Superform v2

June 2025

Superform Labs

Abstract

Superform v2 introduces permissionless yield infrastructure that abstracts multichain complexity into a single, Merkle-verified execution flow. Through a combination of smart accounts and a modular hook engine, Superform enables novel onchain primitives: SuperVaults, infinitely flexible yet secure, validator-attested onchain strategies, and SuperAssets, savings tokens like SuperUSD, SuperETH, and SuperBTC that can pull reference collateral from any chain and react to adverse scenarios. All protocol interactions are coordinated through UP, the system's native utility token. When staked, $UP becomes sUP, granting holders governance ability to influence protocol parameters.

1. Introduction

DeFi is at a crossroads. Breakneck growth over the past three years has come with a critical trade-off: user experience has deteriorated, driven by incentives to build for a small, highly technical audience. This focus on specialized features has led to applications that fall short of expanding crypto's TAM. Without a unified and accessible experience, the Ethereum ecosystem risks alienating a broader audience.

Superform v1 pioneered interoperable DeFi applications through cross-chain yield access and composability but left room for improvement on cost and flexibility. Superform v2 is a ground-up rewrite that removes those pain points through a fully modular account, execution, and product stack.

Five core improvements define Superform v2:

1. **Smart Accounts** now initiate every Superform interaction. These lightweight, ERC-7579 upgradable accounts offer gas abstraction, passkey or social-recovery authentication, and temporary session keys while preserving the traditional security of self-custody. Traditional wallet users are not left behind—the usage of companion smart accounts and EIP-7702 lets any EOA slip seamlessly into the same flow, ensuring both newcomers and power users benefit from the same streamlined experience.
2. **Hook-based execution** pairs a modular hook language with SuperBundler to turn multichain complexity into a single signature. Strategies are expressed as ordered hooks such as—bridge, swap, lend, stake—and translated into ERC-4337 UserOps per chain which are then validated and executed. Adding new functionality no longer requires a protocol upgrade; developers simply deploy a new hook that any account can use.
3. **SuperVaults** deliver flexible yield strategies secured by a proof-of-stake validator set. Logic uses Superform's core hooks for sanitization and validation to showcase composability while validators stake tokens to attest to price-per-share updates and face slashing for misconduct. This layered design finally reconciles flexibility, security, and usability in a single vault architecture.
4. **SuperAssets** create Ethereum L1-native tokens like SuperUSD or SuperETH that aggregate yield from many SuperVaults across chains. They auto-rebalance positions, employ pricing circuit breakers through SuperOracle, and leverage a weighted governance curve so $UP stakers keep allocations balanced and safe. Swap fees, insurance contributions, and incentive budgets all flow through the same policy layer.
5. **$UP** powers the Superform protocol. It is burned as part of strategist upkeep in SuperVaults during PPS updates and can be bonded to reinforce validator accountability through protocol-defined slashing. When staked as sUP, it grants governance rights over core protocol parameters, including fee structures, strategy approvals, and SuperAsset allocations.

2. Superform Core

2.1 Smart Accounts

Superform v2 routes interactions through ERC-7579 smart accounts, minimal contracts whose capabilities are expanded by installing upgradeable modules. Compatibility with existing EOAs is maintained through EIP-7702 or via ERC-7579 companion accounts.

2.1.1 ERC-7579 Modules

Within each account, two Superform-specific modules do the heavy lifting. The SuperExecutor processes ordered "hooks" (bridge, swap, lend, stake) and executes them atomically onchain, while the SuperValidator verifies bundled signatures and Merkle proofs, ensuring that only authorized operations reach the executor.

Users may also install any audited third-party modules such as—dead-man switches, scheduled transfers, flash-loan guards, timelock withdrawals, and more—to customize their security posture. Modules are immutable once deployed, so an attacker cannot silently replace logic; users must explicitly approve any new module hash.

2.2 Hook-Based Execution

Superform v2 translates user intents into ordered onchain "hooks", managed by the SuperExecutor and validated by the SuperValidator. Together with the off-chain SuperBundler service, cross-chain bridge adapters, and the SuperLedger accounting system, these components create a deterministic, multichain execution flow governed by a single user signature.

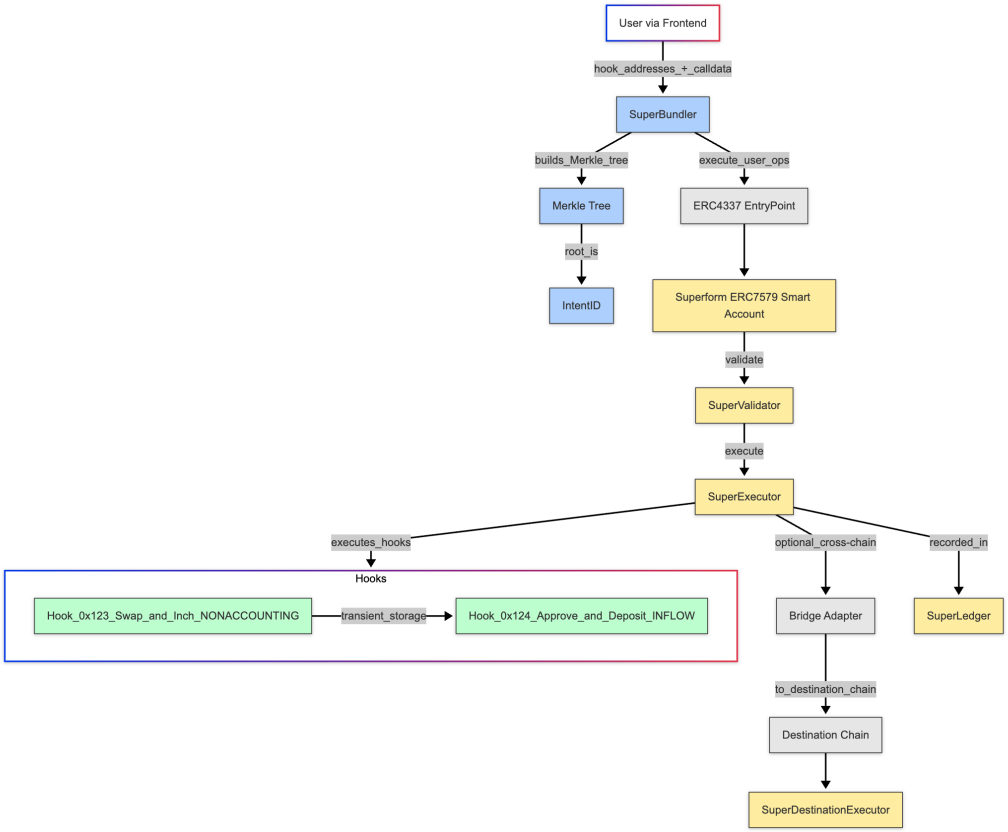

Figure 1: Execution Flow

2.2.1 Hook Bundling

Hooks are modular contracts that enable flexible execution workflows and adhere to a standardized specification. Multiple hooks can be chained together to facilitate custom execution flows. The creation and deployment of hooks are entirely permissionless, allowing developers, protocols, or even AI agents to design and implement sophisticated strategies effortlessly. Key actions that hooks can perform include:

1. **Pre-vault action:** Actions such as bridging, approving, obtaining an LP position, or checking KYC.
2. **Core vault action:** Performing core tasks like depositing or redeeming.
3. **Post-vault action:** Post-deposit operations like staking, looping, supplying as collateral, or other advanced financial workflows.
4. **Helper functions:** Supporting operations such as pricing, oracles, and other auxiliary tasks.

All hooks inherit `BaseHook`, which enforces immutable metadata i.e. a `hookType` (`NONACCOUNTING` | `INFLOW` | `OUTFLOW`) and arbitrary `subType` (e.g., `SWAP_UNISWAPV3`). Hook developers therefore focus only on `_preExecute` (sanity checks & context), `_postExecute` (result persistence), and `build` (returning Execution[]).

The frontend (or any integrator) selects a list of hook addresses plus calldata for each. The SuperBundler deterministically:

1. Splits the list per destination chain, producing one `ExecutorEntry` per chain.
2. Wraps each entry into an ERC-4337 `UserOperation` and places all ops into a Merkle tree whose root is the intent ID.
3. Returns the Merkle root to the wallet for signing and keeps the full tree for proof generation.
4. Optionally simulates each `UserOp`; async ops (e.g., waiting for a bridge receipt) are parked until their dependencies clear.

2.2.2 Validation

When a `UserOp` reaches the EntryPoint, the EntryPoint calls `validateUserOp` on the user's smart account, which delegates to SuperValidator:

- SuperMerkleValidator (source chain) checks the ECDSA/EIP-1271 signature over the Merkle root and verifies the leaf proof for the current `UserOp`
- SuperDestinationValidator (destination chain) performs the same proof check plus extra replay guards—`block.chainid`, `executor` address, and `validUntil` timestamp—because no EntryPoint is involved.
- Both validators support threshold signatures for future decentralization and revert if the caller is not the authorized Bundler or if the proof is stale.

2.2.3 Execution

After validation the smart account invokes `execute(bytes)` on SuperExecutor, passing the ABI-encoded `ExecutorEntry`:

1. `_execute` iterates through `hooksAddresses[]` and `hooksData[]`
2. For each hook `H`, it calls `preExecute`, then executes the low-level calls returned by `H.build(...)`, and finally calls `postExecute`.
3. The hook's internal transient storage (`outAmount`, `usedShares`, `asset`, `spToken`, `vaultBank`, `dstChainId`) is read by the next hook in constant time with zero SSTORE cost.
4. Depending on `hookType`, `_updateAccounting` records inflows or outflows in SuperLedger, queries `YieldSourceOracleConfig`, and, for outflows, auto-transfers protocol fees to the configured `feeRecipient` .

Other core functionality includes:

- **Cross-Chain Locks** – Bridge hooks call `IVaultBank.lock` before invoking any bridge adapters. This allows the permissionless creation of any asset on another chain via proofs, used primarily in SuperAssets.
- **Fee Routing** – Native and ERC20 fees are transferred via delegated calls from the smart account, with balance-before/after assertions.

- **Gas Cost Skipping** — A specialized SuperDestinationExecutor mirrors the same flow on the remote chain, skipping EntryPoint gas costs and permitting account creation if the destination ERC-7579 wallet does not yet exist.
- **Onchain Accounting** — SuperLedger's modular oracle managers calculate APY, track realized yield, and expose manager-defined fee rules to the executor.

2.2.4 End-to-End Guarantees

- **Atomicity** – Any revert inside a hook unwinds the entire `UserOp`; cross-chain actions are ordered by Merkle proofs to prevent partial completion.
- **Extensibility** – New hooks, bridge adapters, or yield-source oracles can be added by anybody permissionlessly without touching validators, executors, or ledger code.
- **Gas Efficiency** – Transient storage, delegate-call batching, and threshold-sig verification keep marginal gas low even with many hooks.
- **Security** – Role-aware validators, caller locking, configurable slashing in adapters/ledger, and immutable module hashes bound the attack surface.

3. Superform Periphery

DeFi has made countless tradeoffs to scale. In this re-architecture of the yield stack, SuperVaults are the programmable base layer, validator-secured vaults that can run any hook-based strategy across chains; while SuperAssets are the user-facing savings wrapper that packages multiple SuperVault positions behind a single ERC-20. Builders compose sophisticated, flexible strategies inside SuperVaults; users simply hold a SuperAsset like SuperUSD to tap those strategies automatically, with oracle-priced swaps, secured by circuit-breakers, with incentives to keep allocations on target. Together they turn complex multichain yield generation into a one-click experience without compromising transparency, security, or composability.

3.1 SuperVaults

Yield seekers today must choose between gas-heavy onchain vaults whose logic evolves infrequently and opaque managed vaults that execute off-chain with minimal transparency. SuperVaults v2 are designed to solve the long-standing "vault trilemma" of flexibility, security, and usability. Strategists can express anything from simple single-chain lending loops to offchain carry trades, while providing depositors with deterministic onchain guarantees around price-per-share (PPS), fees, and withdrawal rights.

3.1.1 Contract Architecture

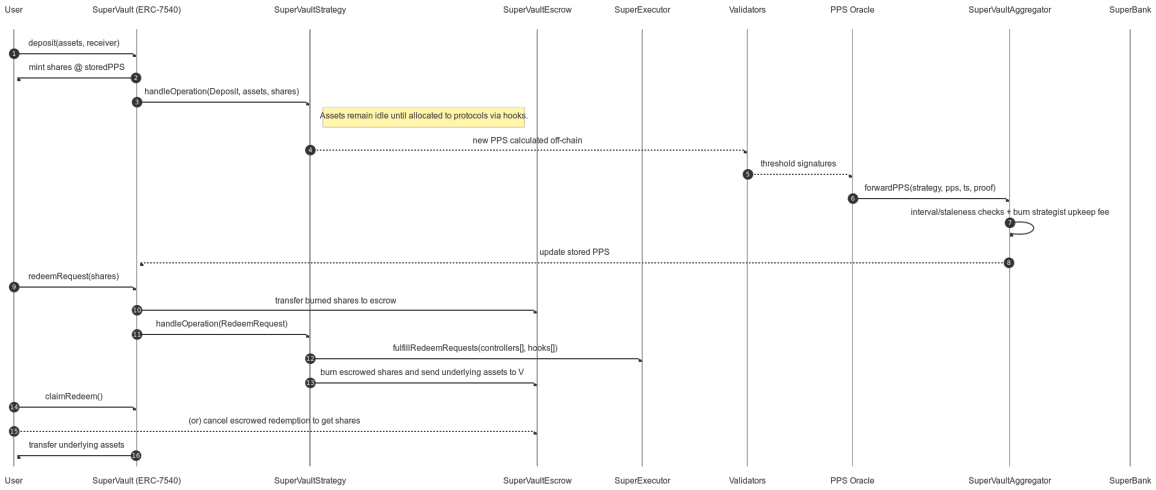

Figure 2: SuperVault Architecture

The architecture achieves this by splitting concerns across four immutable clone contracts—`Vault`, `Strategy`, `Escrow`, and the registry/oracle `Aggregator`.

- **SuperVault** (ERC-7540) mints shares synchronously on deposit and asynchronously returns assets on redemptions.
- **SuperVaultStrategy** consumes Hook bundles, tracks cost-basis, queues and fulfills pending redeems, and enforces fee/slippage policies. Hooks are validated against two Merkle roots (global and per-strategy) served by the Aggregator.
- **SuperVaultEscrow** holds users shares when they initiate a redeem request rather than immediately being burned. This allows users to cancel the pending redemption and retrieve their shares automatically from escrow without any third party if the need arises.
- **SuperVaultAggregator** is the single source of truth for PPS. It can deploy new Vault/Strategy/Escrow pairs, store per-strategy PPS, apply min-update/max-staleness rules, pause misbehaving strategies, and

deduct upkeep fees from strategist-deposited $UP.

- **Validator-secured PPS oracles** allow validators to stake in dedicated oracle contracts and periodically sign `(strategy, pps, ppsStdev, nValidator, totalValidators, timestamp, chainId)`. A whitelisted onchain oracle—e.g., `ECDSAPPSOracle` for ECDSA, verifies the aggregate proof and calls `Aggregator.forwardPPS`. If the update passes interval and staleness checks, the Aggregator writes the new PPS and burns a flat upkeep fee from the strategist's deposit; if not, the strategy is paused and may be taken over. Misbehavior (e.g., invalid signatures or equivocation) may trigger automated slashing, reducing validator-held $UP via immutable onchain logic. Slashed tokens may be redirected to protocol-aligned recovery mechanisms, without discretionary control.

3.1.2 SuperVaults Feature Benefits

For Users:

- **Earn More Securely:** A dual-Merkle tree allowlist of hooks in Superform Core allows for scalable global and strategy-level control of accepted actions that can be taken on funds in the vault without sacrificing functionality. A guardian network watches updates to the root with a configurable window where updates can be vetoed.
- **Transparent Economics:** Every price-per-share update is published onchain, accompanied by threshold consensus reached by an economically-bonded validator set. Users only incur fees on realized profits and strategists must pre-fund upkeep in enforcing consistent reporting.
- **Composable and Efficient:** ERC-7540 architecture allows for costs to be offloaded to the strategists while retaining the convenience of synchronous deposits. Vault products are instantly composable across DeFi protocols given standard compliance.

For Strategists:

- **Permissionless Vault-As-A-Service:** Anyone can launch new vaults, manage them entirely on their own via the SuperVaultSDK, and distribute via the Superform Frontend to bootstrap liquidity and earn performance fees.
- **Leverage Protocol Security:** Strategists can focus on making money, offloading security to the validator set, Aggregator, and Guardian Network which collectively enforce rules before committing updates.
- **Flexible Strategy Design:** Utilize the protocol's capability to combine lending, looping, bridging, and other DeFi actions into complex, tailored strategies for vaults to earn the most they can without writing a line of Solidity.

3.2 SuperAssets

SuperAssets are ERC-20 tokens that act like onchain savings products, offering users instant, diversified yield on a reference asset across all chains while bootstrapping persistent liquidity for new chains and issuers. Each token (e.g., SuperUSD, SuperETH) is composed of pegged asset SuperVault positions on any and multiple chains (e.g. SuperUSDC, SuperUSDT, on Ethereum, Optimism, Base, etc.) and incentivizes rebalances in accordance to a risk-aware configuration enforcing strict circuit-breakers. All allocations are governed by protocol rules and decentralized governance.

3.2.1 Contract Architecture

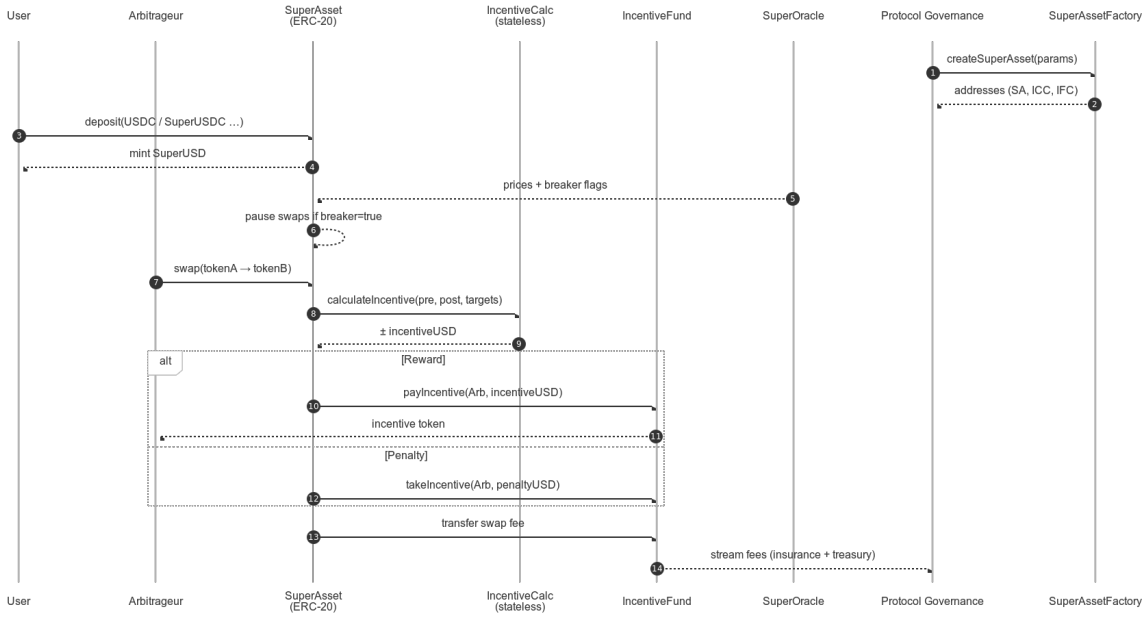

Figure 3: SuperAsset Architecture

The SuperAsset system achieves this through 3 main contracts, which are created through the `SuperAssetFactory`:

- **SuperAsset** is the main contract. It accepts deposits of whitelisted ERC-20 or ERC-7540 vault shares, mints the basket token, executes oracle-priced swaps, and enforces dual circuit-breakers. Each swap applies a configurable fee that is routed by protocol logic across incentive, coverage, and governance-aligned functions. If the SuperOracle flags a depeg, dispersion spike, or stale data, SuperAsset pauses swaps and incentive transfers until conditions normalize. Each SuperAsset is cloned upon instantiation at the factory.
- **IncentiveCalculationContract** (ICC) is a pure-math helper. It compares the live allocation to the governance-set target, applies Ki weights to account for systemic importance, squares the deviation to obtain an "energy" score, and converts the delta into a USD-denominated incentive. Only a few ICCs exist, created by Superform, and strategists can point to which one they want to use.
- **IncentiveFundContract** (IFC) holds the reward budget and the penalties it collects. When SuperAsset calls it, IFC either pays the calculated incentive to the arbitrageur or invoices them for worsening the allocation. It also receives a portion of swap fees (40% by default), of which half seeds the insurance fund and half is earmarked for future governance-directed incentives. All transfers obey the same circuit-breaker checks as SuperAsset. Each IFC is cloned upon instantiation at the factory.

3.2.2 SuperAssets Feature Benefits

For Users:

- **One-Click Omnichain Yield:** Deposit any whitelisted ERC-20 or SuperVault share and instantly mint a SuperAsset (e.g., SuperUSD). Vault Banks across major L2s mint SuperPositions back to Ethereum, so yield earned on any chain streams into the same balance—no bridges, wrappers, or manual rebalancing.

- **Institutional-Grade Protection:** SuperOracle pricing plus depeg and dispersion circuit-breakers freeze markets before losses propagate; while paused, neither incentives nor fees can exit. A continually growing insurance fund underwrites tail risk, and circuit breakers automatically decay depegged assets while recapitalizing the basket.
- **Yield Boosted by Utility:** SuperAssets compound three revenue streams—underlying vault yield, swap fees, and issuer & chain incentives—into a single ERC-20, giving holders outsized returns previously accessible only to large liquidity providers.

For Chains & Issuers:

- **Bootstrap Liquidity:** Permissionlessly create a new SuperAsset or use sUP to influence an existing SuperAsset basket that allocates to a token's SuperVault alongside blue-chip collateral. Doing so gains instant depth, diversification, and TVL without seeding separate pools or bribing AMMs because of inbuilt oracle-priced swaps.
- **Sticky TVL:** Use sUP to secure a lasting weight in the SuperAsset. Because holders value the diversified yield and value in the Superform ecosystem, capital is more likely to stay even after rewards taper, curbing mercenary outflows.
- **Instant Distribution Channel:** Launch a vault on any chain and tap mainnet liquidity instantly—SuperBundler auto-routes deposits, and SuperAssets act as liquidity routers where your asset reaches new users everywhere immediately.

3.3 $UP

UP is the coordination and utility token of Superform and sUP is a non-custodial staked representation that carries onchain governance rights.

3.3.1 Protocol Resource Flows

The protocol coordinates value-aligned behaviors through a permissionless onchain system known as `SuperBank`. `SuperBank` facilitates programmable interactions among autonomous smart contracts. This includes, but is not limited to:

- **SuperVault Parameters:** A performance fee can be taken on SuperVaults managed by Superform along with a portion of the fees other strategists set on their SuperVaults to support protocol operations.
- **SuperAsset Swap Functions:** Governance may signal preferences over routing parameters for post-insurance swap fees to align with system-level functions.
- **Execution Hook Parameters:** A share of flow-based fees in Superform Core may be taken to incentivize system contributions and tooling.

Smart contract logic may, at the discretion of protocol governance, route certain tokens to the `SuperBank`, where they are optionally converted into $UP via programmable hooks. These may then be made available or reserved for protocol use, based on governance outcomes. No distributions are guaranteed or administered by any centralized party.

$UP may also be used in strategist upkeep — a flat amount is burned per PPS update. If an operator's $UP balance reaches zero, the strategy may automatically pause. Participants may optionally escrow $UP as a non-binding signal of reputation. Escrows are subject to automated onchain conditions, and no central entity determines or enforces outcomes.

3.3.2 Benefits of Staking $UP (sUP)

Staking $UP grants participation in decentralized governance processes. Any benefits associated with holding sUP are emergent properties of the protocol and are not financial promises or entitlements. Governance rights may include influence over the following domains:

- **Asset Expansion & Strategy Tuning** — sUP governance may signal preferences over SuperAsset parameters, including asset whitelisting, weight targets, efficiency coefficients (Ki), and risk circuit breakers.
- **Economic Configuration** — Participants can propose and vote on protocol configuration parameters, such as performance fee caps, swap routing preferences, and budgeting system parameters.

3.3.3 Decentralization Roadmap

Phase	**Milestone**	**What Ships**	**$UP utility**
1	v2 Launch	Core stack live (smart accounts, hooks, SuperVaults, SuperAssets, SuperBank).	$UP is used for strategist upkeep; vault performance updates require $UP to execute.
2	Validator Decentralization	Launch permissionless staking so participants may bond $UP to join validator sets across SuperVaults. Slashing is enforced autonomously via onchain rules. Modular oracle adapters with an SDK will enable validator sets to publish comprehensive data to support a broader range of SuperVaults configurations.	sUP holders may vote on coordination parameters (bonding logic, quorum thresholds, max validator share) and signal vault and asset-level preferences.
3	SuperBundler Spec + Reference Implementation	Publish open-source spec and reference implementation for Bundlers. Third-party Bundlers can permissionlessly integrate via a governance-curated allowlist.	sUP holders manage Bundler coordination preferences (e.g. signaling inclusion on allowlist, proposed fee models). $UP may be bonded by Bundlers as a public commitment to uptime or performance.

Glossary

Aggregator Registry/oracle contract where new `Vault`, `Strategy`, and `Escrow` clones are permissionlessly deployed and where validators attest to Price-Per-Share updates.

Circuit Breaker On-chain guard that pauses swaps or vault actions when oracle price dispersion or depeg thresholds are breached.

Depeg Guard Absolute price deviation threshold for an individual asset in a SuperAsset. If breached, swaps pause and the asset weight decays.

Dispersion Guard Parameterized limit on the standard deviation of constituent asset prices inside a SuperAsset; tripping it triggers a circuit breaker.

EIP-7702 Ethereum standard that allows EOAs to act as smart accounts, enabling existing wallets to interact with account abstraction systems without migration.

ERC-4337 Account abstraction standard that enables smart contract wallets to function like EOAs through a UserOperation mempool and bundler system.

ERC-7540 Vault standard that defines interfaces for asynchronous deposits and withdrawals.

ERC-7579 Modular smart account standard that enables plug-and-play functionality through a standardized module interface.

Escrow Immutable contract that holds idle funds between strategy steps and enforces withdrawal rights.

Hook Lightweight, ERC-7579 compliant module that performs a single action (e.g., lend, bridge) inside a strategy call-graph.

Insurance Fund Reserve funded yield earned and swap fees; used to cover shortfalls during black-swan events.

Ki Governance-set efficiency coefficient that weights a vault's importance when computing incentives.

Price-Per-Share (PPS) Cumulative value of a SuperVault denominated in its underlying asset; monotonically increasing and signed by validators.

SuperAsset ERC-20 basket token that auto-rebalances across multiple SuperVaults, compounding yield, swap fees, and issuer incentives.

SuperBank Coordination contract that routes protocol-level fees, burns $,UP for upkeep, and escrow/bond commitments.

SuperBundler Off-chain relayer that batches user deposits and executes hook graphs across chains, returning a single receipt.

SuperGovernor On-chain module where sUP holders propose and vote on protocol events, such as SuperAsset target weights.

SuperOracle Modular oracle adapter set providing fair market prices and volatility metrics to SuperAssets and circuit breakers.

SuperPosition ERC-20 receipt minted by Vault Banks representing a user's position on the source chain; streams yield back to mainnet.

SuperVault Validator-secured vault template that can run arbitrary hook-based strategies while exposing deterministic PPS and fee rules.

$UP / sUP UP is the utility token of the Superform Protocol; staking it yields sUP, granting governance rights and enabling validator / strategist bonding.

Validator Economically-bonded actor that signs SuperVault PPS updates and, in future, other oracle data; slashable for misbehavior.

Vault Bank Chain-specific deposit contract that mints SuperPositions and handles cross-chain locking/unlocking for SuperVaults.

Disclaimer

This document is provided for informational purposes only and does not constitute financial, investment, legal, or tax advice. It is not intended to be, and should not be construed as, an offer to sell, a solicitation to buy, or a recommendation for any digital asset, security, or financial instrument.

The contents of this document are subject to change at any time without notice. No representations or warranties are made as to the accuracy, completeness, or reliability of the information provided herein. Recipients are solely responsible for evaluating the risks and merits associated with any use of the information and are encouraged to consult with their own advisors.

Nothing in this document should be interpreted as creating a contractual relationship or fiduciary duty between any party and the reader. The systems and mechanisms described are intended as general-purpose, non-custodial infrastructure deployed via autonomous smart contracts. The Foundation does not offer custodial services, manage assets, or provide financial guarantees.

Token Notice: UP is designed as a coordination asset for decentralized infrastructure. It does not grant any rights to dividends, profit, or financial return. All functionality is governed by immutable smart contracts and decentralized governance. There is no central entity promising or administering returns, and the token is not marketed or intended as an investment.