string as defined in the brand kit, and SHALL not imply affiliation beyond what is contractually established.

9.4 Theme management. Where themes are managed via API or controlled distribution, theme packages SHALL include lockfiles or hash manifests to prevent drift and to provide an auditable basis for consistency.

10. Audit, Retention & Examination

10.1 Retention. Records and sealed artifacts SHALL be retained per the bank's retention schedule. The system SHALL preserve verifiability throughout the retention period and SHALL not invalidate prior evidence by in-place edits.

10.2 Examiner retrieval. The system SHALL support evidence-ready retrieval by record_id and chain_id, including the ability to produce sealed packs and verification outputs on demand.

10.3 Audit trail. Every approval, denial, revocation, and supersession SHALL be logged as an immutable event. Audit logs SHALL include actor identity, role context, timestamps, and payload fingerprints.

10.4 Incident response. Security incidents impacting integrity or availability SHALL be recorded and shall reference impacted records and policy snapshots where applicable.

Appendix A. ID & Check-Digit Examples

Identifier	Example (display form)
meeting_id	MTG-20260129-2045-TCBON-ASV-0047K
record_id	TCBON-ACH-20260207-000128-PROD-Q
plan_id	PLAN-ACH-2026-PAYROLL-0032M
chain_id	CHN-TCBON-ACH-20260207-000128-PROD-ROOT-7
key_id	KEY-SVC-PROD-202601-0042C

Example check-digit suffix characters are illustrative. Production generation and validation SHALL follow the TCBON Fibonacci Check Digit System implementation used by the canonical registry.

Appendix B. Error Code Catalog (Normative)

Code	HTTP	Meaning	Required Fields
invalid_id_check_digit	422	Identifier fails Fibonacci check-digit validation	code, message, field
invalid_state_transition	409	Requested lifecycle transition is not permitted	code, message, from, to
record_not_found	404	Record identifier does not exist	code, message, record_id
record_expired	409	Record is expired and cannot authorize new actions	code, message, record_id, expires_at
record_revoked	409	Record is revoked and cannot authorize new actions	code, message, record_id, revoked_at
dual_control_required	403	Action requires two distinct role attestations	code, message, required_roles
dual_control_not_satisfied	403	Dual control is configured but approvals are incomplete or invalid	code, message, missing
already_redeemed	409	One-time token has already been redeemed	code, message, token_id, redeemed_at
forbidden	403	Caller lacks required capability	code, message, capability

Implementations SHALL return these codes exactly as specified. Additional codes MAY be added by supersession of this standard.