Bitcoin Asset Protocol (BAP)

White Paper v0.1

Author: Lorenzo Sebastían Wynberg Carmona

LORENZO SEBASTIAN WYNBERG CARMONA (FIRMA)

Digitally signed by LORENZO SEBASTIAN WYNBERG CARMONA (FIRMA)
Date: 2025.10.02 16:27:34 -06'00'

---

1. Abstract

A universal, peer-to-peer registry would allow assets, governance, and proofs to operate directly on top of Bitcoin **without creating a new monetary token or base-layer consensus system.** Instead, BAP inherits security from Bitcoin and adds a coordination layer on top. While Bitcoin provides incorruptible money and truth through proof-of-work, it does not natively support asset issuance, verifiable governance, or collective coordination.

We propose a protocol that anchors event logs into Bitcoin, using satoshis as the only unit of payment and security. Nodes perform useful work — validating events, sequencing them into blocks, and anchoring Merkle roots — forming a record that cannot be changed without redoing the work and re-anchoring into Bitcoin. Security rests on the same majority-honest assumption as Bitcoin: as long as most nodes perform their tasks faithfully, the ledger resists tampering, and once a root is confirmed on Bitcoin, history cannot be rewritten without also attacking Bitcoin itself. The network requires minimal structure; nodes can enter or leave freely, while reputation, bonds, and optional identity proofs can strengthen accountability as the system matures.

---

2. Introduction

The Internet has enabled global communication and exchange, but coordination at scale still relies on platforms that issue their own tokens and operate under their own rules. While these systems can provide asset issuance, governance, and proofs, they suffer from the weaknesses of the token-based model: inflationary currencies, fragile governance, and trust assumptions that fall short of Bitcoin's durability.

Bitcoin solved the problem of incorruptible money. **Bitcoin the coin** is 21 million units of digital scarcity, divisible into satoshis, secured by the most powerful proof-of-work system ever deployed. **Bitcoin the**

network is the most secure distributed ledger in history, but it was intentionally designed to be minimal: it records BTC transfers, and nothing more. This narrow scope has preserved Bitcoin's resilience, yet it leaves a gap. There is no native way to issue assets, to run verifiable governance, or to anchor collective proofs directly into Bitcoin.

What is needed is a neutral, universal, decentralized notary: a layer that records the existence and sequence of events, secured by Bitcoin's proof-of-work and paid for only in sats.

The Bitcoin Asset Protocol (BAP) provides this layer. It anchors event logs directly into Bitcoin, using sats as the sole unit of payment and security. Nodes perform useful work — validating client events, sequencing them into blocks, anchoring Merkle roots into Bitcoin, and executing treasury payouts. The result is a universal registry: a cryptographic notary that can issue assets, organize governance, and anchor proofs, without introducing new tokens or new consensus assumptions.

In this way, BAP builds on Bitcoin's incorruptible money to provide incorruptible records. Where Bitcoin perfected digital cash, BAP establishes the universal notary for assets and coordination — a transparent, tamper-resistant registry fully interoperable with Bitcoin itself.

---

3. Design Philosophy

The Bitcoin Asset Protocol (BAP) is built on a simple principle: **preserve Bitcoin's security, extend its utility**. Bitcoin remains the monetary foundation; BAP provides the coordination layer on top.

BAP makes three uncompromising commitments:

1. **Bitcoin is untouched.** The base layer remains unchanged — no forks, no patches, no new opcodes. Bitcoin continues to do what it does best: secure money through proof-of-work.

2. **Sats are the only currency.** No new tokens are introduced for consensus or gas. Every action in BAP is paid in sats, aligning the protocol directly with Bitcoin's monetary standard.

3. **Extensibility without fragility.** BAP is designed to grow without breaking. All functionality is expressed as deterministic, typed events. New features — whether financial (assets, stablecoins, derivatives), organizational (governance, voting, DAOs), or operational (supply-chain attestations, digital IDs, copyrights) — are added by defining new event types. This modular approach allows the protocol to evolve organically, without rewriting consensus rules.

This combination of **minimalism and extensibility** ensures that BAP does not compete with Bitcoin, nor attempt to reinvent it. Instead, it complements Bitcoin by providing the missing coordination layer — one that adapts to new use cases while remaining firmly rooted in the most secure monetary network ever created.

---

4. Proof of Useful Work

Bitcoin solved double spending with proof-of-work: a distributed timestamp server that proves when transactions existed, making it infeasible to spend the same coin twice. Each block was both a time marker and a seal of economic cost.

BAP extends this principle from transactions to events. Every action — creating an asset, casting a vote, logging a supply-chain checkpoint, oracle update, etc. — is an event. The "work" in BAP is not raw hash-searching for difficulty targets, but the deterministic labor the network requires to stay coherent: validating events, sequencing them into blocks, serving them for audit, and anchoring their Merkle roots into Bitcoin. This is Proof of Useful Work: Bitcoin's security model applied to event consensus and coordination.

This is **Proof of Useful Work**: every epoch produces a timestamped log of peer-validated events, made irreversible once anchored to Bitcoin's proof-of-work.

4.1 Nodes as Bonded Workers

Participation requires a bond, slashable for misbehavior. Bonded nodes perform the tasks that keep consensus alive:

- **Sequencers** order validated events into blocks.
- **Validators** re-check signatures, nonces, and rules before attesting.
- **Data Availability providers** serve block bodies and proofs for auditability.
- **Anchorers** commit Merkle roots into Bitcoin at epoch boundaries.

The bond ensures dishonesty carries cost (invalid blocks, withheld data, conflicting anchors), while honest performance earns rewards through fee distribution.

4.2 Events, STATE, and AUX

All BAP activity reduces to events, but events play two different roles:

- **STATE events** define authoritative facts: who owns what, which keys control it, what supply exists.
- **AUX events** express processes and context: recoveries, votes, timelocks, checkpoints, or oracle feeds.

STATE is the canonical record. AUX is the interaction layer around it.

4.3 The Always-Conclude Rule

Within AUX, there are two categories:

- **One-off informational events** (oracle prices, certifications, supply-chain checkpoints) that are final once included in a block.
- **Chained processes** (recoveries, DAO votes, conditional transfers) that open state which must deterministically resolve.

For chained processes, the rule is strict: they must end in exactly one terminal outcome:

- **SUCCESS** — quorum reached, conditions satisfied.
- **FAIL** — conditions not met, invalid, or deadline expired.

Agents guarantee closure. They watch open processes, emit the conclusion event, and claim a bounty for valid resolution.

This guarantees:

- **Determinism** — processes always close.
- **Integrity** — no dangling outcomes that could enable double claims.
- **Auditability** — every resolution carries a reason code, leaving a timestamped trail.

By requiring chained processes to always conclude, BAP generalizes Bitcoin's core innovation: just as a coin cannot be spent twice, a recovery cannot remain unresolved twice, and a vote cannot be both open and closed. There is always one canonical outcome, timestamped and sealed.

4.4 Incentivized Useful Work

Fees in sats are distributed to those who perform useful work:

- Sequencers, validators, and DA providers are rewarded for block production and attestation.
- A dedicated Anchor Pool covers Bitcoin fees.
- Agents are rewarded for valid conclusions, regardless of outcome.
- Complex processes (like governance votes) escrow extra fees to guarantee closure.

Nothing is burned or wasted. Every sat funds useful work that the system requires.

4.5 Anchoring and Double-Spend Resistance

At the end of each epoch, Merkle roots of blocks are anchored into Bitcoin by quorum signature. Anchoring is both timestamp and seal:

- **Validator consensus** ensures no two conflicting events (two transfers of the same units, two votes with the same nonce) can both be accepted.
- **Bitcoin anchoring** ensures that once a block is confirmed on-chain, its order cannot be reversed without both redoing the useful work and attacking Bitcoin itself.

Together, validator consensus + Bitcoin anchoring prevent the digital equivalent of double spending — not only for money, but for all event types. Assets cannot be transferred twice, votes cannot be duplicated, and proofs cannot be replayed. Each event has exactly one canonical timestamp, secured by Bitcoin's chain of time.

---

5. The Event Machine

At the heart of BAP is the **event machine**. Every action — creating an asset, transferring units, casting a vote, recording a checkpoint — is represented as a typed **event**. Events form the atomic language of the protocol. By reducing everything to events, BAP achieves extensibility without fragility: new features are added by introducing new event types, not by rewriting consensus rules.

5.1 Everything Is an Event

An event is a structured record, defined by a schema (type, inputs, outputs, proofs, metadata).

Examples include:

- **ASSET.CREATE** — define a new asset with metadata, controllers, supply rules.
- **ASSET.TRANSFER** — move asset units from one party to another.
- **ASSET.TRANSFER_WITH_FEES** — transfer that distributes a portion of value to a designated fee sink (e.g. treasury, royalties).
- **ASSET.TRANSFER_CONTROLLER** — change the controller set of the asset itself (rotate multisig, add/remove signers).
- **GOVERNANCE.VOTE** — cast a vote tied to a proposal.

- **TREASURY.PAYOUT** — release funds under defined conditions.
- **ASSET.SUPPLYCHAIN.CHECKPOINT** — log audit or transport data without concluding state.

Events are validated deterministically, sequenced into blocks, and anchored to Bitcoin at regular epoch intervals.

5.2 STATE and AUX

Events fall into two complementary tracks:

- **STATE**: the canonical record of ownership and control. It defines who holds an asset, what its supply is, and which keys can act. Rights and obligations are updated here directly through conclusive events such as transfers or controller rotations.
- **AUX**: the logic and interaction layer. It captures processes and context — votes, recoveries, timelocks, or checkpoints — that reference or eventually modify STATE. AUX events come in two forms:

  • **Processes** — multi-step flows (e.g., a DAO vote, recovery attempt, conditional transfer) that must eventually resolve into a terminal outcome (SUCCESS or FAIL).

  • **Informational updates** — one-off logs (e.g., oracle data, supply-chain checkpoints) that are final once included in a block, without requiring resolution.

This separation mirrors law and execution: **STATE is the authoritative record of rights, while AUX provides the process layer and context around them.** Processes always resolve deterministically; informational AUX simply enriches the ledger without opening long-lived claims.

5.3 Composability

Events are modular and composable. They can be chained together to form higher-level processes:

- A GOVERNANCE.VOTE feeds into a TREASURY.PAYOUT.
- A RECOVERY.SIGN feeds into a RECOVERY.CONCLUDE.
- A SUPPLYCHAIN.CHECKPOINT can trigger a conditional ASSET.TRANSFER.

This design makes BAP a **Lego system for coordination**: builders don't need new blockchains or contract languages, only new event types that follow open schemas.

5.4 Developer Experience: JSON, Not VMs

Unlike platforms that require new scripting languages, BAP is **data-first**. Events are defined in JSON schemas that any developer familiar with APIs can use. To build on BAP is to emit valid events; no custom VM or bytecode is required.

This lowers the barrier for enterprises, SaaS developers, and governments: they can treat BAP like an **event log with Bitcoin-backed finality**.

5.5 Extensible by Design

Unlike Bitcoin's soft-fork signaling, where old nodes may continue following obsolete rules at their own risk, in BAP consensus is explicit. Once an event type or parameter is activated by Protocol AUX and node quorum, all bonded nodes must enforce it. Nodes that fail to update or enforce the confirmed ruleset are excluded from consensus and may be slashed if they attest incorrectly. This guarantees one canonical registry, with no forks.

The event machine is open-ended:

- **New event types and event chains** can be standardized (similar to ERC proposals).
- **No base-layer fork** is required: the foundation (STATE/AUX, bonding, anchoring, mempool rules) remains unchanged.
- **Capability bits**: every node advertises a bitmap of the event types it can fully validate.
- **Unknown events**: until upgraded, a node may relay them as opaque data but must not attest to blocks whose validity depends on rules it cannot check.

5.5.1 Activation workflow for new types

1. **Proposal** — a GOVERNANCE.PROPOSE_EVENT_TYPE defines the TypeID, schema, and criticality (core or optional).
2. **Signaling** — nodes set a capability bit preview (willing to support) and publish it on AUX.
3. **Activation** — an activation epoch E* is chosen by governance, with a grace window G epochs.

5.5.2 By criticality

- **Optional types**: may appear anytime; quorums for blocks containing them are computed only among validators that advertise support. Non-supporters abstain (not slashed).

- **Core types**: after E* + G, any block containing a core-type event requires validator attestations from a quorum that supports that TypeID. Validators that do not support the type must abstain; if they attest blindly, they are slashable.

5.5.3 Slashing conditions

- **False attestation**: validator attests to a block containing TypeID X after E* + G without advertising support for X.
- **Misrepresentation**: validator advertises support for X but fails to enforce its schema/semantics (provable via fraud proof).
- **Invalid inclusion (sequencer)**: sequencer includes an invalid event of TypeID X while advertising support.
- **Anchor insufficiency (anchorer)**: anchorer commits a block whose filtered attestation set (supporting X) does not meet quorum.

This model keeps the protocol evolutionary: nodes can upgrade on a schedule or abstain; slashing only applies to dishonesty, not to lateness.

5.6 Finality

All validated events flow into blocks and epochs whose Merkle roots are anchored to Bitcoin; see §7.5 for cadence and finality.

---

6. Assets on BAP

The Bitcoin Asset Protocol is named for its most important use case: **assets as first-class citizens**. Assets on BAP are not contracts bolted onto a chain, but dedicated event lanes with their own identity, rules, and continuity. By modeling assets directly in the protocol, BAP provides a foundation for stablecoins, securities, DAOs, supply-chain records, digital identities and more — all inheriting Bitcoin's security while remaining fully interoperable.

6.1 Assets as First-Class Citizens

Each asset maintains its own **STATE lane**, a canonical chain of events that defines:

- **Ownership** — who currently controls it.
- **Controllers** — the keys with signing authority.
- **Metadata** — descriptive fields like name, category, supply, and divisibility.
- **Recovery setup** — guardians and backup keys for restoring access.

Assets are self-contained: their **rules and metadata** live in **STATE**, while their **logic and interactions** are expressed in **AUX tracks**.

6.2 STATE and AUX for Assets

- **STATE** anchors the facts: ownership, supply, control keys.
- **AUX** expresses processes around those facts: governance votes, recoveries, conditional transfers, or open-ended logs like supply-chain checkpoints.

This separation keeps assets deterministic and auditable: STATE defines what is true, AUX records how other specific actions came to be.

6.3 Keys and Roles

Control is cryptographic and flexible:

- **Controller keys** — primary authority, able to sign transfers and updates.
- **Guardian keys** — secondary authority for recovery, co-signing reassignment attempts.
- **Asset key** — a derived identity for the asset itself, formed from its controllers (e.g., via multisig or threshold schemes). The asset key rotates automatically when controllers change.

Asset keys allow organizations to act as single entities: a DAO can control a treasury, own intellectual property, or sign certificates, even as its membership evolves.

6.4 Asset Creation

Creating an asset is deterministic. The initiator defines metadata, controller keys, guardians, recovery rules, divisibility, and supply cap. An asset key is derived automatically from the controller set.

Examples:

- A **DAO share asset** with fixed supply and decimals.
- A **real estate title asset** with one indivisible unit which could belong to an organization.
- A **music royalty asset** divisible into multiple whole units or micro-units, with supply capped for predictable payouts.

6.5 Recovery and Continuity

Assets may be recovered. Guardians and backup keys provide redundancy, and every recovery attempt must resolve deterministically with **SUCCESS** or **FAIL**. Agents finalize these processes, and the asset key ensures organizational hierarchies remain intact — sub-assets always follow their parent's control.

6.6 Interoperability and Composition

Assets are composable like events themselves:

- One asset can reference another (e.g., a DAO vote triggering a treasury payout).
- Composite assets can bundle treasuries, licenses, or governance rights.
- Nested hierarchies emerge naturally: an organization controls a treasury; the treasury controls shares; the shares link to licenses or contracts.

This model turns assets into **building blocks for higher-order coordination**, without the need for complex contract languages.

6.7 Finality

Asset state changes inherit Bitcoin's finality via anchoring of epoch roots (§7.5).

---

7. Consensus & Security Model

BAP's consensus is modular and layered. Each role performs specific useful work, every event is checked multiple times before inclusion, and every block is ultimately anchored to Bitcoin. The result is a deterministic, auditable pipeline that is secure as long as the majority of nodes perform their roles honestly.

7.1 Roles in the Network

Consensus is divided across specialized but overlapping roles. A single node may run multiple roles, but separation improves resilience:

- **Gateways** — accept events from users, perform basic pre-validation, and forward them to the mempool.
- **Validators** — independently check events and blocks against protocol rules.
- **Sequencers** — order validated events into candidate blocks on a rotating or weighted schedule.
- **Anchorers** — commit Merkle roots of blocks into Bitcoin at epoch boundaries.
- **Agents** — monitor AUX processes and emit deterministic conclusions (e.g., votes, recoveries, timelocks).

Together, these roles form a pipeline where no single actor has unilateral control.

7.2 The Mempool & Attestation Layer

The **mempool** is the staging ground for events. Before an event is accepted into consensus, it is validated twice:

1. **Gateway pre-validation**
    - Schema check (is the event well-formed and typed?).
    - Signature check (does it match an authorized controller key?).
    - **Nonce check** (does the event carry the next expected nonce for that key, preventing replay or out-of-order actions?).
2. **Mempool validation (peer validators)**
    - Validators cross-check the same fields independently.

    - Invalid or replayed events are dropped.
    - Only events confirmed by multiple validators remain in the pool.

This layered approach prevents a malicious gateway from injecting bad data and ensures all sequencers draw from the same validated pool.

7.3 Sequencing & Block Formation

Sequencers select from the validated mempool and order events into blocks. Block assembly follows clear rules:

- Only validated events may be included.
- Nonces must advance correctly for each account.
- Ordering respects fairness (round-robin or weighted rotation).

The block is then proposed to validators for attestation.

7.4 Block Revalidation & Attestation

Before anchoring, the block itself is revalidated:

- Are all included events valid according to protocol rules?
- Are nonces consistent across accounts?
- Are quorum thresholds or predicates satisfied?

Validators that confirm correctness **attest** to the block. Attestation is required for a block to advance.

7.5 Anchoring to Bitcoin

Once attested, the block's Merkle root is committed to Bitcoin by anchorers. Anchoring cadence balances:

- **Cost efficiency** — batching events into epochs reduces on-chain fees.
- **Latency** — regular anchors provide timely finality.

Validity requirements: an anchor is accepted in BAP only if the root was pre-published, carries validator attestations ≥ threshold, and is signed by the designated anchor quorum.

Finality policy: anchors are accepted immediately upon inclusion; *economic finality* is reached after *N* Bitcoin confirmations (tunable by policy), with clear handling of Bitcoin reorgs.

After sufficient Bitcoin confirmations, history becomes economically irreversible — rewriting it would require not only subverting BAP but also attacking Bitcoin itself.

7.6 Consensus Flow

1. **Submit** — a user signs and submits a JSON-typed event to a gateway.
2. **Pre-Validate (gateway)** — schema, signature, and nonce are checked before forwarding.
3. **Validate (mempool)** — peer validators independently confirm event validity.
4. **Sequence** — sequencers assemble validated events into a candidate block.
5. **Revalidate (block)** — validators re-check the block as a whole and attest.
6. **Anchor** — anchorers commit the block root into Bitcoin.
7. **Conclude** — agents close AUX processes into deterministic SUCCESS or FAIL, updating STATE.

This flow ensures **determinism at every step** and **irreversibility once anchored**.

7.7 Security Assumptions

BAP inherits Bitcoin's model: security rests on majority-honest participation.

- **Majority-honest nodes** — as long as most bonded nodes perform their roles faithfully, consensus advances correctly.
- **Bonding & slashing** — dishonest behavior (double-signing, replay, withholding DA) results in bond loss.
- **Rotation & diversity** — sequencers and validators rotate, preventing capture.
- **Reputation** — long-lived correct participation strengthens trust and reward share.
- **Sybil resistance** — bonds and attestations make cheap identities uneconomical.

- **Anchoring** — once on Bitcoin, history cannot be rewritten without also attacking Bitcoin.

---

8. Treasury & Incentives

Consensus in BAP is not secured by wasted computation but by **directed economic incentives**. Every sat paid into the system funds the useful work that sustains it: validation, sequencing, anchoring, and process resolution. Nothing is burned; nothing is wasted.

8.1 Fee Model

All events in BAP include a fee, denominated in satoshis.

- Fees are proportional to the useful work required, not arbitrary "gas."
- Every sat flows back to the network actors who performed that work.
- Philosophy: **every sat is productive.**

8.2 Bonds and Participation

To join consensus, nodes post a **bond** — collateral that aligns incentives with honest participation.

- Misbehavior (double-signing, censorship, invalid anchors) is penalized by slashing.
- Bond size scales with network maturity: smaller at genesis, larger as throughput grows.
- Bonds ensure that dishonesty carries economic cost, while honest participation earns steady rewards.

8.3 Role-Based Rewards

Collected fees flow into a **treasury pool**, distributed to roles according to measurable contributions:

- **Validators** — attestation and revalidation.
- **Sequencers** — block assembly.
- **Anchorers** — committing roots into Bitcoin.
- **Agents** — deterministically resolving AUX processes.
- **Gateways** — optional share for user-facing infrastructure.

👉 *Illustrative distribution:*

- 40% Validators
- 25% Sequencers
- 20% Anchorers
- 10% Agents
- 5% Gateways

(Percentages are subject to governance and market calibration.)

8.4 Vesting & Clawbacks

Rewards are not spendable immediately. They vest over a short horizon (e.g., 1–2 epochs), creating a window for:

- **Fraud proofs** — dishonest actions can still be challenged.
- **Clawbacks** — if a node is slashed during this window, pending rewards are forfeited.

This ensures accountability without penalizing honest nodes.

8.5 Anchoring Bonuses

Because anchorers incur direct Bitcoin fees, the protocol provides **bonuses for consistent, timely anchoring**.

- Healthy anchors are reimbursed and rewarded.

- If anchorers fail, the treasury can subsidize the next anchor to ensure liveness.

8.6 Economic Philosophy

BAP's treasury model reflects three principles:

1. **Every sat is productive** — fees directly fund useful work.
2. **No waste** — unlike proof-of-work mining, no energy is externalized; all value stays inside the system.
3. **Self-balancing** — as usage grows, total rewards grow; as node count grows, rewards are split thinner. This keeps margins healthy without creating runaway profits.

---

9. Launch & Decentralization Roadmap

BAP is designed for a **mature, permissionless state** where anyone can participate honestly by bonding sats and performing useful work. Reaching that state requires staged decentralization, so that security mechanisms can be tested, hardened, and scaled responsibly.

9.1 Genesis Phase (Bootstrapping)

The network begins with a **small validator set** operating in a development-run mode.

- **Symbolic bonds** — minimal collateral is required, mainly to establish mechanics.
- **Active fees and rewards** — kept modest, but functional from day one.
- **Primary goal** — test anchoring, block flow, AUX logic, and agent behavior under controlled conditions.

Early builders — SaaS developers, DAO experiments, supply-chain pilots — begin integrating assets and testing real-world use cases.

9.2 Transition Phase (Progressive Decentralization)

Once the mechanics are proven, decentralization broadens:

- **Identity & reputation** — dev -> mainnet passports and zk attestations help establish trust in new nodes.
- **Warm-up period** — new participants first run on testnet before mainnet promotion.
- **Bond scaling** — collateral requirements increase with network maturity, creating stronger economic skin in the game.
- **Committee expansion** — sequencers and validators rotate more frequently, lowering centralization risk.
- **Treasury sophistication** — vesting and slashing mechanisms activate fully.
- **Governance maturing** — AUX-based votes, recoveries, and DAO-like flows become standard, finalized by agents.

9.3 Mature Phase (Full Permissionless State)

In its end state, BAP operates as a **fully open system**:

- **Permissionless entry** — any participant can bond and join consensus.
- **Randomized committees** — sequencing and validation rotate unpredictably, ensuring diversity and Sybil resistance.
- **Decentralized treasury** — distribution and parameters governed by BAP-native processes.
- **Anchoring rotation** — duties spread widely, with redundancy for resilience.
- **Global diversity** — nodes across regions ensure no single entity or jurisdiction can dominate.
- **Continuous evolution** — new event standards emerge and are adopted through open coordination, never forks.

9.4 Philosophy of the Roadmap

The roadmap reflects a balance between pragmatism and principle:

- **Security-first** — begin small, prove mechanisms, then scale.

- **Progressive openness** — guardrails like passports, warm-ups, and scaled bonds prepare the system for full permissionless entry.

- **Decentralization as a spectrum** — the network evolves from Genesis → Transition → Mature, just as Bitcoin itself grew from a handful of nodes into a global base layer.

BAP's goal is not speed of launch but durability of trust. With careful bootstrapping, the system moves from fragile beginnings to a robust, incorruptible state — anchored to Bitcoin, paid in sats, and open to all.

---

10. Use Cases

10.1 Governance & Voting

- DAOs, cooperatives, or organizations can run **governance flows** natively on BAP.

- AUX tracks handle proposals, votes, and quorum checks.

- Agents finalize outcomes and update STATE (e.g., rotating controllers, vote finalizations, etc.).

- All votes are anchored to Bitcoin, giving incorruptible audit trails.

10.2 Stablecoins & Synthetic Assets

- Assets can represent **stablecoins** pegged to fiat or commodities.

- Controllers can be custodians, or decentralized governance can manage collateralized models.

- Synthetic assets (e.g., gold-pegged tokens, index baskets) can be created with transparent supply caps and recovery rules.

10.3 DeFi Primitives

- **AMMs (Automated Market Makers)**: trading pools represented as assets with logic defined in AUX.

- **Lending**: assets can be locked as collateral, with AUX flows tracking repayments and liquidations.

- **Derivatives**: structured instruments (futures, options) encoded as AUX events.

- All **anchored** into Bitcoin for settlement certainty.

10.4 Metadata Proofs

- Assets may carry **metadata commitments** (hashes, Merkle roots).
- Use cases:
    - **Copyright & IP** – provable authorship and licensing.
    - **Medical Records** – timestamped proofs without revealing private data.
    - **Supply Chain** – provenance checkpoints for goods.
- AUX tracks allow ongoing updates without bloating STATE.

10.5 Recovery & Safety

- Native recovery flows with **guardians and quorum signatures**.
- Flexible: supports Shamir secret-sharing, multisig guardian sets, or SaaS-managed recovery services.
- Expired or failed recoveries are logged as deterministic outcomes, ensuring clarity and safety.

10.6 NFTs & Collectibles

BAP supports issuance of **unique or semi-fungible assets** directly, without contracts. These represent not files, but **verifiable proofs of uniqueness and ownership**, anchored into Bitcoin.

- **Collectibles & Creative Works** — artists, musicians, or game developers can mint NFTs as proofs of authenticity. The underlying media (art, audio, game item) is stored off-chain, while its hash and metadata are anchored in BAP.
- **Official Titles & Certificates** — governments and universities can issue NFTs for birth certificates, diplomas, licenses, or credentials. Each title is an event chain with metadata and hashes, recoverable and verifiable without storing sensitive data on-chain.
- **Organizational Control** — issuing institutions maintain controller keys; recoveries allow revocation/reissuance if keys are lost or data must be corrected.

- **Composability** — unlike isolated ERC-721 tokens, BAP NFTs can interact with governance, treasuries, and AUX flows (e.g., voting restricted to verified diploma-holders).

10.7 Experimental Tokens

BAP also enables fast, lightweight issuance of **fungible tokens** for culture, community, or experimentation:

- **Experimental Economies** — tokens for games, loyalty programs, or small markets can be created and retired with minimal cost.
- **Memecoins** — community-driven tokens are as simple as an ASSET.CREATE with supply metadata; no custom contracts required.
- **Low Fees** — issuance and transfer fees are small enough to support viral or grassroots adoption.
- **Composability** — meme tokens can integrate with DAOs, treasuries, or AUX logic (votes, bonding curves), while inheriting anchoring, recovery, and auditability from Bitcoin.

---

11. Metrics & Power of the Network

11.1 Security Metrics

- **TBAR (Time Between Anchored Roots)** – average time between Bitcoin anchors. A shorter TBAR = higher security cadence.
- **Cost of Capture** – economic cost to compromise a quorum of nodes, factoring in bonds, slashing, and treasury size.
- **HCP (Honest Consensus Probability)** – probability that a randomly selected committee remains honest given bond distribution and node diversity.
- **Operator Concentration** – percentage of consensus power held by the top X validators/sequencers. Lower concentration = stronger decentralization.

11.2 Throughput Metrics

- **EPS (Events per Second)** – base unit of throughput; how many validated events the network processes.
- **Attestations/sec** – measures revalidation speed and mempool health.
- **Blocks/minute** – how frequently sequencers finalize blocks.
- **Anchors/hour** – cadence of roots being committed into Bitcoin.

These metrics together show BAP's **real throughput and settlement efficiency**.

11.3 Latency Targets

- **Block inclusion latency** – time from user event → sequenced block.
- **Validation latency** – time from submission → cross-node attestation.
- **End-to-end finality** – time from user event → anchored into Bitcoin.
- Targets can scale with network load: e.g., sub-10s block inclusion, sub-60s BAP finality, ~30min BTC finality.

11.4 Transparency & Dashboards

- BAP prioritizes **radical transparency**.
- Real-time dashboards expose:
    - Node participation, bond sizes, and slashing events.
    - Treasury inflows/outflows.
    - Live EPS, latency, and anchor cadence.
- Users, SaaS builders, and auditors can monitor trust **without black boxes**.

11.5 Power of the Network

Together, these metrics demonstrate:

- Security is measurable (not hand-waved).
- Throughput is transparent and auditable.
- Latency targets give users confidence in UX.
- Dashboards keep power in the hands of the community, not hidden operators.

The result: **a network where trust is earned continuously, visible in real time, and rooted in Bitcoin's incorruptibility.**

---

12. Feasibility

The Bitcoin Asset Protocol (BAP) is not theoretical. Every component of its design can be implemented with primitives that already exist in Bitcoin and Lightning:

- **Anchoring** — Taproot, Schnorr signatures, and SegWit enable Merkle roots of BAP blocks to be committed directly into Bitcoin. Threshold signing (MuSig2, FROST) is production-ready in current libraries.
- **Payments in sats** — The Lightning Network already supports multi-part payments, splicing, and BOLT-12 offers, allowing deterministic low-fee settlement. Submarine swaps bridge Lightning ↔ on-chain BTC for anchoring fees.
- **Event consensus** — Deterministic state machines and Merkleized logs are straightforward to implement in Rust and other ecosystems. Validators can re-check events in mempools with today's toolkits.
- **Incentives & bonds** — Taproot UTXOs and LN channels can secure bonded collateral. Slashing can be implemented as reductions in payout entitlements, redistributed to honest participants.
- **Security** — Randomized committees, sat-only payments, and event logging ensure auditability, prevent unpaid work, and align incentives with honesty.

Bottom line: BAP does not wait on new cryptography or protocol changes. It is an **integration challenge** — combining Bitcoin and Lightning into a sat-only, event-driven execution layer. With careful engineering, it can be deployed today.

---

13. Conclusion

The Bitcoin Asset Protocol (BAP) reimagines what it means to build on Bitcoin.

Where other platforms bolt on contracts or issue separate tokens, BAP takes a different path:

- **Events, not contracts** — typed, composable events are the primitives for assets, governance, treasuries, and proofs.
- **Anchored truth** — every block of activity resolves into Bitcoin's proof-of-work, producing incorruptible audit trails.
- **Useful work** — validation, sequencing, anchoring, and agent execution are economically incentivized; nothing is wasted.
- **Recovery & safety** — guardians, quorum signatures, and asset keys provide resilience for both individuals and organizations.
- **Evolution without forks** — new event types and standards can be added openly, without rewriting the protocol.

BAP's role is not to replace Bitcoin, but to extend it carefully. Bitcoin remains the foundation of incorruptible money and truth; BAP becomes the coordination layer where assets, governance, and proofs can operate securely, all paid for in sats.

The result is a new design space: a network where assets are composable and recoverable, where validators and users share aligned incentives, and where every sat contributes directly to consensus work.

BAP complements Bitcoin. It strengthens the ecosystem by providing an execution layer that inherits Bitcoin's finality while unlocking new forms of digital trust and coordination.