The Boli Whitepaper

BOLI WHITEPAPER · THE-BOLI-WHITEPAPER

The Boli Whitepaper

An operating layer on the Canton Network for regulated tokenized assets.

V0.9 — DRAFT FOR REVIEW, MAY 2026

Published by Boli Association (Switzerland).

Status of this document

This is a draft for review. It describes the Boli Platform as designed and currently being implemented at v0.9. Sections marked (specification) in Part III are authoritative for implementations; other sections are explanatory. Material changes between v0.9 and v1.0 will be tracked in the change log at the back of this document.

The whitepaper is maintained at `github.com/boliassociation/whitepaper`, alongside the Boli Standards Proposals (BSPs) repository at `github.com/boliassociation/bsps`. It is licensed under CC BY 4.0.

Abstract

Boli is an operating layer on the Canton Network for regulated tokenized assets. It ships three Daml asset patterns — Tradeable, Registry-mirror, Credential — which together carry every regulated asset class encountered in production: securities, funds, sukuk, real-estate share classes, environmental credits, sovereign registries (land, vehicles, IP), identity attestations, and central-bank instruments. A chain-level compliance-pack engine enforces the licensed party's policy on every transfer; the rule runs on Canton, not in a frontend. Multi-VM mirroring keeps the asset canonical on Canton — where institutions transact under sub-transaction privacy and atomic settlement — and emits distribution mirrors on EVM and Solana for retail and DeFi reach. The TDIP identity bridge, anchored on Tenzro, binds Canton parties to W3C Verifiable Credentials and DIDs. Agentic asset operations under DID-bound, on-chain mandates run the operational lifecycle: NAV reconciliation, MRV verification, compliance monitoring, redemption orchestration.

The whitepaper sets out the architecture, the three asset patterns, the regulatory boundary, and the specifications.

BOLI WHITEPAPER · PART-I-STRATEGIC

Part I — Strategic

1. Introduction

1.1 What Boli is

Boli is an operating layer on the Canton Network for regulated tokenized assets. The platform is built around three Daml asset patterns; a compliance-pack engine that runs at the chain level; multi-VM mirroring that keeps Canton canonical while emitting distribution surfaces on EVM and Solana; an identity bridge to Tenzro's W3C-Verifiable-Credential foundation; and an agentic runtime that executes operational workflows under on-chain delegated authority.

The chain is Canton. Boli rests on Canton protocol properties: sub-transaction privacy, named-party accountability, atomic delivery-vs-payment, programmable multi-party workflows in Daml, instant finality. Boli turns those primitives into products a licensed party can ship.

Boli does not issue assets. Boli does not custody assets. Boli does not hold keys for any party. Boli does not operate regulated venues. Licensed parties — banks, asset managers, registrars, sovereigns — issue, custody, and operate. Boli ships the standard the licensed party uses to do so on Canton.

1.2 Why now

Three currents converge in 2026.

Institutional tokenization is now the operating model. DTCC and Digital Asset announced in December 2025 the tokenization of DTC-custodied U.S. Treasuries on Canton. The HKMA tokenized HK$10 billion in digital green bonds in late 2025. BIS Project Agorá entered its testing phase in January 2026 with seven central banks and forty-one financial institutions. MAS Project Guardian moved from fund tokenization through to government-bills pilots with wholesale

CBDC. India's e-Rupee crossed 6 million users in production. Canton's Splice Token Standard V1 stabilised in early 2026. The shift from "experiment" to "production" has happened inside the past eighteen months.

Sovereign DPI is being built right now. SL-UDI on MOSIP, India Stack, the EU Digital Identity Wallet under eIDAS-2, GovStack, MOSIP-derived programmes across more than thirty countries — the identity, payment, and document rails that tokenization depends on are under active construction in jurisdictions that, three years ago, were not on the institutional finance radar. Tokenization on top of these rails is the platform-defining opportunity of the next decade.

The regulatory perimeter is converging on Canton-shaped properties. MiCA in the EU, the FSRA framework in ADGM, the UK's Digital Asset Bill 2025, and the broader migration of regulated tokenization onto permissioned settlement infrastructure all converge on what Canton already provides natively: privacy at the protocol level, named-party operators, atomic settlement. Boli is built for that regulatory direction.

1.3 Reading this document

The whitepaper is organised so that a reader from any of three audiences can stop at a natural boundary.

— Strategic readers — central bank policy staff, ministry advisors, multilateral programme managers — should read Parts I and IV end-to-end and skim Part II.
— Architectural readers — institutional integration leads, regulators' technical staff, Canton Foundation contributors — should read Parts I and II in full and consult Part III for the specific patterns relevant to their work.
— Implementation readers — engineering teams shipping Boli-compatible packages, custodians integrating against Boli's interfaces — should treat Part III as the authoritative reference and use Parts I–II as motivation.

---

2. The institutional rail: Canton Network

2.1 What Canton is

The Canton Network is a privacy-preserving public blockchain operated as a federation of named institutions. Any participant can join and run a validator; every transaction's privacy domain is bounded to the parties with a need to know. The combination of public access, sub-transaction privacy, and named-party accountability is what makes Canton the rail regulated finance settles on.

The protocol layer is Canton, written in Scala and Daml. The smart-contract layer is Daml, a contract language for multi-party workflows: every Daml template specifies the parties whose authority is required for a state transition, and the Canton runtime enforces that authorisation atomically. If the required signatures are not present, the transition does not occur. Regulated transfers settle on this property.

Settlement is mediated by Splice, a tokenized utility layer on Canton governed under the Linux Foundation through the Global Synchronizer Foundation (GSF). The Splice Token Standard V1 — stabilised in early 2026 — defines the interfaces a Daml asset implements to settle atomically across counterparties under the `AllocationV1` model. `AllocationV1` is delivery-vs-payment as a Canton primitive: the transfer of the asset and the transfer of the settlement asset are a single atomic action, or neither happens.

2.2 Who is on Canton

Goldman Sachs, BNY Mellon, J.P. Morgan, BNP Paribas, DTCC, HSBC, Standard Chartered, Cboe, Deutsche Börse, Broadridge, Tradeweb, the Hong Kong Financial Markets Infrastructure, Moody's Ratings, Digital Asset, Circle, DRW, Cumberland, and the Global Synchronizer Foundation run on Canton. Validator operators and Super Validators include Deloitte, Paxos, and others under the GSF.

These institutions participate in Canton via the Linux Foundation–hosted GSF. The relevance for a licensed party shipping a regulated asset on Boli is architectural: issuing on Canton puts the asset on the same network where the largest custodians, transfer agents, and tokenization desks transact. The conversation with a regulator, a counterparty, or an investor begins from that fact.

2.3 The Global Synchronizer Foundation

Canton's governance sits with the Global Synchronizer Foundation, hosted by the Linux Foundation. The GSF governs the synchroniser layer that orders transactions across the network's privacy domains, defines the Splice token standards, and oversees protocol evolution. Membership is open to financial institutions, technology providers, and standards bodies under the Linux Foundation's neutral-governance model.

Boli composes with GSF-governed standards. The Splice Token Standard V1, the `AllocationV1` settlement model, and the Canton protocol roadmap are inputs to the Boli specification. Where Boli introduces conventions on top — the structure of a compliance pack, the semantics of a registry-mirror transfer — the conventions are documented in the Boli Standards Proposals (BSPs) maintained alongside this whitepaper, and compose with GSF-governed standards.

2.4 Why a privacy-preserving public chain matters for regulated assets

A regulated asset has three properties Canton accommodates natively.

Holder lists are confidential. A bond issuer's investor list, a fund's limited partners, a real-estate development's beneficial owners — these are confidential to the issuer, the regulator, and the holder. On Canton, the holder list is visible only to the parties whose Daml authority is required to read it; the public ledger records that a transfer occurred and that the relevant parties signed it; identities and holding amounts stay inside the privacy domain.

Compliance is per-jurisdiction and per-asset. A real-estate share class issued under Sri Lankan law has different transfer restrictions from one issued under UAE rules; a fund interest issued to a UAE investor has different rules from one issued to a U.S. accredited investor. The compliance rule is encoded in the Daml template and runs as a precondition of every transfer; the chain enforces the rule regardless of which frontend initiated it.

Settlement is atomic and final. A regulated transfer is a delivery against payment, settled in one indivisible action. Canton's `AllocationV1` model treats DvP as a primitive: the asset moves, the settlement asset moves, both or neither. Finality is instant.

Confidentiality, jurisdictional compliance, atomic finality. Canton is the rail Boli is built on, and a Boli-issued asset is regulator-ready by construction.

---

3. The three asset patterns

Every regulated asset class collapses to one of three Daml patterns. A licensed party shipping any of the asset types below selects one of three patterns and configures its parameters; the Daml template is shared, parameterised, and re-used across deployments.

3.1 Pattern A — Tradeable

Motivation. Securities, funds, sukuk, real-estate share classes, environmental credits with secondary markets, structured products. Anything where the asset has a primary issuance, a holder population, and a secondary market in which holders trade with one another under issuer-defined rules.

Shape. A Tradeable asset has:

- A fungible holding contract parameterised by the issuer party and the holder party.
- A transfer interface that, on every transfer, invokes the issuer's compliance pack as a precondition.
- An `AllocationV1` settlement interface for atomic delivery-vs-payment against any settlement asset the licensed party permits.
- Optional redemption, dividend, and corporate-action interfaces parameterised by the issuer.

The compliance pack carries the rules — investor accreditation, jurisdictional restrictions, holder limits, lock-ups, sanctions screening — and runs at the chain level on every transfer. The rule is the asset's own property, enforced by Canton.

Example assets. A Sri Lankan real-estate development's share class. A UCITS fund interest. A Saudi sovereign sukuk. A tokenized U.S. Treasury Bill (when a licensed issuer ships it). A structured note. A blue-carbon credit with a secondary market.

The full specification is reserved for a forthcoming BSP-0001 (Asset Pattern: Tradeable), summarised in §11 of this whitepaper.

3.2 Pattern B — Registry-mirror

Motivation. Land titles, vehicle titles, IP rights, professional licences — anything for which a sovereign register is the source of truth. The asset on Canton does not replace the sovereign record; it mirrors it, with cryptographic verifiability and atomic transfer semantics that the sovereign record's underlying database does not provide.

Shape. A Registry-mirror asset has:

— A mirror contract signed by the registry party (the sovereign authority) and the holder party (the title-holder).

— A transfer template in which both parties' Daml authority is required for any state transition. The registry party retains operational control of corrections and revocations; the holder party retains control of voluntary transfers; both are atomic.

— A link back to the sovereign record's identifier — a folio number, a chassis number, a registration ID — so the on-chain mirror is auditable against the off-chain authoritative record.

— An encumbrance interface that allows liens, mortgages, or restraints to be recorded as additional Daml contracts requiring multi-party authority for release.

Why a mirror, not a replacement. The state remains sovereign over the underlying register. The Canton mirror provides the verifiability properties — uniqueness, atomic transfer, cryptographic audit trail — that the underlying paper or database lacks. The two are kept in sync by an integration adapter operated by the registry authority or its technology partner.

Example assets. Sri Lankan land titles under the Bim Saviya framework. Vehicle registrations under a national DMT. Trademark or patent registrations under a national IP office. Beneficial-ownership filings under a national companies register.

The full specification is reserved for a forthcoming BSP-0002 (Asset Pattern: Registry-mirror), summarised in §12 of this whitepaper.

3.3 Pattern C — Credential

Motivation. eIDAS-2 identity wallets, KYC attestations, professional licences, environmental MRV verifier credentials, accreditation claims, attestations of any form. Anything that is issued by one party to another and verified by a third — without the holder's underlying data being disclosed.

Shape. A Credential is a W3C Verifiable Credential, anchored on Canton with the following Boli additions:

— The issuance contract is signed by the issuer party (e.g. a national identity authority, a professional body, an MRV verifier).
— The holder party controls disclosure: a credential can be presented selectively, revealing only the predicates required by a verifier (e.g. "is over 18", "holds a valid practising certificate", "is the verified MRV operator for this hectare") without disclosing the underlying birthdate, certificate number, or coordinates.
— A revocation contract allows the issuer to revoke a credential atomically; revocation is observable to verifiers without requiring round-trips to the issuer.
— The credential is cryptographically anchored to the holder's DID — issued and managed via the TDIP identity bridge — so the credential is portable across identity wallets that comply with the bridge specification.

Why on Canton. Credentials issued by sovereign authorities require the same privacy and named-party properties as regulated financial assets. A national identity attestation is not data that the issuing state wants observable on a public block explorer; nor is the cryptographic revocation status of every citizen's credential. Canton's privacy-by-default model fits the credential pattern natively.

Example credentials. SL-UDI verifiable credentials anchored to MOSIP. EU Digital Identity Wallet credentials under eIDAS-2. Professional licence attestations issued by a national bar council, medical council, or engineering board. Blue-carbon MRV verifier credentials issued by an accredited body. KYC-claim credentials issued by a regulated bank for re-use by other regulated parties.

The full specification is reserved for a forthcoming BSP-0003 (Asset Pattern: Credential), summarised in §13 of this whitepaper.

3.4 Mapping asset classes to patterns

Selecting any regulated asset class and asking which pattern carries it:

— A central-bank wholesale CBDC is Pattern A, with the CBSL as the issuer and the compliance pack scoped to permitted counterparties.
— A digital green bond is Pattern A, with the green-finance covenant encoded in the compliance pack.
— A digital identity wallet credential is Pattern C.

— A national land registry mirror is Pattern B.

— A blue-carbon credit standard is Pattern A for the credit itself (it trades) plus Pattern C for the underlying MRV attestations (they verify).

— A real-estate share class is Pattern A.

— A trademark registration is Pattern B.

— A KYC-claim re-use programme is Pattern C.

— A tokenized fund interest is Pattern A.

— A vehicle registry is Pattern B.

The patterns compose. A blue-carbon credit programme uses Pattern A for the credit and Pattern C for the verifier attestations that underpin it. A real-estate development uses Pattern A for the share class and references Pattern B (the underlying land title mirror) as the asset backing.

The three-pattern model is the simplification Boli ships. The pattern is parameterised, audited as a unit, and re-used; a licensed party configures rather than authors a new Daml template for every new asset class.

---

4. The regulatory line

Boli's design is governed by an explicit boundary between what it ships and what licensed parties carry. This section sets out the boundary; subsequent sections rely on it.

4.1 What Boli ships

Boli ships technology, standards, and ecosystem-design support. Concretely:

— Daml packages for the three asset patterns.

— A compliance-pack engine that licensed parties configure with their jurisdictional rules.

— Multi-VM mirroring infrastructure that keeps Canton canonical while emitting EVM and Solana mirrors.

— The TDIP identity bridge specification and its reference implementation.

— The Boli Standards Proposals (BSPs) that document the protocol-level conventions on which the platform is built.

— An agentic runtime specification for autonomous operations under DID-bound mandates.

— A marketplace and intent-routing protocol for discovery across listed partners.

4.2 What Boli does not do

Boli does not:

— Issue assets. Issuance is the licensed party's act under the licensed party's authority.

— Custody assets or hold keys for any party. Custody is performed by the licensed party or its custody provider.

— Operate regulated venues. Order books, RFQ systems, or matching engines for regulated assets are operated by appropriately licensed venues.

— Hold settlement assets. Settlement assets are issued and held by the parties that issue and hold them — central banks, regulated stablecoin issuers, tokenized-deposit issuers — not by Boli.

— Take regulatory positions on behalf of any party. Boli does not represent that any specific use of the platform satisfies any specific regulator's requirements; that determination is the licensed party's, made with its own counsel and its own regulator.

4.3 The contributing entities

Three distinct entities sit behind what is conversationally called "Boli".

Boli Association (Switzerland) — a Swiss non-profit. The Association supports research and ecosystem-design: maintaining this whitepaper, the BSPs, the standards process, and the academic and multilateral relationships that surround the platform. The Association does not ship technology. It is the publishing entity for this document.

The Boli Platform — the technology layer, associated with but separate from the Association. The Platform is the operating layer on Canton: the Daml packages, the compliance-pack engine, the multi-VM mirroring infrastructure, the TDIP integration, and the agentic runtime. The Platform is operated by an appropriately incorporated commercial company based in the UAE. The Boli Foundation, when established (see §15), will govern the platform's treasury and the $BOLI ecosystem token.

Tenzro Labs (Singapore) — the infrastructure provider. Tenzro Labs is a partner of the Canton Foundation and operates a Canton validator node. Tenzro provides the infrastructure layer integrated with the Canton Network: digital identity (TDIP), secure key handling (MPC custody primitives), and verifiable execution of automated workflows (the agentic runtime). The Boli Platform is deployed on Tenzro's Canton validator node. Tenzro's mandate is the broader Canton ecosystem; Boli is its primary operating-layer consumer.

The Association publishes standards and engages in academic and policy work. The Platform ships technology and integrates with licensed parties. Tenzro operates the permissionless infrastructure beneath both. Each constituency is governed and accountable on its own terms.

4.4 What licensed parties carry

In every Boli deployment, a licensed party carries:

— Issuance authority for any asset issued through the platform.
— Compliance authority — the policy expressed in the compliance pack is the licensed party's policy, and the licensed party is responsible for its correctness against its regulator.
— Custody and key control — keys never leave the licensed party's custody architecture; Boli operates as a software layer the licensed party uses, not as a counterparty.
— Venue authority — order books, RFQ systems, and matching engines run on licensed venues, not on Boli infrastructure.
— Settlement-asset selection — the licensed party chooses the settlement asset (a tokenized deposit, USDC, EURC, RLUSD, AED-Coin, a Sri Lankan stablecoin, a future LKR wholesale CBDC). Boli does not editorialise on the settlement-asset choice.

This boundary is the basis on which a Boli deployment is regulator-ready in any jurisdiction. The licensed party is regulated; Boli is the standard the licensed party uses.

---

5. The eight audiences

The platform serves eight audiences. The three patterns carry all of them.

Asset and fund managers — issuers of tokenized fund interests, structured products, real-estate share classes, and bespoke private placements. Pattern A. The licensed party's compliance pack carries investor accreditation, lock-ups, holder limits, and jurisdictional rules.

Banks — issuers of tokenized deposits, fixed-income instruments, and structured notes; integrators of tokenized assets into their custody and treasury workflows. Primarily Pattern A; Pattern C for KYC-claim re-use across institutions.

Sovereigns — ministries of justice (land), home affairs (vehicle and personal registries), digital economy (national identity), finance (tokenized treasury issuance, governance bond reporting). Pattern B for registries; Pattern A for treasury issuance; Pattern C for identity attestations.

Citizens — holders of identity credentials, professional licences, real-estate titles, and tokenized fund interests. Citizens interact with the platform through their existing wallets — the SL-UDI eLocker, the EU Digital Identity Wallet, a bank's mobile app — not through a Boli-branded interface.

Environmental markets — issuers of carbon credits, biodiversity credits, water-quality credits, and the verifier attestations that underpin them. Pattern A for the credits, Pattern C for the MRV attestations. Continuous MRV under agentic mandates is a primary use case.

Corporates — issuers of trade-finance instruments, supply-chain documents, working-capital facilities, and treasury instruments. Pattern A for tradeable instruments; Pattern C for supply-chain attestations.

Identity issuers — national identity authorities (MOSIP-derived programmes, EU eIDAS-2 issuers, sectoral identity schemes), KYC providers, professional bodies, and accreditation organisations. Pattern C, with the issuer party held by the appropriate authority.

Central banks — issuers of wholesale CBDC instruments and operators of settlement infrastructure for tokenized assets. Pattern A for the CBDC instrument itself; the broader rail interaction is at the Canton protocol level via the Splice settlement-asset interface.

For each audience, the integration shape is the same: the licensed party selects a pattern, configures the compliance pack, integrates with its custody and identity architecture, and ships. The work is parameterised and re-used across audiences.

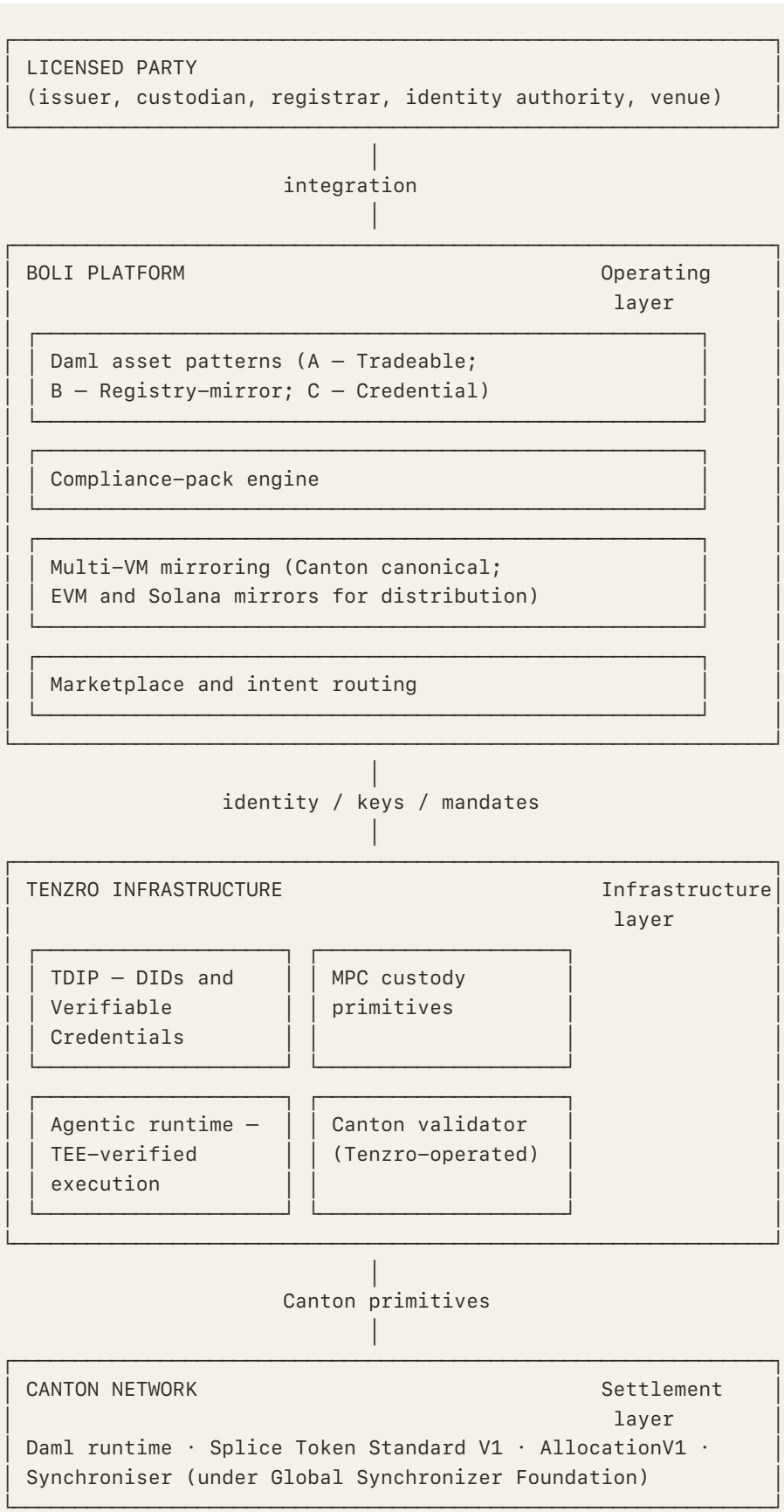
BOLI WHITEPAPER · PART-II-ARCHITECTURE

Part II — Architecture

6. System overview

6.1 The three layers

The Boli architecture is a three-layer model. Each layer has a defined interface to the layer above and below; each layer is independently auditable; no layer assumes responsibilities that belong to another.


LICENSED PARTY
(issuer, custodian, registrar, identity authority, venue)
integration
BOLI PLATFORM
Operating layer
Daml asset patterns (A — Tradeable; B — Registry-mirror; C — Credential)
Compliance-pack engine
Multi-VM mirroring (Canton canonical; EVM and Solana mirrors for distribution)
Marketplace and intent routing
identity / keys / mandates
TENZRO INFRASTRUCTURE
Infrastructure layer
TDIP — DIDs and Verifiable Credentials
MPC custody primitives
Agentic runtime — TEE-verified execution
Canton validator (Tenzro-operated)
Canton primitives
CANTON NETWORK
Settlement layer
Daml runtime · Splice Token Standard V1 · AllocationV1 · Synchroniser (under Global Synchronizer Foundation)

The settlement layer is Canton, governed under the Linux Foundation through the Global Synchronizer Foundation — the public institutional rail on which the architecture settles.

The infrastructure layer is Tenzro. Tenzro provides the identity primitives (TDIP, DIDs, Verifiable Credentials), the secure key-handling primitives (MPC custody), the agentic runtime (TEE-verified execution under DID-bound mandates), and operates a Canton validator node. The Boli Platform is deployed on Tenzro's validator. Tenzro's mandate covers the broader Canton ecosystem; Boli is its primary operating-layer consumer.

The operating layer is the Boli Platform: the Daml asset patterns, the compliance-pack engine, the multi-VM mirroring infrastructure, the marketplace. This is where a licensed party's policy meets Canton primitives.

The licensed party — issuer, custodian, registrar, identity authority — sits at the top. The integration is with the operating layer.

6.2 Why this layering

Each layer has a different audience, a different lifecycle, and a different governance constituency.

The settlement layer evolves under multilateral governance. The Splice Token Standard, the `AllocationV1` model, and the Canton protocol roadmap are owned by the GSF and its institutional members. Boli composes with these standards. When Splice V2 ships, Boli adopts it.

The infrastructure layer evolves under Tenzro's ecosystem mandate. Tenzro's TDIP, MPC custody, and agentic runtime serve the broader Canton developer ecosystem. The interfaces Boli uses — DID resolution, credential anchoring, mandate registration, validator operation — are stable, documented, and used by other Canton-native projects. Boli consumes the infrastructure; the roadmap is Tenzro's.

The operating layer evolves under Boli's own governance. The three asset patterns, the compliance-pack engine, the multi-VM mirroring choices, and the marketplace protocol are Boli's specifications, governed through the BSP process described in §16.

The separation lets each layer be audited on its own terms. A regulator examining a Boli deployment asks three separate questions — Is the settlement rail credible? Is the identity infrastructure credible? Is the operating layer correctly configured for our jurisdiction? — and gets three separate answers, from three separate constituencies, against three separate bodies of evidence.

6.3 Interfaces between layers

The interfaces between layers are explicit and documented. Implementations on either side of an interface are independently replaceable.

Boli ? Canton. Boli's Daml packages compose with Splice Token Standard V1 and the `AllocationV1` settlement interface. The interface is fully specified in the Splice repository and in the BSPs maintained at `github.com/boliassociation/bsps`.

Boli ? Tenzro. Boli's identity bridge consumes TDIP-issued DIDs and Verifiable Credentials. The interface is reserved for a forthcoming BSP-0005 (TDIP integration), with the TDIP specification as the underlying reference. Boli can, in principle, run against any TDIP-compliant identity infrastructure; Tenzro's implementation is the reference, not the only permissible one.

Licensed party ? Boli. The licensed party integrates against the operating layer through (a) Daml package configuration — instantiating the asset pattern with the licensed party's parameters; (b) compliance-pack authoring — encoding the licensed party's policy into the chain-level engine; (c) the licensed party's existing custody and key-handling infrastructure, via TDIP-anchored DIDs.

---

7. The compliance-pack engine

7.1 What a compliance pack is

A compliance pack is the chain-level encoding of a licensed party's transfer policy. It runs as a precondition of every transfer the asset performs, and a transfer that fails the pack does not occur. Policy enforcement is a property of the asset on Canton.

A pack contains:

— Eligibility predicates over the parties to a transfer (e.g. transferee is a credentialled accredited investor; transferee is not in a sanctions list; transferee is in a permitted jurisdiction).

— Quantity constraints (e.g. holder limit of 5%; minimum holding lot; maximum aggregate position).

— Temporal constraints (e.g. primary-issuance lock-up of 365 days; coupon-period transfer freeze).

— Settlement-asset constraints (e.g. only specific tokenized deposits, USDC, and EURC are permitted as settlement assets for this share class).

— Disclosure obligations (e.g. transfers above threshold X must be observable to a designated regulator party).

Each predicate is a Daml expression over party attributes, asset state, and transfer parameters. The pack is itself a Daml contract, with the licensed party as the policy authority — the only party that can amend it — and an explicit audit trail of every amendment.

7.2 How a pack is authored

Compliance packs are authored using a structured policy DSL that compiles to Daml. The DSL is intentionally constrained: it expresses predicates and constraints, not arbitrary computation. A pack is a finite set of rules, each of which is auditable in isolation.

The licensed party authors the pack against a specification published by Boli for each asset pattern. The specification enumerates the predicates that pattern accepts, their parameters, and the semantics of each. A pack that does not match the specification is rejected by the platform — the engine refuses to load a pack with unknown predicates.

Once authored, a pack is signed by the policy authority and published as a Daml contract on Canton. The asset's transfer template references the pack contract; on every transfer, the runtime evaluates the pack against the transfer parameters, atomically with the transfer itself.

7.3 Properties of chain-level enforcement

The rule is the asset's own property. The pack runs on every transfer regardless of the surface that initiated it.

— Frontend uniformity. A frontend deployed by a counterparty, an integrator, or a venue executes against the same on-chain rule set as the issuer's own. The rule is identical across every surface.

— Direct invocation. A transfer initiated by direct contract call evaluates the same pack as a transfer initiated through a UI.

— Cross-venue consistency. An asset trading on multiple venues behaves identically on all of them, defined by the canonical pack.

7.4 Audit and amendment

Every compliance-pack amendment is a Daml transaction signed by the policy authority. The audit trail is the transaction history of the pack contract — observable to the parties with read access (the issuer, the relevant regulator, the licensed party's auditor) and immutable.

Amendments take effect from a specified effective date. Transfers in flight at the time of amendment are evaluated against the pack version in force at the time of transfer initiation. There is no retroactive policy change.

A licensed party may operate multiple packs for different products under the same legal entity, or different jurisdictional rules under the same product. Pack composition — applying multiple packs to the same asset, with explicit precedence — is supported; the precedence rules are part of the asset's Daml configuration.

8. Multi-VM mirroring

8.1 Canonical and mirror

Canton is the canonical settlement layer for Boli assets. EVM and Solana — the two largest distribution surfaces in the public-chain market — are mirror layers. Institutional settlement happens on Canton under sub-transaction privacy and atomic settlement; retail and DeFi distribution happens on EVM and Solana, where the wallet ecosystems are largest.

A Boli asset's authoritative state is on Canton. A mirror on EVM or Solana is a downstream reflection of that state, with documented divergence semantics. The mirror is, by construction, downstream of the canonical leg.

8.2 Why mirror at all

Three reasons.

Distribution reach. A retail investor with a MetaMask or Phantom wallet does not have a Canton wallet. Mirroring the asset onto EVM and Solana means the investor's existing infrastructure works.

DeFi composability. Lending, AMM market-making, and structured-product composability happen on EVM today. A Boli-issued asset that is mirrored to EVM is composable with that ecosystem; one that is Canton-only is not.

Cross-jurisdictional listing. Some venues operate on EVM. Some operate on Solana. A licensed party that wants its asset listed on multiple venues without re-issuing it on each chain mirrors once, lists everywhere.

8.3 What "canonical on Canton" means

Three properties define the canonical-leg invariant.

Source of truth. The Canton ledger is the authoritative record of holdings. If a mirror's state disagrees with Canton, the mirror is wrong.

Settlement. Atomic delivery-vs-payment for the asset happens on Canton, against the licensed party's chosen settlement asset, via `AllocationV1`. A trade settled on a mirror is a trade against a mirror representation; the canonical position has not moved until the canonical leg settles.

Compliance. The compliance pack runs on Canton. A mirror cannot transfer in a way that the canonical pack would have rejected — the mirror's transfer logic enforces the same rule, or the mirror's state diverges and is reconciled against Canton.

The relationship between canonical and mirror is mediated by an anchor contract on each mirror chain, signed by a Boli-operated mirror operator and the asset's issuer. Mirrors are minted against canonical holdings escrowed on Canton; mirrors are burned to release canonical holdings. The mirror operator is operationally responsible for the synchronisation; the asset's authority remains the issuer's, expressed canonically on Canton.

8.4 Divergence and reconciliation

Mirror state can diverge from canonical state in two ways.

Operational lag. A mirror reflects canonical state with bounded latency — typically sub-second, but not zero. During the lag, the mirror's state is a slightly stale view of canonical state. Trades executed on the mirror against this stale view are reconciled at canonical settlement.

Adversarial divergence. Defence is structural: the canonical escrow contract is co-signed by the issuer; mirrors cannot be minted without the escrow being incremented; the escrow's state is observable to the issuer's auditor. A mirror minted without escrow is an instrument the canonical layer will refuse to redeem, and the divergence is visible.

The canonical state is the reference. The mirror operator is auditable. The issuer's authority is preserved.

8.5 Mirror semantics for each pattern

The three asset patterns mirror differently.

Pattern A — Tradeable. Mirrored as ERC-20 (EVM) and SPL (Solana) tokens, with on-chain compliance hooks calling the canonical pack via a cross-chain message. Trading on the mirror is permitted; canonical settlement is required for atomic DvP.

Pattern B — Registry-mirror. Generally not mirrored to EVM or Solana — the registry-mirror pattern is already a mirror, of a sovereign record onto Canton, and a further mirror would compound the indirection. Where required (e.g. a tokenized real-estate share class collateralised by a registry-mirror title), the share class — Pattern A — is mirrored, not the title.

Pattern C — Credential. Credentials are presented, not transferred. The presentation protocol is wallet-side, on the holder's chosen identity wallet, against any verifier on any chain. The credential's revocation status is canonical on Canton; verifiers on other chains query the canonical state via the TDIP bridge.

9. The TDIP identity bridge

9.1 What TDIP is

The Tenzro Decentralized Identity Protocol (TDIP) is the identity foundation Tenzro Labs ships as part of its infrastructure layer. TDIP issues W3C Decentralized Identifiers (DIDs), anchors them to verifiable credentials, manages the cryptographic key material associated with each DID, and bridges DIDs to Canton parties.

For Boli, TDIP solves three problems:

— Naming. Every Daml party in a Boli deployment is named by a DID, not a chain-specific identifier. The licensed party's DIDs are portable across implementations; an asset issued under issuer-DID `did:tdip:lk:bank-of-ceylon` on Canton can be referenced under the same DID on a mirror chain.

— Authentication. Authority over a Daml party is exercised by signing with the keys bound to the party's DID. TDIP manages this — the keys are held in MPC custody (Tenzro's primitive), the DID document records the public-key material, and the signing protocol is portable across wallet implementations.

— Credential anchoring. Pattern C (Credential) issues W3C Verifiable Credentials anchored to TDIP DIDs. The credential's holder is a DID; the credential's issuer is a DID; the credential's revocation status is a Daml contract on Canton referenced by the DID.

9.2 The bridge

The TDIP-to-Canton bridge is the operational integration that maps between DIDs and Canton party-ids. Concretely:

— A DID resolves to a DID document (the standard W3C resolution).

— The DID document includes a Canton party-id binding as a verification method, with the public key authorised to act on behalf of the party.

— Canton's identity provider (IDP) accepts the DID-bound key as a valid authenticator for the party.

— A signing operation on Canton is performed by the key bound to the DID; the signature verifies against the DID document.

The bridge is reserved for a forthcoming BSP-0005 (TDIP integration). The reference implementation lives in Tenzro's identity stack; Boli's web layer integrates against it via a thin client library.

9.3 Wallet portability

End-user wallets are not Boli-specific. A user holding a tokenized real-estate share class issued by a Boli-integrated bank holds the asset in:

— Their bank's mobile app (the bank's white-labelled wallet integrates the Boli SDK).
— Their existing identity wallet (SL-UDI eLocker, EU Digital Identity Wallet under eIDAS-2, a national-identity wallet).
— A standard Web3 wallet (where the asset's mirror is held).

The Daml party — the on-chain entity that owns the holding — is the same across all three. The wallet differences are presentation only; the authority is bound to the DID.

This matters for sovereign-scale deployments, where the citizen-facing surface is a national identity wallet. The citizen's existing wallet, conformant with the bridge specification, is sufficient to participate.

9.4 Key handling

Keys are held in Tenzro's MPC custody primitive. The custody primitive is itself open-source; licensed parties may operate their own MPC custody under the same protocol, or use Tenzro's hosted offering, or use a third-party custodian conformant with the protocol. Boli does not hold keys for any party.

For end-user keys, the standard architecture is custodian-managed MPC: the citizen's bank, identity provider, or wallet provider holds the share material and signs on behalf of the citizen under the citizen's authorisation. This is the model SL-UDI's eLocker uses today, and the model EU eIDAS-2 wallets are converging on.

For licensed-party keys (issuers, registrars, custodians), the standard architecture is institutional MPC under the licensed party's own infrastructure, with TDIP integration via a signing service the licensed party operates.

---

10. Agentic asset operations

10.1 The operational lifecycle

A regulated asset's operational lifecycle — NAV reconciliation, MRV verification, scheduled disclosures, redemption orchestration, dividend distribution, corporate actions, regulatory reporting — runs continuously between transfers. Agentic asset operations execute this lifecycle autonomously, under verifiable mandates, with the same auditability as the on-chain transactions themselves.

10.2 What an agentic mandate is

An agentic mandate is on-chain delegated authority to perform a defined operation under defined constraints. It is a Daml contract signed by the asset's issuer (the principal) granting a Tenzro-runtime agent (the delegate) the authority to perform specific actions on behalf of the principal, subject to:

— Scope — the specific operations the agent is authorised to perform (e.g. recompute NAV daily; submit MRV report monthly; rotate auditor at end of fiscal year).

— Constraints — the conditions under which the agent may act (e.g. only between 09:00 and 17:00 SGT; only for amounts below threshold X; only when the upstream data signature verifies).

— Verifiability — the agent's execution runs in a Trusted Execution Environment (TEE), with the execution attested cryptographically. The audit trail is the TEE attestation plus the on-chain transaction.

— Revocation — the principal can revoke the mandate atomically. Revocation is a Daml transaction signed by the principal.

A mandate is not a private key shared with an agent. It is a Daml authorisation, scoped, time-bounded, and revocable, that the runtime evaluates on every action the agent takes.

10.3 Why TEE-verifiable

The TEE requirement is what makes the agent's autonomy auditable. With TEE attestation, the action is observable as "the agent executed the specific code path on inputs X under attested conditions and produced output Y, then submitted the transaction." The principal, the regulator, and the auditor can verify the attestation independently.

This matters for regulated workflows. A NAV computation that runs in a TEE under a published code commitment, with the input data signed and the output attested, is a reportable artefact.

10.4 What agents do, in practice

Three classes of operation are typical.

Continuous reconciliation. NAV recomputation against custody-side market data; cash position reconciliation between the on-chain settlement record and the off-chain bank account; corporate-action accounting when the underlying issuer pays a dividend.

Continuous verification. MRV report generation for environmental assets, against sensor data; compliance monitoring against transfer streams (e.g. flagging concentrations approaching holder limits); auditor evidence collection on a scheduled basis.

Lifecycle orchestration. Redemption queue processing under bounded SLAs; dividend distribution to the holder list; coupon payment for fixed-income instruments; regulatory disclosure generation on a scheduled basis.

In each case, the agent's actions are bounded by the mandate, attested by TEE execution, and auditable to the principal and the regulator. The operational layer is a verifiable artefact on the same auditability footing as the on-chain layer.

10.5 Boundary between agent and principal

The boundary is explicit. The principal — the licensed party — sets policy, configures the mandate, and bears regulatory responsibility for the agent's actions. The agent — the runtime — executes within the mandate, under attestation, against verifiable inputs. The agent does not exercise judgement outside its mandate; if the inputs to a NAV computation are stale or inconsistent, the agent does not produce a NAV — it raises an exception that the principal handles.

The mandate is the SLA, the TEE attestation is the audit evidence, the chain-level execution is the record. Regulated outsourcing, expressed as cryptographic artefacts.

BOLI WHITEPAPER · PART-III-SPECIFICATIONS

Part III — Specifications

The chapters in this part are specification-grade. They define the Daml interfaces, parameters, and semantics that a Boli-conformant implementation must provide. Where a chapter summarises material that is fully specified in a BSP, the BSP is normative and this chapter is illustrative.

---

11. Pattern A: Tradeable (specification)

11.1 Conformance

A Tradeable asset implementation will conform to the forthcoming BSP-0001 (Asset Pattern: Tradeable) once that BSP is published, by implementing the Daml interfaces defined therein. BSP-0001 will reference CIP-56 (the Canton Improvement Proposal for institutional-grade tokens) and Splice Token Standard V1; conformance with BSP-0001 will imply conformance with both.

11.2 Core interfaces

A Tradeable asset implements the following Daml interfaces:

- `Token` — the holding contract. Parameterised by `issuer : Party`, `holder : Party`, `instrumentId : InstrumentId`, `amount : Decimal`, `lockState : LockState`. The contract represents the holder's position in the instrument.
- `Transferable` — the transfer interface. Implements an `Initiate_Transfer` choice taking the proposed transferee, the amount, the proposed settlement asset, and the proposed settlement amount. The choice produces a transfer proposal; the proposal is accepted by the transferee and atomically settled via `AllocationV1`.
- `Allocatable` — the `AllocationV1` settlement interface from Splice. Atomic delivery-vs-payment is performed via this interface against any settlement asset that itself implements `Allocatable`.
- `Holding` — the holding-list interface, observable by parties with read access (the issuer, the holder, the issuer's auditor, the regulator party if configured).

— `PolicyGoverned` — the compliance-pack interface. Every transfer evaluates the asset's pack as a precondition.

11.3 Compliance-pack predicates for Pattern A

The compliance-pack engine accepts the following predicate categories for Pattern A:

— Eligibility — `transferee_is_credentialled`, `transferee_jurisdiction_in`, `transferee_not_sanctioned`, `transferee_is_accredited`.

— Quantity — `holder_limit_max`, `aggregate_holder_count_max`, `lot_size_min`, `position_concentration_max`.

— Temporal — `lockup_period`, `transfer_blackout`, `coupon_period_freeze`.

— Settlement — `settlement_asset_in`, `settlement_amount_min`, `settlement_amount_max`.

— Disclosure — `regulator_observability_threshold`, `audit_observability_always`.

Each predicate's exact semantics are reserved for the forthcoming BSP-0001 §4.

11.4 Primary issuance flow

1. The issuer instantiates an instance of the Tradeable Daml package, configured with the instrument's parameters and the compliance pack reference.
2. The issuer creates the initial `Token` holding contracts assigning the primary-issuance allocation.
3. Primary investors authenticate via TDIP-bound DIDs; their KYC and accreditation credentials are verified against the issuer's required-credential list.
4. Primary subscriptions execute as `AllocationV1` settlements: the issuer's `Token` holding is allocated to the investor party against the investor's settlement-asset position, atomically.
5. The compliance pack is evaluated on every allocation; primary-issuance lock-up rules apply if configured.

11.5 Secondary settlement flow

1. A holder initiates a transfer via `Initiate_Transfer`, specifying the transferee, amount, settlement asset, and settlement amount.
2. The proposal is signed by the holder; the transferee is offered the proposal as a Daml contract.
3. The transferee accepts; the runtime evaluates the compliance pack against the transfer parameters.
4. If the pack passes, the runtime executes the `AllocationV1` settlement: the holder's holding decrements, the transferee's holding increments, and the settlement asset moves in the opposite direction — all atomically.
5. The transfer is recorded on Canton; if the asset is mirrored, mirror updates are propagated downstream.

11.6 Corporate actions

Pattern A supports corporate actions via the following interfaces, each of which is implemented optionally:

- `Dividend` — the issuer distributes a dividend to the holder list, atomically against the issuer's payment of the settlement asset.
- `Redemption` — the issuer redeems a holding against payment.
- `Reorganisation` — the issuer reissues holdings under a new instrument identifier (e.g. for a corporate split or merger).
- `Coupon` — for fixed-income instruments, scheduled coupon payments.

Each corporate action is a Daml workflow; each evaluates the compliance pack; each settles atomically.

11.7 Mirror semantics for Pattern A

Mirrors of Pattern A assets implement:

- EVM — an ERC-20 contract with a transfer precondition that calls a Boli-operated cross-chain message bus; the message bus relays the proposed transfer to the canonical Canton ledger for compliance evaluation; the transfer either commits on both chains or neither.
- Solana — an SPL token with the same precondition pattern via a Solana program calling the message bus.

The mirror operator is identified by an issuer-signed Daml contract on Canton; the mirror's authority does not exceed the issuer's authority. Burn-to-redeem semantics return canonical holdings to the issuer's escrow.

---

12. Pattern B: Registry-mirror (specification)

12.1 Conformance

A Registry-mirror implementation will conform to the forthcoming BSP-0002 (Asset Pattern: Registry-mirror) once that BSP is published, by implementing the Daml interfaces defined therein.

12.2 Core interfaces

- `MirrorEntry` — the mirror contract. Parameterised by `registryParty : Party`, `holderParty : Party`, `recordIdentifier : Text`, `recordHash : Hash`, `attributes :`

`Map Text Text`. The contract represents the on-chain mirror of a single record in a sovereign register.

— `Transferable` — the holder-initiated transfer interface. Requires the joint authority of the registry party and the holder party for any state transition.

— `Encumberable` — the encumbrance interface. A separate Daml contract (lien, mortgage, restraint, caveat) is created against the mirror entry; the entry's transfer is blocked while encumbrances exist; the encumbrance's release requires its beneficiary's authority.

— `Correctable` — the registry-party-initiated correction interface. The registry retains operational control of corrections, audited by the chain-level transaction record.

— `Synchronisable` — the synchronisation hash interface. Each mirror entry carries a hash of the corresponding off-chain record; the synchronisation adapter is responsible for updating the hash when the off-chain record changes.

12.3 The synchronisation adapter

A registry-mirror deployment requires a synchronisation adapter operated by the registry authority or its technology partner. The adapter:

— Subscribes to the off-chain register's change feed (or polls, where a change feed is unavailable).

— For each change, computes the new record hash and submits a Daml transaction updating the mirror entry's `recordHash` field.

— Verifies that the on-chain mirror's holder party matches the off-chain record's title-holder; flags any divergence.

— Surfaces a public divergence dashboard showing any mirror entries whose hash has not been updated within the SLA window.

The adapter is the operational integration between the sovereign register and Canton. Its correctness is the registry authority's responsibility; the chain enforces the authority of the registry party to make changes, but does not enforce that the registry party's changes match the off-chain record. That match is the adapter's job.

12.4 Encumbrance lifecycle

An encumbrance against a Pattern B asset is a separate Daml contract. The lifecycle:

1. The encumbrance beneficiary (e.g. a mortgagee bank) and the holder party jointly sign an encumbrance contract referencing the mirror entry.
2. The mirror entry's transfer template observes the encumbrance — if any active encumbrance contracts exist, the transfer template's preconditions fail.
3. Release of the encumbrance is a Daml transaction signed by the beneficiary, optionally also by the holder if the contract requires it.

Once all encumbrances are released, the mirror entry is again transferable.

12.5 Correction semantics

Corrections — performed by the registry party to fix errors in the on-chain mirror — are a Daml transaction signed by the registry party alone. Corrections are observable to the holder; the contract's transaction history records the correction's reason, the registry-party signatory, and the resulting state.

A correction does not bypass encumbrances. If the mirror entry has active encumbrances, a correction that would change the entry's identifier or the holder party requires the encumbrance beneficiary's joint authority — preventing the registry from extinguishing a third party's interest unilaterally.

12.6 Mirroring Pattern B further

Pattern B assets are generally not mirrored to EVM or Solana, because Pattern B is itself a mirror. Where downstream tokenization is required (e.g. fractionalising a real-estate title for capital formation), the standard pattern is:

1. The Pattern B mirror entry remains canonical.
2. A Pattern A share class is issued, with the share class's compliance pack referencing the Pattern B entry's identifier.
3. The Pattern A share class is mirrored to EVM and Solana per §11.7.

This composition keeps the registry-mirror's authority structure intact while enabling distribution of derived instruments.

---

13. Pattern C: Credential (specification)

13.1 Conformance

A Credential implementation will conform to the forthcoming BSP-0003 (Asset Pattern: Credential) once that BSP is published, by implementing the Daml interfaces defined therein and emitting credentials conformant with the W3C Verifiable Credentials Data Model 2.0.

13.2 Core interfaces

— `CredentialRecord` — the on-chain anchor for a credential. Parameterised by `issuer : Party`, `holderDid : DID`, `credentialHash : Hash`, `issuedAt : Time`, `expiresAt : Optional Time`, `revocationStatus : RevocationStatus`. The record does not contain the credential's payload — that is held by the holder, off-chain — only the cryptographic anchor.

— `Issuable` — the issuer-initiated issuance interface. The issuer signs a credential payload (off-chain), publishes the corresponding `CredentialRecord` (on-chain), and delivers the signed payload to the holder via TDIP.

— `Revocable` — the issuer-initiated revocation interface. A revocation is a Daml transaction updating the `revocationStatus` field; verifiers observe the status atomically.

— `Presentable` — the presentation protocol. The holder presents a credential to a verifier off-chain, optionally with selective disclosure (zero-knowledge proofs over predicates of the credential payload). The verifier checks the presentation against the on-chain `CredentialRecord` to confirm the credential is current.

13.3 Selective disclosure

Pattern C credentials support selective disclosure over the credential's claims. The holder's wallet computes a presentation that reveals only the predicates a verifier requires, without disclosing the underlying claims.

Examples:

— A holder of an SL-UDI verifiable credential proves they are over 18 to a verifier without disclosing their date of birth.

— A holder of an MRV-verifier credential proves they are accredited to verify a specific methodology without disclosing their other credentials.

— A holder of a KYC-claim credential proves they passed KYC at a regulated bank without disclosing the bank's identity.

The cryptographic primitives are standard W3C VC selective disclosure (BBS+ signatures, in the current implementation).

13.4 Anchoring and resolution

The on-chain `CredentialRecord` serves as the revocation status registry and the issuance anchor. A verifier resolving a credential:

1. Receives the holder's presentation (off-chain).
2. Verifies the issuer's signature against the issuer's DID document.
3. Resolves the credential's `credentialHash` to the corresponding `CredentialRecord` on Canton.
4. Confirms `revocationStatus` is `Active` and the credential has not expired.

The on-chain step (3–4) is the difference between a credential that can be invalidated atomically and one that requires a round-trip to the issuer's CRL. For sovereign and regulated credentials, this matters.

13.5 Privacy

The Canton `CredentialRecord` is observable only to parties with read access — the issuer, the holder (via the holder's DID-bound party), and any verifier the holder has presented to. The credential payload itself is held off-chain by the holder, encrypted in their wallet. The revocation registry does not leak the credential's existence to the public.

Boli's choice of Canton for Pattern C is the same choice as for Pattern A and B: regulated credentials require privacy at the protocol level.

13.6 Mirror semantics for Pattern C

Credentials are not transferred — they are issued and presented. There is no mirror chain for credentials in the same sense as Pattern A. A holder's identity wallet can be on any chain; the wallet presents to a verifier; the verifier resolves the credential's status against Canton via TDIP.

Where a verifier on a non-Canton chain needs to programmatically check credential status, the TDIP bridge exposes a status oracle that emits revocation-status updates to subscribed mirror chains. The oracle is operated under TEE attestation; the canonical state remains on Canton.

---

14. Marketplace and intent routing

14.1 What the marketplace is

The Boli Marketplace is aggregated discovery and intent routing across listed partners. It is not a venue. It does not match orders. It does not custody assets. It surfaces what is available across the partners that have onboarded, routes a user's intent to the partner that can fulfil it, and observes the outcome for analytics and audit.

A "listed partner" is a Boli-onboarded entity — a licensed issuer, a licensed venue, a custodian, a registrar, or an identity provider — that has gone through Boli's onboarding process and elected to make some of its assets or services discoverable through the marketplace.

14.2 Permissionless to use; approval to list

Two distinct permissioning regimes:

— Use is permissionless. Any user with a TDIP-bound DID and the credentials required by the listed asset can browse, subscribe, transfer, or interact with marketplace listings, subject to the listed asset's compliance pack.

— Listing requires approval. A partner that wants to list assets or services on the marketplace goes through Boli's onboarding: legal entity verification, jurisdictional fit, asset-pattern conformance, compliance-pack review. The approval is documented; the listing is reversible.

This split mirrors the architecture of a stock exchange: anyone can buy a listed share (subject to KYC); listing the share requires the issuer to satisfy listing standards.

14.3 Intent routing

A user's interaction with the marketplace is expressed as an intent: "I want to subscribe to X", "I want to transfer Y to Z", "I want to verify the credential C". The marketplace:

1. Identifies the partner(s) that can fulfil the intent.
2. Routes the intent to the partner's integration endpoint.
3. Observes the outcome (settled on Canton, observable via the chain).

The routing is non-custodial. Boli does not take possession of the user's assets, settlement assets, or credentials at any point. The routing is informational; the execution is between the user and the partner, atomically on Canton.

14.4 Onboarding criteria

A partner seeking to list goes through:

— Legal verification — the partner's licensing in its jurisdiction, the legal entity's good standing, beneficial-ownership transparency.

— Asset-pattern conformance — the partner's assets must conform to one of the three Boli patterns. Bespoke asset structures are not listed; they are addressed by extending the patterns through the BSP process.

— Compliance-pack review — Boli reviews the pack for soundness, but does not approve its substance. The substance is the partner's regulatory choice.

— Operational readiness — the partner's integration is tested against Boli's reference platform, including failure modes.

The criteria are reserved for the forthcoming BSP-0006 (Marketplace listing criteria).

BOLI WHITEPAPER · PART-IV-OPERATING-MODEL

Part IV — Operating model

15. Boli Foundation and the $BOLI utility token

15.1 The Foundation

The Boli Foundation will be established ahead of the $BOLI token. The Foundation will govern the platform's treasury, the $BOLI token, and the broader ecosystem programmes. Its mandate is mission-driven, not profit-driven: the Foundation will fund research, ecosystem grants, integration support for sovereign and non-profit deployments, standards-process operations, and the infrastructure required to run the platform at the scale its users need.

The Foundation will be legally separate from the operating commercial entities that run the platform under licence, and from the Boli Association that publishes the standards. The Foundation's governance — its board composition, treasury policy, grant-making process — will be documented in the Foundation's charter, separately from this whitepaper.

15.2 The $BOLI utility token

The $BOLI token is the platform's utility token for ecosystem alignment. It is non-revenue-bearing — holding $BOLI does not entitle the holder to a share of platform revenue, asset issuance fees, or any other cash flow. It is not equity in any of the Boli legal entities.

What $BOLI does:

— Aligns ecosystem contributors. Grants, ecosystem incentives, and standards-process participation are denominated in $BOLI; long-term contributors accumulate $BOLI proportional to their contribution.

— Funds the mission treasury. The Foundation holds a treasury allocation of $BOLI and uses it to fund the activities described in §15.1.

— Provides utility within the platform. Specific platform interactions — onboarding, listing fees, MPC operator service fees, validator-tier benefits — denominate in $BOLI. These uses are payment-for-service, not investment.

The token is issued and governed by the Foundation. Its issuance schedule, treasury composition, and lock-up structure are specified in the Foundation's tokenomics document, separately maintained.

15.3 The regulatory line for $BOLI

The same boundary that governs the platform (§4) governs the token.

— $BOLI is not a security. The token does not entitle the holder to revenue, profit, or an equity claim. The Foundation does not represent it as an investment.

— $BOLI is not a settlement asset. Boli-issued tokenized assets settle against settlement assets the licensed party chooses — tokenized deposits, regulated stablecoins, CBDC, fiat-on-chain. $BOLI is not used as a settlement asset for tokenized regulated assets.

— $BOLI is not a governance share over assets. The token does not grant voting rights over assets issued by licensed parties through the platform. Those assets' governance rests with their issuers.

The narrow scope of $BOLI's utility is what allows the Foundation to operate it as an ecosystem-alignment instrument across jurisdictions.

---

16. Governance and contributing entities

16.1 The standards process

This whitepaper and the BSPs that accompany it are maintained by the Boli Association under a public process. The BSP repository lives at `github.com/boliassociation/bsps`; the only BSP currently published is BSP-0000, which specifies the standards process itself. BSPs 0001 through 0006 referenced throughout this whitepaper are reserved identifiers for forthcoming proposals.

The high-level lifecycle:

1. Authoring. A BSP is drafted by a contributor, in dialogue with the Association's research mandate. The author is named in the BSP.
2. Draft. The BSP is published in the BSP repository with status Draft. Open public comment.
3. Review. Once stable for implementation review, the BSP moves to status Review. Implementations may begin; breaking changes are flagged.
4. Final. When two independent implementations interop against the BSP and no breaking issues remain open, the Association moves the BSP to Final. The BSP is frozen.

5. Superseded. If a later BSP supersedes an earlier one, the earlier moves to Superseded with explicit reference forward.

The process itself is specified in BSP-0000.

16.2 Contributing entities

Three classes of contributor.

Tenzro Labs (Singapore) — infrastructure provider, Canton Foundation partner, operator of a Canton validator. Tenzro authors and reviews BSPs in the identity, key-handling, agentic-runtime, and validator-operation domains; Tenzro contributes the reference implementation of TDIP and the agentic runtime.

The Boli Platform team — the engineering organisation that ships the platform under licence from the Foundation. The Platform team authors BSPs in the Daml-pattern, compliance-pack, mirroring, and marketplace domains; the team contributes reference implementations.

External contributors — institutional integrators, sovereign technical staff, academic researchers, and other Canton ecosystem participants. External contributors author BSPs as their domain knowledge warrants; the BSP-0000 process is open to non-affiliated contributors with no special permission.

The Association does not gate contribution. It does gate the Final status — moving a BSP to Final requires the Association's review of the implementation evidence — but a contributor can author and progress a Draft and Review BSP without prior approval.

16.3 Relationship to the Canton Foundation and GSF

Boli's standards process composes with, and does not duplicate, the Canton Foundation's CIP process and the GSF's standards process. The relationship:

- Canton Improvement Proposals (CIPs) — protocol-level standards for Canton itself. Boli relies on CIPs (notably `cip-0056` for institutional-grade tokens) and contributes to them where Boli's deployment experience is relevant.
- Splice Token Standards — the token-layer standards governed by the GSF. The forthcoming BSP-0001 (Pattern A) will implement Splice Token Standard V1.
- Boli Standards Proposals (BSPs) — operating-layer standards specific to Boli. BSPs do not duplicate or replace CIPs or Splice standards; they extend them with Boli-specific conventions.

A BSP that would conflict with a CIP or a Splice standard is not finalised; either the BSP is revised to compose, or the relevant CIP/Splice change is proposed upstream and Boli waits for the upstream change.

16.4 Decision-making

Decisions on BSP status, on the platform roadmap, and on the Foundation's treasury policy are made by the bodies appropriate to each:

— BSP status decisions — the Boli Association, advised by an editorial board.
— Platform roadmap — the Boli Platform team, with input from the Foundation's grant-making and from licensed-party deployment experience.
— Foundation treasury — the Foundation board, under the Foundation charter.

Each body publishes its decisions transparently. The Association decides standards, the Foundation will decide treasury, and the Platform team decides implementation — the same separation-of-powers logic the Linux Foundation, the GSF, and Canton's contributing implementers operate under.

---

17. Roadmap and current status

17.1 What ships at v0.9

As of this whitepaper's draft date (May 2026), the platform's v0.9 state is:

— Daml asset patterns. Pattern A (Tradeable) and Pattern B (Registry-mirror) base packages are stable and shipped against Daml SDK 3.4.10 with Splice Token Standard V1. Pattern C (Credential) base package is stable for issuance and revocation; selective-disclosure presentation is shipped via the TDIP wallet client.
— Compliance-pack engine. v0.9 supports the predicate categories enumerated in §11.3 for Pattern A, with extension points for additional categories for Pattern B and C in subsequent BSPs.
— Multi-VM mirroring. EVM mirror is shipped against the canonical message-bus protocol. Solana mirror is in integration testing.
— TDIP identity bridge. The DID-to-Canton-party-id binding is shipped; the TDIP wallet client is integrated; the production canonical bridge is wired through Boli's web layer.
— Agentic runtime. Tenzro's runtime ships; on-chain mandate registration is integrated; TEE-attested execution is in pilot deployment.
— Marketplace. v0.9 supports listing, discovery, and intent routing for Pattern A assets; Pattern B and C marketplace surfaces are forthcoming.
— Pack catalog. Four minimal first packs against the asset bases (covering common Pattern A configurations for fund interests, real-estate share classes, environmental credits, and structured products) are available.

17.2 Current network posture

The platform's first launch is on the Tenzro testnet, with Canton mainnet wired from day one for the institutional integrations that require it. The operating company is incorporated in the UAE.

The launch posture is operational, not architectural. The platform's design does not depend on the operating company's jurisdiction; the architecture is portable, and the Foundation's jurisdictional anchoring is independent of the operating company's.

17.3 What is contemplated for v1.0

The v1.0 release is contemplated to include:

— Pattern C selective-disclosure at the on-chain anchor level, beyond the wallet-client level.

— Solana mirror in production.

— The marketplace extended to Pattern B and Pattern C.

— Additional BSPs finalising the bridge specifications (TDIP, mirroring), the marketplace listing protocol, and the agentic-mandate model.

— Sovereign deployments in production — at least one land-registry pilot under Pattern B, at least one environmental-MRV deployment under Pattern C, at least one tokenized real-estate share class under Pattern A.

The exact timing of v1.0 depends on the deployment readiness of the integrating parties, not on the platform's engineering velocity. The Association will publish a v0.95 interim once the Solana mirror and Pattern C selective-disclosure on-chain support are stable.

17.4 Open questions

The principal open questions for the Association at v0.9:

— Pattern composition for hybrid sovereign-and-financial assets. Some asset classes (governance-linked sovereign bonds, sovereign carbon-backed instruments) cross Pattern A and Pattern B in ways the current pattern boundaries handle by composition, but where direct hybrid patterns may simplify deployment. BSP work on this is at the early-draft stage.

— Mirror-chain selection beyond EVM and Solana. Whether and when to support additional mirror chains (Aptos, Sui, Stellar, others) is a tractable engineering question with non-trivial governance implications. The Association's current posture is to add mirrors in response to clear deployment demand, not speculatively.

— The wholesale-CBDC interaction model. Boli is settlement-asset-agnostic; the question of how the platform composes with multiple wholesale CBDC initiatives (mBridge, Project Agorá, Project Ensemble) without privileging any particular initiative is a live policy question, addressed by the Association's research programme.

— Privacy-preserving regulator observability. The compliance-pack engine supports regulator observability under explicit configuration, but the broader question of how regulators'

supervisory needs are served on a privacy-preserving rail — without weakening the privacy properties for the supervised entities — is a live cryptographic and policy question. Active work, with academic partners, is under way.

BOLI WHITEPAPER · BACK-MATTER

Back matter

References

The references below are non-exhaustive; the BSPs referenced in Part III contain their own complete reference lists.

Canton and GSF - Canton Network Documentation. `canton.network/docs`. - Splice Token Standard V1. `canton-network/splice` repository, GSF. - Global Synchronizer Foundation. `canton.foundation`. - Linux Foundation Decentralized Trust. `lfdecentralizedtrust.org`.

Daml - Daml SDK 3.4 Documentation. `docs.daml.com`. - Digital Asset, Daml — A Smart Contract Language for Distributed Multi-Party Applications.

W3C standards - Decentralized Identifiers (DIDs) v1.0. W3C Recommendation. - Verifiable Credentials Data Model v2.0. W3C Recommendation.

Tenzro - Tenzro Network documentation. `tenzro.com/docs`. - TDIP specification. (Forthcoming, to be linked at BSP-0005 finalisation.) - Tenzro open-source SDKs and infrastructure. `github.com/tenzro/`.

Boli - Boli Standards Proposals repository. `github.com/boliassociation/bsps/`. - Boli Whitepaper repository. `github.com/boliassociation/whitepaper/`. - Boli Platform. `boli.technology`. - Boli Association. `boli.org`.

Institutional tokenization precedents - DTCC and Digital Asset, Treasury tokenization on Canton. December 2025. - HKMA, Project Ensemble — wholesale CBDC pilots. November 2025. - HKMA, HK$10 billion digital green bond issuance. Late 2025. - BIS, Project Agorá — testing phase. January 2026. - MAS, Project Guardian. Ongoing. - RBI, e-Rupee — production deployment statistics. 2026. - Project mBridge, cumulative cross-border transactions. 2026.

Glossary

`A llocatable` — the Daml interface in Splice Token Standard V1 that an asset implements to participate in `AllocationV1` atomic settlement.

`AllocationV1` — the Splice Token Standard V1 model for atomic delivery-vs-payment on Canton. The transfer of an asset and the corresponding transfer of a settlement asset are a single atomic action.

BSP — Boli Standards Proposal. A versioned specification document maintained in the `boliassociation/bsps` repository.

Canton — the privacy-preserving public blockchain that is the settlement layer for the Boli Platform. Operated as a federation of named institutions; governed under the Linux Foundation through the Global Synchronizer Foundation.

Compliance pack — the chain-level encoding of a licensed party's transfer policy. A precondition of every transfer; written in a structured policy DSL that compiles to Daml.

Daml — a smart-contract language for multi-party workflows; the contract language used on Canton.

DID — W3C Decentralized Identifier. The naming primitive for parties on the Boli Platform, anchored via TDIP.

GSF — Global Synchronizer Foundation. The Linux Foundation–hosted body that governs the Canton synchroniser layer and the Splice Token Standards.

MRV — Monitoring, Reporting, Verification. The data lifecycle for environmental credits.

Pattern A / B / C — the three Boli Daml asset patterns: Tradeable, Registry-mirror, Credential.

Splice Token Standard V1 — the Canton ecosystem's institutional token standard, governed by the GSF; to be implemented by the forthcoming BSP-0001 (Pattern A).

TDIP — Tenzro Decentralized Identity Protocol. The identity foundation Tenzro Labs ships; provides DIDs, Verifiable Credentials, and the Canton party-id bridge.

TEE — Trusted Execution Environment. A hardware-attested execution context used by the agentic runtime to produce verifiable execution attestations.

$BOLI — the Foundation's non-revenue-bearing utility token; aligns ecosystem contributors and funds the mission treasury.

Change log

VERSION	DATE	NOTES
v0.9	May 2026	First draft for review. Parts I–IV complete; BSP-0000 published, BSPs 0001–0006 referenced as reserved identifiers for forthcoming proposals.

Contributors

The whitepaper is authored by the Boli Association with contributions from the Boli Platform team, Tenzro Labs, and external reviewers. A complete contributor list will be maintained at the v1.0 release; comments on this draft are invited via the `boliassociation/whitepaper` repository's issue tracker.

Boli Association — Switzerland. Maintained at `github.com/boliassociation/whitepaper`, alongside the BSP repository at `github.com/boliassociation/bsps`. Licensed CC BY 4.0.