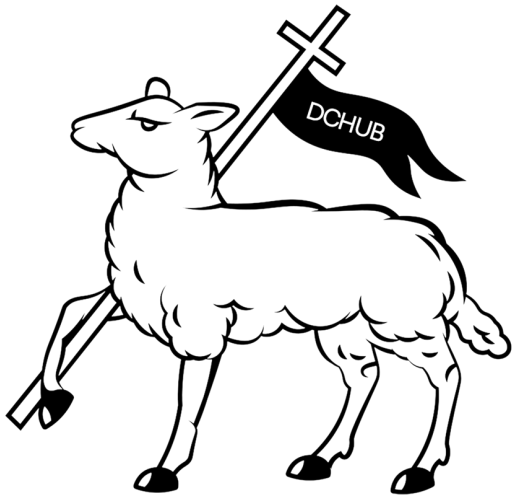

dCorps Hub

Whitepaper (Official)

Digitally native base layer for corporations and nonprofits, born and operated on-chain

Long Version

Document ID: DCHUB-WP-2025-12-21
Version: v1.3.1
Edition: Long
Status: Final v1.3.1
Release date: December 21, 2025
Author: Nicolas Turcotte, Founder
www.dcorps.com · dev@dcorps.com
Last updated: 2026-01-25

***Changelog**: v1.3.1 is a polish pass over v1.3 (section numbering consistency, capitalization, and minor grammar). 2026-01-25: Orbit rollup sanity fixes (fee model alignment, operator roles, canonical stablecoin address language).*

***Editing note (linear read)**: This copy consolidates repeated definitions and uses short reminders plus section pointers, so it reads cleanly from page 1 to the end.*

Disclaimer

Nothing in this document is:

- An offer to sell, or a solicitation of an offer to buy, any token, share, or security.
- Investment, legal, tax, or accounting advice.
- A promise of listing, price performance, or financial return for any asset.

This whitepaper describes a technical and economic design for the dCorps protocol as currently envisioned. Many details will change with engineering work, legal advice, market conditions, and community input. Any token sale, equity financing, or legal agreement will be governed by its own dedicated documentation and terms, not by this whitepaper.

Participants are responsible for obtaining their own professional advice and for complying with applicable laws in all relevant jurisdictions.

0.0 Reader map and core flows (fast path)

If you are reading this for a specific reason:

- **Founders and operators (private corporation)** Read section 0, section 7.1, section 8.3, section 9, and section 9.5B.
- **Nonprofit leaders and donors** Read section 0, section 7.3, section 8.5A, section 9.5, and section 12.6.
- **Rollup operators and infrastructure providers** Read section 6.5, section 10, and section 13.
- **Institutions, policymakers, and legal professionals** Read section 0.3B, section 4.4A, section 4.6A, section 14.3, and section 17.

Document stack and what is normative

This master whitepaper explains the intent, scope, boundaries, and design rationale of dCorps. It is not the protocol specification.

For interoperability and correctness, the dCorps document stack is intended to be:

- **This whitepaper (design intent)**: why dCorps exists, what v1 ships, and the boundary between the kernel and optional adapters.
- **Protocol Specification (normative)**: message types, state machines, event schemas, invariants, and consensus critical rules.

- **Governance Charter (normative for process)**: proposal types, thresholds, councils, upgrade process, emergency controls, and their sunset conditions.
- **Token Policy (normative for operations)**: vesting, lockups, transfer constraints, custody rules, treasury policy, and release caps.
- **Reference specifications (normative for interoperability)**:
    - Module Protocol Standard (docs/spec/SPEC-MODULES.md) and compatibility requirements,
    - Reference Indexer Specification (docs/spec/SPEC-INDEXER.md) and export formats,
    - Reference explorer behavior for entity pages and derived views (see docs/spec/SPEC-INDEXER.md),
    - Compatibility Test Suite (docs/spec/SPEC-CONFORMANCE-TESTS.md) for schema and module conformance.

If there is a conflict between this whitepaper and the Protocol Specification or Governance Charter, the Protocol Specification and the Governance Charter are intended to take precedence.

Digital-native first principles (kernel and adapters)

dCorps treats the Hub as a minimal, stable base layer for digital organizations. Everything that varies by jurisdiction, institution, or sector lives in optional modules that sit on top.

- **Kernel (required)**: canonical entity identity and discovery, ownership and authority, governance actions, treasury primitives and standardized accounting events, and document anchoring.
- **Adapters and modules (optional)**: jurisdiction recognition workflows, institutional reporting views, sector and impact frameworks, and attestations derived from kernel state.
- **Boundary (strict)**: entities must be able to operate without adapters; adapters may publish derived interpretations, but they must not redefine kernel semantics or rewrite history.

v1 mainnet focuses on one strong public container on the Hub. Advanced privacy tooling and optional external integrations are future extensions.

For the formal kernel invariants used to evaluate new features and modules, see section 4.0.

Architecture at a glance (conceptual)

External applications (UIs, markets, payroll, donation portals, dashboards)

v

Optional adapters and modules (jurisdiction recognition, sector frameworks, attestations)

v

dCorps Hub (Arbitrum Orbit rollup, Rollup mode)

- EVM contracts: entity registry, roles, governance, wallets, accounting, anchoring

- Gas: DCHUB

v

Ethereum (settlement + data availability layer)

Future extensions (not required for v1) include additional stablecoins, richer payment rails, and stronger privacy tooling. These extensions must not redefine kernel semantics or rewrite history.

Core v1 flows (three common paths)

1. Register an entity (Hub corporation or Hub nonprofit)

1. Submit an entity registration transaction and pay the registration fee (USDC service fee plus gas in DCHUB or via fee grants).
2. Bind initial roles and wallets (board seats for nonprofits, governance roles for corporations).
3. Anchor baseline governing documents and policies by hash.

2. Operate day to day in stablecoins

1. Receive inflows to canonical wallets (merchant or donation wallets).
2. Execute payouts from canonical wallets using tagged accounting events.
3. Use explorers/indexers to view live, reproducible time-window summaries over tagged flows (cash-based operating view for corporations; allocation view for nonprofits) for any selected timeframe.

3. Optionally attach adapters and derived modules (not required for core operation)

1. Attach optional protocol modules that read Hub state (jurisdiction recognition, sector frameworks, attestations) via governance.
2. Publish derived reports and signals (recognition status, eligibility, impact metrics) without changing the kernel.
3. Integrate external services and counterparties that choose to rely on the entity standard.

---

0. Executive summary

0.1 Overview

dCorps is a digitally native base layer for **corporations** and **nonprofit organizations**.

It provides a shared standard where organizations can be created, owned, governed, and operated entirely on-chain. The Hub is the kernel: it defines canonical identity, ownership and authority, governance actions, treasury control, accounting events, and an auditable history.

v1 in one sentence (minimum lovable product)

In v1, an entity can register, set roles and wallets, run stablecoin operations through tagged accounting events, and view reproducible operating and allocation summaries over any selected timeframe, with optional evidence anchoring.

It provides:

- A **dedicated dCorps Hub chain** built as an **Arbitrum Orbit rollup (Rollup mode)** that acts as the canonical entity registry and execution environment for the dCorps kernel, 197and settles to Ethereum.
- Standardized **on-chain entity containers** for:
    - **Hub corporations** (private ownership units, role and approval governance, structured accounting).
    - **Hub nonprofits** (board-governed, donation and program flows, allocation rules, and transparency).
- A common **wallet and accounting event model** that makes inflows, outflows, approvals, and governance verifiable over time.
- Open **interfaces and data standards** so independent applications and service providers can build interoperable tooling (dashboards, payroll, donation portals, reporting, and markets).

Adapters and external integration

- Jurisdictions, institutional processes, and sector frameworks integrate via optional modules that read Hub state and publish derived interpretations.
- These adapters are not required for a dCorps entity to exist or operate. They exist to reduce friction when an entity chooses to interact with external systems.

v1 focus

- One strong public container on the Hub for corporations and nonprofits, designed to be sufficient for long-lived operation.
- Advanced execution environments and public-market style instruments are future extensions and are explicitly not required for v1 adoption.

dCorps is **infrastructure**, not a bank, broker, exchange, or custodial service. It does not provide legal, tax, or regulatory guarantees. It provides a programmable, auditable entity standard that others can rely on.

A nonprofit **dCorps foundation** is intended to steward public goods over time. Its mission focuses on keeping the Hub minimal and neutral, maintaining reference standards and conformance test suites, and supporting an open ecosystem of applications and modules.

0.1.1 In practice, what dCorps gives entities

In practice, dCorps is an **entity operating ledger** for stablecoin native organizations. It is optimized for entities that keep core operations in crypto, stablecoins, and (where approved) CBDCs only.

Here, **cash-based operating views** means time-window summaries derived from tagged inflow and outflow events, excluding accrual accounting treatments.

dCorps is explicitly optimized for entities that can route meaningful parts of their operations through the protocol, and it does not support bank or fiat rail integrations at any layer.

Concretely, dCorps gives entities:

- A **standard entity container** with canonical identity, roles, and wallets, so any builder can serve any entity without bespoke formats.
- A way to express **ownership and governance as verifiable state** (units, boards and roles, proposals, approvals, and executed resolutions), instead of relying on private systems and ad-hoc reporting exports (spreadsheets/PDFs).
- A **digital-native operating posture** where counterparties can rely on on-chain authority and approvals as the default, even when no legal adapter is attached.
- **Treasury and accounting primitives** (tagged flows, budget categories, standardized view outputs) that make operations auditable and comparable over time.
- **Optional adapters** for external contexts (jurisdiction recognition, institutional reporting, sector frameworks) that can be attached without changing the entity's kernel history or semantics.
- A neutral substrate for service providers and applications (dashboards, payroll, donation portals, analytics), built on shared interfaces and standards.

The protocol defines on-chain facts and a verifiable timeline. Where an entity chooses to interact with external legal or institutional processes, those processes exist outside the kernel and reference on-chain state through anchors and adapter modules.

0.1.1A Why now and the v1 wedge (first users, first workflow, measurable outcome)

dCorps is designed for an era where stablecoins and wallets are already usable as day to day operating accounts for global teams, while entity identity, approvals, and reporting remain fragmented across private tools and jurisdiction specific systems.

Why now

- Stablecoin rails and cross-chain connectivity make it practical for serious entities to route a large share of revenue, payroll, grants, and vendor payments through on-chain wallets.
- Remote first teams and cross border collaborators increasingly need shared, verifiable approval trails that do not depend on a single country, bank, or SaaS vendor.
- Donors, counterparties, and institutions increasingly demand verifiable evidence, not just private reports and manual audits, for governance decisions and financial allocation claims.

The v1 wedge

- **First entities we win**
    - stablecoin native startups and small teams that can route most material inflows and outflows through USDC wallets, and want a clean cash-based operating view without spreadsheet reconciliation overhead.
    - nonprofits and NGOs that want donation and allocation transparency plus board-governed controls, without needing a custom platform.
- **First killer workflow**
    1. Register an entity, bind canonical wallets and roles, and anchor baseline governing documents by hash.
    2. Route operational inflows and outflows through canonical wallets and emit tagged accounting events through typed workflows where possible.
    3. Use explorers/indexers (and optional dApps) to derive cash-based operating and allocation views over any selected timeframe directly from tagged ledger events, with clear coverage metrics and evidence anchors for material items.
- **Measurable outcome**
    - An independent party can verify, using only public chain data and anchored evidence, that:
        - who had authority to act for an entity at a given time is discoverable from the role and governance record,

- a material payment was approved under the entity's policy and linked to a resolution and document anchor, and
- a cash-based operating view or nonprofit allocation view can be reproduced for any selected timeframe from the same underlying ledger inputs.

Illustrative v1 traction benchmark (non-binding)

To make adoption measurable without relying on narratives, v1 traction is tracked using public, reproducible metrics:

- **Active viewable entities**: count of active entities with sufficient tagged coverage for a reference indexer/explorer to reproduce a cash-based operating or allocation view across at least two consecutive measurement windows.
- **Coverage targets (illustrative)**:
  - Inflow coverage ratio: at least 0.90
  - Outflow coverage ratio: at least 0.90
  - Evidence coverage ratio: at least 0.60 for transactions above a defined materiality threshold (default planning threshold: 1,000 USDC, configurable by entity policy)
- **Integrity target (illustrative)**: uncategorized outflows are explicitly surfaced and are expected to be low (for example under 1 percent of total outflows for mature entities), with clear UI warnings when higher.

These are planning benchmarks intended to keep the system honest and measurable. They are not guarantees.

Coverage and assurance vocabulary (used throughout this paper)

- **Inflow coverage**: the percent of total inflows that arrive through canonical on-chain wallets for the entity.
- **Outflow coverage**: the percent of total outflows that occur through canonical on-chain wallets for the entity.
- **Evidence coverage**: the percent of material transactions that include anchored evidence (invoices, receipts, agreements, or other supporting artifacts).
- **Attestation level**: none, self-attested, or third-party attested, as signaled through attestation modules.

This wedge is intentionally narrow. It proves that the Hub can be a neutral entity registry plus a standardized operating ledger for stablecoin native organizations. Jurisdiction adapters and advanced privacy are later layers built on this foundation, not prerequisites for v1 usefulness.

0.1.1B Economic mechanics map (who pays, who benefits, what settles where)

This map describes protocol mechanics, not equity, profit participation, or any promise of returns.

In the dCorps Hub design:

- **Execution** is priced in DCHUB (gas). Interfaces may sponsor or abstract gas for end users, but the underlying execution market settles in DCHUB.
- **Protocol-level actions** may charge DCHUB fees or require DCHUB deposits (for example entity registration and renewals, premium namespaces, and selected module registry actions) to align long-lived protocol usage with the Hub token.
- **Operating flows** (invoices, payroll, grants, vendor payments) use stablecoins through canonical stablecoin contracts on dCorps (initially bridged from Ethereum).

Actor	**What they typically pay**	**What they receive and why they participate**	**Primary settlement asset**
Entities and users	DCHUB gas (often sponsored by apps), DCHUB protocol fees (registration, renewals, premium names, module registry actions), stablecoins for operating flows	predictable execution, registry listing and discovery, optional module services, and a verifiable record of authority and flows	Gas and protocol fees: DCHUB; operating flows: stablecoins
Rollup operators (sequencer and batch poster)	infrastructure and operational costs, including Ethereum L1 costs for posting rollup data	earn a governed share of gas fees and any explicitly governed operator funding; provide ordering and data availability for the rollup	DCHUB (fees on dCorps) + ETH (L1 posting costs)
Protocol Treasury	spends under on-chain policy (grants, audits,	receives a defined share of protocol fees and, if adopted by	DCHUB (protocol fees); stablecoins for

Actor	**What they typically pay**	**What they receive and why they participate**	**Primary settlement asset**
	security operations, and limited liquidity support where permitted)	governance, a capped share of gas fees for public goods and security operations	program disbursements
dCorps foundation	funds ecosystem work and public goods under foundation policy	receives a defined share of protocol fees once established; may administer defined ecosystem programs under strict reporting	DCHUB (protocol fees); stablecoins for program disbursements
Jurisdiction adapters (optional)	maintain recognition modules and related processes	may receive a defined share of adapter participation fees and publish derived recognition signals	typically USDC
Builders and service providers	no protocol-level obligation	benefit from a shared standard and can build products (dashboards, payroll, donor portals) that users choose to pay for	off-protocol

This structure is designed so that no single company has to subsidize the network indefinitely to keep it alive, and so that the Hub can remain neutral and usable even as applications compete at the edges.

0.1.2 v1 scope box (ships on mainnet, explicitly out of scope)

dCorps is a long-term multi layer vision. Mainnet v1 is intentionally narrow: ship a stable Hub kernel that can host complete corporations and nonprofits on a shared public chain.

In scope for v1 mainnet

- Hub chain (Arbitrum Orbit rollup, Rollup mode), DCHUB gas, and basic on-chain protocol governance (timelocked upgrades).
- Entity registry, entity IDs, entity metadata, and lifecycle status.
- Hub corporation module (v1 cap table and governance):
    - Ten thousand unit template
    - Unit issuance, transfers, restrictions, and corporate actions at the Hub entity level
    - Standard pools and claims patterns for finer-grained ownership when needed
- Hub nonprofit module:
    - Board governance
    - Donation and program wallet structure
    - Allocation rules and category level transparency
- Roles, governance primitives, and document anchoring:
    - Proposals, votes, and executed resolutions
    - Hash anchoring of minutes, agreements, audits, and policy documents
- Wallet and accounting primitives:
    - Canonical wallet types
    - Tagged accounting events
    - Reproducible cash-based operating views (derived from tagged events)
- Reference tooling and standards:
    - Explorer and indexer
    - Entity schemas and APIs
    - Conformance test suite for applications and modules (minimum compatibility signals)

Explicitly out of scope for v1

- Any requirement to attach a jurisdiction, legal wrapper, or compliance process.
- Operating or providing:
    - any bank or fiat rail integration (not supported at any layer)
    - on-chain fiat payments or custody of fiat
    - broker or dealer services, exchanges, matching engines, or fundraising platforms
- Automatic legal personhood or guaranteed compliance in any jurisdiction without off-chain legal processes.
- Public market features as a core protocol promise:
    - dShares issuance, primary offerings, or secondary trading infrastructure
    - any guarantee of listing or liquidity
- Advanced execution environments as part of the default path:
    - private execution zones or application-specific execution layers
    - promotion or migration requirements to remain functional
- Mandatory protocol level KYC, KYB, AML, sanctions screening (these live in applications, service providers, and optional adapters).

- Full privacy execution as a default baseline (privacy is supported as optional evolutions and selective disclosure patterns).

Everything out of scope for v1 may arrive later as optional modules, applications, and upgrades once the Hub kernel is mature and operationally proven.

0.2 Mission

The mission of dCorps is:

> Give anyone on earth the ability to create and run a serious, transparent digitally native corporation or nonprofit in a recognized digital ecosystem, whether or not it is attached to a jurisdiction, and to optionally attach legal recognition where and when it is needed.

The core promises are:

- **Access** Founders and nonprofit leaders should be able to form and operate serious structures without needing to be in a small set of favored jurisdictions.
- **Transparency** Governance, cap tables, and financial flows should be anchored in verifiable state, not just in private systems and manual reports.
- **Programmability** Common organizational processes such as vesting, donations, board approvals, and allocation rules should be expressible as code, not only as narrative policies.
- **Digital-only by default, optional external interoperability** Entities should be able to start simple and remain complete inside the on-chain economy. If they later choose to interact with legacy jurisdictions or institutions, they can attach optional adapters and evidence patterns without rebuilding their infrastructure.

0.3 Digital-native kernel, optional adapters for jurisdictions and institutions

dCorps is a **digital-native entity base layer**. It reuses proven organizational abstractions (ownership, units, boards and roles, accounting, approvals) but replaces external enforcement with deterministic execution and auditable state.

That means:

- The **Hub kernel** focuses on:

- Entity registration and identity.
- Ownership and authority (units, roles, approvals).
- Governance actions and document anchors.
- Wallet structure and standardized accounting events.
- DCHUB gas, protocol governance, and (where adopted) protocol-level fees or deposits.
- **Adapters and frameworks** are optional modules that sit on top of the kernel, not hard coded into the base protocol:
    - Jurisdiction adapters interpret on-chain truth for specific legal and institutional contexts.
    - Sector and impact frameworks compute domain specific metrics from Hub state.
    - Attestation and reputation modules publish derived signals about entities, roles, and activity.
- **Applications** are completely separate from the Hub:
    - Payroll tooling, donation portals, fundraising platforms, analytics, and dashboards operate as independent applications.
    - They talk to the Hub through wallets, APIs, and SDKs.
    - They can be built by any developer and surfaced in an open app and module registry.

The **dCorps foundation** is expected to steward public goods, including reference standards, conformance tests, and optional adapter specifications that evolve as institutions, markets, and law evolve. The foundation does not make external systems a dependency of the kernel.

The base layer has one job: **be a neutral, robust, auditable organization kernel**. Everything else is optional and replaceable.

Applications provide user experience on top of the Hub and its adapters, but they do not become part of consensus.

0.3A Digitally native entities and scope

dCorps is a digitally native entity base layer, meaning entities are represented as persistent on-chain objects with explicit roles, wallets, governance, and accounting primitives.

The Hub is designed to be the canonical operational ledger for on-chain entity activity:

- Entity identity and registry state
- Cap tables (Hub units) and board structures
- Governance proposals, votes, and resolutions
- Structured wallet flows and tagged accounting events
- Anchors for off-chain documents and optional anchored-environment summaries (future extension)

dCorps is not a legal wrapper service and does not replace legal responsibility. Legal status, filings, and enforcement remain off-chain and are achieved through optional integrations such as jurisdiction adapter modules plus matching legal processes and documents. The protocol can help express and automate reporting logic and fee obligations, but it does not guarantee compliance, and it does not replace legal, tax, or accounting advice.

dCorps is designed for organizations that operate entirely inside the on-chain economy. The strongest transparency and automation guarantees come from routing operations through canonical on-chain wallets and standardized workflows. Entities may still anchor hashes of off-chain documents (contracts, invoices, policy texts, audits, and similar artifacts) when they want additional evidence or dispute clarity, but the protocol does not assume or require any bank rails, fiat ledgers, or state registries as inputs.

0.3B What a Hub entity is as a digital organization

A dCorps Hub entity is a digitally native organizational object with canonical identity, ownership or board authority, governance history, and standardized wallet and accounting structures recorded on the Hub.

A Hub entity is complete with or without any jurisdiction adapter. Adapters exist only to map on-chain truth into external processes when an entity chooses to interact with them.

A Hub entity can be understood as:

- An on-chain registry entry plus on-chain state machines (ownership or board structure, role and wallet structure, governance events, accounting primitives), and
- Optional off-chain agreements and practices that participants may choose to maintain, which can reference the on-chain state as the canonical source of authority.

dCorps does not grant legal personhood by itself. Where legal personhood is desired, it is achieved through an external process that references Hub identifiers, governance records, and ownership or board state.

Operating without legal recognition may limit enforceability against legacy counterparties in some contexts. This is a choice, not an incomplete state.

At the same time, purely Hub-native entities can still be highly meaningful inside the on-chain economy, to customers, suppliers, contributors, and other entities that choose to rely on on-chain authority.

0.3B.1 Default v1 operating pattern (no jurisdiction adapter)

In v1, many teams can operate purely Hub-native. If a team also wants an off-chain agreement layer that references on-chain authority, a common pattern is:

1. Register a Hub corporation or Hub nonprofit on-chain.
    - For corporations, the default ownership model is 10,000 base units, and it can be expanded in v1 when higher precision is needed (see section 7.1.1).
2. Optionally adopt governing documents off-chain that reference:
    - The entity ID,
    - The role and voting rules used on-chain,
    - The cap table or board state as the canonical source of record for defined decisions, and
    - The document anchoring scheme used for minutes, resolutions, and material contracts.
3. Operate day to day treasury activity through the standardized on-chain wallet structure as much as practical.
4. Use anchored documents and attestations only where needed for external counterparties, audits, or dispute resolution.

Reference templates and checklists for this pattern may be published or recommended through the app and module registry, but these templates (whether provided by dCorps, third parties, or law firms) are not part of the protocol and are not legal advice.

0.3B.1A Legal binding and enforceability kit (reference standard)

This section defines a repeatable documentation and evidence pattern for entities that want off-chain counterparties, auditors, and service providers to treat the Hub record as the canonical source of truth for authority, approvals, and governance history.

This is a reference standard for interoperability and operational clarity. It does not create legal personhood and does not guarantee enforceability in any jurisdiction.

Entity Governance Binding Document (reference artifact)

Entities that want a high clarity posture in a jurisdictionless phase are expected to adopt an off-chain governing document that, at minimum, references:

- The Hub `entity_id` (and chain ID or other canonical chain identifier used at the time of signing).
- The entity type (Hub corporation or Hub nonprofit) and the version of the governance template or module family used.
- The role binding method used for authority (wallet-based roles, DID-based roles, or hybrid), and how role reassignment and key rotation are recognized.
- The voting and approval rules that have binding effect for defined decision classes, including:
    - Which decisions must be approved by board votes versus unit holder votes (or both),
    - Quorum and threshold rules,
    - How abstentions and vacancies are treated.
- The list of decision classes for which the Hub record is treated as canonical evidence of approval, including, at minimum, the following when used:
    - Material contract approval,
    - Treasury policy and limits,
    - Unit issuance, cancellations, and major transfers (for corporations),
    - Allocation rule changes (for nonprofits),
    - Attachment or detachment of protocol modules,
    - Anchored environment recognition related actions (future extension) where applicable.
- The document anchoring scheme, including:
    - What must be anchored (minutes, resolutions, material contracts, audits, and other defined documents),
    - How anchors are referenced (hash, timestamp, on-chain event pointer),
    - How superseded documents and corrections are handled.

Authority evidence package for counterparties (recommended)

For any material off-chain agreement, the entity is expected to produce a compact evidence package that a counterparty can verify without trust in a private inbox or a single individual. A standard package includes:

- `entity_id`, entity name, and current registry status.
- The role wallet or DID asserted to have signing authority, and the current on-chain role binding record for that role.
- A governance resolution that:
    - Approves the agreement (or the signing of a defined agreement class),

    - Identifies the signer role and any limits (amount caps, term limits, scope limits),
    - Anchors the final agreement hash, or anchors a definitive agreement reference ID that maps to an anchored file.
- A document anchor for the final, executed agreement, including its hash and minimal metadata.
- If the agreement was approved through delegation or an internal policy, the evidence package includes the policy anchor and the policy effective date.

This package is designed so that a counterparty can confirm, using public state, that the correct approvals exist and that the signed document matches the anchored commitment.

Disputes, corrections, and signer authority conflicts (reference pattern)

Because the Hub is not a court, disputes are handled by contracts and institutions outside protocol consensus. A standard posture is that:

- The entity's governing documents define how disputes about authority and approvals are resolved and what evidence is admissible.
- Corrections are handled by anchoring superseding documents and publishing explicit correction resolutions, rather than attempting to erase history.
- If a signer's authority is later disputed, the evidence timeline remains visible, including:
    - role binding changes,
    - resolutions and their vote records,
    - anchored documents and their supersession chain.

The protocol does not mandate a particular dispute venue. The goal is that whichever venue applies has a clear, verifiable evidence trail.

Personal liability and limited liability boundary (operational safety checklist)

In a jurisdictionless phase, limited liability should not be assumed. A safety checklist for serious teams includes:

- Use explicit signer roles and anchored approvals for material obligations so counterparties can verify authority.
- Keep a clear boundary between entity wallets and personal wallets, and avoid commingling.
- For high stakes off-chain obligations, use an attached jurisdiction adapter module or an explicit off-chain legal wrapper that references the Hub record, so that signers are not relying on informal assumptions about liability.

dCorps can standardize evidence and authority records. It does not replace legal structure.

0.3B.2 Optional attachment for external recognition

An entity may choose to attach a jurisdiction adapter when it wants to interact with external systems that require local recognition (certain contracts, employment, procurement, or other institutional workflows). The adapter is an overlay: it reads Hub state and publishes a recognition status or proof pointers, while the actual legal steps happen off-chain.

Recognition status can be recorded on-chain as derived state so counterparties and applications can rely on a consistent view. The adapter does not change the kernel, it bridges kernel facts to off-chain legal assertions.

0.3B.3 Digitally native operation (jurisdictionless by default)

dCorps is designed so that entities can be legitimate and useful even without any jurisdiction adapter attached, as long as their relevant counterparties agree to rely on the Hub record.

Common patterns include:

- digitally native service businesses and digital collectives that transact primarily in stablecoins and onboard customers globally.
- Protocol teams and DEX operators that want structured governance, role-based authority, and transparent operational flows, without immediately committing to a jurisdiction regime.
- On-chain ventures whose contracts, treasury policies, and counterparties are primarily on-chain, so enforceability is achieved through:
    - smart contract execution,
    - platform rules and account controls,
    - arbitration or dispute processes referenced by anchored documents, and
    - counterparties choosing to rely on the entity's on-chain authorization rules.

This does not replace law. It is a practical digitally native organizational posture that can be upgraded later into legal recognition through jurisdiction adapters and matching off-chain processes when and where that becomes desirable.

Optional note: when an entity chooses to interact with legacy legal or institutional contexts, it may need external recognition and off-chain enforcement. That is why jurisdiction adapters exist as optional overlays. They are not required for digital-only operation on-chain.

0.4 Who dCorps is for

dCorps is designed for:

- **Founders and small teams**
    - Want a serious, transparent corporation that can operate globally in USDC.
    - Need clear ownership, governance, and accounting from day one.
    - May later need legal recognition or public style financing, but do not want to choose everything on day one.
- **Nonprofit founders and NGOs**
    - Want full transparency on donations and program spending.
    - Need verifiable board based governance and allocation rules.
    - Want to receive donations and grants in stablecoins, with clear reporting.
- **Protocols and DAOs maturing into entities**
    - Started life as informal DAOs.
    - Now need predictable governance, HR, legal contracts, and sustainable funding structures.
- **Jurisdictions and institutions**
    - Want digital friendly regimes for corporations and NGOs.
    - Prefer programmable, auditable bases rather than purely paper based systems.
- **Builders, auditors, and service providers**
    - Want a stable base layer to build accounting tools, dashboards, donation platforms, compliance tools, and analytics.
    - Prefer open standards rather than each client or jurisdiction inventing its own format.

dCorps is **not** for:

- Projects that want opaque structures, cosmetic governance, and minimal transparency.
- nonprofits that are unwilling to expose financial flows on-chain at least at an aggregated category level.
- Actors looking for the chain to replace legal responsibility or to provide guaranteed investment returns.

In short: dCorps is **infrastructure**, not a bank, not a broker, not an exchange, and not a compliance shortcut. Entities and participants remain responsible for law, regulation, and business risk.

0.5 Skeptic FAQ (quick answers)

Is dCorps trying to replace law or act as a legal wrapper service?

No. The Hub is a registry and evidence layer for digitally native entities. Legal personhood, limited liability, filings, and enforcement remain off-chain and are achieved through optional jurisdiction adapter modules plus matching legal documents and processes (see section 0.3B and section 14.3).

Is a Hub entity a legal corporation or charity by default?

Not by default. A Hub entity is an on-chain organizational object with roles, governance, wallets, and history. It becomes legally recognized only when local law and off-chain processes bind recognition to on-chain identifiers and module state (see section 0.3B and section 14.3).

Why a dedicated Hub chain and DCHUB instead of only contracts on an existing chain?

Because dCorps needs a stable, neutral home for entity IDs, governance evidence trails, and anchoring standards, plus a controlled execution environment where gas is paid in DCHUB while still settling to Ethereum for security. These are long-lived primitives that benefit from conservative upgrades, clear timelocks, and a dedicated operational posture (see section 5.6 and section 6.5).

Can entities lie with tags and reporting?

Amounts, timestamps, and transfers for on-chain funds are verifiable. Category codes are interpretations and can be misused. dCorps mitigates this by encouraging typed workflows that emit deterministic categories, evidence anchoring for material transactions, counterparty receipts, and optional third-party attestations and reconciliation signals (see section 9.5A and section 9.5B).

Is dCorps a bank, broker, exchange, or fundraising platform?

No. The protocol does not custody funds, run markets, intermediate capital, or perform KYC at the base layer. Regulated activity lives in external applications, custodians, and service providers that carry their own responsibilities (see section 4.6 and section 4.6A).

What happens if USDC is frozen or disrupted?

Stablecoin issuer actions and rail risk are external to the Hub and cannot be overridden by rollup operators or protocol governance. dCorps treats this as a treasury and continuity planning

issue, surfaced through asset registry risk labeling, treasury segmentation, diversification, and explicit operating continuity policies (see section 9.1B and section 15.8.1).

How do you reduce governance capture risk?

dCorps uses long vesting, non voting treasury and foundation defaults, protected changes with higher thresholds, voting-power age requirements, and execution timelocks for sensitive actions. These guardrails are designed to make hostile changes slower and more visible (see section 10.3A and section 13.3.5).

---

1. Purpose, scope, and digital-native philosophy

1.1 What this whitepaper is

This whitepaper is:

- A **technical and economic design** for the dCorps base layer.
- A description of:
  - The Hub chain and entity models.
  - Token and fee mechanics.
  - Governance and security structures.
  - The roles of the development corporation and future foundation.
- A statement of **design intentions and principles**, not a binding specification.

In this document, “must” and “required” describe compatibility requirements for implementations, modules, applications, and reference interfaces that claim compatibility with the dCorps standards described here. They do not describe legal, regulatory, or market guarantees. The normative rules, interfaces, schemas, and conformance tests are defined in the **Protocol Specification** and the **Module Protocol Standard**.

It is meant to be read together with:

- A Protocol Parameters document.
- Developer documentation and module specifications.
- Governance and Treasury policy documents.

Those will provide more precise numbers and APIs.

1.2 What this whitepaper is not

This whitepaper is not:

- A prospectus, offering memorandum, or fundraising document.
- A legal opinion or classification of DCHUB or any other token under any particular law.
- A full technical specification; important implementation details will live in separate documents and repositories.

Any token sale, equity financing, or legal recognition program will be described in its own documents and will comply with local rules where it takes place.

1.3 Intended audience

The intended readers are:

- **Founders and nonprofit leaders** evaluating dCorps for their own entities.
- **Developers and infrastructure providers** who want to build on dCorps.
- **Rollup operators and node operators** assessing the security and economics of the network.
- **Institutions, policymakers, and legal professionals** who need to understand what dCorps does and does not do.
- **Researchers and observers** interested in on-chain based entity infrastructure.

The language aims to be precise enough for serious readers, without assuming specialist background in cryptography or specific jurisdictions.

1.4 Digitally native by default philosophy

The design of dCorps follows a digital-native philosophy:

- **Neutral protocol, not a product bundle** The Hub is not a commercial incorporation service. It is neutral infrastructure that anyone can build on, including competing incorporation services, jurisdictions, and applications.
- **Self custody and sovereignty by default** Entities and users keep control of their wallets, keys, and governance processes, except where they choose to use custodial or managed services.

- **Transparency as verifiable state** Ownership, governance, and material financial flows are represented as on-chain state, or anchored through hashes and proofs. Reports and dashboards are views over this state, not alternative sources of truth.
- **Clear separation between core, protocol modules, and applications** The Hub focuses on entity registration, entity structure, governance events, and accounting primitives. Jurisdiction rules and sector standards live in protocol modules. User experience, KYC, traditional integrations, and markets live in external applications.
- **Compliance aware, not compliance enforcing** The protocol does not claim to encode the law for every jurisdiction. Instead it provides primitives and interfaces so that legal and regulatory rules can be integrated through optional modules and off-chain processes, where those actors accept responsibility.

The aim is to combine the programmability and auditability of on-chain infrastructure with the durability and seriousness required by real corporations and nonprofits.

Universal base layer, consistent protocol behavior

dCorps does not tune its core entity model for specific countries, sectors, or organization sizes. The Hub aims to expose consistent primitives for identity, roles, wallets, governance, and accounting.

At the same time, legal status and legal effects are not universal. Legal recognition, filings, reporting obligations, and enforcement still depend on jurisdiction, contracts, and institutions. dCorps addresses this by keeping jurisdiction specific differences in **jurisdiction adapter protocol modules** and matching off-chain processes, rather than hard coding them into the core chain.

1.5 How to read this document

This whitepaper is intentionally long and complete. It is the canonical reference for how dCorps is intended to work at the protocol and ecosystem level.

Other documents are, or will be, shorter slices of the same design, for example:

- A **litepaper** (docs/whitepaper/LITEPAPER.md) or **investor brief** (docs/investor/INVESTOR_BRIEF.md) that focuses on vision, architecture, token model, and adoption path.
- A **nonprofit note** (docs/whitepaper/NONPROFIT_NOTE.md) that focuses on nonprofit modules, donor transparency, and jurisdiction adapter patterns for charities.
- Technical docs (docs/engineering/TECHNICAL_OVERVIEW.md, docs/engineering/INTEGRATION_GUIDE.md) and normative specs (docs/spec/) that focus on APIs, schemas, and implementation details.

Those documents do not replace this whitepaper. They present the same structure from different angles. Where there is any doubt, this long form whitepaper is the starting point, and more precise parameter values or legal details live in the separate reference and policy documents listed in section 1.1.

1.6 Terminology and conventions

This whitepaper uses the following conventions to keep wording and technical meaning consistent:

Entity and system terms

- **Hub**: The dCorps Hub chain (an Arbitrum Orbit rollup in Rollup mode), the canonical registry and coordination layer.
- **Hub entity**: An entity that operates directly on the Hub (Hub corporation or Hub nonprofit).
- **Protocol module**: On-chain logic (EVM contracts) that attaches to, reads, and writes Hub entity state (for example jurisdiction adapter modules, sector frameworks, and attestations).
- **Application**: Off-chain software (UI, API services, dashboards, platforms) that interacts with the Hub and its modules via wallets, JSON-RPC, and indexing APIs, but does not become part of consensus.

Writing conventions

- “On-chain” and “off-chain” are written with a hyphen.
- “nonprofit” is written as one word.
- “Must” and “required” indicate compatibility requirements for implementations, modules, applications, and reference interfaces that claim conformance; they are not guarantees about off-chain outcomes.
- “Expected”, “intended”, and “design intention” describe goals, not guarantees.

Money and denominations

- USDC is used as the baseline unit of account for examples and default reporting.
- EVM tokens use decimals. When amounts are shown, they are shown in human units (USDC, USDT, DAI) unless explicitly stated otherwise.
- USDC and USDT typically use 6 decimals; DAI typically uses 18 decimals. DCHUB decimals are defined by the token contract.

- When stablecoins are held or transferred on the Hub, they are represented as canonical ERC-20 contracts on dCorps Hub. In early phases these are bridged assets originating from Ethereum. Each supported stablecoin has a single canonical contract address on the Hub recorded in the asset registry; interfaces should treat any other address as non-canonical. Reference interfaces should present them with mainstream symbols (USDC, USDT, DAI) and disclose bridging/issuer risk separately.

1.7 Key assumptions and dependencies

This whitepaper assumes:

- Ethereum remains a viable settlement and data-availability layer for rollups.
- The dCorps Hub rollup (Arbitrum Orbit, Rollup mode) remains viable as an EVM execution environment and settlement stack.
- Bridged USDC remains available as an operating currency, with issuer controls and bridge risk treated as external constraints (see section 9.1B). Additional stablecoins may be approved over time.
- Native issuer integrations (for example Circle-native USDC mechanisms) may become available later, but are not assumed and are not guaranteed.
- One or more independent indexers and explorers exist and remain available, because most users experience the protocol through indexed views rather than raw node queries.
- Entities that want strong transparency route the large majority of their material activity through canonical on-chain wallets and standardized workflows. If an entity uses multiple chains or external systems, it may publish optional completeness commitments and attestations, but the Hub kernel does not support fiat rails.
- Jurisdiction adapters are optional and may be delayed, limited, or unavailable for long periods; legal recognition remains jurisdiction dependent and off-chain (see section 0.3B and section 14.3).
- Privacy-preserving execution and zero knowledge reporting are optional evolutions; v1 does not assume they are universally available or easy to deploy (see section 8.5).

If these assumptions do not hold, adoption paths and module timelines may change, but the Hub’s core goal remains the same: a minimal, auditable entity registry and operating substrate.

---

2. Vision, problem, and context

2.1 Core mission

The core mission of dCorps is to democratize:

- Serious corporation creation and operation, and
- High trust nonprofit operation and donor reporting

for people in all countries, not only in a small set of financial centers.

In concrete terms:

- A founder in Lagos, Dhaka, or Medellín should be able to run a USDC based, transparent entity with credible governance and accounting, without needing to relocate or rely on opaque intermediaries.
- A small NGO doing critical work in a fragile state should be able to prove allocation and governance quality with the same level of cryptographic assurance as a large foundation in a major capital.

2.2 Problems with current corporate and NGO systems

Current systems have deep structural limits.

Geography and jurisdiction bias

- Founders in good jurisdictions can incorporate quickly, open bank accounts, and access international payment rails.

- Founders in fragile or excluded jurisdictions face:

    - Unreliable registries.
    - High banking risk or rejection.
    - Low trust from foreign counterparties.

Two teams of equal skill and seriousness often face entirely different futures because of their passports, not their work.

Fragmentation and manual reconciliation

- Legal status lives in registries and jurisdiction-bound processes.
- Money lives in bank accounts, payment networks, custodians, and stablecoin wallets.
- Governance lives across private tools (board minutes, internal workflows, service providers, and third-party portals).
- Accounting lives in private ledgers and spreadsheets, reconciled by humans.

Keeping these in sync is manual and error prone. Cap tables drift from reality. NGO reports lag reality by months or years.

Gated access to capital and grants

- Public markets and major private capital are effectively reserved for a tiny percentage of corporations that can afford heavy regulatory and advisory costs.
- Large, brand name NGOs dominate institutional funding. Smaller organizations with real impact cannot prove it in a way that large donors trust.

Transparency that is narrative, not cryptography

- Transparency often means self-reported exports and human audits, selectively compiled.
- Underlying records can often be changed without global visibility.
- Donors and investors see stories, not ledgers.

At the same time, many crypto native organizations are:

- Programmable and transparent in treasury movements, but
- Informal, hard to map to law, and limited in back office functions.

There is a missing layer between traditional systems and raw DAOs.

2.3 Why on-chain infrastructure is an appropriate base

On-chain infrastructure, used carefully, solves some core issues:

- A **shared ledger** across borders offers a common source of truth.
- Smart contracts allow core processes such as vesting, board decisions, and allocation rules to be executed and audited by code.
- Wallets and stablecoins allow entities to transact globally without relying on a particular national banking system.

However, generic smart contract platforms and DAO frameworks alone are not enough. They often lack:

- Clear mapping between wallets and legally meaningful roles such as director or board member.
- Well defined cap tables and corporate actions.
- NGO specific requirements such as allocation rules and board oversight.

- Stable, neutral base layer governance suitable for institutions and conservative counterparties.

dCorps uses on-chain infrastructure not as a speculation machine, but as a **programmable entity operating system**.

2.4 What a digitally native entity stack looks like

A modern entity stack built on on-chain infrastructure should be:

- **Neutral and global** It should not belong to a single country or corporate service provider. Jurisdictions can plug in through protocol modules, not through exclusive control.
- **Transparent but privacy aware** Key structures and flows should be observable and auditable. Sensitive data such as individual salaries, HR records, and beneficiary identities must be protected through controlled zones and off-chain storage.
- **Programmable and composable** Core patterns such as cap tables, payroll, donation flows, and allocation rules should be reusable, auditable modules, not bespoke scripts and spreadsheets.
- **Sovereign entities, not anonymous contracts** Entities should have names, identities, and roles that map to the real world, even when they operate globally.
- **Compliance aware** Laws differ by jurisdiction and evolve. The stack needs clear places where legal and regulatory logic can be attached, updated, and versioned without rewriting the base.

dCorps is designed as such a stack. It keeps entity models at the center and treats finance, governance, jurisdiction logic, and sector standards as composable layers around them, implemented as protocol modules and applications.

2.5 Who dCorps is for and not for

For

- Serious founders who want long term entities, not short term token games.
- NGOs and impact organizations that want donors to see how funds are used.
- Protocol teams that want to graduate from pure DAOs to structured, accountable organizations.
- Institutions, policymakers, and jurisdictions that want to work with transparent on-chain entities, using adapters instead of bespoke reporting.

- Builders of accounting, payroll, compliance, and donation tools who want a stable base to integrate with many entities.

Not for

- Projects that want to hide ownership, governance, or financial flows.
- Entities that want token price as the main narrative, without clear business or impact logic.
- Actors hoping that, because it is on-chain, they no longer have to comply with local laws.
- nonprofits unwilling to show donors how funds are allocated in practice.

2.6 Example use cases

Short vignettes help illustrate how dCorps works in practice.

Remote-first startup

Three founders in three countries want to launch a software product and sell subscriptions globally.

On dCorps they:

- Register a **Hub corporation** and allocate the ten thousand internal units among founders, early contributors, and a contributor pool.
- Receive income into a **merchant wallet** in USDC.
- Pay contractors, infrastructure costs, and distributions from treasury and operating wallets using tagged accounting events.
- At any time, generate a **cash-based operating view** over a selected timeframe, suitable for stakeholder transparency, distinct from GAAP or IFRS reporting.
- If they later need to sign contracts with legacy counterparties, they can attach an optional **jurisdiction or institutional adapter** that references on-chain governance and ownership, without changing the kernel.

Radically transparent nonprofit

A small NGO works on education in a low income region. It is trusted locally but struggles to convince international donors.

On dCorps it:

- Registers as a **Hub nonprofit** with a board represented by DIDs and role wallets.
- Receives donations into a **donation wallet** in USDC, including checkout donations from partner corporations.

- Defines program wallets and **allocation rules as code**, including minimum percentages for direct program spending versus overhead, and restricted fund constraints when needed.
- Allows donors to see, in near real time, how funds move from donation to program costs, and how board decisions are taken.
- Optionally attaches a **jurisdiction adapter** for local charity recognition or tax receipt workflows, while keeping the Hub as the canonical ledger of truth.

This does not replace field level due diligence, but it makes governance and allocation visible in a way traditional tools do not.

Joint venture SPV for a specific project

Two corporations want to co-fund and operate a specific project without merging their main entities.

On dCorps they:

- Register a dedicated **Hub corporation** as a joint venture or SPV, owned by the parent entities, with its own units and wallets.
- Allocate units to the parent entities according to their agreement, and optionally allocate a contributor pool for project work.
- Route project revenues and costs through dedicated wallets, tagged separately from each parent entity's own operations.
- Use on-chain approvals and derived views to give both parents and external financiers a clear picture of the project's performance.

The JV behaves like any other Hub corporation from the protocol's perspective, while remaining ring-fenced from each parent's main operations.

Digital-friendly jurisdiction as an adapter

A jurisdiction wants to offer a digital corporation or nonprofit recognition regime without building a chain from scratch.

Instead of changing the kernel, it can:

- Publish an adapter specification that maps dCorps entity state and proofs into a local recognition process.
- Offer optional services (templates, filings, reporting, tax receipt workflows) that reference anchored evidence and executed resolutions.
- Collect fees through the adapter workflow only when an entity opts in.

In this model, dCorps remains the kernel, and jurisdictions integrate as optional plugins.

2.7 Concrete operating examples with numbers

The following examples are simplified cash flow and tagged ledger views to make the view model concrete. They are not intended to represent GAAP, IFRS, or any local statutory reporting standard.

They illustrate:

- canonical wallets,
- tagged accounting events,
- optional evidence anchors,
- and the reproducible time-window views described later in section 9.5B (corporations) and section 9.5C (nonprofits).

2.7.1 Hub corporation, timeframe operations example (end-to-end)

A Hub corporation routes all material revenue and operating payouts through its dCorps wallets.

Selected timeframe inputs (merchant wallet inflows)

- Subscription revenue (invoicing module, typed workflow): 50,000 USDC

Selected timeframe outputs (operating outflows)

- Salaries (payroll batch, typed workflow): 30,000 USDC
- Contractors (tagged outflows): 5,000 USDC
- Cloud and infrastructure (tagged outflows): 4,000 USDC
- Other operating expenses (tagged outflows): 2,000 USDC
- Jurisdiction compliance fees (optional adapter, typed workflow): 500 USDC
- One uncategorized outflow (missing required category code, surfaced in coverage): 500 USDC

Treasury sweep (internal transfer, not an expense)

- Transfer to treasury wallet for reserves: 8,000 USDC

End of timeframe balances (illustrative)

- Merchant wallet remaining: 0 USDC
- Treasury wallet: 8,000 USDC

Coverage and integrity signals (illustrative)

- Inflow coverage: 1.00 (all inflows arrived through canonical wallets)
- Outflow coverage: 1.00 (all outflows occurred through canonical wallets)
- Evidence coverage: 0.70 (illustrative, based on anchors attached to material items)
- Uncategorized outflows: 500 USDC (explicitly shown and expected to converge toward zero with improved tagging discipline)

Tagged accounting events (simplified excerpt)

Event	Direction	Wallet type	Amount	Category code	Source type	Evidence anchor
Subscription revenue	inflow	merchant	50,000	REV_SUBSCRIPTION	typed_workflow	invoice batch anchor (optional)
Payroll batch	outflow	merchant	30,000	EXP_PAYROLL	typed_workflow	payroll report anchor
Contractor payout	outflow	merchant	5,000	EXP_CONTRACTOR	entity_tagged	invoice anchor
Cloud bill	outflow	merchant	4,000	EXP_INFRA	entity_tagged	statement anchor (optional)
Other opex	outflow	merchant	2,000	EXP_OTHER	entity_tagged	receipt anchors (optional)
Compliance fee	outflow	merchant	500	EXP_COMPLIANCE	typed_workflow	policy or invoice anchor (optional)
Missing category example	outflow	merchant	500	(missing)	entity_tagged	none

Event	Direction	Wallet type	Amount	Category code	Source type	Evidence anchor
Treasury sweep	internal	merchant -> treasury	8,000	TREASURY_MOVEMENT	typed_workflow	none

Derived cash-based operating view (category totals excerpt)


```
{

  "report_type": "cash_based_operating_statement",

  "base_asset": "USDC",

  "base_decimals": 6,

  "income": [

    { "category_code": "REV_SUBSCRIPTION", "amount": "50000000000", "source_type":
"typed_workflow" }

  ],

  "expenses": [

    { "category_code": "EXP_PAYROLL", "amount": "30000000000", "source_type":
"typed_workflow" },

    { "category_code": "EXP_CONTRACTOR", "amount": "5000000000", "source_type":
"entity_tagged" },

    { "category_code": "EXP_INFRA", "amount": "4000000000", "source_type": "entity_tagged" },

    { "category_code": "EXP_OTHER", "amount": "2000000000", "source_type": "entity_tagged"
},

    { "category_code": "EXP_COMPLIANCE", "amount": "500000000", "source_type":
"typed_workflow" }

  ],
```

```
  "coverage": {
    "total_inflows": "50000000000",
    "total_outflows": "42000000000",
    "uncategorized_outflows": "500000000"
  },
  "net_operating_result": "8000000000"
}
```


This view is reproducible from the underlying accounting events and explicitly surfaces missing tags.

2.7.2 Hub nonprofit, timeframe allocation example (end-to-end)

A Hub nonprofit receives donations into its donation wallet and distributes funds across program and support categories with board visibility and enforcement.

Selected timeframe inputs (donation wallet inflows)

- Donations received (donation module, typed workflow): 100,000 USDC

Selected timeframe allocations (donation wallet outflows)

- Program A direct costs (program category, typed workflow): 58,000 USDC
- Program B direct costs (program category, typed workflow): 17,000 USDC
- General and administrative overhead (support category): 15,000 USDC
- Fundraising costs (support category): 5,000 USDC

Treasury retention (internal transfer, not an expense)

- Retained for future buffer (donation wallet -> treasury wallet): 5,000 USDC

Allocation ratios (illustrative, based on distributed funds)

- Total distributed in this timeframe: 95,000 USDC
- Program spending: 75,000 USDC (79 percent)
- Overhead: 15,000 USDC (16 percent)
- Fundraising: 5,000 USDC (5 percent)

Coverage and integrity signals (illustrative)

- Inflow coverage: 1.00
- Outflow coverage: 1.00
- Evidence coverage: 0.65 (illustrative, based on anchors attached to material items)
- Restricted funds coverage (optional): surfaced when restricted tags and consent rules are used

Tagged accounting events (simplified excerpt)

Event	Direction	Wallet type	Amount	Category code	Source type	Evidence anchor
Donation inflow	inflow	donation	100,000	DONATION_GENERAL	typed_workflow	none (optional donor receipt anchor)
Program A spending	outflow	donation	58,000	EXP_PROGRAM	typed_workflow	invoice and receipt anchors (recommended for material items)
Program B spending	outflow	donation	17,000	EXP_PROGRAM	typed_workflow	invoice and receipt anchors
Overhead spending	outflow	donation	15,000	EXP_ADMIN	entity_tagged	receipt anchors (optional)
Fundraising spending	outflow	donation	5,000	EXP_FUNDRAISING	entity_tagged	receipt anchors (optional)
Treasury retention	internal	donation -> treasury	5,000	TREASURY_MOVEMENT	typed_workflow	none

Derived nonprofit allocation view (category totals excerpt)


```
{
  "report_type": "nonprofit_allocation_statement",
  "base_asset": "USDC",
  "base_decimals": 6,
  "donations_in": "100000000000",
  "distributed_out": "95000000000",
  "retained": "5000000000",
  "by_category": [
    { "category_code": "EXP_PROGRAM", "amount": "75000000000" },
    { "category_code": "EXP_ADMIN", "amount": "15000000000" },
    { "category_code": "EXP_FUNDRAISING", "amount": "5000000000" }
  ],
  "ratios": {
    "program_spending_ratio": "0.7895",
    "overhead_ratio": "0.1579",
    "fundraising_ratio": "0.0526"
  }
}
```


Allocation rules can enforce constraints such as minimum program ratios, maximum overhead, board compensation limits, or restricted fund logic. The point is that the allocation view flows directly from the ledger and associated governance events, not from privately compiled reports.

3. Market landscape and competition

3.1 Market map

The landscape dCorps lives in can be simplified into layers:

1. **Traditional legal and registry layer**
    - Corporation and NGO registries.
    - Corporate law and charity law.
2. **Financial rails**
    - Banks, payment processors, card networks.
    - Local and cross border payment systems.
3. **Back office software and services**
    - Accounting, payroll, HR, cap table management, board portals.
    - Corporate secretarial services and law firms.
4. **on-chain infrastructure**
    - Smart contract platforms and app chains.
    - DAO frameworks and treasury tools.
    - Digital identity and DID systems.

Most serious entities today live at layers 1 to 3. On-chain systems often operate separately, focusing on DeFi and speculative tokens.

dCorps sits between these worlds as an **on-chain layer for real entities**. It is not a replacement for law or for all software, it is a shared base that both traditional and on-chain systems can plug into.

3.2 Traditional competition

Traditional competition comes from:

- **Corporation and NGO registries**

    - Government or court operated systems holding official records.
    - Often paper based, slow, and not machine readable.

- **Offshore incorporation and trust providers**

    - Offer fast corporation setup and management for a global client base.
    - Increasingly advertise digital corporations with online dashboards.
    - Still rely on private registries, banks, and bespoke software stacks.

- **Law firms and corporate service providers**

    - Maintain cap tables, board records, and compliance filings as services.
    - Integrate with bank and software systems on a case by case basis.

- **NGO and charity SaaS**

    - Donation platforms, CRM tools, basic reporting and transparency dashboards.
    - Usually closed, not interoperable between organizations or donors.

These actors provide important services and will continue to do so. What they do **not** provide is:

- A neutral, open, programmable entity layer shared across many jurisdictions and providers.
- A single common data model of entities that tools, donors, regulators, and DeFi protocols can use without custom integration each time.

Some offshore providers are experimenting with digital shares and online cap table management. They do not expose those structures as public, composable state.

3.3 On-chain competition

On-chain competition and adjacent projects include:

- **DAO and governance platforms**

    - Aragon, Safe based stacks, Tally, Snapshot, Juicebox, and others.
    - Focus on token voting, multisig treasuries, proposal front ends, and governance automation.

- **Legal wrapper projects for DAOs**

    - Connect DAOs to LLCs, foundations, or similar structures in specific jurisdictions via standardized templates.
    - Often treat the DAO itself as off-chain from a registry point of view.

- **On-chain identity and registry systems**

    - DID frameworks and verifiable credential ecosystems.
    - Name registries and specialized chains experimenting with legal entities.

These tools are valuable and dCorps expects to integrate with them. However, they typically do not:

- Provide a single, opinionated model for the full lifecycle of a corporation or NGO, including cap table, board, accounting, and operational flows.
- Act as a common base for multiple jurisdictions and sectors.
- Offer clear, entity centric data standards that institutions and accountants can read without learning a new DAO dialect.

dCorps is not a replacement for DAO tooling. It is a structured environment that DAOs and protocols can plug into when they want to operate as long term entities.

3.4 Adjacent and complementary projects

Many projects are natural partners rather than direct competitors:

- **Stablecoin providers**

    - USDC issuers and other reputable stablecoins.
    - Provide the monetary unit that entities use on dCorps.

- **DeFi protocols and lending markets**

    - Can use dCorps entity data to inform collateral parameters and product design.

- **Oracles and data providers**

    - Can ingest entity state and financial flows from dCorps and provide analytics, ratings, or risk signals.

- **Identity providers and KYC/KYB services**

- Can issue credentials, verify directors and key stakeholders, and integrate with DID-based identity on dCorps.

dCorps is designed so that these actors can integrate through standard interfaces rather than bespoke per entity contracts.

3.5 Differentiation and positioning

dCorps differentiates itself by:

- Treating **entities as first class objects**, not just tokens or accounts.
- Providing a **global, neutral base layer** focused specifically on corporations and NGOs.
- Defining a **minimal standard data and governance model** that others can build on.
- Keeping **jurisdictions and sector rules as protocol modules**, not core assumptions.
- Staying **non custodial** and **not a market operator**:
    - dCorps does not hold user funds as its core business.
    - It does not run exchanges, brokerages, or crowdfunding platforms.
    - It does not act as an asset manager or investment adviser.

In practice, using dCorps should feel more like using a shared registry and accounting substrate that many tools and jurisdictions can plug into, not like signing up to a single platform that owns everything.

---

4. Design principles and boundaries

4.0 Kernel invariants and the adapter boundary

Section 0.3 introduces the kernel and adapter model. This section makes that boundary explicit as invariants that any protocol change or module must respect. If a feature violates them, it is not part of the kernel.

Kernel invariants

1. **Canonical identity and discovery live on the Hub registry.**

2. **Canonical ownership and authority live on the Hub.** Units, roles, approvals, and executed resolutions are the source of truth.
3. **Canonical treasury and accounting events are recorded on the Hub** using standardized event types and schemas.
4. **Every entity can operate without any adapter** and without any fiat or bank rail dependency (not supported).
5. **Adapters may read kernel state and publish derived interpretations**, but they must not mutate kernel semantics or rewrite history.
6. **External recognition is derived and context-specific.** It can be attached when needed, but it is never required for correctness.
7. **Tooling must be interoperable.** Applications and modules integrate once through stable, versioned schemas and interfaces.
8. **Minimalism is a security feature.** The Hub evolves slowly, and extensions remain modular and replaceable.

These invariants are used as a design test for any proposed module, feature, or roadmap item.

4.1 Neutral and non custodial

dCorps is neutral infrastructure:

- Any entity that meets basic technical requirements can register without needing a special relationship with the core team or foundation.
- Multiple jurisdictions, service providers, and tools can compete and cooperate on top of the same base.

The protocol is non custodial:

- Entities keep control over their wallets and keys.
- The development corporation and foundation do not hold user assets as their core business.
- Custodial services may exist around the protocol as separate, regulated entities with their own responsibilities.

4.1A Digital-only boundary

dCorps is built for organizations that primarily operate in crypto and on-chain systems.

- The protocol is optimized for stablecoin native operations, DeFi-native treasury flows, and on-chain governance and accounting events.
- It does not replicate banking, fiat payment rails, or state-based corporate registries inside the protocol.

- Fiat or bank rail integrations are not supported at any layer.

This boundary is intentional. dCorps is meant to enable a digital economy that functions without fiat rails.

4.2 Separation of base layer and higher layers

The design draws a clear line between:

- **Base layer responsibilities**:
    - Entity registry and identifiers.
    - Entity types and internal models.
    - Cap tables, board structures, and governance events.
    - Wallet structures and accounting primitives.
    - Anchoring of documents and evidence packages.
    - DCHUB gas token and protocol governance (timelocked upgrades).
- **Protocol module responsibilities**:
    - jurisdiction adapter modules that encode local law and recognition logic.
    - Sector frameworks that define domain specific metrics and standards.
    - Other rule sets that read and interpret entity state, for example allocation frameworks and eligibility rules.
- **External application responsibilities**:
    - User interfaces and integrations.
    - KYC and verification flows where needed.
    - Markets, issuance flows, and donation portals.

Keeping these separate allows:

- Jurisdictions to adopt dCorps in different ways, without changing the Hub.
- Builders to innovate in protocol modules and application layers without touching consensus.
- The base layer to remain simple, conservative, and auditable.

The **dCorps foundation** focuses most of its work on protocol modules and ecosystem development, while helping keep the core Hub minimal and stable.

4.3 Transparency and verifiable state

Transparency is not marketing language, it is a property of the system:

- Ownership, governance, and key financial flows are recorded as state transitions, not only as documents.
- nonprofits have clear, on-chain donation and program spending categories.
- Corporations have **cash-based operating views** derived from tagged transaction flows (views over on-chain events).

At the same time, dCorps:

- Uses off-chain anchoring and selective disclosure patterns for sensitive data.
- Enables privacy-preserving approaches such as private zones, encrypted payloads, and zero knowledge proofs for selected use cases.

The aim is not full radical transparency of every detail, but verifiable transparency of what matters for trust.

In practice, chain integrity is only half the problem. Data integrity, meaning correct classification, correct links to evidence, and honest completeness, is harder.

dCorps addresses this by prioritizing typed workflows, evidence anchoring, and explicit coverage and assurance signals that make it clear what is verifiable and what is asserted (see section 9.5A).

4.4 Compliance aware, not compliance enforcing

dCorps is built to be **compliance aware**:

- jurisdiction adapter modules express rules about which types of entities are recognized, what fees apply, and what reporting is expected.
- Sector frameworks can require certain metrics or proof patterns.

The protocol does not claim that:

- Entity structures are automatically compliant anywhere.
- Tokens are automatically non securities or non regulated instruments.
- Using dCorps removes the need for legal advice.

Compliance remains the responsibility of entities and their partners. dCorps gives them better tools; it does not grant legal immunity.

4.4A Sanctions, AML, and KYC boundary

dCorps is neutral infrastructure and does not perform KYC, KYB, AML monitoring, sanctions screening, or customer due diligence at the base protocol level.

- **Who is responsible**
  - Entities, application operators, issuance platforms, custodians, exchanges, payment providers, and jurisdiction adapter module operators are responsible for implementing and complying with any KYC, KYB, AML, sanctions, travel rule, reporting, licensing, and consumer protection obligations that apply to their activities and jurisdictions.
  - Protocol modules may require credentials or attestations as inputs, but the issuance of those credentials and the legal responsibility for their correctness sits with the issuer and the relying party, not the Hub.
- **Stablecoin issuer enforcement and external controls**
  - Many stablecoins and bridges include administrative controls such as blacklisting and freezes. These controls are external to dCorps and may be exercised under issuer policy or legal process.
  - dCorps protocol governance and reference interfaces cannot unfreeze a stablecoin balance and cannot override issuer enforcement.
- **No evasion stance**
  - dCorps is not designed, marketed, or positioned as a tool for evading sanctions, AML obligations, or lawful enforcement.
  - Entities and users remain responsible for lawful use, and applications that facilitate prohibited activity may be blocked or restricted by external venues, issuers, or service providers regardless of protocol neutrality.
- **Practical limits**
- While the Hub avoids protocol level censorship features by design, rollup operators (sequencer and batch poster) and infrastructure providers may face external legal or operational constraints. This can affect transaction inclusion and network access in practice.

These boundaries are part of keeping the Hub minimal, neutral, and non intermediation focused, while enabling compliant actors to build responsible applications and modules on top.

4.5 No hidden super admin keys

Critical protocol components are designed to avoid hidden, unilateral control by any one actor.

- There are no secret keys or privileged backdoors that can silently:
    - Move entity funds,
    - Reassign ownership,
    - Bypass entity approvals, or
    - Rewrite historical state.
- If an emergency or upgrade role exists in early phases, it must be:
    - Explicitly defined in governance documents (scope, limits, and triggers),
    - Publicly disclosed (addresses, signers, and policy),
    - Executed transparently on-chain, and
    - Time bounded, with a clear path to reduction or removal as the network matures.
- Emergency and upgrade mechanisms must be constrained so they cannot be used as a substitute for entity governance (for example, they must not be able to arbitrarily alter cap tables, governance outcomes, or role assignments outside of defined upgrade paths).

This is essential for institutions, entities, and serious users to trust that the base layer behaves predictably over time.

4.6 What dCorps does and does not do

In summary:

- **dCorps does**:
    - Provide a neutral Layer 1 for entity structure and operations.
    - Standardize core models and interfaces.
    - Support optional protocol modules for jurisdictions and sectors.
    - Expose state in a way that auditors, donors, investors, and institutions can analyze.
- **dCorps does not**:
    - Act as a bank, deposit taker, or payment institution.
    - Operate as an exchange, broker, dealer, or asset manager.
    - Guarantee that any entity or token is compliant with any law.
    - Promise returns or financial outcomes.

This boundary is fundamental.

4.6A Regulated activity boundary map (reference)

This section provides a mechanical map of where regulated or high responsibility activity can and cannot live in the dCorps stack. It is not legal advice and does not classify any activity under any specific law.

Base protocol (Hub)

The Hub provides:

- Entity registry and structural state
- Governance records and document anchors
- Wallet structure and accounting primitives
- Anchoring of documents and evidence packages
- DCHUB gas and protocol governance (timelocked upgrades)

The Hub does not provide:

- Custody of user funds
- Fiat account access, payment services, or money transmission
- Brokerage, dealing, exchange, matching engines, or market making
- Underwriting, offering distribution, or investor solicitation
- Protocol level KYC, KYB, AML, sanctions screening, or travel rule compliance

Protocol modules

Protocol modules can provide:

- Jurisdiction recognition logic, fee collection logic, and reporting interfaces
- Sector frameworks, eligibility rules, and metric outputs
- Optional attestation and reputation systems with transparent schemas and dispute signaling

Protocol modules must not be assumed to provide:

- Licensing, legal guarantees, or universal compliance outcomes
- Custodial control of user assets as a condition for using the Hub

Any module operator that performs off-chain regulated functions does so outside protocol consensus, under its own legal obligations.

External applications and service providers

External applications and service providers are where regulated activity may occur, including:

- Custodial wallets and managed key services
- Fiat onramps and offramps, bank integrations, and payment processing
- KYC, KYB, AML, sanctions screening, monitoring, and reporting
- Issuance platforms for units or dShares, including marketing, distribution, and investor onboarding
- Exchanges, brokers, matching engines, and secondary market venues
- Accounting, payroll, and compliance services that touch private or regulated data

dCorps can be used by compliant actors, but it does not make an actor compliant. If an application operator, module operator, or service provider performs regulated activity, compliance and licensing are their responsibility, and the responsibility of the entity using them.

Registry listing is discovery, not authorization

- Listing in the app and module registry is not a license, a jurisdiction approval, or a legal endorsement.
- Official status signals protocol compatibility and governance approval for module standards. It does not certify legal compliance of any off-chain business activity.
- Users and entities remain responsible for due diligence, risk assessment, and legal compliance for the apps, modules, and providers they choose.

---

5. High-level architecture

5.1 Layer overview

Building on the kernel invariants in section 4.0, the ecosystem can be understood as four main conceptual layers:

1. **dCorps Hub chain (kernel)**
    - An Arbitrum Orbit rollup (Rollup mode) that settles to Ethereum and:
        - Hosts the global entity registry and canonical discovery.
        - Stores entity structural state, governance actions, and lifecycle events.
        - Stores standardized wallet primitives, accounting events, and document anchors.

- Uses DCHUB as the native gas token and protocol governance token.
- Enforces the kernel invariants described in section 4.0.

2. **Hub entities (default container)**

    - Corporations and nonprofits that live entirely on the Hub:

        - **Hub corporations** with an internal unit-based cap table and role-based governance.
        - **Hub nonprofits** with board based governance and transparent donation and program flows.

3. **Optional adapters and modules (derived state)**

    - Modules that read Hub state and publish derived interpretations or signals:

        - Jurisdiction recognition adapters and legal wrapper workflows (optional).
        - Institutional reporting and compliance tooling (optional).
        - Sector and impact frameworks that compute metrics from standardized flows.
        - Attestation and reputation modules.

4. **External applications and service providers**

    - Independent tooling built by the ecosystem:

        - Explorers and dashboards.
        - Accounting tools, payroll tooling, donor portals, and procurement tooling.
        - Integrations with markets and protocols that choose to rely on dCorps entity semantics.

The Hub is intentionally conservative and narrow. It defines stable semantics for entities. Adapters and applications provide innovation and context-specific integration without pulling external systems into the kernel.

Future extensions may introduce additional execution environments for extreme scale or specialized privacy, but they are not required for the v1 adoption path.

5.2 Base versus external responsibilities

The architecture separates base protocol responsibilities from optional adapters and applications. This separation is what keeps the kernel neutral and stable.

The Hub is responsible for:

- Assigning entity IDs and maintaining the canonical entity registry.
- Hosting Hub corporation and Hub nonprofit modules as the default entity containers.
- Recording governance actions, document anchors, and lifecycle events as verifiable state.
- Providing the wallet and accounting primitives that all entities share.
- Running DCHUB as the native gas token and coordinating protocol governance for upgrades and parameters.

Optional protocol modules on top of the Hub are responsible for (adapters, derived state):

- Jurisdiction and institutional adapters that interpret on-chain truth for external contexts (optional).
- Sector metrics and eligibility criteria as sector frameworks (optional).
- Attestation, reputation, and monitoring modules (optional).

External applications are responsible for:

- User interfaces and experience.
- Integration into existing workflows such as accounting systems, HR systems, procurement, and donor portals.
- Off-chain services that accept responsibility for local compliance, filings, or reporting when an entity opts in.

Future extensions (not required for v1) may include optional anchored execution environments. If introduced, they would be responsible for their own throughput and custom logic while anchoring standardized summaries back to the Hub.

5.3 Data flows and anchoring

Data flows through the system in consistent patterns:

- **Entity creation and updates**
    - Executed as EVM transactions against the registry and entity module contracts.

    - Entity metadata, roles, and module attachments are reflected as contract state and emitted events for indexers.

- **Operating flows**

    - Once stablecoins are bridged to dCorps, internal treasury, payroll, grants, and intra-wallet movements happen entirely on dCorps using canonical stablecoin contracts (ERC-20) and typed workflow contracts where used.
    - Inbound payments can originate on Ethereum: customers pay invoices in USDC on Ethereum (paying their own ETH gas) to an invoice/router contract, and the entity later bridges and settles the funds into its dCorps wallets, absorbing bridging and operational costs as a merchant-fee model.

- **Bridge gateways (Ethereum to/from dCorps)**

    - Canonical bridge gateways move assets and messages between Ethereum and the dCorps rollup.
    - Bridge interactions are asynchronous and subject to bridge finality and challenge windows; tooling should treat “pending bridge” state as a first-class status.

- **Documents and evidence**

    - Contracts, minutes, audits, invoices, and reports live off-chain.
    - Hashes of these documents (and optional Merkle roots of bundles) are anchored on-chain with minimal metadata for reproducibility.

Protocol modules may:

- Read anchored hashes and bridge events and interpret them for jurisdiction logic, sector metrics, or eligibility rules.
- Write derived state such as recognition status, allocation scores, or compliance signals.

Anchoring creates a verifiable timeline without placing sensitive documents on-chain, and without requiring additional execution environments beyond the Hub rollup.

5.3A Privacy, disclosure, and lifecycle (summary)

Privacy and disclosure are related but not the same. The protocol defines what must be visible (visibility policy), while confidentiality requires privacy-preserving execution or selective disclosure tools.

Each entity declares a disclosure mode at creation. The disclosure mode is public metadata used by explorers, registries, and modules:

- **Mode A**: public operations (maximum verifiability).
- **Mode B**: public structure with aggregate reporting (privacy-aware operations).
- **Mode C**: private execution with public anchoring (private zone).

Choosing a disclosure mode does not automatically provide confidentiality; it signals what is published and how proofs or aggregates are presented. Detailed privacy tiers and guarantees are defined in section 8.5A.

Every entity also carries a lifecycle status in the registry so counterparties can understand standing at a glance:

- **draft**
- **active**
- **suspended**
- **dissolved**

Lifecycle changes are recorded on-chain as governance actions and are part of the kernel record. See sections 5.4 and 5.5 and the Protocol Specification (docs/spec/SPEC-CORE.md) for lifecycle flows.

5.4 Example lifecycle: corporation

A typical corporation lifecycle on dCorps is not linear. The Hub container is complete by default. Entities branch into optional adapters only when needed.

1. **Formation (Hub corporation)**

    - Registers as a Hub corporation on the Hub.
    - Issues an initial unit distribution, assigns initial roles, and anchors baseline governance documents and policies.

2. **Operate on the Hub (default path)**

    - Runs treasury operations in stablecoins using canonical wallets and tagged accounting events.
    - Executes approvals, payroll, vendor payments, and governance decisions as on-chain state transitions.

- Produces reproducible cash-based operating views from the same underlying ledger events.

3. **Optional branches (adapters, when needed)**

    - **External recognition adapter**: attaches a jurisdiction or institutional adapter for a specific context (contracts, regulated counterparties), without changing the entity's kernel history.
    - **Sector frameworks**: publishes domain metrics and eligibility signals derived from standardized flows.
    - **Institutional reporting**: produces reports that external parties can verify against anchored evidence.

4. **Long-lived evolution on the Hub**

    - Restructures governance and cap table through standard corporate actions.
    - Uses pools, vesting, and claims patterns for finer-grained ownership and incentives without changing the base unit template.
    - Participates in group structures and multi-entity ownership graphs while remaining fully on the Hub.

5. **Future advanced execution modes (rare, post v1)**

    - Only if extreme scale or specialized privacy is required, the entity may use an additional execution environment that anchors summarized state back to the Hub. This does not redefine kernel semantics and is not required for adoption.

At no point is a corporation required to attach a jurisdiction adapter or migrate away from the Hub to remain functional.

5.5 Example lifecycle: nonprofit and umbrella sponsorship

A nonprofit in dCorps is a digital-native impact organization. Its mission, governance, and financial flows can be fully expressed and executed on the Hub. Legal charity status is an optional overlay, not the definition.

1. **Formation (Hub nonprofit)**

    - Registers as a Hub nonprofit with a board and governance rules.
    - Anchors baseline bylaws, policies, and program structure.

2. **Operate on the Hub (default path)**

- Receives donations into canonical donation wallets.
- Sets up program wallets, restricted fund rules, and allocation policies.
- Uses dashboards to show donors and supporters how funds are allocated, with category level transparency reproducible from ledger events.

3. **Umbrella sponsorship (optional pattern)**

    - A larger nonprofit can sponsor smaller initiatives:

        - The sponsor is itself a Hub nonprofit with established governance and reporting posture.
        - Sponsored initiatives can operate as dedicated programs and wallets with separate views.
        - Sponsorship is a service relationship, not a change in kernel semantics.

4. **Optional external overlays (adapters, when needed)**

    - A jurisdiction or institutional adapter can be attached for contexts that require external recognition (local charity registration, tax receipt workflows, regulated procurement).
    - Donor identity, attestations, and selective disclosure can be provided by applications and modules without becoming protocol requirements.

5. **Future advanced execution modes (rare, post v1)**

    - If specialized privacy or extreme volume is required, an additional execution environment may be used and anchored back to the Hub. This is not required for the v1 adoption path.

This path supports a long tail of nonprofits while keeping the Hub as the canonical, verifiable source of truth for governance and flows.

5.6 Why a dedicated Hub chain and token, not only contracts

It is natural to ask why dCorps needs its own Hub chain and a native token, instead of being only a smart contract suite on an existing chain.

dCorps runs a dedicated Hub chain as an **Arbitrum Orbit rollup (Rollup mode)** and uses **DCHUB as its native gas token and governance weight**, because the product is a **shared**

organizational standard, not a single application. Standards require stable semantics, predictable costs, canonical discovery, and long-term governance alignment.

Key reasons:

- **Entities are first class objects, not bespoke contract dialects**

  On general purpose chains, an organization is usually a contract or a set of contracts, and every team implements the model differently. That produces incompatible corporate dialects. The Hub treats entities as first class objects with standardized identity, ownership, authority, governance actions, and accounting events.

- **Standardization is the deliverable**

  If the goal is infrastructure for everyone, the main deliverable is shared schemas, indexing, discovery, comparable reporting, and composable permissions. A dedicated Hub makes the entity model canonical by default and enables an ecosystem of interoperable tools.

- **Predictable execution environment for organizational workflows**

  Organizational operations are routine and frequent (approvals, payments, reporting, role changes). A dedicated Hub can tune fees, throughput targets, and performance for these workflows, rather than inheriting unpredictable congestion and fee spikes from a host chain.

- **Governance and long-term stability aligned to organizational infrastructure**

  A standard cannot be hostage to external protocol politics. Running a dedicated chain allows dCorps governance and security posture to be aligned with kernel invariants and long-term stability goals.

- **Interoperability and mainstream stablecoin UX**

  dCorps is designed so that mainstream users can pay with stablecoins they already hold on Ethereum while entities operate day to day on dCorps. The Hub rollup uses canonical bridge gateways to move assets between Ethereum and dCorps, without making any single bridge a dependency for correctness of core entity registry state.

- **Why DCHUB exists**

  dCorps is shared infrastructure, not a single product. The Hub is a neutral registry, a shared execution environment, and a standard that many independent actors rely on,

which requires its own security, coordination, and governance layer rather than being embedded inside one DAO, one SaaS, or one app.

In practice, shared systems that aim for credible neutrality tend to converge on one of three funding and sovereignty paths: (1) a sponsor pays indefinitely, which risks capture over time, (2) the system fully inherits another network's economics and governance, which constrains sovereignty and couples the standard to external politics, or (3) the system operates a native token that prices execution and governs protocol evolution. dCorps chooses the third path.

DCHUB is not equity, profit participation, or entity ownership. It is the mechanism for execution pricing (gas), protocol governance weighting, and (where used) protocol-level fees or deposits for actions like entity registration/renewal and module registry operations. Interfaces may sponsor gas or charge stablecoin service fees for usability, but the Hub's base execution fees are paid in DCHUB.

Future execution and sovereignty extensions (including an eventual dCorps L1) may be explored later, but they are not required to justify the Hub. The Hub exists to make the entity standard canonical, stable, and neutral over decades.

---

6. The dCorps Hub chain

6.1 Role of the Hub

The Hub is the coordination heart of the ecosystem and the canonical home of the kernel.

- It is the **canonical record** of:
    - Which entities exist.
    - Their type, status, and lifecycle events.
    - Their kernel state roots (ownership, authority, governance record).
    - Which optional adapters and modules are attached.
- It provides a secure environment for:
    - Hub corporations and nonprofits that live entirely on the Hub.
    - Protocol level governance and upgrades.
    - Document anchoring and standardized accounting event history.

The Hub is intentionally conservative. Applications and adapters evolve faster, but the kernel must remain stable and predictable.

6.2 Entity registry

The entity registry on the Hub:

- Assigns each entity:
  - A unique ID.
  - A human readable name, subject to uniqueness and naming rules.
  - A type (Hub corporation, Hub nonprofit). (Additional types are reserved for future extensions.)
  - Sector tags.
  - Disclosure mode and reporting preferences.
  - Protocol module attachments (which modules are attached, and their attachment status).
- Emits events for:
  - Creation and termination.
  - Changes of control or key roles.
  - Attachments and detachments from protocol modules.

The registry is intentionally minimal and neutral. It records the existence and structural identity of entities plus their attachment graph. It is designed for protocol level neutrality and censorship resistance goals:

- The registry module does not include admin deletion, blacklists, or other controls intended to remove entities or filter registry content.
- Entities cannot be deleted from the Hub and historical state cannot be rewritten.
- Detaching from a protocol module is always possible according to the entity's governance rules and any module defined exit rules.
- Detaching ends the active relationship going forward, but it does not erase prior history.

This does not guarantee that every transaction will be included under all real world conditions. Transaction inclusion depends on rollup sequencing, network conditions, and the Ethereum settlement path. Sequencer operators and other infrastructure parties may face external legal or operational pressure. dCorps is designed without protocol level censorship features and with an intent toward neutrality, while recognizing these practical limits.

6.2.1 Status meanings and renewal logic (registry standard)

Status is a classification used for discovery, safety, and ecosystem hygiene. It is not a claim of legal standing, and it is not a claim that an entity is legitimate or illegitimate.

- **active**
  - The entity has a current registry listing, meaning it is within its renewal window and has satisfied any required registry renewal fees and rules.
  - Reference explorers may include it in default “active entities” views and indexes.
  - Active status is required for defined registry privileges, such as name lease continuity and participation in default discovery and reference payment routing (where used).
- **inactive**
  - The entity record still exists and remains queryable, but the entity has not been actively renewed within the defined renewal window (after any grace period).
  - This state exists to keep the registry clean and to avoid presenting stale entities as currently operating.
  - Inactive status does not imply dissolution, fraud, or invalidity; it is a registry liveness and renewal signal.
  - An inactive entity can become active again by completing renewal according to registry rules.
- **dissolved**
  - The entity has explicitly ended operations through an on-chain governance action, a jurisdiction adapter action, or another defined termination path.
  - Dissolved does not erase history.
- **retired**
  - A terminal registry classification used when an entity is intentionally decommissioned from active use without implying legal dissolution (for example, a migrated structure that is kept only for historical continuity).
  - Retired does not erase history.

What renews

Renewal applies to the entity’s public registry listing and related registry privileges, not to the existence of the entity record itself. Examples of renewable items include:

- Registry listing lease (participation in “active” classification and default discovery).
- Name lease (continued exclusive claim on the human readable name, if names are leased rather than permanently reserved).
- Optional registry services (for example, premium namespaces or enhanced indexing categories, if adopted).
- Optional module participation requirements, where a module itself requires periodic renewal to remain attached (module local rule, not a Hub deletion rule).

The entity object, event history, governance history, and prior attachments remain permanently on-chain regardless of renewal status.

6.2.1A Anti-spam, name squatting, and abuse handling (registry standard)

A public registry attracts spam, squatting, and impersonation attempts. The Hub addresses this with economic friction and clear interface conventions, while keeping the kernel neutral.

Economic spam resistance (protocol level)

- **Entity registration and renewal fees** create a recurring cost to keep a listing active. This keeps the default discovery set clean without requiring admin deletion.
- **Name leases** (recommended) treat human readable names and premium namespaces as renewable leases rather than permanent property. This reduces permanent squatting and makes abandoned names recoverable over time.
- **Premium name pricing** (where used) can be parameterized to increase costs for scarce names (for example very short names, high-demand namespaces, or reserved words), with transparent on-chain rules.

Expiration and cooldown (recommended)

- Names and registry listings can include a defined **grace period** after expiry.
- After grace, a name can enter a **cooldown window** before it becomes available again, reducing opportunistic front-running and giving the prior holder time to recover.

Neutrality and what the kernel does not do

- The registry module does not include admin deletion or an on-chain blacklist. Records remain queryable and history is not rewritten.
- The protocol does not attempt to adjudicate trademarks or identity disputes at the kernel layer.

How ecosystems handle abuse without changing the kernel

- Reference explorers and wallets are expected to implement clear UI warnings for inactive entities and to support additional labeling signals (for example “verified,” “contested,” or “impersonation risk”) based on optional attestation or reputation modules.
- Sector frameworks, jurisdiction adapters, and third-party attesters can publish signed lists and interpretations. These are opt-in overlays that help users avoid abuse while preserving kernel neutrality.

6.2.2 Payment safety and registry privileges for inactive entities (reference standard)

To reduce the risk of funds being sent to stale or abandoned operations, the registry exposes the entity’s status and standard wallet identifiers (merchant, donation, treasury, program wallets) intended for use by explorers, invoicing tools, and payment interfaces.

Design intentions:

- When an entity is **inactive**, reference interfaces must clearly label it as inactive and should block “one click” sends by default unless the sender explicitly acknowledges the risk.
- Ecosystem payment and invoicing flows should resolve destinations using the entity ID plus wallet type, not by caching addresses indefinitely, so that senders are nudged toward current canonical wallets and current status.

Registry privileges tied to active status (recommended)

Active status is used to gate defined registry privileges that are meaningful even if the protocol cannot globally prevent direct transfers to an address:

- Default discovery and listing in reference explorers (active by default, inactive behind warnings or filters).
- Continued name lease, where naming is lease based.
- Resolution of canonical payment endpoints (entity ID plus wallet type) in reference tooling. When inactive, resolution surfaces an explicit inactive status signal and requires an override path in the UI.
- Optional participation in certain registry services (premium namespaces, enhanced indexing categories), where offered.

Canonical payment routing (optional, recommended for safety critical flows)

For applications that want stronger safety guarantees, the ecosystem can use a standard payment routing pattern:

- A sender submits a payment by calling a standard “pay entity” router contract that specifies:
  - `entity_id`
  - `wallet_type`
  - token contract address (for example USDC)
  - amount
- The router checks the entity’s current status at execution time and can:
  - forward to the current canonical wallet address only if the entity is active, or
  - reject (or route to an explicitly defined escrow pattern) if the entity is inactive.

This pattern is enforceable only for senders that choose to use it. Direct, raw token transfers to an address remain possible at the base ERC-20 level and cannot be universally blocked on an EVM chain. For that reason, reference interfaces must treat canonical payment routing and raw

transfers differently, and must label when a payment was routed using an active status check versus sent directly to a raw address.

The Hub registry does not include a built in reputation system and does not maintain a global “risk flag” bulletin board. Any assurance, scoring, or reputation logic exists only through optional protocol modules that read Hub state and publish their own outputs.

This registry is the main integration point for explorers, analytics, jurisdictions, and external applications.

6.2A Attestation issuer registry and interface weighting

Attestations and reputation are not part of the Hub core registry. They are implemented as optional protocol modules that can be attached to, or used alongside, Hub entities by choice.

This section defines a reference standard for attestation style modules and for how explorers and dashboards should present their outputs.

6.2A.1 Attestation record format (module standard)

An attestation is an issuer signed statement published by an attestation module. Attestation modules must use structured, machine readable records and must not rely on free text accusations.

A compliant attestation record includes, at minimum:

- Issuer identity (DID and signing address).
- Issuer scope (jurisdiction authority, auditor, sector framework operator, oversight body, analytics provider, and similar scopes).
- Subject entity ID.
- Attestation type (enumerated), for example:
    - Recognition active
    - Recognition withdrawn
    - Audit or review anchor published
    - Reporting completeness signal
    - Policy compliance signal
- Reason codes (enumerated), sufficient to interpret the attestation without narrative text.
- Evidence anchors (hashes of documents, reports, or module outputs).
- Validity window:
    - Issued at
    - Effective at
    - Optional expiry

- Lifecycle status (enumerated):
    - active
    - expired
    - superseded
    - disputed
    - withdrawn

Attestation types, reason codes, and evidence anchor schemas are versioned, published, and upgradeable through governance of the module, not by informal UI conventions.

6.2A.2 Issuer classes (module local, optional)

Attestation modules may maintain an issuer registry to make trust assumptions explicit. Issuer registries are module local, not universal, and different modules may use different issuer sets.

- **Registered issuer** An issuer added to a module's issuer registry through the module's governance process, with:
    - DID and signing addresses
    - declared scope
    - public issuer metadata (including contact channels)
    - published correction process
    - optional bond requirements
- **Unregistered issuer** Any other issuer. Unregistered attestations may still be recorded by the module, but they do not receive default weighting in reference interfaces that follow the module's weighting standard.

6.2A.3 Issuer registry governance (module local)

Issuer registry changes are executed through transparent governance.

Adding or removing a registered issuer requires an on-chain proposal that includes:

- Issuer identity and scope
- rationale and supporting evidence
- any required bond amount and conditions
- conflict and affiliation disclosures

Registered issuer status can be suspended or revoked for objective reasons such as:

- proven fraud or misrepresentation
- repeated publication of materially false attestations without timely correction
- failure to follow the issuer's published correction process
- cryptographic key compromise without appropriate rotation

Issuer registry governance does not adjudicate truth. It defines which issuers a specific module treats as default weighted inputs.

6.2A.4 Dispute and correction signaling (module standard)

Attestation modules must support a symmetric dispute and correction pattern:

- Entities can publish a signed dispute record that references a specific attestation and anchors response evidence.
- Issuers can withdraw, correct, or supersede attestations with new signed statements.
- Disputed status is a first class lifecycle state, not a UI convention.

The Hub does not adjudicate disputes. Dispute resolution occurs through:

- the issuer's correction process,
- module governance (for module level integrity issues), and
- off-chain legal processes where jurisdictions or contracts apply.

6.2A.5 Interface weighting standard (reference explorers)

Reference explorers and dashboards that choose to display attestation module outputs must:

- Allow users to select which attestation modules are shown, and support "no module selected" as a valid default state.
- Display issuer identity, issuer scope, registry status (registered or unregistered), attestation type, reason codes, evidence anchors, and validity window.
- Label disputed, expired, and superseded attestations prominently.
- Treat detachment from an attestation or reputation module as a neutral structural fact, not as deletion of history.

No reference interface is consensus. Any interface can be forked or replaced, and any operator can publish an alternative weighting policy. The protocol remains neutral and censorship resistant regardless of interface decisions.

6.2B Spam resistance and deposits for attestations

To keep attestation modules usable and resistant to spam, attestation modules are expected to implement spam resistance rules.

6.2B.1 Publishing limits

- Each issuer may be subject to rate limits per period for selected attestation types.

- Attestation modules enforce maximum payload sizes for attestation metadata, with evidence stored as anchors.

6.2B.2 Deposits and bonds (optional)

For selected high impact attestation types, an attestation module may require a deposit or bond:

- The deposit is denominated in DCHUB or USDC, as defined by module parameters.
- Deposits are designed to price spam and abuse, not to create an endorsement market.
- Any forfeiture conditions must be objective, verifiable, and narrowly defined (for example invalid schema, duplicate flooding, or proven key misuse), not based on the content or popularity of an attestation.

Registered issuers may be exempt from per attestation deposits but may be required to maintain a module level issuer bond.

6.2B.3 Dispute rights

Entities can publish dispute records without deposits. Disputes must reference the specific attestation and include evidence anchors. Reference interfaces must surface disputes prominently for any disputed attestation they display.

6.2C Reputation and scoring modules (optional)

Reputation modules are optional protocol modules that compute scores from defined inputs. Reputation is not a core Hub function.

A reputation module is a metric module, not a narrative system:

- Outputs are numeric scores and subscores, plus enumerated reason codes and source anchors.
- No free text accusations are required or expected.
- Inputs may include:
    - Objective on-chain behavior (anchoring cadence, governance participation, disclosure mode, reconciliation anchors present).
    - Outputs from jurisdiction adapter modules (recognition active or withdrawn).
    - Outputs from sector frameworks (allocation constraints met or missed).
    - Attestations from selected attestation modules and issuer sets, if the reputation module chooses to incorporate them.

Entities can attach to, or detach from, reputation modules. Detachment does not erase history; it ends new score updates for that entity under that module unless reattached. Counterparties, applications, and markets decide what module participation they require for trust.

6.3 Structural state on the Hub

In addition to the registry, the Hub stores structural state for:

- **Hub corporations**
    - Unit balances and cap table changes.
    - Governance parameters and voting rules.
    - Attachments to jurisdiction and sector modules.
- **nonprofits**
    - Board composition and roles.
    - Allocation rules for donation and program wallets.
    - Governance events and resolutions.

Structural state is designed to be:

- Compact and indexable.
- Sufficient for high-level analysis, even for entities that run complex operations across multiple applications and external settlement rails.

6.4 Future extension: specialized execution modes (post v1)

This section is a future extension. v1 does not require any additional execution environments for entities to operate.

Some organizations may eventually require specialized confidentiality, scale, or domain-specific execution. These needs can be addressed through:

- Protocol modules that verify proofs and anchor commitments.
- Privacy-preserving applications that keep sensitive logic off-chain while anchoring evidence and hashes on-chain.
- External systems that publish attestations to the Hub under explicit, reviewable rules.

These extensions must treat the Hub as canonical and must not redefine kernel semantics or rewrite history. Official interfaces may surface optional references and attestations, but the base entity registry remains complete on the Hub.

6.5 Rollup security and operations model

The Hub is deployed as an **Arbitrum Orbit rollup (Rollup mode)** that settles to Ethereum.

- **Execution**: EVM smart contracts define the kernel (registry, entities, governance, accounting primitives, anchoring).
- **Ordering**: a sequencer orders transactions and publishes batches.
- **Settlement**: rollup state is ultimately settled on Ethereum; Ethereum is the final settlement and data-availability layer for the rollup.
- **Gas**: transactions on the Hub pay gas in DCHUB.

This model replaces a sovereign proof-of-stake consensus design. It introduces a different set of operational and governance risks, which dCorps treats as first-order design constraints.

6.5A Chain owner reality, upgrades, and timelocks

In early phases, an Orbit rollup has an explicit administrative control surface (often described as the *chain owner*) for things like:

- Protocol upgrades and configuration changes.
- Emergency response controls (where implemented).
- Bridge gateway configuration and security parameters.

dCorps treats these powers as a temporary operations layer, not a hidden backdoor:

- Privileged actions must be on-chain and executed through a publicly disclosed multisig and timelock.
- Privileged actions can change future behavior (for example upgrading contracts), but they do not retroactively erase the public record of events.

6.5B Progressive decentralization plan (accurate, staged)

The intended path is progressive decentralization that reduces concentrated control over time:

- **Stage 0 (launch)**: a small multisig acts as chain owner behind a timelock; the upgrade policy and emergency posture are public.
- **Stage 1**: transfer chain owner and critical admin rights to on-chain governance controlled by DCHUB, keeping a conservative timelock and Protected Change rules.
- **Stage 2**: broaden operations (sequencing, infrastructure, monitoring) to multiple independent operators where feasible; minimize emergency powers; keep governance transparent.
- **Stage 3 (future)**: if dCorps launches its own L1, introduce staking-based security and validate any migration path through governance.

Even under early centralized operations, protocol governance cannot erase contracts or prevent generic asset transfers. Governance can change official registry labels, module eligibility, and reference-interface policy under transparent rules.

6.5B.1 Sequencer operations and liveness

The rollup's sequencer orders transactions and publishes batches to Ethereum. If the sequencer is offline or degraded, transaction inclusion may be delayed.

Design intentions:

- Publish clear liveness expectations and incident playbooks for sequencer operators.
- Prefer multiple independent operators over time where feasible within the Orbit stack.
- Use transparent monitoring so outages are visible to users and to governance.

6.5B.2 Bridge gateways and asset entry

Stablecoins and other assets reach dCorps through canonical bridge gateways between Ethereum and the Hub rollup.

- Deposits from Ethereum are confirmed on Ethereum (ETH gas paid by the sender) and credited on dCorps.
- Withdrawals back to Ethereum follow the rollup's standard withdrawal path and timing.

Bridges are high-risk components. Core entity registry state and governance history remain correct even if a bridged asset fails.

6.5B.3 Operating costs and fee model (rollup reality)

Although users pay L2 gas in DCHUB, the rollup incurs Ethereum costs (data posting, settlement) that are paid in ETH by operators.

dCorps expects these costs to be covered through a mix of:

- Protocol service fees denominated in DCHUB (quoted in USDC-equivalent terms for UX or covered via fee grants).
- Application and merchant-fee models where entities absorb bridging and operational costs to keep payment UX mainstream.
- Governance-managed budgets for core infrastructure, audits, and monitoring.

This is an operations model, not a guarantee of fee levels.

6.5B.4 Governance limits: what can and cannot be changed

Protocol governance (DCHUB-weighted) can:

- Update registry parameters and official registry labels.
- Add, deprecate, or flag modules in the official module registry.
- Upgrade protocol contracts through timelocks (when empowered to do so).

Protocol governance cannot:

- Delete historical events or erase the existence of an entity.
- Prevent generic asset transfers between addresses at the base token level.

This separation keeps registry signaling powerful for safety and discovery without turning the base layer into a censorship mechanism.

6.5B.5 Module and interface behavior under registry and module policy (v1 standard)

Registry labels and module registry status are operational inputs for official modules and reference interfaces:

- **Reference interfaces** must display entity status and module status prominently and show warnings when an entity is inactive, deprecated, or flagged.
- **Protocol modules** may declare eligibility requirements based on registry status and required attestations (for example only act on entities that are active).
- **Governance changes** to these policies must be transparent, time-locked, and reflected as on-chain events so downstream tooling responds deterministically.

6.5B.6 Fault attribution and incident response under rollup operations

In a rollup architecture, most critical failures are contract bugs, bridge failures, sequencer outages, or governance/key misuse.

Response patterns include:

- Conservative upgrades with timelocks and clear disclosure.
- Emergency controls (where implemented) that are explicit, time-bounded, and governed.
- Independent indexers and transparency tooling so that incidents are auditable.

There is no staking-based security or penalty mechanism in the Orbit rollup configuration. Accountability is expressed through governance processes, operational transparency, and the ability for users and integrators to change exposure based on visible risk signals.

6.6 Upgrade and minimalism

The Hub strives to be:

- **Minimal**
    - Only core modules needed for entity registry, base entity models, and protocol governance live on the Hub.
    - Experimental or high-risk features are developed on testnets, staging environments, or as optional protocol modules before any move to the Hub.
- **Upgradable with care**
    - Protocol upgrades follow a structured path:
        - Specification and code review.
        - Testnet deployment.
        - Governance proposal with clear rationale and risk analysis.
        - Staged rollout with monitoring.
- **Resilient**
    - Avoids tight coupling between unrelated features.
    - Uses well understood components and patterns where possible.

This approach balances the need for evolution with the reliability that entities, NGOs, and institutions require.

The foundation's role is to propose and iterate on protocol modules and tooling as needs evolve, while aiming to keep the Hub's core logic as stable and minimal as practical.

---

7. Entity models on the base layer

7.1 Hub corporations

7.1.1 Ten thousand unit model

By default, each Hub corporation uses a **10,000 base unit** cap table on the Hub.

Ten thousand is the default interoperability profile for simple mental math and comparability. It is not meant to limit real share structures.

v1 clarity: the base unit count is not forced

In v1, a corporation may choose an **expanded base unit count** when it is created, or it may adopt one later through a one-time **unit expansion** corporate action approved under its own governance thresholds.

- The default base unit profile is **10,000** base units (1 unit = 0.01 percent).
- Expanded base unit counts are allowed in v1 as **multiples of 10,000** (for example 100,000 or 1,000,000), which increases precision while preserving the same mental model.
- A unit expansion is a mechanical action similar to a split: it does **not** change relative ownership or voting percentages, it only increases the number of base units used to represent them.
- Tools and modules must read an entity's declared base unit count from the registry and should display ownership primarily as percentages, with units as a secondary view.
- The protocol does not enforce a maximum base unit count, but v0.1 templates and reference tooling assume a practical guardrail: **recommended maximum 1,000,000 base units** for interoperability and UI performance.

This design keeps the Hub simple for small teams while allowing advanced private corporation structures to remain Hub-first on a public chain.

Ten thousand is a deliberate default:

- It keeps the mental model simple. One unit is 0.01 percent of the corporation. A partner who owns 34.54 percent holds 3,454 units in the default base unit profile.
- It makes Hub corporations easy to compare and reason about, since most private corporations on the Hub share the same default unit profile.
- It discourages unit inflation games and awkward decimals, which are common problems with arbitrary token supplies.

These base units are the internal equivalent of ownership in a traditional LLC or Inc style structure:

- Units represent:
    - Economic rights such as entitlement to distributions and exit proceeds.
    - Voting power in unit-holder decisions.
    - Exposure to dilution when new units are issued.

Units are an internal ledger representation of rights defined by the corporation's governing documents and chosen governance templates. Their legal effect, if any, depends on agreements and law outside the protocol.

- By default, a simple **one unit, one vote** model applies for unit-holder votes.

- Corporations can adopt alternative models through governance templates, such as:
    - Non voting units.
    - Weighted voting structures.
    - Multiple classes with different rights or preferences.

Units are internal to the corporation:

- They are not global fungible tokens.
- Transfers and balances are tracked in the corporation's own state on the Hub.
- External agreements may refer to unit holdings for calculations, but units do not circulate outside that entity context.

Units are a protocol cap table primitive. They are on-chain state scoped to a single entity, used to represent ownership and voting relationships as defined by the entity's governing documents and, where applicable, by attached optional modules. Whether a particular unit arrangement is treated as equity, a security, or another regulated instrument depends on facts and law outside the protocol. Transfer restrictions and approval workflows enforce on-chain process and evidence trails; they do not, by themselves, guarantee legal compliance.

Optional precision for grants and vesting (v1 default)

Some entities need allocations smaller than 0.01 percent or need to model fine grained vesting and entitlements without changing the base 10,000 unit model.

To keep "small entity" onboarding simple while supporting precision when required, dCorps supports an optional, additive module pattern:

- **Allocation pool** The corporation can allocate a defined number of units to a pool address (for example an employee or contributor pool). The pool ownership remains on-chain and visible.
- **Claims ledger** Inside the pool, the corporation can track fine grained economic claims using a fixed point claims ledger (for example micro units or basis points accounting), typically non transferable by default.
- **Event mapping** At economic events (distributions, exits, redemptions, or other defined events), the pool maps claims to outcomes according to published rules and governance approvals, with anchors to supporting documents where required.

This pattern preserves the simplicity and comparability of the 10,000 unit cap table while enabling precise grants and vesting for startups and contributor heavy entities. Entities that do not need this precision never have to touch it.

When structures become more complex, there are three main paths, all Hub-first:

- Use standardized pools and claims on top of the 10,000 units for precision without changing the base model.
- Use optional Hub extensions that define class-like rights, vesting schedules, conversions, and conditional instruments, while keeping the entity kernel and discovery on the Hub.
- Only in rare future cases, use an additional execution environment for extreme scale or specialized privacy, anchored back to the Hub, without making it a requirement.

7.1.2 Ownership, voting, and economic rights

Ownership of units is expressed as balances associated with wallets and DIDs:

- Corporations can set:
    - Who is allowed to hold units.
    - What consent is required for certain transfers.
    - How votes are counted for different resolution types.

Economic rights tied to units can include:

- Profit distributions.
- Exit proceeds, for example sale of the corporation.
- Participation in rights issues or other capital actions.

Voting rights can be:

- Simple majority with defined quorum.
- Supermajority for structural changes.
- Weighted or class specific where templates allow.

On-chain governance actions link resolutions to:

- The units that participated.
- The DIDs and role wallets that initiated or approved them.
- Anchored legal documents where applicable.

7.1.3 Transfers, restrictions, and approvals

Units can be:

- Transferred between holders.
- Issued for new capital or compensation.
- Cancelled or bought back.

Transfers can be subject to:

- **Restrictions**:
    - Lockups for founders and employees.
    - Waiting periods for investors.
    - Whitelist rules where law requires only certain investors or jurisdictions.
- **Approvals**:
    - Board or shareholder approval for certain transfers.
    - Automatic checks by protocol modules where jurisdiction rules apply.

The protocol can enforce:

- That a unit transfer passes through an approval workflow.
- That a transfer is allowed only for whitelisted holders.
- That locked or unvested units cannot be transferred.

Compliance with law is not guaranteed by these patterns, but they make it much easier to implement and audit restrictions.

7.1.4 Corporate actions and group structures

Hub corporations are designed to support both simple LLC-like structures and advanced private corporation structures. The Hub records corporate actions as standardized on-chain events, linked to resolutions, approvals, and document anchors.

Common corporate actions that can be expressed on the Hub include:

- Issuance of new units (new financing, compensation, contributor grants).
- Cancellation, repurchase, and redemption of units (when the entity chooses to model them on-chain).
- Mechanical cap table changes that preserve proportional ownership:
    - Unit splits and consolidations.
    - Base unit count expansion (a precision increase), where desired.

- Transfers governed by restriction and approval policies (lockups, vesting locks, whitelists, right-of-first-refusal style flows, and board or unit-holder approvals).
- Creation, modification, and conversion of unit classes through governance templates:
    - Voting versus non-voting classes.
    - Weighted voting or protected matters.
    - Class conversions (for example preferred-to-common).
- Conditional instruments and entitlements expressed as standardized templates, such as:
    - Option-like grants and warrants (typically non-transferable by default).
    - Convertible claim templates (note-like or SAFE-like), expressed as conditional conversion rules and on-chain claims.
- Pools and claims for vesting and fine-grained allocations (v1 standard optional precision).
- Distributions and dividend-like payouts, including on-chain waterfall rules when the distribution itself occurs on-chain.
- Mergers, asset sales, restructurings, and dissolutions recorded as governance actions with anchored agreements and resulting ownership changes when executed on-chain.

Not every economic term can be enforced purely by the kernel. The design intent is:

- Enforce what is purely on-chain (voting, restrictions, vesting, distribution math), and
- Record and anchor what depends on external venues or counterparties, while preserving a clear evidence trail.

Group structures are supported by allowing:

- Entities to hold units or dShares of other entities.
- Clear mapping of multi entity ownership chains.

This same mechanism supports joint ventures and SPVs. Two or more corporations can hold units or dShares in a dedicated entity that runs on the Hub. The JV or SPV uses the same corporate primitives as any other dCorps corporation, but its wallets, governance rules, and reporting are tailored to a specific project or asset and are clearly separated from each parent entity’s own operations.

Nonprofits do not have equity and are not “owned” by units. They can still participate in group structures by holding units in Hub corporations (for-profit subsidiaries) and by anchoring off-chain control documents where relevant (for example membership or appointment rights), while using board governance as their primary on-chain control surface.

Legal and tax consequences of group structures depend on jurisdiction and must be handled by advisors, but dCorps provides a precise map of the ownership relationships.

7.2 Future extension: public instruments (dShares) and regulated distribution

Note: this section describes a future extension. v1 mainnet focuses on Hub corporations and Hub nonprofits as complete, long-lived containers. Public instruments are not required for v1 adoption and are not part of the default entity lifecycle. This material is included for completeness and may evolve.

7.2.1 Why public instruments are optional

Not every entity needs public instruments. Public issuance and trading introduces overhead:

- Disclosure and reporting obligations.
- Eligibility rules, transfer restrictions, and market-venue constraints.
- Higher security expectations and operational complexity.

For many small and medium entities, **remaining a Hub corporation is the right long term choice**.

7.2.2 dShares as an equity style mechanism

Public-instrument issuers may issue **dShares** as on-chain tokens (for example ERC-20-style instruments) under their own legal regime:

- dShares represent the corporation’s equity style interests on-chain.
- dShares can carry:
    - Governance rights such as voting on board elections or major corporate actions.
    - Economic rights such as dividends or buybacks, where allowed by law and corporation policy.

dCorps:

- Does not design dShare terms.
- Does not issue, custody, or manage dShares for entities.
- Provides standards for how public instruments are represented on the Hub and how required disclosures and key supply/cap-table state are surfaced through on-chain events and anchors.

Issuers and their advisers must ensure that dShare design and distribution comply with applicable law.

7.2.3 Protocol boundary for dShares and law

The boundary is:

- dCorps defines:
  - How dShares are represented technically on the Hub (token contracts plus registry references).
  - How cap table snapshots, supply summaries, and disclosure anchors are published as on-chain state and events.
  - How governance events and corporate actions are exposed as state.
- Law and regulators define:
  - Whether a given dShare is a security or another regulated instrument.
  - What disclosures and approvals are required.
  - Which investors can hold or trade them and in which venues.

jurisdiction adapter modules can link these two worlds:

- A jurisdiction, or a delegated service provider that acts under its supervision, can encode in a module:
  - Which types of dShare offerings it is willing to recognize.
  - What minimum disclosure standards apply.
  - Which categories of investors are eligible.
  - How ongoing reporting should work.
- Modules can require specific **off-chain documents** and **on-chain anchors**, for example:
  - A jurisdiction may require that any dShare offering under its regime file a prospectus or information document off-chain and publish a hash of that document on the Hub before new holders can be added under that module.
  - Periodic financial statements can be anchored by hash and period, so that regulators and investors can verify they have the right version.

Even when a jurisdiction adapter module exists, the **legal effect** comes from local law and contracts that refer to the module, not from the chain itself. The module is a technical expression of rules that the jurisdiction or its delegated providers choose to enforce.

dCorps does not certify that any dShare or module is legally compliant. It provides the primitives and interfaces so that jurisdictions, issuers, and service providers can implement their obligations more transparently and programmatically.

7.2.4 Registry listing and interface policy for public instruments

To be listed in the official registry as a public-instrument issuer, an entity must:

- Register the instrument in the Hub registry (token contract address plus metadata).
- Anchor required disclosure documents and any required issuer attestations.
- Declare the transfer and eligibility posture (for example which jurisdiction adapter modules apply).
- Accept that official interfaces may apply conservative defaults (warnings, eligibility filters) based on module outputs.

Entities are free to issue tokens on other networks or under other standards. Those instruments are outside the scope of dCorps official registry signals and do not receive automatic support from official modules and reference interfaces.

The Hub then:

- Records instrument metadata and links it to the issuing entity.
- Exposes relevant events and anchors to explorers, modules, and tools.

Governance can de-list or flag an instrument in the official registry, but it cannot erase deployed contracts or prevent generic token transfers at the base layer.

Registry listing is technical and data-oriented. It does not mean a regulator has approved the corporation or that dShares are suitable for any user group.

7.3 Nonprofit entities

Nonprofits on dCorps are **digital-native impact organizations**. Their governance and financial flows are expressed as verifiable on-chain state and executed through programmable rules.

A Hub nonprofit is complete on the Hub. External charity registration, tax receipt workflows, and local compliance are optional overlays implemented through applications and adapters. They are not required for the nonprofit to exist, raise funds in stablecoins, allocate budgets, or publish transparency.

The goal is to make nonprofit operations legible and auditable by default, while allowing selective disclosure patterns when needed.

7.3.1 Board based governance

nonprofits on dCorps:

- Have **boards** represented by DIDs and wallets.
- Define roles such as chair, treasurer, secretary, and ordinary members.
- Configure rules for:
    - Quorum.
    - Majority and supermajority thresholds.
    - Term lengths and rotation.

Board decisions are recorded as proposals and votes:

- Each vote is tied to a board seat and DID.
- Proposals include links to anchored minutes or documentation where appropriate.

This gives donors, partners, and auditors a clear picture of how decisions are made.

7.3.2 Donation and program flows

nonprofits use:

- A **donation wallet** as the primary income wallet for donations and grants.
- Optional **program wallets** for specific initiatives.

All inflows and outflows are:

- Denominated primarily in approved stablecoins (USDC as the baseline reporting currency).
- Tagged with:
    - Type (donation, grant, program cost, overhead, fundraising cost, internal transfer, and similar categories).
    - Counterparty type (donor, beneficiary, supplier, staff, partner NGO, jurisdiction, and similar categories).

This supports:

- Real time or near real time views of allocation.
- Yearly reports built directly from on-chain data.
- Cross NGO flows that are transparent rather than hidden.

7.3.3 Allocation rules as code

nonprofits can encode **allocation rules** as contracts, for example:

- Minimum program spending ratio over a rolling period.
- Maximum overhead or fundraising ratio.
- Upper limits on board compensation or per category expenses.

Changing these rules requires:

- A board vote explicitly tied to the rule.
- A recorded governance event with an anchor to the revised policy.

Sector frameworks or other protocol modules can also:

- Monitor adherence.
- Raise alerts when entities drift from their own rules.
- Apply consequences such as flagging entities in dashboards or affecting eligibility for certain programs.

7.3.4 Transparency guarantees

The core nonprofit module guarantees:

- Public visibility of:
    - Incoming donation and grant amounts.
    - Outgoing spending at least at category level.
    - Board governance events and allocation rule changes.

nonprofits can still keep:

- Beneficiary identities and sensitive details off-chain or in controlled zones.
- Detailed breakdowns private except when shared with specific auditors or regulators.

The guarantee is that money and high-level categories are visible, not that every detail is public.

7.3.5 Donors, identity, and umbrella NGO pattern

Donor identity and privacy are handled carefully:

- Donors can be individuals, corporations, or pooled funds.
- Donors may appear on-chain only as addresses or DIDs.
- Where regulations require KYC, protocol modules or off-chain services can link donors to verified identities.

Umbrella NGOs:

- Can manage multiple program wallets that correspond to smaller partner NGOs or project teams.
- Provide governance, compliance, and reporting infrastructure on top of dCorps.
- Help smaller organizations plug into the system without managing everything themselves at first.

This supports a diverse ecosystem of nonprofits while keeping flows coherent.

7.3.6 Practical digital-only operating pattern

A Hub nonprofit is designed to operate fully on-chain.

A practical operating pattern looks like:

1. The nonprofit receives donations and grants directly in one or more approved stablecoins into its **donation wallet** on the Hub. Structured giving (grants, sponsorships, memberships) may use invoices or recurring plans, while ad-hoc donations do not require any invoice.
2. The nonprofit allocates funds to program wallets and operational categories through **board approved allocation rules**, ideally using typed workflows that emit deterministic categories.
3. The nonprofit executes program spending, payroll, grants, and partner disbursements on-chain, with evidence anchors for material items where appropriate.
4. Explorers and dashboards can derive reproducible allocation views over any selected timeframe from the same ledger, including inflow coverage, outflow coverage, and evidence coverage, so donors can interpret transparency honestly.

5. If the nonprofit uses multiple chains or other on-chain venues, it can publish optional completeness commitments and third-party attestations that reconcile those external positions to the Hub time-window view for the selected timeframe.

This pattern is complete inside the digital economy. dCorps does not support fiat rails or bank integrations; any legacy fiat or charity processes are outside the system.

7.4 Initial entity templates and first deployment patterns

To make the design concrete, dCorps starts with a small set of canonical entity templates. These are opinionated defaults that cover common use cases and form the wedge for early adoption.

In v1, the intended default is **Hub-first**: both simple and advanced structures can live on the public Hub through standard modules and governance templates. Additional execution layers (for example private execution zones, off-chain systems with anchored commitments, or other networks) are optional future extensions used only for extreme scale or specialized privacy, not a requirement for complex share structures.

Templates are grouped under Hub corporation and Hub nonprofit. Each template has a public-facing name and a code identifier used by tooling and specs.

Hub corporation templates

- **Solo operator (CORP-SOLO)** *Deployment:* dCorps Hub (shared) *Complexity:* Low A single-signer corporation with a 1-of-1 treasury, a 10,000 base unit cap table, and minimal role structure for small owner-operators.

- **Private standard (CORP-PRIVATE-STD)** *Deployment:* dCorps Hub (shared) *Complexity:* Low to medium A small or LLC-style private corporation with the default 10,000 base unit model, role-based governance, and stablecoin wallets for revenue and expenses (USDC at launch).

- **Venture-grade (CORP-VENTURE)** *Deployment:* dCorps Hub (shared) *Complexity:* Medium A private corporation with board approvals, unit pools, vesting schedules, and stricter transfer rules, while remaining Hub-first.

- **Complex private (CORP-COMPLEX-PRIVATE)** *Deployment:* dCorps Hub (shared) *Complexity:* High A private corporation with venture-grade capital controls (rounds, investor consents, pools/vesting), multi-class units, committees, advanced treasury policy, and multi-entity holdings or group structures.

Hub nonprofit templates

- **Nonprofit simple (NONPROFIT-SIMPLE)** *Deployment:* dCorps Hub (shared) *Complexity:* Low A board-governed nonprofit with donation and program wallets, clear allocation categories, and the default transparency floor.

- **Nonprofit board (NONPROFIT-BOARD)** *Deployment:* dCorps Hub (shared) *Complexity:* Medium A nonprofit with board and committee structures, multi-program operations, and tighter allocation policies for donors and partners.

- **Nonprofit complex (NONPROFIT-COMPLEX)** *Deployment:* dCorps Hub (shared) *Complexity:* Medium to high A nonprofit with designated funds, umbrella program structures, and selective disclosure patterns while preserving category level transparency (often used for foundations and fiscal sponsorship/umbrella programs).

7.4A Tag schema by template

All templates use the same required tags:

- `category_code`
- `counterparty_type`
- `reference_id` (when applicable)
- `reference_type` (when `reference_id` is present)

Template-specific context tags:

- **CORP-SOLO**
  - Operating/org: `program_tag` or `business_unit_tag`, `department_tag`, `cost_center_tag`, `project_tag`, `product_tag`, `item_id`, `channel_tag`, `region_tag`, `counterparty_tag`.
  - Equity context: `equity_class_tag`, `vesting_schedule_tag`, `option_pool_tag`.
- **CORP-PRIVATE-STD**
  - Operating/org: `business_unit_tag`, `department_tag`, `cost_center_tag`, `project_tag`, `product_tag`, `item_id`, `channel_tag`, `region_tag`, `counterparty_tag`.
  - Equity context: `equity_class_tag`, `vesting_schedule_tag`, `option_pool_tag`.
- **CORP-VENTURE**
  - Operating/org: `business_unit_tag`, `department_tag`, `cost_center_tag`, `project_tag`, `product_tag`, `item_id`, `channel_tag`, `region_tag`, `counterparty_tag`.

    - Capital/financing: `round_tag`, `security_type_tag`, `equity_class_tag`, `vesting_schedule_tag`, `option_pool_tag`, `debt_instrument_tag`, `loan_id`.
- **CORP-COMPLEX-PRIVATE**
    - Operating/org: `business_unit_tag`, `department_tag`, `cost_center_tag`, `project_tag`, `product_tag`, `item_id`, `channel_tag`, `region_tag`, `counterparty_tag`.
    - Capital/financing: `round_tag`, `security_type_tag`, `equity_class_tag`, `vesting_schedule_tag`, `option_pool_tag`, `debt_instrument_tag`, `loan_id`.
    - Treasury/asset: `wallet_tag`, `treasury_bucket_tag`, `asset_tag`, `custody_tag`.
- **NONPROFIT-SIMPLE**
    - Program/fund: `program_tag`, `fund_tag`, `restriction_tag`.
    - Donor/grant: `grant_id`, `donor_tag`, `campaign_tag`, `item_id`, `region_tag`, `counterparty_tag`.
- **NONPROFIT-BOARD**
    - Program/fund: `program_tag`, `fund_tag`, `restriction_tag`.
    - Donor/reporting: `grant_id`, `donor_tag`, `campaign_tag`, `item_id`, `beneficiary_tag`, `impact_area_tag`, `region_tag`, `project_tag`, `counterparty_tag`.
- **NONPROFIT-COMPLEX**
    - Program/fund: `fund_tag`, `restriction_tag`, `program_tag`, `project_tag`.
    - Donor/impact: `grant_id`, `donor_tag`, `campaign_tag`, `item_id`, `beneficiary_tag`, `impact_area_tag`, `region_tag`, `counterparty_tag`.
    - Treasury/asset: `wallet_tag`, `treasury_bucket_tag`, `asset_tag`, `custody_tag`.

See section 9.8A for the full tag taxonomy and definitions.

Optional future extension: anchored environments

If an organization needs extreme throughput, specialized privacy, or bespoke execution logic, it may use an additional execution environment that anchors standardized summaries back to the Hub. This is not a v1 dependency and does not change kernel semantics.

These templates are not a fixed catalog. Governance can introduce new templates, refine existing ones, or deprecate patterns that do not match real usage. The Hub and module architecture is designed so that new templates can be added as modules or standard entity types without rewriting core consensus logic. Examples could include sector-specific corporation templates (for example a DeFi market operator or a CBDC settlement desk) as the ecosystem matures.

7.5 Migration and adoption paths

dCorps is designed for organizations born on-chain and operating inside the digital economy. Migration of existing legal entities is possible, but it is treated as an optional edge path. The core protocol does not depend on legacy registries, and many entities will never attach a jurisdiction adapter.

The key idea is that adoption should not require a single irreversible jump. Entities can adopt the Hub kernel first and attach external overlays only when needed.

7.5.1 Registering an existing entity on the Hub (parallel ledger)

An existing legal corporation or nonprofit can register a Hub entity and treat it as its canonical on-chain operating ledger:

1. Register a Hub corporation or Hub nonprofit that corresponds to the existing entity.
2. Anchor governing documents and key evidence (bylaws, board resolutions, signing policies) by hash.
3. Map real-world roles to on-chain roles and wallets so authority is explicit and verifiable.
4. Route stablecoin flows through canonical wallets and tagged accounting events to gain reproducible reporting and auditability.

This does not magically transfer legal personhood to the chain. It creates a verifiable, programmable operating layer that can be referenced by counterparties and institutions that choose to rely on it.

7.5.2 Attaching a jurisdiction adapter for external recognition (optional)

When external recognition is needed, the entity can attach a jurisdiction adapter as an overlay:

1. The entity passes a governance resolution to attach the adapter and specify any required declarations or evidence links.
2. The adapter workflow (off-chain where necessary) processes the recognition steps for that specific context.
3. The adapter publishes a recognition status or proof pointers back to the Hub as derived state.

This approach keeps the kernel neutral: the Hub records canonical truth, and the adapter expresses context-specific recognition without rewriting history or changing semantics.

7.5.3 Future extension: using an anchored execution environment (if ever needed)

Some organizations may eventually require specialized execution for extreme scale or specialized privacy. If introduced, anchored execution environments would be a future extension and would not be required for ordinary Hub entity operation.

If used, the design constraints are:

- The Hub remains the canonical entity registry and authority log.
- The environment must anchor standardized summaries and proofs to the Hub.
- Tooling can safely ignore the environment and still have a complete Hub entity view.

7.5.4 Future extension: specialized privacy or volume modes for nonprofits

Large nonprofits may have privacy or volume constraints (beneficiary privacy, large donor lists, high-frequency microdonations). Selective disclosure patterns and privacy-preserving reporting may be introduced as future extensions, while keeping the Hub as the canonical governance and accounting reference layer.

---

8. Identity, roles, and data architecture

8.1 DIDs and identities

Identity in dCorps is based on **wallet addresses** and optional **decentralized identifiers (DIDs)**:

- Wallet addresses are the minimal, universal identity primitive used by the Hub.
- DIDs are an optional, higher level identity layer used for roles, credentials, and cross system linkage.

Each DID can represent an actor such as:

- Founder or shareholder.
- Board member or director.
- Officer, for example CEO, CFO.
- Auditor or reviewer.
- Regulator or oversight body.

A DID can be linked to:

- One or more wallets.
- Verifiable credentials and attestations issued by external identity and KYB providers.

dCorps does not invent a proprietary identity standard:

- The protocol aligns with the **W3C DID model**.
- dCorps treats DIDs as an alias and credential compatibility layer on top of the wallet first base.

v1 identity stance

- Wallet only mode is fully supported. An entity can operate using addresses without any DID at all.
- DIDs are optional and additive. Roles may be bound to wallets directly, or to DIDs that are linked to wallets.

v1 default DID method

- The default DID method for v1 is **`did:pkh`**, which maps to blockchain accounts and fits a multi-chain EVM environment.
- Additional DID methods may be added over time through standards and modules.

Optional institutional DID method

- dCorps may support **`did:web`** as an additive method for institutions that want identity anchored to a domain they control (for example auditors, foundations, or providers).

Credential and issuer integration stance

- Credentials enter the system as issuer signed attestations that reference either a wallet, a DID, or both.
- Additional identity and credential ecosystems can be integrated later as optional modules, without changing the v1 requirement that wallet only operation remains valid.

Identity is designed to be pluggable and upgradable:

- Entities can migrate role bindings from one DID method to another without losing their entity history, because the canonical record is the entity's state and event timeline, not any single DID method.
- Protocol modules and applications can apply their own identity policies, for example requiring specific credentials for attaching to a jurisdiction adapter module or for holding specific roles.

8.2 Role and permission model

Roles are explicit constructs in the protocol:

- Examples:
    - Board seat.
    - Director or officer role.
    - Treasurer or CFO.
    - Committee member.
    - Auditor access role.
    - Protocol Council member.

Roles have:

- Bound DIDs and wallets.
- Specific permissions, such as:
    - Proposing or approving transactions.
    - Initiating governance proposals.
    - Accessing controlled data.
    - Attaching or detaching modules.

Roles are separate from individuals:

- When a person leaves, governance actions reassign the role to a new DID or wallet, preserving history.

This structure:

- Avoids single wallet choke points.
- Aligns with how boards and management actually operate in law.

8.3 Wallet structure for entities

Entities use a structured wallet model that separates **authority wallets** (role control) from **operational wallets** (USDC flows):

- **Merchant wallet (corporations)** Canonical income wallet for operating revenue. Default destination for invoices and customer payments.

- **Donation wallet (nonprofits)** Canonical income wallet for donations and grants.
- **Treasury wallet** Holds reserves and long term holdings. Not used for frequent operational payments.
- **Program wallets** Represent specific programs or projects, especially in NGOs. Have budgets, allocation rules, and separate dashboards.
- **Role (authority) wallets** Bound to roles such as director or CFO. Used to sign governance actions and approvals, not to receive customer payments.

This structure makes it possible to:

- Analyze flows by function.
- Apply policies per wallet type.
- Provide clear, consistent views for dashboards and auditors.

Protocol modules can refer to wallet types, not just raw addresses, when applying rules.

8.3A Commerce primitives (items, invoices, recurring plans)

The Hub includes minimal commerce objects so businesses can run end-to-end on-chain:

- **Catalog item / service** On-chain item ID with a label, price, and optional cost baseline. Used for sales and invoices.
- **Invoice (payment request)** On-chain request that resolves to a canonical payment wallet and amount. Includes line items, counterparty reference (wallet or pseudonymous ID), due date, and status.
- **Recurring plan** On-chain schedule that creates invoices on a fixed cadence.

Invoice status states: `draft`, `open`, `partial`, `paid`, `overdue`, `waived`, `canceled`.
Item status states: `active`, `paused`, `retired`.
Plan status states: `active`, `paused`, `canceled`.

Applications generate payment links from invoice IDs, and merchant integrations use SDK/API calls to create invoices and read status.

Nonprofit donation flows are open by default: donors can send directly to the donation wallet without an invoice. Payment requests or recurring plans are optional for grants, sponsorships, memberships, or pledged giving. When confirmations are needed, entities anchor donation receipts and reference them in the accounting event (`reference_type=donation_receipt`).

8.4 Data categories: public, role gated actions, private, and off-chain

dCorps distinguishes four broad data categories.

Public on-chain data

- Entity IDs, names, types, and tags.
- Cap tables and ownership breakdowns for Hub corporations.
- Board compositions and governance events for nonprofits.
- High-level donation and spending categories for nonprofits.
- Anchors of off-chain documents.
- Protocol module outputs and signals where applicable, including jurisdiction recognition status, sector framework outputs, and optional attestation or reputation module outputs.
- Counterparty references may be stored as wallet addresses or pseudonymous IDs; private mappings stay off-chain.

Role gated actions and permissions (on-chain)

- Certain actions are restricted by role, policy, or module checks, for example:
    - Issuing or transferring Hub units under restrictions.
    - Changing allocation rules for nonprofits.
    - Executing treasury movements that require multi role approvals.
    - Attaching or detaching protocol modules.
- These restrictions control who can change state. They do not, by themselves, make state unreadable.

Private data commitments and encrypted payloads (on-chain, optional)

- When an entity needs to anchor sensitive facts without revealing raw details, it can store:
    - Hash commitments to documents, ledgers, or datasets.
    - Encrypted blobs intended for specific recipients.
    - Proof artifacts or verification outputs from privacy-preserving systems.
- The Hub records what was anchored and when. Decryption and access are handled by the entity and its counterparties, not by consensus.

Off-chain data

- Stored in external systems, with integrity anchored on-chain:
    - Contracts and detailed legal documents.
    - HR and payroll records with personal data.
    - Beneficiary lists and sensitive field level program records.
    - Full audit reports and supporting workpapers.

- Full general ledger exports and reconciliations where an entity chooses to publish them as anchors.

This separation allows:

- Public verifiability of key facts and category level flows.
- Stronger control over sensitive personal and commercial data.
- Evolution of privacy and storage technology without breaking core logic.

8.5 Privacy design and selective disclosure

Privacy mechanisms include:

- **Private contract zones**
    - Specialized environments where detailed logic and data can live, while only publishing commitments or aggregates to the Hub.
- **Zero knowledge proofs**
    - Allow entities to prove certain properties, such as:
        - Meeting allocation rules.
        - Maintaining solvency thresholds.
    - Without revealing raw transaction details.
- **Selective disclosure**
    - Permissioned dashboards and data rooms for auditors, regulators, or major donors.
    - Signed attestations by auditors or oversight bodies that confirm alignment between on-chain state and off-chain records.

The protocol records that proofs or attestations exist and were verified. It does not decide which level of disclosure is sufficient for any particular law or contract.

Protocol modules and applications can integrate these tools and require proofs or attestations as part of their logic.

8.5A Privacy tiers and disclosure modes

dCorps is transparent by design, but organizations have legitimate privacy needs. Privacy in an on-chain system has two distinct meanings:

- **Visibility policy**, meaning what official interfaces and standards require an entity to publish and how data is presented.
- **Confidentiality**, meaning whether raw facts are technically hidden from the public.

Visibility policy can be expressed through protocol rules and standards. Confidentiality generally requires privacy technology (encryption, private execution zones, and zero knowledge proofs) and often implies different execution environments.

To make choices explicit, each entity declares a disclosure mode at creation. The disclosure mode is public metadata and is used by explorers, registries, sector frameworks, and jurisdiction adapters.

8.5A.1 Disclosure modes (v1)

Mode A, Public operations (maximum verifiability)

- Hub corporations: cap table balances (pseudonymous addresses and optional DIDs), governance events, and tagged operational flows are public on the Hub.
- Hub nonprofits: board composition and governance events are public; donation inflows and category level outflows are public; program wallet structure is public.

Mode B, Public structure with aggregate reporting (privacy aware operations)

- Structural state remains public (entity identity, wallet types, governance events).
- Operational detail may be kept off-chain or in controlled zones, while the Hub records:
    - Commitments (hash anchors) to detailed ledgers or datasets,
    - Encrypted payloads intended for specific recipients (auditors, regulators, major donors), and
    - Time-window aggregates and standardized category totals for public views (where published).

Mode B establishes a standard for publishing aggregates and proofs instead of raw line items where privacy tools are used.

Mode C, Private execution with public anchoring (private zone required)

- Sensitive operations execute in a private execution zone (for example a private contract zone or controlled execution environment).

- The Hub records:
    - Governance checkpoints,
    - Time-window aggregates and standardized category totals (where published), and
    - Commitments to detailed records and, where applicable, proof verification artifacts.

Mode C is intended for entities that require stronger confidentiality for beneficiaries, payroll, commercial terms, or regulated data.

8.5A.2 Minimum transparency guarantees by entity type

Hub nonprofits must meet a minimum transparency floor in all modes. This floor is a protocol visibility policy, independent from whether raw line items are public.

Minimum requirement	**Hub nonprofit transparency floor (all disclosure modes)**
Timeframe	Views are available live; reference tooling can compute totals over any selected timeframe
Inflows	Total donation inflows and total grant inflows over selected timeframe, plus total other inflows if used
Outflows	Total spending over selected timeframe for, at minimum: program spending, general and administrative overhead, and fundraising costs; additional categories from the minimal chart of accounts may be exposed as the entity chooses
Governance	Board composition, proposals, votes, and allocation rule changes
What may remain private	Beneficiary identities, payroll line items, vendor invoices, field level program data, and any personally identifiable information

In Mode A, the floor can be computed directly from on-chain line items. In Mode B and Mode C, the floor can be met by publishing time-window aggregates and category totals, plus commitments and optional third-party attestations as described in section 8.5A.3.

Hub corporations have no universal mandatory public expense disclosure by category. Corporations choose Mode A, B, or C, and counterparties can require a specific mode by contract or by module participation.

8.5A.3 Selective disclosure standards

Where selective disclosure is used:

- Every private dataset referenced for assurance is anchored by hash and timeframe.
- Access is granted through encrypted payloads or separate data rooms controlled by the entity.
- Auditors and other issuers can publish signed attestations that a private dataset matches anchored commitments and reconciles to on-chain aggregates.

8.5A.4 Limits of protocol privacy

Choosing a disclosure mode does not automatically provide confidentiality. Confidentiality exists only when an entity uses privacy-preserving execution or disclosure tools. Official interfaces surface this distinction clearly by labeling which views are raw on-chain data, which are aggregates, and which are claims supported by attestations.

8.5A.5 Transparency tiers for discovery and comparability (reference interface standard)

To prevent “trust gaps” between raw disclosures and aggregate disclosures, reference explorers and dashboards surface a mechanical transparency tier alongside the disclosure mode. The tier is a presentation standard derived from observable facts and published signals, not an editorial judgment.

Illustrative tiers include:

- **Tier 1, Raw on-chain detail** Time-window views are derived directly from on-chain line items for the relevant wallet flows.
- **Tier 2, Public aggregates with evidence** Time-window views are derived from published aggregates plus supporting commitments, proofs, or third-party attestations.
- **Tier 3, Minimum transparency floor** The entity meets the minimum nonprofit floor (where applicable) but does not publish additional detail or assurance beyond that floor.

Reference interfaces must clearly distinguish:

- Deterministic outputs from typed workflows
- Self-reported aggregates
- Aggregates supported by third-party attestations or proof verification artifacts

This makes transparency a measurable advantage for entities that choose stronger disclosure, while allowing privacy aware organizations to participate without being presented as equally verifiable.

8.6 Example entity state representation (illustrative)

The following examples are intentionally simplified and non normative. They are meant to show the kinds of structured state and events the Hub exposes. IDs, addresses, and dates are placeholders.

```
{
  "schema_version": "1.0",
  "entity_id": "dcorp:hub:entity:00001234",
  "entity_type": "hub_corporation",
  "name": "Acme Labs",
  "status": "active",
  "tags": ["software", "remote_first"],
  "created_at": "2025-06-15T12:00:00Z",
  "attachments": [
    {
      "module_id": "dcorp:module:jurisdiction:example_v1",
      "status": "attached",
      "attached_at": "2025-07-01T00:00:00Z"
    }
  ]
}
```

```
{
  "schema_version": "1.0",
  "entity_id": "dcorp:hub:entity:00001234",
  "wallets": [
    { "wallet_type": "merchant", "address": "0xabc123..." },
    { "wallet_type": "treasury", "address": "0xdef456..." },
    { "wallet_type": "fee_reserve", "address": "0x987654..." }
  ]
}
{
  "schema_version": "1.0",
  "event_type": "accounting_event",
  "entity_id": "dcorp:hub:entity:00001234",
  "tx_hash": "0xabc123...",
  "timestamp": "2025-08-01T18:30:00Z",
  "from_wallet_type": "merchant",
  "to_address": "0xfeedbeef...",
  "amount": { "asset": "USDC", "decimals": 6, "value": "30000000000" },
  "tags": {
    "category_code": "EXP_PAYROLL",
    "counterparty_type": "employee",
```

```
    "reference_id": "payroll_batch:2025-08",
    "reference_type": "payroll_batch"
  }
}
{
  "schema_version": "1.0",
  "event_type": "governance_resolution",
  "entity_id": "dcorp:hub:entity:00005678",
  "resolution_id": "res:00000042",
  "timestamp": "2025-09-20T10:00:00Z",
  "resolution_type": "nonprofit_allocation_rule_update",
  "result": "passed",
  "links": {
    "minutes_anchor": "doc:ipfs:Qm...",
    "policy_anchor": "doc:ipfs:Qm..."
  }
}
```


Note: in practice, stablecoins on the Hub are represented by canonical ERC-20 contracts that are backed by bridged assets from Ethereum. Reference interfaces should present them with mainstream symbols (USDC, USDT, DAI) and disclose bridge/issuer risk separately.

8.7 Operational security (non-custodial)

Most real world failures come from compromised signers and process breakdowns, not from consensus bugs. dCorps is designed to support organization-grade operational security through explicit roles, separation of duties, policy-aware wallet structures, and auditable governance actions.

Custody and wallet security procedures remain the responsibility of users and entities and are not specified by the public documentation.

---

9. On-chain operations

9.1 Assets and operating currency

dCorps separates:

- **DCHUB** as the gas token for the Hub rollup and the governance token for protocol-level decisions (and, where used, protocol-level fees or deposits for actions like entity registration/renewal and module registry operations).
- **Stablecoins** (USDC is required; other approved stablecoins may include USDT, DAI, and others) as operating currencies for:
    - Invoices and revenue.
    - Salaries and contractor payments.
    - Vendor payments.
    - Grants and donations.
    - Jurisdiction fees and certain protocol fees.

In the initial implementation, stablecoins are represented on dCorps as canonical bridged ERC-20 contracts, originating from Ethereum via the Hub's bridge gateways. Reference interfaces should present them with mainstream symbols (USDC, USDT, DAI) without chain-suffix labels, while clearly disclosing bridge/issuer risk.

Treasury and reserve wallets may also hold DCHUB for gas buffers or protocol exposure; DCHUB holdings should be labeled with `asset_tag` and the `BAL_DCHUB` category.

v1 mainnet expectation management

- Mainnet may launch with a small approved operating set (at minimum USDC), so that reliability, monitoring, and integrations are tight.

- Additional stablecoins (including decentralized stablecoins) can be added later through the asset registry process after risk review and governance approval.
- Entities can choose one primary operating stablecoin (or a small approved set) at creation; canonical wallets and reporting must record the denom and token contract for every flow.
- Entities can still accept and hold other digital assets, but v1 reference reporting uses USDC as the baseline unit of account for comparability.

Mainstream payment entry (Ethereum-first)

- Customers can pay invoices using USDC they already hold on Ethereum, paying their own Ethereum gas.
- Entities absorb bridging and operational costs as a merchant-fee model and credit the resulting funds to their canonical dCorps wallets.
- Once bridged, internal treasury/payroll/intra-wallet movements happen entirely on dCorps using the canonical on-chain stablecoin contracts.

dCorps is optimized for entities that treat dCorps wallets as their **primary digital operating accounts**. Incoming revenues and donations are received in stablecoins, primarily USDC, into merchant or donation wallets. Salaries, contractor payments, vendor payments, grants, and program costs are paid out from dCorps wallets. The on-chain view is the transparent, verifiable part of an entity's activity.

The protocol does not support fiat rails or bank payments at any layer. dCorps is complete inside the on-chain economy. Any fiat activity is outside the system and is not integrated.

Entities may hold and use other digital assets, but reports and metrics use USDC as the baseline unit of account.

9.1A Gas payment, fee abstraction, and sponsored transactions

dCorps uses DCHUB for gas at the Hub protocol level, but the ecosystem is designed so that entities can operate in a stablecoin first manner without requiring every end user to directly hold DCHUB at all times.

Design intentions include:

- **Paymasters and gas sponsorship**
  - Entities, applications, and service providers can sponsor gas for specific wallets or roles, covering routine activity such as governance actions and reporting anchors.

- **Relayers and meta-transactions**
    - Relayers can submit transactions on behalf of entities using explicit, revocable authorizations (for example signed intents), enabling stablecoin denominated service billing off-chain while preserving on-chain transparency of the underlying actions.
- **Clear UI requirements**
    - Official interfaces and registry listed applications are expected to disclose when a transaction is sponsored, which party paid gas, and what policy controlled the sponsorship.

Non custodial requirements (hard boundary)

- The protocol is non custodial. Entities keep control of their wallets and keys.
- Relayers and sponsors must not take custody of entity or user funds as part of gas abstraction. They pay gas from their own accounts. They may charge off-chain fees for service, but they do not control client assets inside dCorps as part of this flow.
- Any delegation used to enable relayed execution must be message scoped, time bounded, and revocable, and must not grant discretionary control over an entity treasury.

These mechanisms improve usability and reduce operational friction for small teams and nonprofits, while preserving DCHUB's role in network security and governance.

9.1B Stablecoin issuer risk, freezes, and treasury continuity (recommended practices)

dCorps is optimized for stablecoin native operation, but stablecoins introduce issuer and rail risk.

Many fiat backed stablecoins include administrative controls such as blacklisting and freezes. When a freeze happens, reversal typically depends on the stablecoin issuer's policy and the administrative controls of the token's home chain. dCorps protocol governance and reference interfaces cannot unfreeze a stablecoin balance and cannot override issuer enforcement.

dCorps addresses this risk by design and by recommended operating practices:

- **Asset registry risk labeling (protocol level)**
    - The approved asset registry records issuer and rail risk factors, including freeze and redemption mechanics, and official interfaces surface these risk labels prominently.
- **Multi stablecoin support (protocol and ecosystem)**

    - Entities may diversify operating balances across multiple approved stablecoins over time, reducing dependency on any single issuer.
- **Treasury segmentation (recommended)**
    - Keep operational liquidity in the stablecoin most counterparties use.
    - Keep reserves segmented and diversified, for example across multiple approved stablecoins and venues, with clear policy controls and internal limits.
- **Decentralized stablecoins as partial reserves (optional)**
    - Overcollateralized stablecoins (for example DAI style designs) can reduce issuer freeze exposure, but they introduce different risks, such as collateral volatility, peg stress, governance risk, and liquidity constraints.
    - They should be treated as diversification instruments, not as a universal replacement for fiat backed stablecoins.
- **Avoid “freeze bypass” wrappers as a core strategy**
    - Wrapping a stablecoin does not remove the underlying issuer and redemption risk, and it adds smart contract and liquidity risk.
    - dCorps does not position itself as a system for evading issuer controls or legal enforcement; entities remain responsible for compliance with applicable laws and sanctions regimes.
- **Operating continuity planning (recommended)**
    - Entities should maintain a documented continuity plan for stablecoin disruption, including:
        - approved alternative settlement assets,
        - minimum reserve buffers,
        - wallet and policy controls that limit blast radius, and
        - clear internal escalation and incident response procedures.

9.2 Income and invoicing

Entities can:

- Issue invoices or payment requests denominated in USDC.
- Receive payments directly into their merchant or donation wallets.
- Tag income with:
    - Type (product, service, subscription, donation, grant, other).
    - Counterparty type (customer, donor, grant maker, affiliate, jurisdiction, other).
    - References to contracts or agreements anchored on-chain.

This enables:

- Automated aging and receivables tracking where integrations exist.
- Simple views of revenue by type and counterparty over time.

Entities that commit to routing all material income and payouts through dCorps wallets obtain the strongest possible transparency guarantees, because their public reports can be derived directly from on-chain flows.

9.3 Payroll, compensation, and vesting

Corporations can:

- Run payroll in USDC:
    - Scheduled payments from merchant or treasury wallets.
    - Batch operations for many employees at once.
- Manage contractor payments with clear categorization.
- Implement **vesting schedules** for:
    - Hub units.
    - dShares, once enabled as a future extension.
    - Other incentive instruments.

Vesting logic can include:

- Cliff periods.
- Linear or stepwise vesting.
- Accelerated vesting conditions linked to governance events or external triggers.

nonprofits can:

- Pay staff and contractors through tagged transactions.
- Keep board compensation within contract enforced caps that are visible to donors.

9.4 Expense management and approvals

Expense management follows a standard pattern:

1. **Proposal**
    - An authorized role creates an expense proposal with:
        - Amount and currency.
        - Beneficiary.
        - Category and subcategory.
        - Links to anchored invoices or contracts.
2. **Approval workflow**
    - Depending on policy, approvals may require:
        - Single signer for small amounts.
        - Multiple signers or board votes for larger or sensitive items.
        - Additional checks from protocol modules, for example jurisdiction or sector rules.
3. **Execution**
    - Once approved, a payment is executed and tagged.
    - The link between proposal, approval, and execution is recorded.
4. **Reporting**
    - Dashboards can show budget versus actuals at many levels.
    - Auditors can follow chains of proposals and approvals.

Policies can be implemented in ecosystem applications that use these primitives.

9.5 Views and dashboards

Because flows are on-chain and structured, explorers and dashboards can derive:

- Timeframe-selected summaries and views, including:
    - Cash-based operating summaries for corporations (views over tagged events).
    - Allocation summaries for nonprofits (views over tagged events plus allocation rules).
    - Cash-flow and category trend views for any selected timeframe.

- Stakeholders can see:
    - Revenue and expense trends.
    - Allocation ratios, for example program versus overhead.
    - Governance activity.

Public entities are expected to meet baseline disclosure standards. Private entities have more flexibility but still benefit from internal reporting built directly from state.

Protocol modules and external analytics tools can consume this data and provide scores, alerts, and comparisons.

9.5A Tag integrity, assurance, and audit signals

Tagged accounting events improve machine readability, but tags are not automatically truthful. Amounts, timestamps, and wallet movements are verifiable on-chain for on-chain funds. Category tags and descriptions are interpretations created by entities or by the workflows they use.

dCorps strengthens tag reliability by encouraging constrained, typed workflows instead of free form tagging:

- **Typed modules emit deterministic tags (recommended)** Core operating workflows (payroll, invoicing, grants, donation allocation, vendor pay flows) are expected to be implemented as typed modules or standardized message families that emit category codes as part of execution. In this model, the category is not merely an assertion, it is the output of a constrained process.
- **Free form tags remain possible, but are clearly labeled** Where entities use custom flows, events may include custom tags. Reference interfaces must label these as entity supplied tags and distinguish them from tags produced by typed workflows.

dCorps separates:

- **Verifiable movements** (what happened on-chain), and
- **Interpretations and completeness** (what it means, and whether the view is complete).

9.5A.1 Evidence anchored tagging (recommended)

For material flows above a parameterized threshold, entities are expected to attach at least one of the following:

- An anchored invoice, contract, or resolution hash

- A standardized reference ID that maps to an off-chain document in a controlled data room
- A signed counterparty receipt or acknowledgement where practical

Evidence anchors improve auditability without forcing sensitive content on-chain.

9.5A.2 Counterparty receipts (recommended)

To raise integrity without requiring heavy third-party audits, dCorps supports optional counterparty receipts:

- A vendor, contractor, partner, or recipient signs a receipt referencing the payment or invoice identifier and the relevant category and period.
- Receipts can be stored as on-chain attestations or as anchored documents with a signed payload.

Counterparty receipts make systematic misclassification harder without turning the protocol into an adjudicator.

9.5A.3 Completeness and cross-chain reconciliation signals (optional)

dCorps does not require reconciliation to function. However, some entities operate across multiple on-chain venues (multiple chains, multiple DEXs, multiple custody models). These entities may voluntarily publish:

- Time-window commitments (hashes of standardized exports, position snapshots, or reconciled inventory lists).
- Third-party attestations that specified external balances or DeFi positions reconcile to the Hub time-window view for a selected timeframe.

These signals are optional. They exist primarily to support large donors, risk-aware counterparties, and auditors who want a clear coverage story across venues. Reference dashboards must clearly distinguish self-reported views from views supported by third-party attestations.

9.5A.4 Assurance signals and module outputs (optional)

Assurance signals and scoring are optional. They come from protocol modules and independent issuers, not from the Hub core.

Examples include:

- Auditor attestations that tagged aggregates match underlying records.
- jurisdiction adapter outputs that required reports and fees were satisfied, or that recognition is active or withdrawn.

- Sector framework outputs that allocation rules or reporting commitments were met.
- Reputation module scores derived from defined inputs, published as metrics with reason codes and source anchors.

Reference dashboards should distinguish clearly between:

- Views produced from deterministic, typed workflows
- Self-tagged and self-reported views
- Views supported by independent attestations
- Views supported by rule outputs or privacy proof verification artifacts

---

9.5B Cash-based operating view standard (v1)

This section defines the v1 standard for cash-based operating views derived from dCorps accounting primitives.

Cash-based operating views are time-window summaries derived from cash-like inflows and outflows recorded through entity wallets, excluding accrual accounting treatments. They are designed for operational clarity and comparability. They are not GAAP, IFRS, or statutory financial statements.

Important boundary: these views are derived by explorers/indexers (and optional dApps) from on-chain events. The kernel does not require a reporting cadence and does not store periodic statements as native state.

9.5B.1 Time window definition and currency

- View time windows are defined as half open intervals: `[period_start, period_end)`.
- Window timestamps use UTC for canonical views in reference tooling.
- The baseline reporting currency is USDC.
- When a window includes multiple stablecoins, reference views may display:
  - Native denomination totals per denom, and
  - An optional USD equivalent view using transparent conversion rules and explicit price feed sources, where supported by tooling.

9.5B.2 Required inputs and category mapping

A cash-based view is derived from:

- Accounting events that record cash-like inflows and outflows for the entity's canonical wallet types (merchant, donation, program, treasury, and other configured wallet types).

- The `category_code` tag on each accounting event, which must map to the minimal standard chart of accounts (section 9.8) or to an entity scoped extension that maps to a parent category in the minimal chart.

Events that omit required tags are treated as uncategorized and must be surfaced explicitly in coverage metrics.

9.5B.3 Source labeling and integrity signals

To keep views honest about their inputs, cash-based view exports include a source label for each aggregate:

- **typed_workflow** Category produced deterministically by a constrained workflow module (for example payroll, invoicing, donation allocation).
- **entity_tagged** Category provided directly by the entity or an external application using free form or custom logic.
- **anchored_aggregate** Aggregate derived from an anchored dataset/export commitment rather than raw on-chain line items.

Reference interfaces must distinguish these inputs clearly. This does not adjudicate truth; it makes the provenance of categories visible.

9.5B.4 v1 derived view export (reference schema excerpt)

The following object is a reference excerpt for interoperability between explorers, indexers, and dApps. It is not an on-chain object. Exact field names are defined in reference formats, but implementations are expected to produce an equivalent structure.

```
{

  "schema_version": "1.0",

  "entity_id": "dcorp:hub:entity:00001234",

  "report_type": "cash_based_operating_statement",

  "period": {

    "period_start": "2025-11-10T00:00:00Z",

    "period_end": "2025-12-15T00:00:00Z",

    "timezone": "UTC"
```

```
},
"base_asset": "USDC",
"base_decimals": 6,
"coverage": {
  "total_inflows": "50000000000",
  "total_outflows": "42000000000",
  "uncategorized_inflows": "0",
  "uncategorized_outflows": "500000000",
  "uncategorized_event_count": 2
},
"income": [
  {
    "category_code": "REV_SUBSCRIPTION",
    "amount": "50000000000",
    "source_type": "typed_workflow"
  }
],
"expenses": [
  {
    "category_code": "EXP_PAYROLL",
    "amount": "30000000000",
    "source_type": "typed_workflow"
```

```
},
{
  "category_code": "EXP_CONTRACTOR",
  "amount": "5000000000",
  "source_type": "entity_tagged"
},
{
  "category_code": "EXP_INFRA",
  "amount": "4000000000",
  "source_type": "entity_tagged"
},
{
  "category_code": "EXP_OTHER",
  "amount": "2000000000",
  "source_type": "entity_tagged"
},
{
  "category_code": "EXP_COMPLIANCE",
  "amount": "500000000",
  "source_type": "typed_workflow"
}
```

```
  ],
  "net_operating_result": "8000000000",
  "notes": {
    "mode": "Mode A",
    "derivation": "derived_from_accounting_events",
    "disclosure": "cash_based_view_only"
  },
  "anchors": [
    {
      "anchor_type": "policy_or_minutes",
      "reference": "doc:ipfs:Qm..."
    }
  ]
}
```


Reference explorers are expected to show cash-based operating views only as a view over the ledger and to include clear labels that it is not an audited statement and not accrual accounting.

9.5C Nonprofit allocation view standard (v1)

Nonprofits in dCorps are expected to publish a minimum transparency view that is meaningful for donors and oversight without forcing disclosure of sensitive beneficiary details.

A nonprofit allocation view is a reproducible, cash-based time-window view derived from tagged accounting events and the nonprofit's allocation rules.

9.5C.1 Required inputs and category mapping

An allocation view is derived from:

- Donation and grant inflows to canonical nonprofit wallets (for example the donation wallet and designated program wallets).
- Tagged outflows that map to:
  - program categories (for example `EXP_PROGRAM` with `program_tag`),
  - support categories (for example `EXP_ADMIN`, `EXP_FUNDRAISING`),
  - and any restricted fund categories when restriction logic is used.
- Internal transfers to treasury wallets (for example retained buffers), which are recorded but are not treated as “distributed” spending.

As with the corporate cash-based view, events that omit required tags are treated as uncategorized and must be surfaced explicitly in coverage metrics.

9.5C.2 v1 allocation view export (reference schema excerpt)

The following object is a reference excerpt for interoperability between explorers, indexers, and dApps. It is not an on-chain object. Implementations are expected to produce an equivalent structure.

```
{

 "schema_version": "1.0",

 "entity_id": "dcorp:hub:entity:00005678",

 "report_type": "nonprofit_allocation_statement",

 "period": {

  "period_start": "2025-11-10T00:00:00Z",

  "period_end": "2025-12-15T00:00:00Z",

  "timezone": "UTC"

 },

 "base_asset": "USDC",

 "base_decimals": 6,

 "coverage": {
```

```
    "donations_in": "100000000000",
    "distributed_out": "95000000000",
    "retained": "5000000000",
    "uncategorized_outflows": "0",
    "uncategorized_event_count": 0
  },
  "by_category": [
    { "category_code": "EXP_PROGRAM", "amount": "75000000000" },
    { "category_code": "EXP_ADMIN", "amount": "15000000000" },
    { "category_code": "EXP_FUNDRAISING", "amount": "5000000000" }
  ],
  "ratios": {
    "program_spending_ratio": "0.7895",
    "overhead_ratio": "0.1579",
    "fundraising_ratio": "0.0526"
  },
  "notes": {
    "derivation": "derived_from_accounting_events",
    "disclosure": "category_level_allocation_view"
  }
}
```

Reference explorers are expected to show this as a view over the ledger, to clearly label its disclosure mode, and to surface coverage and uncategorized amounts so donors can interpret transparency honestly.

9.6 Governance actions and records

Key governance actions are:

- Recorded as proposals and votes.
- Linked to:
    - Roles and DIDs that participated.
    - Anchored documents such as minutes or resolutions.
    - Resulting state changes.

Examples:

- Board approval of an annual budget.
- Shareholder vote on a major transaction.
- nonprofit board decision to adjust allocation rules.

This creates a tamper evident record of how decisions were made.

9.7 Integrated donation features for corporations

Corporations can support nonprofits natively:

- **Checkout donations**
    - Customers can add a donation that goes directly from the customer wallet to the NGO donation wallet, with separate tagging.
- **Revenue share commitments**
    - Corporations can commit a percentage of revenue or profits to specific NGOs.
    - Smart contracts route funds periodically from merchant wallet to NGO donation wallets.
- **Matching programs**

- Corporations can match employee or customer donations up to a defined cap, with matching logic enforced by contracts.

These patterns:

- Do not turn corporations into banks or custodians of donated funds.
- Allow donors and NGOs to verify commitments through on-chain data.

Protocol modules and applications can make these flows visible and comparable.

9.8 Minimal standard chart of accounts

To make the data standard claim concrete, dCorps defines a **minimal default chart of accounts**. Entities can extend it, but common categories ensure that tools and dashboards can compare entities consistently.

Income categories (default codes)

- `REV_PRODUCT` – Product revenue.
- `REV_SERVICE` – Service revenue.
- `REV_SUBSCRIPTION` – Subscription revenue.
- `REV_GRANT` – Grants.
- `DONATION_GENERAL` – Unrestricted donations (for NGOs and foundations).
- `DONATION_RESTRICTED` – Restricted donations.
- `REV_INTEREST` – Investment and interest income.
- `REV_OTHER` – Other income.
- `REV_NONOPERATING` – Non-operating revenue (fallback bucket).

Expense categories (default codes)

- `EXP_PAYROLL` – Salaries, wages, and payroll taxes.
- `EXP_CONTRACTOR` – Contractors and freelancers.
- `EXP_VENDOR` – Vendors and suppliers.
- `EXP_RENT` – Rent and facilities.
- `EXP_INFRA` – Cloud and infrastructure.
- `EXP_MARKETING` – Marketing and sales.
- `EXP_RND` – Research and development.
- `EXP_COMPLIANCE` – Jurisdiction and compliance fees.
- `EXP_TAX` – Taxes.
- `EXP_GRANTS` – Grants to NGOs (for corporations and foundations).
- `EXP_PROGRAM` – Program spending (for NGOs).

- `EXP_FUNDRAISING` – Fundraising costs (for NGOs).
- `EXP_ADMIN` – General and administrative overhead.
- `EXP_TRAVEL` – Travel and events.
- `EXP_OTHER` – Other expenses.

Capital, financing, and treasury (default codes)

- `CAPEX` – Capital expenditures.
- `FIN_EQUITY_RAISE` – Equity financing inflows.
- `FIN_DEBT_DRAW` – Debt financing inflows.
- `FIN_DEBT_REPAY` – Debt repayments.
- `FIN_DIVIDEND` – Distributions/dividends.
- `FIN_OWNER_DRAW` – Owner draws.
- `TREASURY_MOVEMENT` – Internal treasury rebalancing.

Balance sheet categories (optional tags)

- `BAL_CASH` – Cash and stablecoins (USDC and others).
- `BAL_DCHUB` – DCHUB holdings.
- `BAL_TOKEN` – dShares and other tokens.
- `BAL_AR` – Accounts receivable.
- `BAL_PREPAID` – Prepaid expenses.
- `BAL_AP` – Accounts payable.
- `BAL_DEFERRED_REV` – Deferred revenue.
- `BAL_DEBT` – Loans and other liabilities.

This baseline is not a full accounting standard and does not replace local GAAP or IFRS. It is a neutral minimum schema that allows:

- Standard dashboards across many entities.
- Easier mapping into local accounting systems.
- Sector and jurisdiction adapters to express additional rules without losing comparability.

Balance sheet categories in this schema are an optional tagging and dashboard view standard; they are not audited financial statements and do not, by themselves, imply GAAP, IFRS, or statutory reporting compliance.

The foundation is expected to maintain this minimal schema, propose extensions where needed, and work with the ecosystem when mapping to local accounting standards.

9.8A Tag taxonomy (required + optional tags)

Every tagged accounting event MUST include:

- `category_code` – required category code from the minimal chart (section 9.8).
- `counterparty_type` – who the counterparty is (`customer`, `vendor`, `contractor`, `employee`, `donor`, `grantee`, `beneficiary`, `investor`, `lender`, `government`, `bank`, `custodian`, `payment_processor`, `affiliate`, `exchange`, `nonprofit`, `foundation`, `other`).
- `reference_id` – external reference when applicable (invoice, contract, grant, payroll batch, policy, resolution).
- `reference_type` – recommended when `reference_id` is present; standard values include `invoice`, `payment_request`, `receipt`, `contract`, `purchase_order`, `payroll_batch`, `expense_report`, `grant_agreement`, `donation_receipt`, `subscription_plan`, `bank_statement`, `board_resolution`, `cap_table_update`, `policy`, `tax_filing`, `audit_report`, `other`.

Standard optional tags (v0.1) add context and should be used where applicable:

Operating and org context:

- `program_tag` – nonprofit program or operating unit.
- `business_unit_tag` – corporate business unit (optional alternative to `program_tag`).
- `department_tag` – department or team owner.
- `cost_center_tag` – internal cost center or budget code.
- `project_tag` – project or initiative.
- `product_tag` – product or service line.
- `item_id` – on-chain catalog item/service ID (optional).
- `channel_tag` – revenue or distribution channel.
- `region_tag` – ISO country or region code.
- `counterparty_tag` – pseudonymous counterparty identifier (hashed or coded when privacy is required).

Nonprofit allocation context:

- `fund_tag` – designated or restricted fund (nonprofits).
- `restriction_tag` – restriction code or policy label.
- `grant_id` – grant or award identifier.
- `donor_tag` – donor identifier (hashed or coded).
- `campaign_tag` – fundraising campaign identifier.
- `beneficiary_tag` – beneficiary group or cohort identifier.
- `impact_area_tag` – impact theme or program domain.

Capital and financing context:

- `round_tag` – financing round label (Seed, Series A).

- `security_type_tag` – equity/SAFE/note type.
- `equity_class_tag` – equity class label (e.g. Class A, Class B).
- `vesting_schedule_tag` – vesting schedule identifier.
- `option_pool_tag` – option pool identifier.
- `debt_instrument_tag` – debt instrument type (note, credit line).
- `loan_id` – loan or facility identifier.

Treasury and asset context:

- `wallet_tag` – internal wallet label (ops, reserves, pass-through).
- `treasury_bucket_tag` – treasury segmentation bucket.
- `asset_tag` – asset or token grouping (USDC, DCHUB, other).
- `custody_tag` – custody or provider identifier.

Entities may add custom tags, but custom tags should be namespaced to avoid collisions.

9.9 Multi stablecoin and CBDC support

USDC is the baseline unit of account for early dCorps reporting, but the architecture supports multiple stablecoins and, where technically and legally possible, tokenized fiat instruments such as CBDCs.

Design intentions include:

- An on-chain approved asset registry that lists which stablecoins are supported for:
  - Entity operations
  - Protocol fees
  - Reporting and metrics conversions
- Clear risk criteria for approving, limiting, or retiring an asset, including:
  - Issuer and legal structure
  - Redemption, blacklisting, and freeze mechanics
  - Availability on Ethereum and through the canonical Ethereum to/from dCorps bridge gateways
  - Liquidity and market depth in relevant venues
  - Operational history and transparency of administrative control policies
- Reporting that remains consistent across assets:
  - Transactions are recorded with explicit `asset` (token contract) and amounts
  - Dashboards can produce USD denominated summaries using transparent conversion rules, price feeds, and reconciliation anchors
  - Entities can choose which assets they accept while still producing comparable reports

Not all CBDCs will be compatible with open on-chain systems. Many may be permissioned, non transferable, or restricted in ways that prevent direct integration. dCorps supports CBDC-style assets only where their technical rails and legal constraints allow safe, auditable use.

CBDC rails and integration patterns (non-normative)

- **Open token rails**: CBDC-style assets that exist as transferable tokens on open networks can be treated similarly to stablecoins, subject to the same asset registry risk review and disclosure of administrative controls.
- **Permissioned token rails**: CBDCs on permissioned networks (or with strict whitelisting) typically require regulated gateways or delegated operators to move funds and to publish attestations. Any on-Hub representation is treated as a separate approved asset with explicit counterparty and audit risk.
- **Account or API rails**: where a CBDC is not available as an on-chain transferable asset, dCorps treats it as an external settlement rail; the Hub can still record tagged accounting events and anchor evidence or provider attestations, without attempting to mirror balances inside the kernel.

Eligibility constraints (residency, local-agent verification, wallet whitelisting, and similar rules) are handled as optional adapter logic and off-chain processes. The kernel remains non-custodial and does not perform KYC/KYB.

Because CBDC integration depends on issuer and jurisdiction cooperation, dCorps treats it as an adoption track rather than a kernel dependency. Where a jurisdiction or issuer is willing to interoperate, the foundation and ecosystem-funded adoption research groups (such as ResCo, if established) are expected to coordinate feasibility research, partner mapping, and government-relations work (where lawful), then translate that work into governance-reviewable module proposals and pilot implementations.

---

10. Token model and economic design

10.1 Separation of DCHUB, Hub units, dShares, and NGO governance

The dCorps stack separates tokens and governance concepts by layer:

- **DCHUB**
    - Native gas token of the Hub rollup.

    - Used for protocol governance and (where used) protocol-level fees or deposits for registry and module-registry actions.
    - Does not represent ownership in user entities or in the development corporation or foundation.
    - Staking-based security is not part of the Orbit rollup configuration; staking is a possible future dCorps L1 feature.

- **Hub units**

    - Ten thousand per Hub corporation.
    - Represent ownership and voting rights inside that corporation.
    - Not global tokens; entirely scoped to that entity.

- **dShares**

    - Equity style tokens of public instrument issuers (future extension).
    - Represent that corporation's equity and governance rights.
    - Designed and issued by each corporation under its own legal regime.

- **nonprofit governance**

    - Board and allocation rules, not equity.
    - No token that represents ownership of a nonprofit.

This separation reduces confusion between protocol participation and entity level ownership.

10.2 DCHUB utility and non equity nature

dCorps uses two main assets with distinct roles.

DCHUB

- Native gas token of the Hub rollup (Arbitrum Orbit, Rollup mode).
- Used for protocol-level governance, within the scopes defined in this whitepaper and related governance charters.
- May be required for certain protocol-level fees or deposits (for example registry actions and module registry operations) as defined by governance.
- Staking is not part of the Orbit rollup configuration; if dCorps launches its own L1, staking incentives may be introduced under separate policy.

Stablecoins (bridged)

- Primary operating currencies for entities using dCorps.
- Represented as canonical bridged ERC-20 contracts on dCorps, initially backed by Ethereum-originated assets via bridge gateways.
- Used for invoices and revenue, salaries and contractor payments, vendor payments, grants, donations, and other operating flows.
- Used by applications or adapter services for optional service fees where they choose to price in stablecoins; protocol-level fees remain DCHUB-denominated.
- Native issuer integrations (for example Circle-native USDC mechanisms) are a future possibility, not guaranteed.

In high-level economic terms:

- More entities and more protocol modules increase utilization of the Hub.
- Higher utilization can increase demand for blockspace and can affect the DCHUB gas market and rollup operating budgets, but market outcomes are uncertain and parameter dependent.
- Protocol fees collected in DCHUB can fund operations and ecosystem public goods; optional adapter and service fees in stablecoins can support program operations without relying solely on DCHUB distribution.

These relationships describe protocol mechanics, not promises about price, liquidity, or any market outcome.

DCHUB does not entitle holders to dividends, profit distributions, or rights to donations from any entity. It does not represent ownership of user entities, the development corporation, or the foundation. The design intention is that DCHUB is a protocol level utility token. How any given jurisdiction classifies DCHUB or related instruments is a matter of local law and facts. This whitepaper does not take a legal position on classification.

10.3 Total supply and allocations

DCHUB has a hard cap maximum supply of **1,000,000,000 (one billion)** tokens.

“Hard cap supply” means the protocol is designed so that total DCHUB supply **cannot exceed 1,000,000,000**.

This section defines the cap and allocations. Vesting and lockups are described in section 10.4, and circulating supply release caps are described in section 10.4A.

v1 supply implementation stance

- At TGE, the full cap supply is created once in genesis and assigned into on-chain vesting accounts and governance controlled module accounts that correspond to the allocations below.
- After genesis, there is no discretionary inflation path intended to create additional DCHUB beyond what was created at genesis. “Emissions” in this document refers to tokens entering circulation from these pre-funded allocation accounts (for example the network incentives reserve), not to new supply creation beyond the hard cap.
- Total supply may decrease over time through burns or other sink mechanisms if adopted by the protocol, but it cannot rise above the genesis hard cap.

An indicative allocation is:

- **Founder**: 15 percent (150,000,000)
- **Core team and future contributors**: 8 percent (80,000,000)
- **Investors**: 15 percent (150,000,000)
    - Seed: 2.5 percent (25,000,000)
    - Series A: 3.5 percent (35,000,000)
    - Series B: 3 percent (30,000,000)
    - Series C: 2 percent (20,000,000)
    - Public sale / ICO (if any): 4 percent (40,000,000)
- **Community and ecosystem programs**: 33 percent (330,000,000)
- **Network incentives reserve (future L1 + operations)**: 18 percent (180,000,000)
- **Protocol Treasury**: 4 percent (40,000,000)
- **dCorps foundation**: 4 percent (40,000,000)
- **Liquidity bootstrap (operational liquidity)**: 3 percent (30,000,000)

Founder, team, and investor allocations together represent 38 percent of supply. Community and ecosystem programs plus the network incentives reserve represent 51 percent of supply, distributed over time through open programs and governed budgets. Protocol Treasury, foundation, and liquidity bootstrap allocations are operational and stewardship pools and are designed to be non capturing by policy and by default governance configuration (see section 10.3A and section 10.7A).

Indicative sub allocation plan for Community and ecosystem programs (33 percent)

Community and ecosystem programs are intended to be programmatic and transparent, with clear buckets that map to public goods needs. An indicative split is:

- **Security and audits**: 7 percent (70,000,000)
    - Audits, monitoring, bug bounties, incident response tooling, safety critical infrastructure.
- **Core tooling and developer grants**: 9 percent (90,000,000)

    - SDKs, indexers, explorers, accounting and payroll tooling, compatibility tooling, module development grants.
- **Jurisdiction pilots and module development**: 6 percent (60,000,000)
    - jurisdiction adapter pilots, standards work, legal research support, jurisdiction adapter implementation grants.
- **NGO onboarding and support**: 6 percent (60,000,000)
    - NGO onboarding programs, nonprofit fee waivers where adopted, training, reporting tooling, local capacity building.
- **Ecosystem growth and incentives**: 4.95 percent (49,500,000)
    - Limited incentives for meaningful adoption, integrations, ecosystem reliability, recruiting incentives, and targeted airdrops, structured as time bounded programs with transparent criteria.
- **Gas-free onboarding credits**: 0.05 percent (500,000)
    - Time-boxed fee grants that cover entity onboarding gas at scale, with per-entity caps, governance approval, and public reporting.

Entity onboarding gas support may be issued as time-boxed fee grants rather than direct transfers, capped per entity and governed through the community program pool.

These are budget buckets for governance and reporting. They do not imply that the full bucket amount is released early. Release caps and a circulating supply schedule are defined in section 10.4A.

Exact vesting, lockups, unlock mechanics, custody, and distribution controls are defined in section 10.4 and section 10.4A and will be finalized in a separate Token Policy and related legal documents before any public sale.

10.3A Governance, concentration, and continuity

Token distribution and governance are designed with two goals that must be balanced carefully:

- Avoid permanent capture of the protocol by a small insider bloc.
- Avoid destabilizing core development and security without a serious, deliberate process.

Key design intentions and baseline rules are:

- Over time, as the network incentives reserve and community allocations are distributed, active builders, entities, operators, and contributors should hold a growing share of effective governance power. Governance parameters and program design should support this by prioritizing distributions tied to real participation and contribution.
- **Community allocation custody and release (genesis stance)**

    - Community and ecosystem program allocations are intended to be held in governance controlled module accounts or contracts with explicit release rules, not in discretionary personal wallets.
    - Program releases are executed through published program categories (for example builder grants, audits and security, integrations, jurisdiction pilots, NGO onboarding) with transparent reporting of recipients, milestones, and outcomes.
    - Where an administrator or operator role exists for program execution in early phases, it must be publicly disclosed, time bounded, and contestable through governance, with a clear path toward more automated and decentralized program administration.
- **Protocol Treasury and foundation voting policy (default non voting)**
    - DCHUB held in Protocol Treasury and dCorps foundation controlled accounts is **non voting by default** in protocol governance tallies.
    - Treasury and foundation accounts are expected to be held in non delegating module accounts or contracts (or equivalent mechanisms) so these balances cannot be used to generate governance voting power through delegation.
    - Governance can explicitly enable limited voting behavior for narrowly defined proposal classes if required for legal or operational reasons, but any such enablement must be:
        - Time bounded
        - Scope limited (explicit proposal types)
        - Publicly disclosed as policy
        - Enforced at the governance contract level where technically feasible
- **Vesting weighted governance for locked allocations**
    - DCHUB held in on-chain enforced vesting contracts is voting eligible only to the extent it is vested.
    - The unvested portion of a vesting allocation is non transferable and non voting.
    - This rule applies symmetrically to founder, team, and investor vesting contracts.
    - Vesting weighted voting is designed to ensure that governance power emerges over time as long term commitments vest, while preserving the ability of locked allocations to participate in future L1 security incentives if dCorps launches its own L1 under separate policy.
- **Community and ecosystem allocation is programmatic, not discretionary**
    - Community programs should be structured as published budgets and program categories with transparent reporting.
    - Program design should prefer open calls, milestone based grants, and objective evaluation criteria, with public reporting on allocations and outcomes.
    - Programs should include conflict and affiliation disclosures for decision makers and recipients.
- **Anti capture guardrails for protected changes**
- Governance may require voting-power age requirements for protected proposal classes, where only DCHUB locked for a parameterized minimum age is counted toward quorum and passing power for those proposals.

    - Governance may require execution timelocks for protected proposal classes, giving the ecosystem time to react before a passed proposal takes effect.

At the same time:

- The development corporation is expected to remain a core engineering and product provider for the protocol for a long period.
- Any decision to significantly reduce its role, or to move core development to a different primary provider, should require strong quorums and supermajorities in governance and be paired with a credible alternative development arrangement.

Exact numerical thresholds, voting-power age parameters, timelock durations, and the detailed governance module configuration will be documented in the Governance Charter and related policy documents and can evolve through governance as the network matures.

10.4 Vesting, lockups, and emissions

In this whitepaper, the **Token Generation Event (TGE)** is the same event as **mainnet launch**, defined as the first production block of the dCorps Hub mainnet, when initial DCHUB allocations become live on-chain and DCHUB becomes usable for gas, protocol governance, and protocol-level fees where enabled. Staking applies only if a future dCorps L1 launches under separate policy.

In this whitepaper, “emissions” refers to **tokens entering circulation** from predefined allocations (for example the network incentives reserve), not the creation of supply beyond the fixed cap defined in section 10.3.

This section focuses on vesting and lockups. For circulating supply definitions, release caps, and an illustrative unlock schedule, see section 10.4A.

dCorps uses long, on-chain enforced vesting schedules to align insiders with the long term health of the protocol. Indicative schedules are:

- **Founder allocation (15 percent)**
    - Total allocation: 150,000,000 DCHUB
    - Cliff: 24 months after TGE
    - Vesting: linear vesting from month 24 to month 96 after TGE, with monthly unlocks
    - Transfer guidelines: during at least the first four years after TGE, only a limited portion of vested founder tokens may be transferred per year, except where tokens are used as stake or bond for protocol security under a future dCorps L1

policy. Detailed limits will be defined in a Token Policy and mirrored in legal agreements.
    - Enforcement: founder vesting and any transfer limits will be enforced on-chain where technically feasible, with matching legal documentation.
    - Governance voting: voting eligibility is limited to vested amounts (see section 10.3A).
- **Core team and future contributors (8 percent)**
    - Total allocation: 80,000,000 DCHUB
    - Typical schedules: 18 month cliff followed by 48 month linear vesting for most roles, with longer schedules for senior roles
    - Vesting is enforced on-chain using standardized vesting contracts. Leaving early may trigger clawback of unvested portions, subject to contracts.
    - Governance voting: voting eligibility is limited to vested amounts (see section 10.3A).
- **Investors (15 percent)**
    - Total allocation: 150,000,000 DCHUB
    - Seed: 12 month lockup, then 36 month linear vesting
    - Series A: 9 month lockup, then 36 month linear vesting
    - Series B: 6 month lockup, then 30 month linear vesting
    - Series C: 3 month lockup, then 24 month linear vesting
    - Public sale (if any): 10 percent available at TGE; remaining 90 percent linear vesting over 12 months
    - Governance voting: voting eligibility is limited to vested amounts (see section 10.3A).
- **Community and ecosystem programs (33 percent)**
    - Total allocation: 330,000,000 DCHUB
    - Held in governance controlled module accounts or contracts with explicit release rules, not discretionary personal wallets
    - Released gradually through open programs and milestone based grants subject to governance
    - Release caps are defined in section 10.4A to reduce circulating supply shocks and to make program releases legible and predictable
- **Protocol Treasury (4 percent) and dCorps foundation (4 percent)**
    - Total allocations: 40,000,000 DCHUB for the Protocol Treasury and 40,000,000 DCHUB for the foundation
    - Held in Treasury and foundation controlled accounts or contracts under governance and, where required, by the legal entities that represent them
    - Funds are released only through on-chain governance and, where needed, matching off-chain execution
    - Governance voting: Treasury and foundation balances are non voting by default (see section 10.3A).
- **Network incentives reserve (18 percent)**

    - Total allocation: 180,000,000 DCHUB reserved for future L1 security incentives and/or governance-approved decentralization and safety programs
    - Not required for Orbit rollup v1 consensus; releases are controlled by explicit schedules and governance policy
    - Governance can adjust any release parameters within predefined bounds (see section 10.6B), with stronger requirements for changes that materially increase near-term releases
- **Liquidity and market support (3 percent)**
    - Total allocation: 30,000,000 DCHUB
    - A portion may be available at or shortly after TGE to seed DEX liquidity for operational usability
    - The remainder is subject to internal policies that restrict its use to liquidity provisioning and operational market infrastructure, with transparent reporting and capped budgets
    - Governance voting: balances reserved for liquidity and market support are expected to be non voting by default and held under policy constraints consistent with section 10.3A.

All schedules and limits in this section are indicative. Final details will be documented in a Token Policy and legal agreements before any sale. A circulating supply schedule and release cap standard is defined in section 10.4A to make supply availability legible and predictable for entities, operators, and the ecosystem.

10.4A Circulating supply schedule and release caps (illustrative, v1 planning)

This section defines a standard way to talk about circulating supply and unlocks in dCorps. It is designed to prevent ambiguity and to make supply availability legible for entities, operators, and the ecosystem.

10.4A.1 Definitions

- **Minted supply** DCHUB that exists on-chain, including tokens held in vesting contracts, module accounts, and distribution pools.
- **Circulating supply** DCHUB that is transferable by its holder without time locks or vesting restrictions and without a governance controlled release step.
- **Liquid supply** The portion of circulating supply that is actively available for typical market transfers. Liquid supply is a market condition, not a protocol guarantee.

Reference explorers must publish minted supply and circulating supply explicitly as separate numbers.

10.4A.2 Genesis custody stance (design intention)

At TGE, major allocations are expected to be held as follows:

- Founder, team, and investor allocations are held in on-chain enforced vesting or lockup contracts.
- Community and ecosystem programs allocation is held in governance controlled module accounts or contracts with explicit release rules.
- Network incentives reserve allocation is held in a governance-controlled distribution contract and becomes circulating only as releases are executed under policy.
- Protocol Treasury and foundation allocations are held in non delegating accounts or contracts, non voting by default, with governed release paths.
- Liquidity bootstrap allocation may be deployed for operational liquidity under a published liquidity plan and Treasury policy.

10.4A.3 Maximum intended transferable supply at TGE (cap)

The design intention is that TGE circulating supply is kept low.

At TGE, the maximum intended transferable supply is:

- Liquidity bootstrap: up to 30,000,000 DCHUB (3 percent of total supply)
- Optional community launch distribution: up to 10,000,000 DCHUB (1 percent of total supply), only if approved through governance for launch programs

All other major allocations are expected to be non transferable at TGE due to vesting, lockups, or governance controlled custody.

10.4A.4 Insider vesting unlock schedule (illustrative)

The schedules below are illustrative, based on the indicative vesting patterns in section 10.4:

- Team allocation assumed as an 18 month cliff followed by 48 month linear vesting.
- Investor allocation assumed as round-based lockups and vesting:
    - Seed: 12 month lockup + 36 month vesting.
    - Series A: 9 month lockup + 36 month vesting.
    - Series B: 6 month lockup + 30 month vesting.
    - Series C: 3 month lockup + 24 month vesting.
    - Public sale: 10 percent at TGE + 12 month linear vesting for the remainder.
- Founder allocation assumed as a 24 month cliff followed by linear vesting from month 24 to month 96.

Illustrative cumulative unlocked amounts for insider allocations (end of year, in millions of DCHUB)

- End of year 1: 56.4 (public sale unlocks and early investor vesting)
- End of year 2: 108.4 (team partial vesting and investor unlocks)
- End of year 3: 187.9 (team and investor unlocks plus founder partial unlock)
- End of year 4: 250.0 (investors fully unlocked plus founder and team partial unlocks)
- End of year 5: 295.0
- End of year 6: 330.0
- End of year 7: 355.0
- End of year 8: 380.0 (founder fully unlocked)

Illustrative insider unlock curve (cumulative, millions of DCHUB)

Year: 0 1 2 3 4 5 6 7 8

Unlocked: 0 56.4 108.4 187.9 250.0 295.0 330.0 355.0 380.0

Reference explorers must show actual on-chain vesting contracts and the exact unlock timelines for each vesting schedule once deployed.

10.4A.5 Network incentives reserve schedule (genesis default)

To make releases legible, the network incentives reserve uses a published schedule that sums to the pool cap of 180,000,000 DCHUB.

In the Orbit rollup configuration, this reserve may remain unused in early phases; any activation or use is explicitly governed and must be disclosed through on-chain events.

A genesis default schedule (illustrative, per year after TGE) is:

- Year 1: 40,000,000
- Year 2: 35,000,000
- Year 3: 30,000,000
- Year 4: 25,000,000
- Year 5: 20,000,000
- Year 6: 15,000,000
- Year 7: 10,000,000
- Year 8: 5,000,000

Total: 180,000,000

Governance may adjust the schedule within predefined bounds (see section 10.6B). Changes that materially increase near-term releases beyond those bounds are designated as Protected Changes under the governance framework.

10.4A.6 Community release caps (initial guardrails)

To reduce sudden changes in circulating supply, community program releases from the Community and ecosystem programs allocation follow an explicit cap.

Initial cap (design intention for v1):

- Maximum release into circulating supply: 50,000,000 DCHUB per year (5 percent of total supply), measured over a rolling four quarter window.
- Maximum release in any single quarter: 20,000,000 DCHUB.

These caps apply to tokens leaving governance controlled community custody into transferable user addresses. Internal movements within governance controlled custody do not count as release.

Raising these caps is designated as a Protected Change under the governance framework.

10.5 Protocol fees (DCHUB-denominated)

In addition to gas, the protocol supports protocol-level fees for specific services. These protocol fees are denominated in DCHUB; interfaces may quote USDC-equivalent estimates or sponsor fees via grants.

Core protocol fees (DCHUB)

Core protocol actions may charge DCHUB fees (or require DCHUB deposits), such as:

- Entity registration and registry listing renewal (including name lease renewal where applicable).
- Premium names and namespaces.
- App and module registry listing, upgrades, and renewals (where used).

Optional service fees (stablecoins, off-protocol)

Applications or adapter operators may charge separate service fees in stablecoins for off-chain operations (for example jurisdiction filings, attestations, or monitoring), but these are not protocol-level fees and are handled outside protocol consensus.

DCHUB protocol fees are routed according to predefined rules, for example:

- A share to the **Protocol Treasury**.

- A share to the **dCorps foundation** once established.
- A share to participating jurisdictions where jurisdiction adapter modules are used.

Protocol fees help fund operations and grants without relying only on emissions or discretionary Treasury drawdowns.

10.5A Use of DCHUB in module economics and Treasury operations (policy aligned)

USDC is the primary operating currency for entities and for most operating flows. DCHUB is the gas and coordination asset for the Hub rollup. dCorps uses DCHUB in registry and module economics to align long-lived protocol usage with the governance token, while avoiding any implication of investment return.

10.5A.1 DCHUB in protocol and module fee design

Selected protocol services may include a DCHUB component, such as:

- Application and module registry listing, upgrade, or renewal fees.
- Deposits or bonds required for official jurisdiction adapter modules and sector frameworks.
- Spam resistance deposits for selected on-chain actions.

Where bonds are used, bond terms must be explicit:

- The bond amount, duration, and release conditions are defined by governance parameters.
- Forfeiture conditions apply only to objective, verifiable violations of published module integrity requirements, as defined in module standards.
- Bonds are not used to punish policy disagreements or unpopular decisions.

10.5A.2 Treasury operations policy and prohibitions

The Protocol Treasury may hold and manage assets for operational continuity, security, grants, and ecosystem development under a governance approved Treasury policy.

Treasury policy explicitly prohibits:

- Price targeting, informal pegs, or discretionary market operations intended to raise or defend the market price of DCHUB.
- Public or private commitments to buy back DCHUB as a form of return to holders.

- Any program that could reasonably be interpreted as a dividend, profit distribution, or yield promise to DCHUB holders.

Treasury policy may permit, within strict budget bounds and transparency requirements:

- Liquidity provisioning in DCHUB pairs to support:
  - Predictable transaction fee markets,
  - Routine on-chain operations, and
  - Ecosystem usability for entities and builders.

Any liquidity provisioning must be executed under an on-chain approved plan that discloses:

- Maximum budget and duration,
- Venues and custody assumptions,
- Risk controls and withdrawal conditions, and
- Public reporting cadence for positions and outcomes.

10.5A.3 No implied guarantees

Nothing in protocol fee design or Treasury policy is a guarantee of value accrual. These mechanisms exist to fund security and public goods and to improve operational reliability, not to promise any financial outcome.

10.6 Long term sustainability and rollup operations funding

The long-term economic goal is:

- Fund rollup operations and safety work without relying on perpetual token emissions.
- Keep costs legible: users pay DCHUB gas on dCorps; operators pay ETH on Ethereum for data posting and settlement.
- Avoid hidden subsidies and unclear “security budgets” that only exist as narratives.

Unlike a sovereign proof-of-stake chain, the Orbit rollup does not have staking rewards for consensus security. Sustainability therefore comes from transparent fee policy and budgets rather than a staking reward program.

10.6A Operating budgets and fee routing (explicit)

dCorps tracks two ongoing budgets:

- **Rollup operations budget (primarily ETH)**: sequencing infrastructure, batch posting, Ethereum data-availability costs, RPC/indexing infrastructure.

- **Security and public-goods budget (primarily stablecoins)**: audits, monitoring, bug bounties, incident response, ecosystem grants.

Funding sources can include:

- DCHUB gas fees (user-paid on dCorps), routed by on-chain policy.
- Protocol fees (DCHUB and/or stablecoins) for registry and module-registry actions.
- Treasury and foundation budgets approved by governance.

If costs exceed funding, governance levers include:

- Adjust gas pricing parameters and paymaster policy.
- Adjust protocol fees within policy bounds.
- Fund the gap from Treasury under a disclosed budget.
- Reduce non-critical spending while keeping core safety work funded.

10.6.1 Illustrative rollup cost model (non-binding)

In a rollup, marginal cost is dominated by Ethereum data posting:

- Operators pay ETH on Ethereum to post transaction data (or compressed batches) and to interact with rollup contracts.
- Users pay DCHUB gas on dCorps; operators can convert fee revenue into ETH to cover L1 costs.

A simple accounting frame:

- **Annual_ops_cost** ≈ (L1_data_cost_in_ETH + infra_costs) + security_ops_budget
- **Annual_ops_funding** ≈ (gas_fee_revenue + protocol_fee_revenue) ± (Treasury subsidy)

This section is conceptual and not a prediction of any fee level.

10.6B v1 protocol parameters snapshot (initial ranges)

This section provides initial target ranges for v1 planning and launch configuration. Genesis values and ongoing adjustments are governed by protocol governance within bounded ranges defined in the Protocol Parameters and Economics reference.

These are targets and bounded ranges, not guarantees.

Rollup operations and governance (v1 targets)

- Timelock delay for upgrades and protected parameter changes (public at genesis).
- Conservative upgrade policy and explicit emergency roles (if any), with a published plan to reduce powers over time.
- Published operator roles (sequencer and batch poster) and key-rotation procedures.
- Published bridge gateway addresses and admin/pausing roles for transparency.

Gas and fee targets (Hub, usability targets)

- DCHUB is required for Hub gas at the base protocol level.
- Target median fee range for common entity operations under normal load (measured in USDC terms and shown in UIs as an estimate):
    - Standard operations: 0.05 to 0.25 USDC equivalent
    - Heavier operations (batch actions, larger payload governance actions): 0.25 to 2.00 USDC equivalent
- Rollup operators configure fee parameters in practice. Governance publishes a recommended band intended to keep typical operations within these usability targets while preserving anti-spam conditions.

Protocol service fees (v1 initial ranges)

Protocol service fees are separate from gas and are designed to fund public goods, security operations, indexing, and ecosystem development without relying only on emissions. For budgeting and comparability, ranges below are expressed in USDC-equivalent terms; protocol service fees are DCHUB-denominated and may be covered by fee grants or app sponsorship for stablecoin UX (see section 10.5).

- Entity registration fee (one time, USDC equivalent):
    - Hub corporation: 50 to 250 USDC equivalent
    - Hub nonprofit: 0 to 100 USDC equivalent (with optional waivers per policy)
- Registry listing renewal (annual, per entity, USDC equivalent):
    - Hub corporation: 25 to 150 USDC equivalent
    - Hub nonprofit: 0 to 50 USDC equivalent (with optional waivers per policy)
- Premium names and namespaces (where used):
    - Initial lease and renewal ranges are expected to be set to price scarce namespaces and reduce squatting, with values set by governance in DCHUB and expressed as USDC-equivalent bands for UX.
- App and module registry listing fees (where used):
    - Annual listing and renewal fees are set by governance in DCHUB with risk tiering (for example higher for custody related apps, issuance apps, and critical infrastructure modules), and expressed in USDC-equivalent bands for UX.

Governance change constraints (v1)

- Adjustments within these ranges follow standard on-chain governance.
- Changes to the existence of these bounds, or changes that materially weaken security alignment requirements, are designated as Protected Changes under the governance framework.

10.6C Rollup operations and security budget logic (v1)

Rollup security is anchored to Ethereum, but the system still has operating and safety costs. dCorps uses an explicit budget framing to keep those costs visible and governed, rather than implicitly subsidized or ignored.

Budget components (conceptual)

- **Rollup operations (primarily ETH)**: Ethereum data posting, sequencing and RPC infrastructure, monitoring, and incident response capacity.
- **Security operations (primarily stablecoins)**: audits, bug bounties, and safety-critical tooling and indexer maintenance.

Funding sources (v1)

- DCHUB gas fees and protocol fees.
- Treasury and foundation spending under disclosed policies.
- Optional application or adapter service fees in stablecoins for off-chain services (where used).

These figures are planning anchors, not promises.

Illustrative scenarios (not predictions)

- **Low usage**: fee revenue may not cover full ops/security budgets; governance may subsidize from Treasury or constrain spending.
- **Moderate usage**: fees cover most operations costs; Treasury focuses on audits and ecosystem public goods.
- **High usage**: fees can cover operations costs comfortably; governance may reduce subsidies and expand public-goods budgets within policy limits.

Governance response when costs and funding diverge

If operating costs exceed funding, governance can respond through explicit levers within bounded ranges:

- Adjusting minimum gas pricing and fee market rules.
- Adjusting protocol service fees within policy constraints.
- Funding additional safety work from Treasury, not by changing core custody boundaries.

The intent is legibility: the community can see what costs exist, what is funded, and what tradeoffs are made.

10.7 Liquidity strategy for DEX and CEX (reliability focused)

dCorps does not operate exchanges or matching engines. Liquidity exists to support usability and network operations, not as a promise of market performance.

10.7.1 DEX liquidity (permitted scope)

The Protocol Treasury may support limited DEX liquidity under Treasury policy to improve:

- Predictable transaction fee conditions,
- Onboarding and day to day usability for entities and builders, and
- Reduced slippage for operational conversions related to protocol activity.

DEX liquidity activities must:

- Use capped budgets and time bounded programs,
- Be executed transparently with public reporting, and
- Avoid messaging that implies price support or return expectations.

10.7.2 CEX listings

Centralized exchange listings are at the discretion of exchanges. Any integration work or listing fees, if ever paid by a dCorps controlled entity, must be:

- Approved through governance where required by policy,
- Disclosed transparently to the extent legally possible, and
- Structured as infrastructure enablement, not as a performance promise.

dCorps makes no promises of listing, liquidity, or any trading conditions.

10.7A Liquidity bootstrap policy (v1)

The liquidity bootstrap allocation (3 percent of total supply) exists to support operational usability for gas and routine entity operations. It is not intended to target price, imply return, or act as a market support commitment.

Custody and control

- The liquidity bootstrap allocation is held in a dedicated on-chain account labeled for liquidity operations and controlled under the Treasury policy.
- The liquidity bootstrap balance is non voting by default and is not delegated for governance power (see section 10.3A).
- Deployments and withdrawals follow an on-chain approved Liquidity Plan that specifies:
    - maximum budget and duration,
    - permitted venues and custody assumptions,
    - position sizing and risk limits,
    - authorized operators (if any), and
    - transparency requirements (public disclosures and reproducible time-window views).
- In early phases, execution may be delegated to a time bounded operational multisig with a narrow mandate (liquidity provisioning only). The multisig's addresses, signers, and mandate must be publicly disclosed, and the delegation is revocable through governance.

Permitted venues and activities (default stance)

Permitted use cases are limited to actions that improve predictable on-chain usability:

- Seeding and maintaining DEX liquidity in DCHUB paired pools (for example DCHUB and USDC) to reduce slippage for routine conversions needed by entities, operators, and service providers.
- Rebalancing and consolidating liquidity positions within the same approved venues, within the risk limits and budget caps of the Liquidity Plan.
- Supporting bridge and routing usability where applicable, under the same transparency and custody constraints.

Prohibited use cases include:

- Discretionary market making intended to move price.
- Private OTC activity where the effective terms are not publicly auditable.
- Any messaging or program design that frames liquidity activity as a return mechanism for DCHUB holders.

Transparency requirements (v1 standard)

Liquidity operations must be transparent and reproducible:

- All liquidity positions controlled by the liquidity bootstrap account must be discoverable on-chain.
- Any liquidity summary published by the Treasury MUST be reproducible from tagged on-chain activity and MUST state the timeframe covered (no fixed monthly/periodic cadence is required by the kernel), including:
    - current positions and venues,
    - assets deployed and withdrawn over the timeframe,
    - fees earned and losses realized (if any),
    - policy compliance attestations (budget caps, permitted venues), and
    - a clear statement that results are not guaranteed and are not a promise of future performance.

CEX interaction boundary

dCorps does not operate exchanges or matching engines. If any centralized exchange integration is pursued, it follows the same reliability posture:

- Any transfer of liquidity bootstrap tokens to a centralized venue must be explicitly authorized under a Liquidity Plan or a separate governance action and must be disclosed as infrastructure enablement, not as a market performance action.
- Custody and counterparty risk is treated as a separate, explicit risk decision under Treasury policy.

This policy exists to make liquidity actions legible, bounded, and auditable, while preserving a clear non intermediation and non promise posture.

10.8 Grants and incentives

The **community and ecosystem** allocation and part of the USDC Protocol Treasury will fund:

- Developer grants for:
    - Tools and applications.
    - jurisdiction adapter modules.
    - Sector frameworks.
    - Security and monitoring tools.

- NGO and nonprofit support:
    - Onboarding and training.
    - Beneficiary management tools.
    - Local capacity building.
- Security and research:
    - Independent audits.
    - Bug bounty programs.
    - Academic and applied research.

The **dCorps foundation** is expected to administer a significant portion of these programs, in coordination with protocol governance. Grants will:

- Be milestone based.
- Use clear criteria and public reporting.
- Not create employment or agency relationships unless accompanied by separate contracts.

Treasury and grants policy is described further in the governance section.

10.9 Incentives by actor

dCorps is designed as an ecosystem where different participants have aligned incentives that are not dependent on speculative narratives.

- **Entities (corporations and nonprofits)**
    - Gain standardized governance, transparent operating flows, and reusable tooling
    - Gain reputational benefits from verifiable state, clearer audits, and credible histories
- **Founders and core contributors**
    - Are incentivized by long vesting schedules, reputation, and the long term health of the network
- **Rollup operators and infrastructure providers**
    - Earn governed shares of L2 fees and/or explicit operator budgets; cover L1 posting costs in ETH
    - Are incentivized to keep sequencing, batch posting, and RPC reliable for serious entities
- **Builders (apps, modules, tooling)**

    - Can earn revenue from software, integrations, and services
    - Can receive grants and incentives tied to real adoption and milestones
- **Jurisdictions and service providers**
    - Can publish jurisdiction adapter modules and earn fees from entities that opt in
    - Reduce integration costs by using a shared registry and standardized data models
- **Auditors, reviewers, and oversight bodies**
    - Can provide attestations and services with clearer evidence trails and lower reconciliation overhead
- **Donors and grant makers**
    - Gain real time visibility into allocation and governance, improving decision quality and reducing reporting friction

None of these incentives imply any promise of returns. They describe why participants might choose to use and support the network.

---

11. Go to market, target users, and adoption targets

11.1 Target user segments

Primary target segments:

- **Early stage remote first startups and small teams**
    - Technology and services corporations with global customers.
- **Crypto native protocols and DAOs**
    - Especially those seeking to professionalize operations and governance.
- **nonprofits and NGOs**
    - Initially, organizations comfortable with high transparency and basic crypto exposure.
- **Jurisdictions and corporate service providers**
    - Looking for digital native products and programmable regimes.

Secondary segments:

- **Auditors, accountants, and law firms**
    - Can build services and tools on top of dCorps for their clients.
- **Donors and grant makers**
    - Seeking better data about NGOs and projects.

11.2 First wave focus (first one to two years)

In the first one to two years, dCorps focuses brutally on a narrow wedge of real use:

- **Private corporation baseline on the Hub (CORP-PRIVATE-STD)**

  Remote first startups and small teams that are willing to run the large majority of their operating stack in USDC on the shared Hub. They use the CORP-PRIVATE-STD template: ten thousand internal units, standardized wallets, and on-chain accounting.

- **Nonprofit baseline on the Hub (NONPROFIT-SIMPLE)**

  NGOs and nonprofit teams that want transparent donations and program spending, board based governance, and verifiable allocation ratios, again on the shared Hub without running their own chain.

- **At least one jurisdiction attachment pilot (phased path, post mainnet)**

  Because direct jurisdiction integration and key custody by a jurisdiction typically takes time, dCorps targets a phased path that can start shortly after mainnet launch:

    1. A **temporary delegated filing provider bridge** that allows real formations, renewals, and registry updates to be completed off-chain by local providers and reflected on-chain through standardized attestations and labels.
    2. A **disclosure, fee, and reporting module** that integrates with Hub entity state, collects fees in USDC, and produces machine readable oversight signals.
    3. A **direct jurisdiction integration step** where legal recognition and registry actions are bound to on-chain module state under keys controlled by the jurisdiction (or an explicitly disclosed delegated operator under its supervision).

Alongside these entities, the first wave of builders focuses on:

- Explorers and dashboards that read the entity registry, wallets, and flows.
- Accounting and payroll tools that use the standard chart of accounts and wallet structure.
- NGO reporting tools that turn donation and program flows into usable reports for donors and grant makers.
- Typed workflow modules that improve tag integrity by producing deterministic categories for common operating actions.

Success in this phase is measured by:

- Working software used by a small but serious set of entities.
- Clear evidence that stablecoin native startups can run most or all of their operations from dCorps wallets.
- NGOs that demonstrate real transparency benefits and better donor trust.
- Reliable rollup operations (sequencing, batch posting, and RPC) with visible uptime and incident response.

The development corporation will play a leading role initially, shipping core modules and integrations. The foundation takes on more responsibilities for modules, grants, and registry operations as it is established.

11.3 Five year adoption targets (illustrative)

Adoption projections are **aspirational scenarios**, not promises. A reasonable internal target by year five could be:

- More than 1,000 active Hub corporations.
- Hundreds of NGOs and foundations.
- Between 10 and 30 public instrument issuers (dShares), if/when enabled as a future extension.
- Several jurisdiction adapter modules in production.
- A healthy ecosystem of applications using dCorps data.

These numbers help guide planning. They are forward looking and subject to change. Nothing in this section guarantees that they will be achieved.

11.4 Partner strategy

Key partners include:

- **Back office and accounting providers**
    - Integrate their tools with dCorps data structures.
    - Offer migration paths from traditional systems.
- **Legal and corporate service providers**
    - Use dCorps as a substrate for modern corporate and NGO services.
    - Help clients choose jurisdictions and structures.
    - Often interact with jurisdiction adapter modules.
- **NGO infrastructure organizations**
    - Foundations and intermediaries that work with many NGOs.
    - Use dCorps to enhance transparency and trust.
- **Infrastructure providers and rollup operators**
    - Cloud and bare metal providers that support sequencing, batch posting, RPC, and indexing operations.
    - Security firms that support audits and monitoring.

Partnerships are expected to be **open** and competitive. The protocol does not enshrine any exclusive provider.

The foundation is expected to work with many partners, especially jurisdictions and sector bodies, while remaining neutral and protocol focused.

11.5 Success metrics beyond token price

Success is judged by measurable adoption, operational usefulness, and security, not by token price.

Token price and trading volume are not used as primary indicators of protocol success.

For the structured metrics set used in implementation planning and ongoing reporting, see section 16.4.

---

12. Capital formation, markets, and donations

dCorps does not run exchanges or fundraising platforms. It provides the primitives and state that independent platforms can use.

12.1 Private ownership transfers and cap table updates

Private transactions include:

- Secondary sales between investors.
- Founder liquidity events.
- Employee option exercises.
- Buybacks and redemptions.

Pattern:

- Legal agreements are negotiated off-chain.
- Agreements are anchored on-chain via hashes and metadata.
- Unit or dShare balances are updated according to agreed terms.
- Cap tables are adjusted and versioned with links to the relevant approvals and documents.

This creates a verifiable history of ownership changes, even for private dealings. These patterns also apply to joint venture and SPV structures, where parent entities or investors subscribe to units or dShares in a dedicated dCorps entity that is ring fenced from their main operations.

12.2 Primary issuance modules

Primary issuances are handled by **modules and platforms**, not by the core protocol.

Modules and platforms can be designed for:

- **Private placements**
    - For qualified or professional investors.
    - Enforce eligibility through KYC, KYB, and credential based allow lists.
- **Community offerings (where legal)**
    - With jurisdiction specific constraints and protections.

    - Enforce per user caps, geofencing, and other rules.

- **NGO campaigns and recurring donations**

    - Structured flows into program wallets.
    - Clear commitments about allocation and reporting.

These modules and platforms:

- Can be built and operated by independent teams.
- May receive support from the dCorps foundation or ecosystem grants.
- Are not part of the base consensus layer.

Legal compliance for offerings remains with the module operators and participating entities.

12.3 Listing and secondary markets

dShares and DCHUB may be:

- Listed on **decentralized exchanges** that choose to support them.
- Supported by **centralized exchanges** that decide to list them.

dCorps:

- Does not operate exchanges or matching engines.
- Does not guarantee listing or trading conditions anywhere.
- Does not guarantee that any market is suitable or accessible to any given user.

Venues retain responsibility for:

- Eligibility rules and compliance.
- KYC and AML requirements.
- Listing and delisting decisions.

12.4 DeFi integrations and portfolio products

DeFi protocols and structured product providers can:

- Use DCHUB or selected dShares as collateral or portfolio elements.

- Construct indices or baskets based on:
    - Sector tags.
    - Jurisdiction attachments.
    - Transparency and allocation metrics for NGOs.

Examples:

- Lending markets that accept dShares of corporations with stable revenue and governance behavior.
- Impact oriented portfolios that share a portion of yield with NGOs meeting defined criteria.

These products are independent from dCorps. They carry their own smart contract, market, and regulatory risks.

12.5 Investor safeguards

Capital formation on dCorps can incorporate safeguards, including:

- **On-chain vesting and lockups**
    - Founder, team, and early investor allocations vest over time.
    - Transfers of locked or unvested tokens are blocked at contract level.
    - Changes to vesting parameters require explicit governance actions.
- **Transfer controls where required**
    - Allow lists to ensure certain tokens are held only by eligible investors.
    - Blacklists where law requires restriction of particular addresses, managed by regulated intermediaries.
- **Governance change constraints**
    - Rules that prevent sudden changes to voting rights or major protections without supermajority approval and notice.

These patterns do not guarantee outcomes, but they reduce the risk of sudden, opaque changes.

12.6 Donor safeguards

nonprofit and donor protections include:

- **Allocation rules as code**
    - Program versus overhead ratios and board compensation limits are enforced by contracts.
    - Any change is visible and requires board approval.
- **Restricted and designated funds**
    - Donations can be tagged for specific programs.
    - Contracts enforce that these funds are spent only for that purpose unless donors explicitly consent to changes.
- **Transparent cross NGO flows**
    - When funds move between NGOs, those flows are visible.
    - Double counting and layering are easier to detect.

These safeguards help donors and grant makers make more informed decisions but still require human judgment and oversight.

12.7 Roles in capital and donation flows

Roles are clearly separated:

- **dCorps protocol**
    - Supplies primitives and state.
    - Does not intermediate capital or donations.
- **Entities (corporations and NGOs)**
    - Decide how and where to raise funds or accept donations.
    - Bear legal and fiduciary responsibility.
- **Platforms and venues**
    - Exchanges, issuance platforms, and donation portals.

- Provide interfaces, KYC, compliance checks, and matching.
- Are regulated or unregulated according to their own jurisdictions and activities.

This structure is essential to keep dCorps as neutral infrastructure rather than an all in one financial platform.

---

13. Protocol governance

13.1 Governance goals

Protocol governance aims to:

- Maintain the security and reliability of the Hub.
- Keep the base layer neutral and predictable for entities and jurisdictions.
- Allocate shared resources, such as Treasury and foundation funds, responsibly.
- Allow evolution when needed, while limiting arbitrary or short term changes.
- Keep the Hub core minimal, while evolving protocol modules and ecosystem tools as real world needs change.

13.2 Governance actors

Actors in protocol governance include:

- **DCHUB holders**
  - Participate in voting on protocol level proposals.
  - Provide signaling on app and module registry entries.
- **Rollup operators and infrastructure providers**
  - Operate sequencing, batch posting, and RPC/indexing under governance-defined operator policies.
  - Provide liveness and incident transparency; they do not provide staking-based consensus security in Orbit v1.
- **Protocol Council** (once formed)
  - A group of technical and ecosystem experts.
  - Reviews and recommends on protocol upgrades, module approvals, and registry policies.
  - Includes a reserved Founding Steward seat for an initial period, described in section 13.2A.
- **Founding Steward**

    - The individual recognized in protocol documents as the founding steward of dCorps.
    - Holds the reserved Protocol Council seat for an initial period and participates in governance through that role.
- **Development corporation**
    - Implements code and supports early governance processes.
    - Has no special rights over protocol state beyond what is defined in governance charters.
- **dCorps foundation** (once established)
    - Administers parts of the Treasury and grant programs.
    - Acts as legal steward for certain assets and contracts.
    - Coordinates development and maintenance of protocol modules, especially jurisdiction adapter modules and sector frameworks.
    - Operates or delegates operation of the app and module registry according to governance instructions.
    - Advocates for keeping the Hub core minimal and stable.
- **Security and Continuity Council (Guardians)** (if adopted)
    - A limited, time bound council with a narrow mandate: respond to critical security incidents and prevent clearly harmful protocol changes during defined emergency windows.
    - Actions are limited to temporary pause or parameter freeze within predefined bounds, plus publishing signed incident statements.
    - Any use of these powers must be followed by on-chain ratification within a short window, otherwise the action automatically expires.
    - Guardians have no authority to move entity funds, edit cap tables, or override entity governance outcomes.

13.2A Founding Steward seat on the Protocol Council (continuity, accountability, and sunset)

To support continuity and accountable delivery during the formative years, one seat on the Protocol Council is reserved for the founding steward of dCorps for a limited initial period, under explicit constraints. The purpose is to reduce early coordination risk and hostile takeover risk while the ecosystem is still fragile, not to create permanent control. All meaningful authority remains with on-chain governance, and any privileged operational powers are time bounded and contestable.

13.2A.1 Scope and limits

- The Founding Steward seat is a single Council seat with one vote, no unilateral veto power, and no direct control over protocol state.

- Council membership confers no special access to entity level funds, cap tables, role assignments, or entity governance outcomes.
- The Council's authority is advisory except for narrowly delegated functions explicitly approved by DCHUB governance.

13.2A.2 Duration, reaffirmation, and conversion

- The reserved seat exists for three years after mainnet launch.
- An annual reaffirmation vote is held each year during this period.
- Failure to reaffirm converts the seat immediately into a standard rotating Council seat.
- At the end of the three year period, the reserved seat automatically converts into a standard rotating Council seat.

13.2A.3 Removal standards and thresholds (baseline)

Early removal during the initial period is permitted only for:

- Permanent incapacity,
- Explicit resignation, or
- Proven fraud, serious misconduct, or material breach of Council duty related to the protocol.

A successful early removal proposal requires, at minimum:

- Quorum: 20 percent of voting power participating, and
- Passing threshold: 67 percent yes of participating voting power.

Final numerical thresholds, if adjusted, are documented in the Governance Charter.

13.2A.4 Conflict and affiliation disclosure

All Council members, including the Founding Steward, must publish and maintain:

- Material affiliations with entities, applicants, modules, and service providers,
- A conflict of interest policy statement, and
- Recusal commitments for votes where direct commercial benefit exists.

13.3 Governance phases (explicit emergency sunset and protected changes)

Governance evolves through phases, with explicit constraints and automatic sunsets for emergency powers.

13.3.1 Phase 0: Early emergency and upgrade period (time bounded)

A publicly disclosed multisig may hold narrowly scoped emergency and upgrade powers while the network stabilizes.

Permitted actions

- Rapid security patches and bug fixes for Hub consensus and core modules.
- Emergency halts or parameter freezes within predefined bounds to prevent active exploitation.

Mandatory expiry and ratification (recommended)

- Any emergency halt or parameter freeze must automatically expire after a short window (for example 24 to 72 hours) unless extended or ratified through an on-chain vote.
- Emergency actions must publish a structured incident record with scope, affected modules, and a remediation plan.

Hard prohibitions

- No movement of entity funds by this multisig.
- No cap table edits, role reassignment, or entity governance outcome overrides.
- No silent changes; actions are executed on-chain and accompanied by public incident reporting.

Automatic sunset

Phase 0 emergency powers expire at the earliest of:

- 12 months after mainnet launch, or
- The first successful upgrade executed purely through standard on-chain governance, or
- A governance proposal that explicitly retires emergency powers sooner.

After expiry, the emergency mechanism is disabled at the protocol level and cannot be re enabled except through a Protected Change process.

13.3.2 Phase 1: Hybrid governance with Council

- A Protocol Council is formed through governance approved selection and rotation rules.
- Protocol upgrades follow public specification, Council review, on-chain vote, and transparent execution.

13.3.3 Phase 2: Full on-chain governance (post Phase 0)

- All protocol upgrades and parameter changes are executed through standard on-chain governance processes.
- The Council operates under charters approved by governance and provides public review, risk analysis, and recommendations.
- Any remaining privileged operational roles are limited to non consensus tasks (for example, reference interface operations) and remain contestable through governance.

13.3.4 Phase 3: Mature governance

- Governance authority rests primarily with DCHUB holder voting.
- The Council operates under charters approved by governance.
- Any remaining emergency mechanisms are narrow, transparent, contestable, and time bounded.

13.3.5 Protected changes (higher thresholds)

Certain actions are designated as Protected Changes and require higher thresholds, including:

- Re-enabling emergency powers after the Phase 0 sunset.
- Changing issuer registry governance rules in ways that reduce transparency or due process.
- Changing hard limits related to non custodial boundaries and censorship resistance commitments.
- Material changes to rollup settlement and bridge gateway assumptions or to the scope of chain-owner powers.

A Protected Change requires, at minimum:

- Quorum: 25 percent of voting power participating, and
- Passing threshold: 67 percent yes of participating voting power.

In addition, Protected Changes are expected to include two safety mechanisms:

- **Token age / lock requirement (anti raid)** Only voting power that satisfies a parameterized minimum holding period or governance lock age is counted toward quorum and voting power for Protected Changes. For Protected Changes, quorum and vote weighting are computed using only voting power that satisfies this age requirement.
- **Execution timelock** After a Protected Change passes, execution occurs only after a parameterized delay unless a narrowly defined emergency path applies under the governance charter.

Final numerical thresholds, voting-power age parameters, and timelock durations, if adjusted, are documented in the Governance Charter.

13.4 Scope and hard limits

Governance can:

- Adjust parameters such as:
    - Release schedules for governance-controlled allocations within limits.
    - Fee schedules for protocol services within policy bounds.
    - Sequencer, batch posting, and bridge gateway parameters within bounded change rules.
    - Criteria for official module status and app registry categories.
- Approve or remove:
    - Official jurisdiction adapter modules.
    - Sector frameworks that want recognition.
    - Policies for the app and module registry.
- Allocate:
    - Treasury and foundation funds for programs and grants.

Governance cannot, where technically enforceable:

- Arbitrarily seize individual user funds outside of predefined penalty modules.
- Silently rewrite historical ownership records.
- Unilaterally override clear legal prohibitions for entities or participants.

Hard limits will be encoded wherever possible. Where not possible, they will be expressed in governance charters and legal documents.

13.5 On-chain governance processes

Governance processes follow defined steps:

1. **Proposal drafting**

- Proposers publish detailed rationales and specifications.
- Early discussion takes place in public forums.

2. **Council review**

    - The Protocol Council assesses technical and risk implications.
    - Provides a recommendation and any suggested changes.

3. **On-chain vote**

    - DCHUB holders vote within a defined window.
    - Quorum and majority thresholds are defined in governance parameters.

4. **Execution**

    - If approved, changes are executed through upgrade handlers or Treasury transactions.
    - Results and changes are documented and anchored.

Smaller parameter changes may use simpler paths. Large structural changes follow longer and more cautious processes.

Registry changes, such as marking a module as official or flagging an app as unsafe, follow similar patterns but may use lighter processes, for example shorter voting windows or delegated authority to the Council with veto rights for token holders.

13.6 Execution by legal entities

Some governance decisions require legal execution, for example:

- Signing contracts with vendors or auditors.
- Setting up and managing bank accounts for the foundation.
- Hiring staff or entering commercial agreements.

These acts are carried out by:

- The development corporation.
- The foundation.
- Other organizations, as appropriate.

Their charters and internal governance documents define how they reflect on-chain decisions in off-chain acts. Where legal constraints conflict with on-chain instructions, those conflicts must be disclosed and resolved case by case.

13.7 Grants, ecosystem funding, and Treasury policy

Governance sets the principles for:

- **Treasury use**
    - What categories of spending are allowed.
    - How much should be kept in reserves.
    - Risk constraints on asset holdings and liquidity provision.
- **Grant programs**
    - Focus areas such as:
        - Core tooling and infrastructure.
        - jurisdiction adapter modules and sector frameworks.
        - NGO tooling and impact reporting.
        - Security, audits, and monitoring.
    - Processes for:
        - Proposals and evaluation.
        - Milestones and reporting.
        - Revocation or adjustment when necessary.

The foundation is expected to administer these programs within the bounds of governance set policy, and to publish regular reports on grants, results, and remaining reserves.

Treasury and grants policy will be documented in a dedicated policy that can evolve through governance, subject to legal requirements and the foundation’s charter.

---

14. Ecosystem, developer tools, and external modules

14.1 Developer tools and builder experience

Developers interact with dCorps through:

- **EVM JSON-RPC endpoints**
    - For sending transactions and reading contract state (`eth_call`).
    - For reading protocol event logs for registry, governance, and accounting streams.
- **Contract ABIs and typed clients**
    - For integrating the registry, entity templates, and modules from standard web3 stacks.
    - For generating TypeScript clients and building apps with libraries such as `ethers` or `viem`.
- **Indexing and data exports**
    - Event-based indexing for derived views and time-window reports.
    - Reference exports for cash-based operating and allocation statements.
- **Command line tools and scripts**
    - For local devnets, deployments, and integration tests.

A developer portal will provide:

- Quickstart guides for common flows.
- Example applications.
- ABI and event schema references.

The goal is that a competent developer familiar with web APIs can build proof of concept integrations in days, not months.

The foundation and development corporation are expected to contribute reference implementations and tooling, but the ecosystem remains open to any developer.

14.1A Module execution environments (v1 stance)

Protocol modules are implemented as EVM smart contracts and registered in the module registry.

v1 distinguishes two practical execution categories:

- **Kernel contracts (system contracts)**: conservative, high-assurance contracts for the core registry, entity templates, governance primitives, accounting primitives, and document anchoring.
- **Module contracts (EVM)**: optional contracts for jurisdiction adapters, sector frameworks, attestations, and other extensions, deployed and upgraded under explicit governance and security policy.

Modules must declare:

- The schema versions they read and write,
- Their upgrade policy and compatibility commitments,
- Their accepted issuer inputs where applicable, and
- Their anchoring and reporting outputs where applicable.

14.1B Minimal v1 protocol surface (contracts, events, and reads)

The protocol defines a minimal, stable surface area that applications and tools can rely on. Exact ABIs and function names are defined in developer specifications, but the v1 surface includes the following functional families.

14.1B.1 Core contract families

- **Entity registry** Entity creation, metadata updates, status updates, and namespace leases and renewals.
- **Roles and governance** Role binding and reassignment, proposal submission, voting, execution of resolutions, and document and evidence anchoring.
- **Hub corporation** Unit issuance and cancellation, transfers, restrictions, and corporate actions.
- **Hub nonprofit** Board seat management, allocation rule changes, and program wallet management.
- **Accounting primitives** Standardized accounting event emission for relevant flows and tag schema compliance.
- **Module registry and attachments** Module registration and metadata, attaching and detaching modules, and querying module attachment status.

Attestations, assurance, and reputation are not required parts of the v1 core surface. They are provided by optional protocol modules that define their own interfaces and schemas.

14.1B.2 Required event streams

Applications and indexers rely on stable events, including:

- Registry events: entity created, updated, and status changed.
- Role events: role bound and unbound.
- Governance events: proposal submitted, vote cast, resolution executed.
- Corporation events: units issued, transferred, and cancelled.
- nonprofit events: donation received, spending recorded with category codes, allocation rule changed.
- Document and evidence events: document anchored, evidence bundle anchored.
- Module registry events: module registered or updated, module attached, module detached.

Optional protocol modules may emit additional event streams, such as attestation published, disputed, or superseded, or reputation score updated, according to their own standards.

14.1B.3 Canonical read paths

The Hub exposes canonical read paths through:

- On-chain view calls (`eth_call`) for registry lookup, entity structural state, governance state, module metadata, and module attachment status.
- Event logs for chronological state transitions and accounting/event streams.
- Reference indexers that transform logs into derived views and time-window exports (see section 9.5B and section 9.5C).

Optional protocol modules may expose additional view calls and derived indexer outputs, for example attestations and disputes by entity and by issuer.

14.1C Schema versioning and extensibility rules

To support long term compatibility and safe evolution, dCorps uses explicit schema versioning rules.

14.1C.1 Versioned schemas

- Every structured object type includes a `schema_version` field.

- Schema versions are monotonic and backward compatible unless a major version bump is approved through governance.

14.1C.2 Tag schema rules

- A baseline required tag set exists for accounting and reporting compatibility, including `category_code`, `counterparty_type`, and a `reference_id` or anchor pointer where applicable. `reference_type` is recommended when `reference_id` is present.
- Entities may add custom tags, but custom tags should be namespaced to avoid collisions.

14.1C.3 Chart of accounts extensibility

- The protocol maintains a minimal default chart of accounts as a global reference.
- Entities may extend the chart through entity scoped namespaces.
- For comparability, extended categories should map to a parent category in the minimal schema unless an entity explicitly opts out of comparability through its disclosure mode and reporting policy.

14.1C.4 Deprecation policy

- Deprecations are announced through governance with replacement schemas, migration guidance, and a defined support window.
- Official tools and reference indexers support both old and new schemas throughout the support window.

14.2 App and module registry (dCorps App Store)

An **app and module registry** catalogs:

- Applications that integrate with dCorps.
- jurisdiction adapter modules.
- Sector frameworks.
- Tools such as explorers and analytics platforms.
- Optional attestation, assurance, and reputation modules.

The registry is presented as a kind of **dCorps App Store**:

- Any developer who meets minimal technical and legal requirements, defined in the Registry and Module Policy, can submit an app or module for listing.
- Listings include metadata such as:
    - Category and use case.
    - Links to source code, where available.

    - Audit status.
    - Usage metrics and age.

The registry is:

- Open to publication by developers.
- Not a guarantee of security or compliance.

Instead, the registry surfaces **signals**, for example:

- Whether code has been audited and by whom.
- Usage levels.
- Age and update frequency.
- Voluntary reviews or ratings.
- Governance status, for example:
    - **Official module** for jurisdiction adapter or sector modules that have been approved through DCHUB governance.
    - **Recommended** for apps and tools that are widely used and reviewed.
    - **Experimental** for early stage projects.
    - **Flagged** for entries that governance or the Council consider clearly malicious, insecure, or seriously misleading.

The **dCorps foundation**:

- Operates or oversees operation of a reference registry and reference interfaces.
- Maintains and publishes reference standards for listings, metadata, and signals.
- Stewards the protocol module layer (module standards, interfaces, security requirements, and upgrade policies), in this document referred to as the **Module Protocol**.
- Develops and maintains a set of **official protocol modules**, especially jurisdiction adapter modules and sector frameworks.
    - Official modules may be implemented by the foundation directly or by third-party teams funded or commissioned by the foundation.
    - In all cases, stewardship, versioning, and upgrade responsibility for official modules sits with the foundation under protocol governance oversight.
- Implements governance decisions in the reference registry.

Importantly, the registry is a **discovery and signaling layer**, not a hard gate:

- Apps and modules can exist and be used even if they are not listed, or if they are listed as experimental or flagged.
- Multiple registries and explorers can coexist. Any team can publish a registry view and any user can choose which registry and interface they trust.

- Users and entities remain responsible for their own due diligence and for any legal or regulatory obligations that apply to their use of specific apps or modules.

Emergency actions at the registry level:

- Affect how entries are displayed in reference interfaces and default lists.
- Do **not** delete or disable the underlying smart contracts or code.
- Are subject to later review through transparent governance, with clear records of:
    - Why the action was taken.
    - Who requested or executed it.
    - How and when it will be revisited.

14.2A Registry due process and safety actions (reference standard)

The app and module registry is a discovery and signaling layer. Registry labels affect how entries are presented in reference interfaces and default lists. They do not disable underlying code, do not prevent deployment, and do not prevent users from choosing to interact with an entry.

To avoid arbitrary or politicized labeling, the reference registry follows a due process standard for safety actions.

14.2A.1 Label taxonomy (reference)

Registry entries may carry one or more labels:

- **Official** Approved through DCHUB governance for protocol module standards and compatibility expectations.
- **Recommended** A positive signal based on usage maturity and objective transparency signals (for example audits, open source, time in production), using criteria defined in Registry and Module Policy.
- **Experimental** Early stage, limited assurance, or rapidly changing entries.
- **Quarantined** A time bounded, emergency warning label used when there is credible evidence of active exploitation or imminent harm.
- **Flagged** A persistent warning label used when there is credible evidence of serious risk, malicious behavior, or materially misleading claims.

Labels are signals, not judgments of legal compliance. Listing and labeling do not certify that an app, module, or provider is compliant, safe, or appropriate.

14.2A.2 Eligible initiators (who can trigger a review)

A registry safety review can be initiated by:

- Any DCHUB holder through an on-chain proposal.
- The Protocol Council through a publicly logged request.
- The foundation (or delegated registry operator) through a publicly logged request, limited to security and misinformation concerns that meet the evidence threshold below.
- A security reporter submitting a disclosure package through a published security reporting process, where the reporter's claims are anchored or otherwise made verifiable.

Initiation does not determine outcome. It triggers a review record.

14.2A.3 Evidence requirements and record format

A safety review request must include a structured record with:

- Entry identifiers (name, version, chain addresses, repository references where applicable).
- One or more enumerated reason codes, such as:
    - Critical exploit or active drain risk
    - Malware, phishing, key theft, or impersonation
    - Material misrepresentation of protocol facts (for example claiming endorsement, recognition, or custody guarantees that do not exist)
    - Repeat security negligence without remediation
- Evidence anchors:
    - audit reports, incident reports, exploit proofs, reproducible demonstrations, or other verifiable artifacts
- A conflict and affiliation disclosure by the initiator, including whether the initiator is a competitor or has a direct commercial stake.

Reference registries must publish these records in a consistent, indexable format.

14.2A.4 Emergency quarantine (time bounded)

Quarantine is used only for imminent harm scenarios.

- A Quarantined label can be applied immediately when credible evidence indicates active exploitation, imminent asset loss, or high probability of user harm.
- The Quarantined label must include:
    - a reason code,
    - a short factual rationale, and
    - evidence anchors (or a responsible disclosure record with a defined disclosure schedule).
- Quarantine is time bounded:

- It automatically expires after a short window (recommended 72 hours) unless extended by an on-chain governance action.
- Extensions must include updated evidence anchors and a clear scope statement.

Quarantine affects presentation in reference interfaces. It does not disable the underlying code.

14.2A.5 Standard Flagged process (notice, response, decision)

A Flagged label requires a standard process:

1. **Notice** The registry publishes a notice to the listed contact channels for the entry maintainer (or issuer), and anchors the notice record.
2. **Response window** The maintainer can publish a response record with evidence anchors, correction actions, and version plans.
3. **Review** The Protocol Council may publish an assessment record with explicit reasoning and recusal statements.
4. **Decision** Final application or removal of the Flagged label is executed through an on-chain governance action or through a delegated registry authority explicitly authorized by governance for narrow categories.

14.2A.6 Appeals, reinstatement, and label expiry

To reduce permanent, stale warnings and to keep registry status responsive to remediation:

- A maintainer may appeal a Flagged label by publishing:
  - remediation evidence,
  - a versioned fix description,
  - audit or verification anchors where applicable.
- Reference registry policy may require periodic renewal of Flagged status (for example every 90 days) through an explicit reaffirmation record. If not reaffirmed, the label expires and reverts to Experimental, unless a new review is initiated.

14.2A.7 Neutrality and non exclusivity

Due process is about interface integrity and user safety, not control:

- Any developer can publish an alternative registry.
- Any user can choose a different interface.
- The Hub remains neutral and does not enforce registry labels as transaction censorship.

14.3 Jurisdiction adapter modules (optional overlays)

Jurisdiction adapter modules are optional overlays. They are protocol modules that attach to the Hub, not separate execution layers.

They are not part of the kernel and are not required for v1 operation or for the primary adoption path. dCorps is designed to be complete inside the on-chain economy without them.

Adapters exist only to map Hub truth into external legal or institutional processes when an entity chooses to interact with them. Many entities will never attach any jurisdiction adapter.

Early pilots, if any, are expected to focus on a small number of DAO-friendly jurisdictions and clearly disclosed delegated operators, coordinated through the dCorps foundation. None of this is a protocol dependency.

Jurisdiction adapter modules implement the Module Protocol standards and can be developed by:

- The **dCorps foundation** as official modules (or funded and commissioned by the foundation as official modules), and
- Jurisdictions, service providers, and third-party builders as independent modules.

They are frameworks where:

- Jurisdictions encode:
    - Which entity types they recognize and under what conditions.
    - Required disclosures and reporting intervals.
    - Fee and tax related obligations, including how fees are collected in USDC.
    - Eligibility rules for jurisdiction-scoped settlement instruments (for example a local stablecoin, tokenized deposits, or a CBDC), including any delegated operator or gateway requirements.
- Entities opt in by:
    - Attaching to the module on-chain.
    - Accepting its terms off-chain where needed, for example by signing local incorporation or registration documents.

These modules:

- Run as EVM contracts on the Hub rollup (and may optionally rely on off-chain services that publish signed attestations to the Hub under explicit rules).
- Read entity state directly from the Hub:
    - Ownership and cap table snapshots.

    - Governance events and key resolutions.
    - High-level financial aggregates and allocation metrics.
- Write additional state such as:
    - Recognition status (where applicable).
    - Eligible settlement rails or assets (where applicable), with expiry windows and required disclosures.
    - Fee schedules and obligations.
    - Compliance signals or alerts.

Jurisdiction recognition is controlled by the jurisdiction adapter's keys and rules, not by the dCorps DevCo corporation, rollup operators, or any reference interface. For official modules, the design intention is that operational recognition keys are held by the relevant jurisdiction or its explicitly disclosed delegated operator, not by the foundation, except where a jurisdiction explicitly delegates key custody under a publicly disclosed arrangement.

- A jurisdiction adapter can withdraw recognition according to its published rules.
- The Hub records this as module state and history.
- If an entity disputes a recognition decision, recourse is in that jurisdiction's legal system and processes.
- If a court or authority compels a correction, the jurisdiction or its delegated operator is the party that updates the module state.

Because direct jurisdiction integration often takes time, dCorps is designed to support a clear transition phase after mainnet launch: a temporary bridge that can provide immediate operational capacity and evidence for jurisdictions, while preserving the end goal of full, direct jurisdiction integration.

14.3.1 Adoption path (phased, recommended)

Jurisdiction adoption, if it happens, is expected to happen in phases. The optional path below exists for entities that choose to pursue external recognition. It is not required for digital-only operation, and it must never become an implied protocol milestone.

The intent is to document a realistic, politically complex road and to keep interface labeling honest about what is and is not recognized.

14.3.1.1 Pilot Step 0 (temporary): Delegated filing provider bridge (post mainnet)

Purpose

Pilot Step 0 is a temporary bridge between mainnet launch and direct jurisdiction integration. It exists to:

- Enable real formations, renewals, and registry updates now, even before a jurisdiction operated module is live.
- Generate auditable lifecycle data and operational evidence that supports and accelerates jurisdiction adoption.
- Standardize how off-chain legal actions are reflected on-chain, without implying recognition by the Hub itself.

How it works

- Providers (corporate service providers, registrars, law firms, or other operators) perform off-chain filings, renewals, and registry interactions under their own legal responsibilities and direct contracts with the entity.
- The provider then publishes standardized on-chain attestations that reference:
    - The subject entity ID,
    - The jurisdiction and registry context,
    - The filing type (formation, renewal, amendment, dissolution, other),
    - Evidence anchors (receipts, confirmations, or document hashes),
    - Validity windows and status (submitted, accepted, rejected, superseded, withdrawn, disputed), and
    - A required disclosure marker stating that this is a Step 0 provider attestation and not jurisdiction integrated recognition.
- Reference interfaces label the entity status clearly as **Filed via Provider (Pilot)** (or equivalent), distinct from any “jurisdiction integrated” or “recognition active” label.

Provider listing, weighting, and non endorsement stance

Step 0 may maintain an optional issuer registry used for discovery and default interface weighting only.

- “Listed provider” status means that the provider has published required identity metadata and agrees to the module’s disclosure, correction, and dispute signaling standards.
- “Listed provider” status is not legal authorization, not jurisdiction approval, and not a guarantee of compliance or licensing.
- The foundation and reference registry do not recommend or endorse a provider as “approved.” Multiple providers can coexist, and entities remain responsible for due diligence and for selecting their own providers.

Provider conduct rules for default weighting (reference policy)

To be eligible for default weighting and “Listed provider” signals in reference interfaces, a provider must:

- Publish identity and contact metadata sufficient for accountability, including:

    - provider name and jurisdiction of operation,
    - public contact channels,
    - the signing DID or keys used for attestations, and
    - a published key rotation and compromise response plan.
- Publish a correction and dispute process:
    - how errors are corrected (withdrawal and supersession),
    - how entity disputes are received and addressed,
    - expected response windows, and
    - how final outcomes are recorded on-chain.
- Follow strict marketing and labeling constraints:
    - must not market Step 0 attestations as legal recognition or as an official jurisdiction integrated status,
    - must not present any “dCorps listed” signal as a government endorsement or regulatory approval,
    - must include clear disclosure language in client facing materials that Step 0 is an evidence bridge and not automatic legal personhood.

Violations of these conduct rules result in:

- removal from default weighting and “Listed provider” signals in reference interfaces, and
- application of a warning label for the provider entry in the registry under the due process rules in section 14.2A.

This does not prevent the provider from publishing attestations. It changes how reference interfaces present their signals.

Dispute, correction, and supersession

Step 0 uses the dispute and correction signaling standard defined in section 6.2A.4. The Hub records the signed dispute and correction statements, but it does not adjudicate disputes.

Step 0 adds the following interface requirements for provider attestations:

- Reference interfaces must display:
    - issuer identity and registry status,
    - evidence anchors,
    - validity windows,
    - disputed or superseded flags, and
    - the most recent active attestation for the same filing context where applicable.

Foundation role in Step 0

The foundation's role is stewardship and standardization, not legal authority and not incorporation as a service:

- Operate or oversee the reference Step 0 module and publish the attestation schemas, dispute signaling standards, and interface labeling requirements.
- Maintain an optional issuer registry used for default interface weighting and discovery only. It is not a gate and does not prevent any provider from publishing attestations on-chain.
- Ensure the bridge is non exclusive and supports multiple providers.
- Ensure entities can switch providers, and that switching is reflected transparently on-chain.

The foundation does not grant legal status and does not act as the registry. Any legal effect comes from jurisdiction law and from the provider's filings and contracts, not from the Hub.

Boundaries

- Step 0 is not jurisdiction recognition and is not presented as such.
- Step 0 does not make the foundation a wrapper provider, agent, or coordinator for entity formation.
- Providers may be regulated or licensed depending on their jurisdiction; compliance is their responsibility and the entity's responsibility.
- Step 0 does not require custody of entity funds by the foundation. Any provider fees are handled directly between the entity and the provider under their own arrangement.

14.3.1.2 Pilot Step 1: Disclosure, fee, and reporting module

- The module defines eligibility, required disclosures (via anchored documents), reporting cadence, and fee collection in USDC.
- The module produces clear, machine readable status outputs that regulators and counterparties can observe.
- Legal effect may remain partially off-chain in this step, but the jurisdiction gains an operational dashboard and an auditable fee and reporting pipeline.
- This step can be operated by a jurisdiction directly or by an explicitly disclosed delegated operator under the jurisdiction's supervision, depending on the jurisdiction's readiness.

14.3.1.3 Pilot Step 2 (target end state): Direct jurisdiction integration and recognition binding

When local law, processes, and operational readiness support it, the jurisdiction binds legal recognition and registry actions to module state:

- The jurisdiction (or an explicitly disclosed delegated operator under its supervision) controls the operational keys for recognition and registry status actions.

- The module writes recognition active or withdrawn as on-chain status, with the legal effect arising from local law and contracts that reference that status.
- Registry interactions become directly integrated, meaning the jurisdiction recognizes the on-chain module state as part of the official process, rather than relying on third-party attestations as the primary bridge.

This is the intended final state for a mature jurisdiction integration.

14.3.1.4 Graduation, sunset, and interface labeling (reference standard)

To make the temporary nature of Step 0 unambiguous, the ecosystem follows explicit graduation rules.

Graduation conditions (Step 0 deprecation triggers)

Step 0 is deprecated for a jurisdiction when any of the following becomes true:

- A jurisdiction operated or jurisdiction supervised jurisdiction adapter module (Step 1 or Step 2) is live and available for that jurisdiction, or
- A formal arrangement (MoU, delegation contract, or equivalent) exists that binds registry actions to the jurisdiction adapter under jurisdiction controlled keys, or
- The jurisdiction publishes an official integration path and begins onboarding entities through direct module attachment.

Interface labels (required for reference explorers)

Reference explorers and dashboards must clearly distinguish:

- **Filed via Provider (Pilot)** (Step 0)
- **jurisdiction adapter Attached (Reporting and Fees)** (Step 1)
- **Jurisdiction Integrated (Recognition Active)** (Step 2, where applicable)
- **Recognition Withdrawn** (module output, where applicable)

Interfaces must never present Step 0 as recognition. They must also show issuer identity, evidence anchors, validity windows, and dispute status for provider attestations.

Non exclusivity and portability (required)

- Multiple providers can exist per jurisdiction and per entity type.
- Entities must be able to switch providers.
- Switching providers does not erase history; it changes the active provider relationship going forward, with a recorded timeline and evidence anchors.

This phased approach allows jurisdictions to adopt dCorps without requiring an all or nothing jump from day one, while producing practical value early. The success and legal effect of any jurisdiction adapter module depends on the **law and practice** of the relevant jurisdiction, not on this whitepaper or on-chain mechanics.

Modules can:

- Automatically collect fees in USDC from entities that opt in, splitting revenues between jurisdiction wallets, the Protocol Treasury, and possibly other public purpose sinks.
- Provide structured data feeds and dashboards for regulators, corporate registrars, and tax administrations.
- Serve as technical hooks for local compliance interfaces, such as filing portals and automated reporting pipelines.

The **dCorps foundation** has a specific mission around these modules. It is expected to:

- Design and steward the Module Protocol standards that jurisdiction adapter modules follow.
- Research and co design **official** jurisdiction adapter modules in collaboration with local experts, regulators, and service providers.
- Propose official modules to the community and governance for review and formal approval.
- Maintain and update official modules as laws and regulations change, including funding third-party teams to implement updates where appropriate.
- Support a broader ecosystem where third-party jurisdiction adapter modules can be built, listed, audited, and adopted by entities, even when they are not official.

14.4 Sector and impact frameworks

Sector frameworks are:

- Protocol modules focused on domains such as:
    - Climate.
    - Education.
    - Public health.
    - Other thematic areas where impact and standards matter.

They define:

- Standard metrics and indicators.

- Reporting intervals.
- Eligibility criteria for participation in certain funding or recognition programs.

Entities can adopt these frameworks to:

- Signal adherence to sector norms.
- Access specialized funding or partner networks.
- Provide more meaningful reporting to donors and investors.

The dCorps foundation is expected to:

- Work with sector experts, NGOs, and funders to design reference frameworks.
- Sponsor pilots and experiments.
- Maintain a set of official or recommended frameworks through governance.

Frameworks are not laws. They are shared standards for their domains. Adoption is voluntary, though some funders or platforms may require participation as a condition for support.

14.5 External issuance and fundraising platforms

Issuance and fundraising platforms:

- Build on dCorps entity state and modules.
- Handle user onboarding, KYC, and marketing.
- Implement offering structures and payment flows.

They remain independent entities with:

- Their own legal and regulatory obligations.
- Their own risk models.
- Their own users and governance.

dCorps provides them with:

- Reliable entity data and governance state.
- Hooks for allocation of tokens, units, or dShares.
- Anchoring of offering documents and investor consents.

Platforms may be listed in the app registry along with their audit and compliance status, but listing is not an endorsement or guarantee.

14.6 DeFi integrations and indices

DeFi integrations and index providers:

- Use dCorps data to inform:
    - Collateral acceptability and parameters.
    - Inclusion criteria for indices and baskets.

They may rely on:

- Governance and cap table metrics.
- Revenue and allocation history.
- NGO transparency and allocation scores.
- Jurisdiction and sector framework participation.

They are not part of the core protocol and carry their own risk profiles. dCorps aims to make their work easier and safer by providing better data, not by controlling their design.

These integrations can also be listed in the registry, with DCHUB governance able to flag obvious abuses or misrepresentations.

14.7 Code and data licensing, forks, and compatibility

dCorps is intended to be an open ecosystem.

- **Core code**
    - Hub chain code, standard entity modules, and reference tooling are intended to be open source under licenses that support broad use and contribution.
    - Specific license choices and contribution policies will be published in the governance and repository documentation.
- **Modules and applications**
    - Third-party apps and modules may be open or closed source, but the registry surfaces transparency signals such as source availability and audit status.
- **Public data and privacy**
    - The protocol minimizes personal data on-chain by design. Public data focuses on entity level facts, governance records, and category level aggregates.
    - Controlled zones, encryption, and anchoring patterns are used for sensitive data and private disclosures.
- **Forks and compatibility**

- The code can be forked. Independent networks may exist and evolve under their own governance.
- The dCorps name, marks, and official registry branding are intended to be protected so that users can distinguish official networks and interfaces from forks.
- Compatibility standards, including schemas and interfaces for anchoring and modules, are published so that tools can interoperate across implementations where desired.

---

15. Security, privacy, and interoperability

15.1 Threat model and key risks

Key risks include:

- Software bugs in Hub or modules.
- Rollup operator outages, censorship, or misconfiguration (sequencer and batch poster).
- Exploits of smart contracts in the kernel, modules, or bridge gateways.
- Bridge and interoperability failures (cross-chain asset risk).
- Key compromises and poor operational security.
- Malicious or low-quality third-party apps.

dCorps does not claim to remove these risks. It aims to reduce them and to make them more manageable.

15.2 Secure development practices and audits

Core components are developed with:

- Code review and testing standards.
- Unit, integration, and property based tests.
- Fuzzing and adversarial testing.

Independent audits:

- Are conducted for:
    - Hub core modules.
    - Standard entity modules.
    - Official jurisdiction adapter and sector frameworks.

- Critical infrastructure around the app and module registry.

Audits are published with:

- Scope and findings.
- Remediation status, subject to responsible disclosure.

A bug bounty program will encourage researchers to report vulnerabilities within defined rules.

The foundation is expected to coordinate audits for official modules and core tooling, while other developers remain responsible for their own code.

15.3 Rollup operations security

Rollup operators (sequencer, batch poster, RPC, and reference infrastructure) are encouraged to:

- Use hardened infrastructure and hardware security modules where appropriate.
- Separate signing keys from general purpose machines.
- Maintain monitoring, alerting, and clear incident response procedures.
- Publish operational transparency commitments so downtime and abnormal behavior are visible.

dCorps can:

- Publish operator security guidance and minimum disclosure standards for reference infrastructure.
- Configure on-chain roles and timelocks to limit the impact of key compromise or operational mistakes.
- Progressively decentralize operations by expanding the set of independent operators over time.

It cannot enforce security practices by law; operators remain responsible for their setups.

15.4 Privacy tools and private zones

Privacy tools include:

- Private contract zones or chains that hold sensitive logic.
- Encryption and access control for controlled data.

- Zero knowledge proof systems for selected metrics.

These tools are used to:

- Protect personal data such as salaries and beneficiaries.
- Keep commercially sensitive terms confidential.
- Still allow proofs of compliance or performance.

Which zones and tools to use is up to entities and builders. Protocol modules and apps can specify which privacy tools they require or support.

15.5 Zero knowledge proof patterns

Zero knowledge proofs can be used to:

- Prove that nonprofits meet allocation rules without revealing every single payment.
- Prove that an entity maintains certain financial ratios or risk limits.
- Prove that specific jurisdiction rules have been followed.

dCorps is agnostic about specific proof systems, but it provides:

- Anchoring for proofs.
- Verification hooks in protocol modules.
- Ways to link proofs to entities and selected time windows.

Adoption of advanced zero knowledge techniques will grow over time as tooling matures.

15.6 Interoperability via Ethereum bridge gateways and selected bridges

dCorps is an Orbit rollup that settles to Ethereum. Its primary interoperability surface is the canonical Ethereum to/from dCorps bridge gateways.

- Stablecoins and other ERC-20 assets are bridged from Ethereum into canonical ERC-20 contracts on dCorps.
- Withdrawals back to Ethereum follow the rollup’s standard withdrawal path and timing.

dCorps may also integrate additional bridges or cross-chain messaging protocols over time, but they are treated as optional and high-risk components.

Bridges are high-risk components. The design intent is that:

- **Core entity accounting and governance on the Hub do not depend on any external bridge for correctness.** The canonical record of entities, Hub corporations, and NGOs lives on the dCorps rollup.
- Cross-chain assets are always used at your own risk and logically separate from core entity state. A bridge exploit can affect assets that have crossed that bridge, but it should not silently corrupt the basic records of who owns which units or dShares, how boards are composed, or how nonprofit flows are categorized.
- Preference is given to well audited, widely used bridges with transparent security models.
- Protocol modules and apps that rely on cross-chain data must clearly communicate their reliance and associated risks.

Entities and users ultimately choose their own risk exposure to external chains. dCorps aims to make that exposure explicit and optional, not hidden inside the base protocol.

15.7 Incident response and decentralization path

Incidents can and will happen. dCorps intends to handle them with:

- Minimum necessary intervention.
- No arbitrary confiscation or silent rewrites.
- Transparent communication of any emergency actions.
- Clear sunset and limitation on early emergency powers.

In early phases:

- More direct control is available to the core team for urgent issues.

Over time:

- Control is shifted to on-chain governance and broader stakeholders.
- Emergency powers are narrowed and eventually retired where possible.

The goal is a path toward robust decentralization, with realistic handling of the risks of young networks.

The foundation and Council are expected to play key roles in incident response, within boundaries set by governance.

15.8 Example failure scenarios and responses

dCorps assumes failures will occur and aims to make them visible, containable, and auditable.

15.8.1 Stablecoin disruption or issuer actions

Scenario: a stablecoin used by entities depegs, faces redemption issues, or enforces freezes or blacklists on specific addresses.

Response patterns:

- Asset registry governance can:
    - Limit or retire affected assets for protocol fees and default UI flows
    - Introduce warnings and risk labels in official interfaces, including explicit labels for freeze and redemption mechanics
    - Accelerate support for alternative approved settlement assets where safe and feasible
- Entities can:
    - Diversify treasury holdings across approved assets and venues
    - Use wallet segmentation and policy controls to limit operational exposure to any one issuer or rail
    - Maintain continuity plans for payroll, vendor payments, and NGO disbursements under stablecoin disruption scenarios

Issuer actions are external enforcement and policy decisions. The Hub cannot override them; remediation depends on issuer policy and, where relevant, the token’s administrative controls on its home chain.

15.8.2 Bridge exploit or cross-chain asset failure

Scenario: a bridge used for non core assets is exploited.

Response patterns:

- Core entity registry, cap tables, and governance state remain correct on the Hub.
- Affected bridged assets are treated as external risk; dashboards and tools surface exposure.
- Interfaces can label compromised assets and advise on containment without rewriting entity history.

15.8.3 Sequencer outage, batch posting failure, or censorship

Scenario: the sequencer is offline or degraded, the batch poster stalls, or transaction inclusion becomes unreliable.

Response patterns:

- Official interfaces surface liveness warnings and “pending bridge” states clearly.
- Operators rotate keys and restore service under a transparent incident process.
- Where supported by the rollup protocol, users can submit transactions via Ethereum as a fallback path (exact mechanics are chain-configuration dependent).
- Entities maintain continuity plans (for example emergency withdrawal procedures and ETH reserves for L1 actions).

15.8.4 Entity treasury key compromise

Scenario: an entity treasury is drained through compromised keys.

Response patterns:

- Recommended wallet separation and limits reduce blast radius.
- Role reassignment and wallet rotation can be executed through recorded governance procedures.
- Entities can use recovery committees, time delays, and external custody integrations to reduce the probability and impact of compromise.

15.8.5 Malicious or insecure third-party application

Scenario: an app listed in the registry is found to be malicious or critically insecure.

Response patterns:

- The registry can quarantine or flag the listing in official interfaces with public rationale.
- Governance can downgrade status signals and require security remediation for re listing.
- The underlying contracts or code remain outside protocol control; users retain choice and responsibility.

15.8.6 Governance capture attempt or chain-owner abuse

Scenario: a concentrated group attempts to push harmful parameter changes, or early administrative powers are misused.

Response patterns:

- High visibility governance processes, Council review, and timelocks reduce surprise changes.
- Protected Changes and explicit bounds reduce the blast radius of parameter changes.
- Optional guardian or veto mechanisms, if adopted, provide narrow time-bounded protection against clearly harmful proposals.

- Users and entities can treat governance and operator risk as a visible signal and adjust exposure, including exiting bridged assets back to Ethereum.

15.8.7 jurisdiction adapter module becomes invalid due to legal change

Scenario: a jurisdiction changes policy, withdraws support, or issues legal orders that affect a module.

Response patterns:

- Module status can be updated, including warnings and deprecation plans.
- Entities can detach from the module and attach to alternatives, with a recorded timeline.
- Records remain on-chain, preserving auditability of what was recognized and when.

---

16. Implementation status and roadmap

16.1 Current state

At the time of this master whitepaper revision (v1.3, December 21, 2025):

- Architecture and core designs are specified at a conceptual level.
- Prototype implementations of:
    - The entity registry.
    - Hub corporation and nonprofit modules.
    - Wallet and accounting primitives exist in internal or limited test environments.
- No production mainnet exists yet.
- No public token sale or listing has occurred.
- No foundation has been incorporated yet.

All details are subject to change based on engineering work, testing, legal analysis, and community feedback.

16.1A Public artifacts and verification checklist (v1 readiness signals)

dCorps is designed to be verifiable. The goal is that serious users can assess readiness without private access, informal assurances, or reliance on any single party.

Before mainnet v1 is treated as production ready, dCorps intends to publish a concrete set of artifacts and verification signals.

Code and specifications (public, versioned)

- Hub chain implementation source code and build instructions.
- Protocol Specification and module standards referenced in this whitepaper:
  - core contract interfaces, state machines, and event schemas (docs/spec/SPEC-CORE.md),
  - Module Protocol Standard and compatibility requirements (docs/spec/SPEC-MODULES.md),
  - Anchoring Standard and anchor schema versions (docs/spec/SPEC-ANCHOR.md).
- Reference tooling source code:
  - reference indexer behavior and export formats (docs/spec/SPEC-INDEXER.md),
  - reference explorer behavior for entity pages and derived views (docs/spec/SPEC-INDEXER.md),
  - Compatibility Test Suite for schema and module conformance (docs/spec/SPEC-CONFORMANCE-TESTS.md).

Testnet and reproducibility

- A public testnet with:
  - published chain ID, genesis file, rollup contract addresses, and node/operator setup steps,
  - tagged releases and reproducible build guidance,
  - published upgrade rehearsals for at least one major upgrade path.
- A deterministic way for third parties to:
  - run a node,
  - run the indexer,
  - reproduce entity views and derived outputs from raw chain data.
- A public example entity package on testnet that includes:
  - at least one corporation example and one nonprofit example,
  - a complete time window of tagged accounting events with a few evidence anchors,
  - the expected derived view export objects (cash-based operating statement and nonprofit allocation statement),
  - and a simple reproduction script or notebook that verifies the derived outputs against raw chain data.

Security and audits (public scope and remediation)

- Independent audits for:
  - core Hub modules,
  - entity modules (Hub corporation and Hub nonprofit),
  - critical reference tooling used for reporting and registry presentation.
- A bug bounty program with:
  - scope, rules, and disclosure workflow,
  - clear remediation commitments and public post mortems where appropriate.

Governance and operational policy artifacts

- Governance Charter and Operator Charter (or equivalent) describing:
  - governance processes, protected changes, thresholds, and timelocks,
  - operator expectations and objective accountability rules.
- Treasury Policy, including:
  - permitted uses and prohibitions,
  - reporting cadence,
  - liquidity bootstrap policy (see section 10.7A).

These artifacts are designed to turn “trust us” claims into observable behavior, while keeping the protocol neutral and open.

16.1B Team, contributors, and advisors (disclosure policy)

dCorps makes no claim that a whitepaper replaces execution. Team, governance, and security posture are therefore treated as first class public signals.

Current authorship and role

- **Founding Steward and Author**: Nicolas Turcotte, Founder

Disclosure commitments

To preserve credibility and reduce hidden conflicts, dCorps intends to maintain clear public disclosures for:

- The legal identity and officers of the development corporation once incorporated, including its jurisdiction, directors, and controlling governance documents to the extent appropriate.

- The legal identity and governance of the dCorps foundation once incorporated, including charter, board membership, and core policies, with finances and grants reported under published transparency commitments.
- Material advisors and service providers, when engaged in protocol critical roles, including:
    - auditors and security firms,
    - core infrastructure providers for reference tooling,
    - legal and regulatory counsel acting on protocol level structures, subject to confidentiality constraints that may apply in specific engagements.

Conflict of interest and recusal standard

- Protocol Council members and any delegated registry operators must publish affiliation and conflict disclosures.
- Where direct commercial benefit exists related to a module, app, provider, or grant decision, the relevant reviewer or decision maker is expected to recuse under the Governance Charter and registry policy.

This section is about clarity of accountability. It does not imply endorsement of any party and does not modify the protocol's neutral, non custodial boundaries.

16.2 Roadmap principles

The roadmap is guided by a small set of principles. These are not marketing statements, they are constraints on what ships and in what order.

1. **Kernel invariants first**

   Every protocol change must preserve the kernel invariants in section 4.0. If a feature requires external authority to be correct, it is not part of the kernel.

2. **Hub-first standardization**

   The v1 adoption path is one strong public container on the Hub. Most entities should never need anything else. Optional modules and applications exist to extend the Hub, not to replace it.

3. **Security and correctness before breadth**

   The Hub is long-lived shared infrastructure. Audits, clear upgrade processes, conformance tests, and operational monitoring come before adding new surface area.

4. **Adapters are optional and replaceable**

   Jurisdiction and institutional integration is an adapter layer. It must never become a protocol dependency or a hidden milestone in the entity lifecycle.

5. **Ecosystem enablement over vertical integration**

   The protocol should make it easy for independent teams to build dashboards, payroll tooling, donor portals, and reporting modules. The foundation provides standards, reference tooling, and test suites, not a monopoly product suite.

6. **Conservative upgrades, explicit versioning**

16.3 Phased rollout

This rollout plan is structured around one idea: **ship a stable Hub kernel first, then make it operationally complete, then add optional adapters.**

The phases below start at **mainnet launch** and continue through a definition of **fully operational** infrastructure.

Phase 1: Mainnet launch (Kernel v1)

Objective: launch a stable Hub that can host complete Hub corporations and Hub nonprofits as the default entity containers.

Key deliverables:

- Hub chain genesis/config, deployed rollup contracts on Ethereum, and runtime stability.
- DCHUB gas and protocol governance primitives (timelocked upgrades).
- Entity registry with IDs, types, metadata, and lifecycle status.
- Hub corporation module v1:
    - Ten thousand unit template
    - Role and approval based governance
    - Basic corporate actions (issuance, transfers, restrictions, pools and claims patterns)
- Hub nonprofit module v1:
    - Board governance
    - Donation and program wallet types
    - Allocation rules and restricted fund patterns
- Canonical wallet types and standardized accounting event schemas.
- Document anchoring and evidence timelines (hash anchoring of bylaws, minutes, audits, policies).

- Reference explorer and indexer for entity discovery and event timelines.

Exit criteria:

- Core modules audited and deployed with reproducible builds.
- Upgrade process and on-chain governance path tested on testnet and exercised in controlled mainnet upgrades.
- First real entities can register, operate, and produce reproducible reports using only the Hub.

Phase 2: Operational completeness and standards hardening

Objective: make the Hub reliable enough that external builders can treat it as infrastructure, not an experiment.

Key deliverables:

- Conformance test suite for entity modules, event schemas, and indexing compatibility.
- Stable APIs and SDKs for common operations (entity creation, role changes, approvals, reporting queries).
- Monitoring, alerting, and incident processes for rollup operators and core services.
- Security hardening: bug bounty, audit extensions, and formalized threat models for the kernel.
- Fee grants and UX primitives so entities can cover DCHUB protocol fees via stablecoin sponsorship while the chain still prices execution in DCHUB.

Exit criteria:

- Independent builders can integrate against published schemas and pass conformance tests.
- A first cohort of entities operates end to end on mainnet for real value flows.

Phase 3: Ecosystem bootstrapping and stablecoin rails

Objective: make the Hub easy to use for real organizations and easy to integrate for service providers.

Key deliverables:

- Ethereum bridge gateway stablecoin connectivity and standard treasury patterns for stablecoin native operations.
- App and module registry with clear metadata, versioning, and security posture disclosures.

- Reference templates for common entity setups (basic corporation, basic nonprofit, umbrella sponsorship pattern).
- Indexer redundancy and data availability patterns so the ecosystem is not dependent on a single hosted service.
- Reference governance UI and reporting UI to validate the end user experience and reduce integration friction for new tools.

Exit criteria:

- Multiple independent applications and service providers operate in production.
- Stablecoin based operations work reliably for real entities across a range of workflows (inflows, approvals, payouts, reporting).

Phase 4: Adapter layer and institutional legibility

Objective: enable optional external integration without making external systems a kernel dependency.

Key deliverables:

- Jurisdiction adapter framework (schemas, proofs, and reference workflows) so external recognition can be attached as an overlay.
- Reference patterns for regulated payment rails (including CBDC-style instruments where feasible), including gateway disclosure, eligibility signals, and audit-ready evidence anchoring.
- Institutional reporting modules that derive verifiable reports from standardized accounting events and document anchors.
- Attestation modules and selective disclosure patterns (where an entity can prove statements about its state without publishing everything).
- Nonprofit specific overlays such as donation receipt workflows and sponsorship frameworks, implemented as adapters and applications.

Exit criteria:

- At least one high quality adapter implementation demonstrates external recognition or institutional integration while leaving the kernel unchanged.
- Entities can choose to remain purely Hub-native or attach adapters based on their needs, without any concept of mandatory graduation.

Phase 5: Fully operational maturity

Objective: reach a state where the Hub is a long-lived, self-sustaining public utility for organizational infrastructure.

Fully operational means:

- The chain is stable under a sufficiently decentralized operator set, with minimized and transparent administrative powers.
- The standard is stable, versioned, and supported by conformance tests and multiple independent tooling stacks.
- Governance and upgrades are predictable and do not depend on a single team or company.
- The ecosystem has enough applications and service providers that real entities can operate without bespoke support.
- Economics are sustainable, meaning fees and emissions policies can support security and public goods without constant external subsidy.

Key deliverables:

- Progressive decentralization of governance participation and operator diversity.
- Multiple independent indexers and reference implementations for critical components (indexing, APIs, explorer tooling).
- Mature protocol operations: upgrade cadence, emergency procedures, and long-term maintenance policies.
- Sustainable foundation processes: standards stewardship, audits, grants, and ecosystem support aligned with kernel invariants.

Exit criteria:

- No single organization is required for the Hub to continue operating and evolving safely.
- Entity creation and operation are routine, with predictable costs and predictable semantics.

Optional future phase: Advanced execution environments and public instruments (only if justified)

Advanced privacy execution, private execution zones, and public instrument models are only explored if real adoption proves they are needed. Any such work must preserve the kernel invariants and must not become a requirement for ordinary Hub entity operation.

Dates and details will depend on progress, adoption, legal developments, and resources.

16.4 Key metrics

Important metrics include:

- **Usage and adoption**
    - Number and diversity of active entities.
    - Retained activity over time (cohort retention and reactivation).
    - Volume and nature of on-chain operations (typed workflows and tagged events, not only transfers).
    - Real-world use in programs and operations (donations and program spending for nonprofits; payroll and operating flows for corporations).
- **Security and decentralization**
    - Number and distribution of independent operators and full nodes.
    - Governance participation rates (proposal and voting participation).
    - Incident profile (downtime, bridge incidents, and security incidents).
- **Ecosystem**
    - Number and quality of applications and modules.
    - Presence and adoption of jurisdiction adapters and sector frameworks.
    - Activity in the app and module registry.
- **Financial sustainability**
    - Protocol fee revenue (entity registrations and operations fees).
    - Treasury and foundation reserves.
    - Operator and safety budget mix (fee-based funding vs scheduled releases and Treasury subsidies).

These metrics are more relevant to protocol health than token price.

---

17. Legal position, BVI to Switzerland or ADGM, and risk

17.1 Neutral infrastructure summary

dCorps is intended to be **neutral infrastructure**. Functionally, using dCorps should feel more like using:

- A public registry, plus
- A cloud accounting and governance platform

than like buying a managed fund or financial product.

The protocol:

- Records and helps execute organizational choices.
- Does not pool client funds into discretionary portfolios.
- Does not offer guaranteed returns or principal protection.

Entities retain their own legal existence. dCorps is a shared technical substrate that many independent actors use.

17.2 Development corporation in BVI

Initially, core development and early integrations are expected to be handled by a **development corporation incorporated in the British Virgin Islands (BVI)** or a similar jurisdiction.

This corporation:

- Builds and maintains the protocol code and reference implementations.
- Provides integration and support services to early adopters.
- Enters contracts with audit firms, infrastructure providers, and partners.

BVI is considered for pragmatic reasons:

- It has experience with globally oriented technology and blockchain projects.
- It can offer clearer and more predictable treatment for a development corporation whose revenues come from software and services related to a utility style token, compared to some larger jurisdictions.
- It is relatively fast and cost effective to set up, which matters in the earliest phases of the project.

The development corporation is a software and services company, not:

- A bank.
- An exchange, broker, or asset manager.
- A corporate or NGO service provider for all dCorps entities.

Commercial relationships between the development corporation and entities will be governed by separate contracts.

17.3 Foundation jurisdiction (Switzerland or ADGM)

Once dCorps reaches greater maturity, the intention is to:

- Add a **nonprofit foundation** in a reputable jurisdiction. Two leading candidates are Switzerland and ADGM (Abu Dhabi Global Market).
- Gradually shift stewardship of shared resources and protocol governance processes to that foundation.

Switzerland is attractive because:

- It has a long history of rule of law and predictable treatment of foundations.
- It has practical experience with crypto and on-chain projects, including token foundations and nonprofit stewards.
- It is easier for serious regulators, NGOs, and institutional partners to trust a Swiss based foundation than a purely offshore structure.

ADGM is attractive because:

- It offers a modern foundation framework with remote-friendly setup and lower early overhead.
- It is a credible international jurisdiction and can remain lean while governance and reporting standards mature.

Decision process and status:

- The foundation is not incorporated yet.
- ADGM is a leading candidate for an initial, lean setup; Switzerland remains a strong option for maximum institutional signaling.
- The jurisdiction choice will be made before incorporation based on credibility, governance needs, operational overhead, and the ability to publish transparent reporting.
- The final decision will be published in foundation filings and governance communications.

The foundation will:

- Steward parts of the Protocol Treasury and community allocations.

- Administer grants and ecosystem programs.
- Support long term protocol development and maintenance.
- Coordinate the design, implementation, and maintenance of protocol modules that connect dCorps to the real world, especially:
    - jurisdiction adapter modules.
    - Sector and impact frameworks.
    - Other non core features that interpret Hub state in terms of law, regulation, and societal standards.
- Operate and evolve the app and module registry.
- Promote neutrality and resist capture by any single corporate or jurisdictional interest.
- Actively work with jurisdictions and institutions to co-design and validate jurisdiction adapter modules, using the Hub as a shared base layer.

A core design principle is:

- The **Hub** should remain pure, minimal on-chain infrastructure for entities.
- Everything that links the Hub to the actual world, including legal regimes and sector specific rules, should be expressed as protocol modules and applications.
- These modules must adapt as society, law, and technology evolve, without forcing changes to basic Hub logic.

The foundation is the natural home for this adaptive work. It sponsors research, consults with stakeholders, proposes modules through governance, and retires or replaces modules when they no longer fit current realities.

This is a **good faith design intention**, not a fixed commitment to move or incorporate by a specific date. The exact timing and structure of this transition will depend on legal, regulatory, financial, and operational considerations. Mainnet is gated on transferring protocol and brand IP stewardship to the foundation once formed; until that transfer is complete, mainnet does not proceed. Details will be documented in public filings and governance proposals.

17.3A Relationship between development corporation and foundation

The development corporation and the foundation have complementary roles.

The **development corporation**:

- Is the primary engineering and product organization for dCorps.
- Employs the core team that designs and implements the Hub, reference modules, and critical tooling.
- Enters commercial contracts with entities and partners for integration and custom work.

- Is expected to be one of the first Hub corporations on dCorps, using the same structures that other entities use.

The **dCorps foundation**:

- Is the neutral steward for long term public goods:
  - Parts of the Protocol Treasury and community allocations.
  - Official jurisdiction adapter and sector modules.
  - The app and module registry.
- Acts as a bridge to jurisdictions, regulators, NGOs, and other public stakeholders.

The relationship between them is expected to be formalized through **framework agreements**, for example:

- The foundation can recognize the development corporation as an **authorized development provider** for core protocol work and ecosystem projects.
- The foundation can fund the development corporation to deliver specified milestones, while keeping intellectual property and governance structures aligned with the protocol.
- The foundation can also fund other teams for specific modules, tools, or research, to avoid single vendor risk and to foster a broader ecosystem.

Replacing or significantly downgrading the development corporation as the primary provider of core protocol work is possible, but it should be:

- Governed by clear criteria and processes.
- Subject to strong governance thresholds.
- Paired with a credible alternative development arrangement.

This balance aims to:

- Give the founding team enough stability to build a serious, multi year project.
- Ensure that, in the long run, the protocol is not dependent on a single private company if that company stops performing or shifts priorities.

17.3B Development corporation business model and neutrality

The development corporation is expected to operate as a normal software and services provider, not as a protocol gatekeeper.

Typical revenue sources may include:

- Engineering services for entities and partners:
    - Integrations, custom workflows, and deployment support
    - Migration assistance for adopting dCorps as an operating layer
- Maintenance and support contracts for infrastructure and tooling:
    - Explorer and dashboard operations
    - Enterprise grade APIs and monitoring services
- Delivery of funded milestones:
    - Foundation or Treasury funded work under transparent proposals and milestone reporting
    - Work for jurisdictions or service providers building jurisdiction adapter modules, subject to clearly disclosed terms

The protocol remains open to other development providers:

- The foundation can fund multiple teams to reduce single vendor risk.
- Entities can commission work from any provider.
- Governance can change funding priorities and provider arrangements through documented processes.

This structure is intended to support long term development capacity while keeping the Hub neutral, minimizing conflicts of interest, and avoiding exclusive control by any single private company.

17.4 Foundation as first nonprofit on dCorps

The foundation is expected to be:

- One of the first nonprofit entities registered on dCorps.
- A user of the NGO module, with:
    - Board based governance recorded on-chain.
    - Transparent handling of its own funds.
    - Allocations to programs visible to the community.

This dogfooding reinforces the seriousness of dCorps for nonprofits.

17.5 Development corporation as first Hub corporation

Similarly, the development corporation is expected to be:

- One of the first Hub corporations on dCorps.
- Using:
    - Hub units for its internal cap table.
    - Merchant and treasury wallets for operations.
    - Governance modules for key decisions.

This aligns the incentives of the core team with the robustness of the infrastructure.

17.6 User responsibilities

Participants remain responsible for:

- **Entities (corporations and NGOs)**
    - Choosing appropriate legal forms and jurisdictions.
    - Maintaining compliance with corporate, charity, tax, and other laws.
    - Ensuring governance and financial practices match their obligations.
- **Rollup operators and infrastructure providers**
    - Understanding the technical, operational, and legal risks of sequencing, batch posting, RPC, and bridge operations.
    - Complying with local regulations applicable to infrastructure and service operation.
- **Builders and service providers**
    - Ensuring their tools, apps, and platforms are legally compliant.
    - Managing security and operational risks for their users.

Using dCorps does not remove any legal responsibilities that would otherwise exist.

17.7 Risk factors and acceptance

Participation in dCorps involves significant risks, including:

- **Technology and protocol risks**

    - Bugs, exploits, or failures in code, consensus, or dependencies.
- **Governance and organizational risks**
    - Concentration of voting power.
    - Low participation.
    - Conflicts of interest.
    - Coordination failures in incident response.
- **Market and economic risks**
    - Volatile and potentially low or zero market prices for DCHUB, dShares, or other assets.
    - Illiquidity and slippage.
    - Competition from other protocols or technologies.
- **Regulatory and tax risks**
    - Changes in how authorities treat tokens, entities, and activities.
    - New licensing, reporting, or tax obligations.
    - Restrictions or bans on certain activities or assets.

Even though DCHUB is designed and intended as a protocol utility token, there is **no guarantee** that regulators in every jurisdiction will treat it that way. Some regulators may classify DCHUB, dShares, or other instruments associated with dCorps as securities or as falling under other categories of financial regulation. Classifications may differ across countries and may change over time as laws, regulations, and case law evolve. Participants should factor this uncertainty into their risk assessments and seek independent legal and tax advice.

Other risks include:

- **Counterparty and third-party risks**
    - Failures, misbehavior, or insolvency of exchanges, custodians, or service providers.
    - Poor quality or malicious third-party apps or modules.
- **nonprofit and donor risks**
    - Misinterpretation of on-chain data.
    - Misuse of transparency to create misleading narratives.
    - Local hostility in some regions toward NGOs using crypto infrastructure.

Participants should only engage with dCorps, run entities, or hold related tokens if they are prepared to accept the possibility of partial or total loss and if they can comply with all applicable laws that apply to them.

17.8 Separation of roles and responsibilities

To recap:

- **dCorps protocol**
    - Neutral infrastructure.
    - Provides entity, governance, and financial primitives.
- **Development corporation and foundation**
    - Develop, maintain, and steward parts of the protocol and ecosystem.
    - Execute decisions in the legal realm.
    - Coordinate protocol modules, grants, and registry operations.
- **Entities**
    - Operate their businesses or NGOs.
    - Bear legal, fiduciary, and operational responsibility.
- **Market and service providers**
    - Exchanges, issuance platforms, custodians, DeFi protocols, auditors, and others.
    - Have their own legal and regulatory responsibilities.

Maintaining these separations is essential for clarity, accountability, and long term trust.

---

18. Glossary and open decisions

18.1 Selected glossary

dCorps The protocol and ecosystem described in this whitepaper; an on-chain base layer for entities.

Hub / dCorps Hub chain The Arbitrum Orbit rollup (Rollup mode) that settles to Ethereum and hosts the entity registry, core entity templates, governance, and accounting primitives.

DCHUB The native token of the dCorps Hub, used for gas on the Hub rollup and protocol governance, and optionally for protocol-level fees or deposits. Not equity in user entities or in the development corporation or foundation.

Hub corporation A corporation operating entirely on the Hub, with ten thousand internal units forming its cap table. Hub corporations can represent operating companies, holding companies,

or joint venture and SPV style structures, depending on how their units, wallets, and governance are configured.

Hub units Internal units of a Hub corporation that represent economic and voting rights.

cash-based operating view Time-window summaries derived from tagged inflow and outflow events, excluding accrual accounting treatments.

dShares Equity style tokens issued by public instrument issuers (future extension), representing governance and economic rights for that issuer under its chosen legal regime.

nonprofit / NGO entity An entity registered on the Hub as a nonprofit, using board based governance and transparent donation and program flows.

Protocol module An on-chain module implemented as EVM contracts on the Hub rollup that reads entity and financial state and applies additional logic. Includes jurisdiction adapter modules, sector frameworks, and other rule sets. Does not change kernel semantics, but builds on top of them.

jurisdiction adapter module A protocol module encoding how a specific jurisdiction treats certain dCorps entities, including fees, recognition, and reporting expectations.

Sector framework A protocol module defining metrics and standards for a specific domain such as climate, education, or public health.

App and module registry / dCorps App Store The on-chain and off-chain registry that lists applications and protocol modules that integrate with dCorps, along with status, audits, and governance signals.

DID (Decentralized Identifier) An identifier under the W3C DID model that represents an entity, person, or role in a self sovereign way.

Protocol Council A multi stakeholder group that reviews protocol changes and module approvals and advises token holder governance.

Protocol Treasury The pool of assets governed by the protocol and foundation for long term development, security, and ecosystem support.

dCorps foundation The intended nonprofit foundation that will steward the protocol, manage parts of the Treasury, coordinate protocol modules and the app registry, and help keep the Hub core minimal and neutral.

18.2 Open technical parameters

Several parameters will be finalized in separate documents and may change over time, including:

- Sequencer and batch poster configuration and decentralization plan.
- Bridge gateway parameters (admin roles, pause powers, and withdrawal timing assumptions).
- Block time, throughput targets, and gas limits.
- Governance thresholds, timelocks, and Protected Change rules.
- Asset registry criteria and the initial set of supported stablecoins.
- Specific interfaces and schemas for protocol modules.

These will be documented in a **Protocol Parameters and Economics** reference and adjusted via governance as needed.

18.3 Open legal and organizational points

Open points include:

- Final legal structure and jurisdiction of the development corporation.
- Exact legal form and jurisdiction of the dCorps foundation.
- Details of the foundation's charter, board composition, and initial membership.
- Exact selection and rotation mechanisms for Protocol Council members.
- The first set of jurisdictions to adopt jurisdiction adapter modules and the legal instruments they use.
- Details of app and module registry policies, including categories and processes.

These will be documented in charters, articles, and governance proposals and may evolve with experience and advice.

18.4 Future living documents

This whitepaper will be complemented by living documents, including:

Protocol core

- **Protocol Specification**
    - Normative rules, contract interfaces, state machines, and event schemas.
- **Protocol Parameters and Economics**

    - Concrete configuration values, bounded ranges, and economic assumptions.
- **Module Protocol Standard**
    - Attachment rules, interfaces, upgrade requirements, and compatibility tests.
- **Anchoring Standard**
    - Required commitments, cadence, proofs, and failure handling rules for documents and reporting commitments.
- **Data Standards**
    - Schema versions, chart of accounts, tags, and reporting cadence rules.

Governance and policy

- **Governance Charter**
    - Roles, phases, protected changes, thresholds, timelocks, and council processes.
- **Treasury Policy**
    - Permitted uses, prohibitions, reporting cadence, liquidity bootstrap rules.
- **Registry and Module Policy**
    - Listing rules, labels, signals, dispute handling, and deprecation rules.
- **Operator Charter**
    - Operational expectations, security guidance, and objective accountability rules.
- **Foundation Charter**
    - Mission, board processes, stewardship scope, and transparency commitments.

Token

- **Token Policy**
    - Vesting contracts, voting restrictions, unlock schedules, and distribution controls.
- **Genesis Distribution Plan**
    - Genesis custody, lockups, and programmatic controls for major allocations.
- **Emissions and Security Budget Notes**
    - Operator cost assumptions, scheduled release policies, and sustainability targets.

Security

- **Security Policy**
    - Development practices, audit requirements, and release processes.
- **Audit Plan**
    - Audit scope, sequencing, and disclosure policy.
- **Bug Bounty Program**
    - Scope, rules, payouts, and disclosure workflow.
- **Incident Response Playbook**
    - Communication commitments and response procedures.

Legal and risk disclosures

- **Risk Disclosure** (docs/legal/RISK_DISCLOSURE.md)

    - Technical, governance, market, and regulatory risks in plain language.
- **Non-custodial and Non-intermediation Statement** (docs/legal/DISCLAIMERS.md)
    - Bright-line boundaries for custody, relayers, and service providers.
- **Provider Attestation Framework** (docs/spec/SPEC-ATTESTATIONS.md)
    - Step 0 schemas, labels, dispute rights, and roles.

Developer and ecosystem

- **Developer Documentation** (docs/engineering/TECHNICAL_OVERVIEW.md, docs/engineering/INTEGRATION_GUIDE.md)
    - SDKs, APIs, indexing guides, and example integrations.
- **Compatibility Test Suite** (docs/spec/SPEC-CONFORMANCE-TESTS.md)
    - Module conformance tests and schema compliance tests.
- **Reference Indexer Specification** (docs/spec/SPEC-INDEXER.md)
    - Canonical indexing behavior and data export formats.
- **Pilot Showcase**
    - Living dashboards and case studies (published separately from this repo's normative documents).

These documents will reflect real world experience and will be updated through governance and legal processes.

Orbit Rollup Architecture Notes

This document reflects the current Arbitrum Orbit rollup (Rollup mode) architecture.

Key assumptions:

- dCorps Hub is an Arbitrum Orbit rollup settling to Ethereum.
- Secondary chains are out of scope for v1; optional private execution zones may exist but are not required for entity operation.
- DCHUB is the native gas token and the protocol governance token; Orbit v1 does not use staking-based security (staking is only a future L1 possibility).
- Stablecoins (USDC at launch; USDT/DAI etc) are canonical bridged ERC-20 contracts on dCorps; UIs should show mainstream symbols (USDC/USDT/DAI), not chain-suffixed labels.
- Payment UX assumes customers can pay invoices with USDC they already hold on Ethereum (paying ETH gas); entities absorb bridging/ops costs and then operate internally on dCorps stablecoin contracts.
- Governance can de-recognize entities/modules in the official registry, but cannot erase contracts or prevent generic transfers; decentralization follows a staged chain-owner -> DAO plan with timelocks.

- Native issuer integrations (for example Circle-native USDC) are a future possibility, not guaranteed.

END