Agoric tokenomics litepaper

The cryptoeconomy has entered its next phase: a multichain economy where value, liquidity, and applications naturally span multiple networks. No single blockchain is the center of gravity anymore.

Agoric's orchestration allows smart contracts to operate across chains, over time, without forcing users to juggle gas tokens or complex workflows.

Agoric's orchestration-first model uniquely supports *multi-block*, *multi-chain*, *asynchronous* applications — software that can coordinate actions over time and across ecosystems, all powered by one token.

In this document, we describe the economic characteristics of orchestration, and propose a revamp of BLD tokenomics so that the BLD token and fee model can support long-lived cross-chain contracts and to ensure that BLD directly captures value from orchestration. It provides:

- **Value capture for BLD holders:** Orchestration activity increases gross demand for BLD as the network's fee and settlement token, while net value accrual to holders depends on governance-directed use of orchestration fee surplus.
- **Simplicity for users:** Instead of unpredictable execution costs, users just pay a clear, predictable postage fee.
- **Developers empowerment:** Orchestration enables new applications and flexible business models, whether they come from Web2 or Web3. Agoric's orchestration-first design enables use cases not possible on other chains.
- **Network sustainability:** Contract stakes, gas purses, and the Purser system ensure operations are funded and accountable.

By positioning BLD as the center of the ecosystem and firmly linking it to orchestration, Agoric is laying the foundation for a more scalable, flexible, and user-friendly network that can drive the growth of the multichain digital economy.

The cryptoeconomics of orchestration

The Agoric L1, a decentralized network of validator nodes, supports multi-block, asynchronous programs through a combination of distributed object-capability security, asynchronous message passing, and durable contract execution within the JavaScript runtime. Each smart contract runs in an isolated vat—a deterministic event loop that processes messages and event responses sequentially.

Multi-block behavior emerges naturally because messages between vats are durable and replayable; a contract can send a message to another vat or chain and later receive its response in a future block without halting execution or requiring a new transaction from the user.

The implications of this architecture extend beyond the boundaries of the Agoric L1, allowing for interactions across chains and the ability to exercise control over, and interact with, remote accounts. This makes the Agoric L1 an enabler of cross-chain economies by reducing frictions while also promoting on-chain secure computing.

Cross-chain orchestration extends this model beyond the local chain. Outbound actions are encoded as IBC packets or routed through bridge adapters (*e.g.*, Axelar, CCTP). The orchestrator manages these remote calls as asynchronous promises; it queues the operation, tracks confirmation or failure, and resumes local logic when results return.

The architecture of the Agoric L1 allows programs to behave like distributed workflows – reacting to events, performing remote operations, and maintaining consistency across chains while remaining fully verifiable and deterministic. The result is a generalized framework for on-chain automation and cross-chain state continuity, built directly into the Agoric platform.

A new fee structure for orchestration

On most blockchains, users pay directly to run contracts. This made sense when contracts and exchanges were limited to just one blockchain at a time. But in a multichain world of increasing institutional adoption, the user-pays model creates serious problems. For example, users must hold gas tokens for every chain and service they touch, just to interact with an app. This shifts all the costs (time, effort, and money) directly on to users, even though contracts and developers are the ones consuming resources and have the specialized capabilities to deal with these costs efficiently. These issues make blockchain apps hard to use and limit their potential scale.

Orchestration introduces an asynchronous fee model that reflects real-world business needs, where users pay postage and contracts pay execution and orchestration fees. This model aligns incentives with Agoric's orchestration-first architecture – where long-lived, cross-chain contracts power real applications at scale.

The new fee structure is designed to put funding responsibility where it belongs: on the services that consume the resources.

- Users pay a small "postage" fee to deliver a message, which helps prevent spam
- Contracts cover the costs of execution and orchestration

Contract fees and purses

The orchestration services used by contracts include:

- Agoric execution costs such as gas and storage fees
- Creation of remote accounts to be controlled by the contract
- Cross-chain transfers such as via CCTP, Axelar, etc.
- Swaps between tokens, typically executed on remote DEXs
- Invocation of remote operations, such as invoking an Aave `supply` operation on a remote account
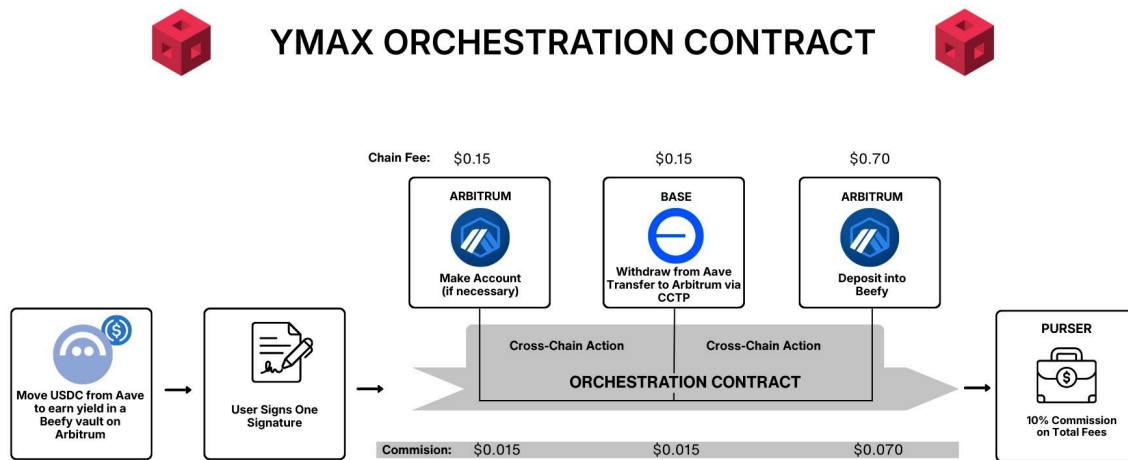
Contracts pay for these services through a fee account: each contract must pre-fund its fee account for orchestration and gas. Anyone (including DAOs or community funds) can top the fee account up. If that account runs dry, synchronous operations like local execution will continue for a brief (governance determined) period while async operations like remote messages pause.

In addition to the fee account, contracts must also maintain a stake balance (which functions as a performance bond). Synchronous operations executed after the fee account is exhausted temporarily accrue a debt. The debt must be repaid or governance can enforce penalties or slashing against the contract stake. This simplifies resource management and error handling within contracts. Additionally, the system may deploy the funds under stewardship for (low-risk) returns. Yields from contract stakes are treated as additional orchestration fees.

Each remote operation typically requires fees to be paid at the destination (e.g., gas fees on Arbitrum, relayer fees for Axelar). The contract pays those in BLD, and orchestration or intermediate services convert that to the appropriate gas token (see, "Coordination and governance", below).

The orchestration services charge commission on those remote operation fees. This is an important new source of revenue. For example, as of this writing (2025), an Orchestration transaction to request an Aave supply operation costs less than a penny to execute on Agoric, whereas the remote operation it invokes costs approximately $0.18. Given that many transactions are likely to incur multiple remote operations, a 10% commission on the total fees accrued will be an order of magnitude higher than Agoric chain fees. As a result, the BLD economy taps into the economic value of those other chains.

YMAX ORCHESTRATION CONTRACT

How it works:
- One Signature, Full Automation - User signs once; orchestration contract executes the entire sequence automatically
- Sequential Cross-Chain Flow - Operations happen in order: Arbitrum → Base → Arbitrum (not simultaneously)
- Chain Fees Applied - Each blockchain charges for execution (e.g. $0.15 + $0.18 + $0.70 = $1.03 total)
- Purse collects 10% commission on total chain fees ($1.03 in chain fees = $0.103 commission), returned to the Agoric chain

Coordination and governance

This tokenomics design also introduces two critical functions to support the fee account and contract stake and coordinate and facilitate exchanges: stewards and the purser.

· **Contract Stewards**: Each app has an infrastructure agent, its Steward, which manages its gas fee account and ensures orchestration actions are funded with the right tokens. The contract prepays for services in any allowed token to the Steward, and the Steward converts it to BLD (using local or remote on-chain markets).

· **Purser**: The Purser is a chain-level infrastructure service for orchestration. It manages fee routing/burning, governance enforcement, and yield on contract stakes. This structure allows iterative upgrades and more sophisticated financial operations.

Governance determines how the fee surplus is disbursed. For example, BLD holders might choose to direct the surplus to a community pool, to burn BLD, to deploy it into defi contracts to earn interest, or to otherwise return reward to BLD stakers. The Purser holds contract stakes, aggregates fees, and implements how network revenues are handled.

Orchestration pricing can be denominated in USD or BLD equivalents. The system dynamically adjusts the BLD amount to maintain stable real-world costs. Over time, as data accumulates, fees are likely to evolve toward market-driven pricing rather than governance-set rates.

Governance may also reprice orchestration and gas fees monthly or as needed based on volatility.

Enhanced value capture for BLD

As applications make use of orchestration, they require BLD to pay for execution and cross-chain operations, creating sustained demand for BLD. This demand provides the foundation for BLD tokenomics to capture value from real network usage, with the specific mechanisms and parameters determined through community governance.:
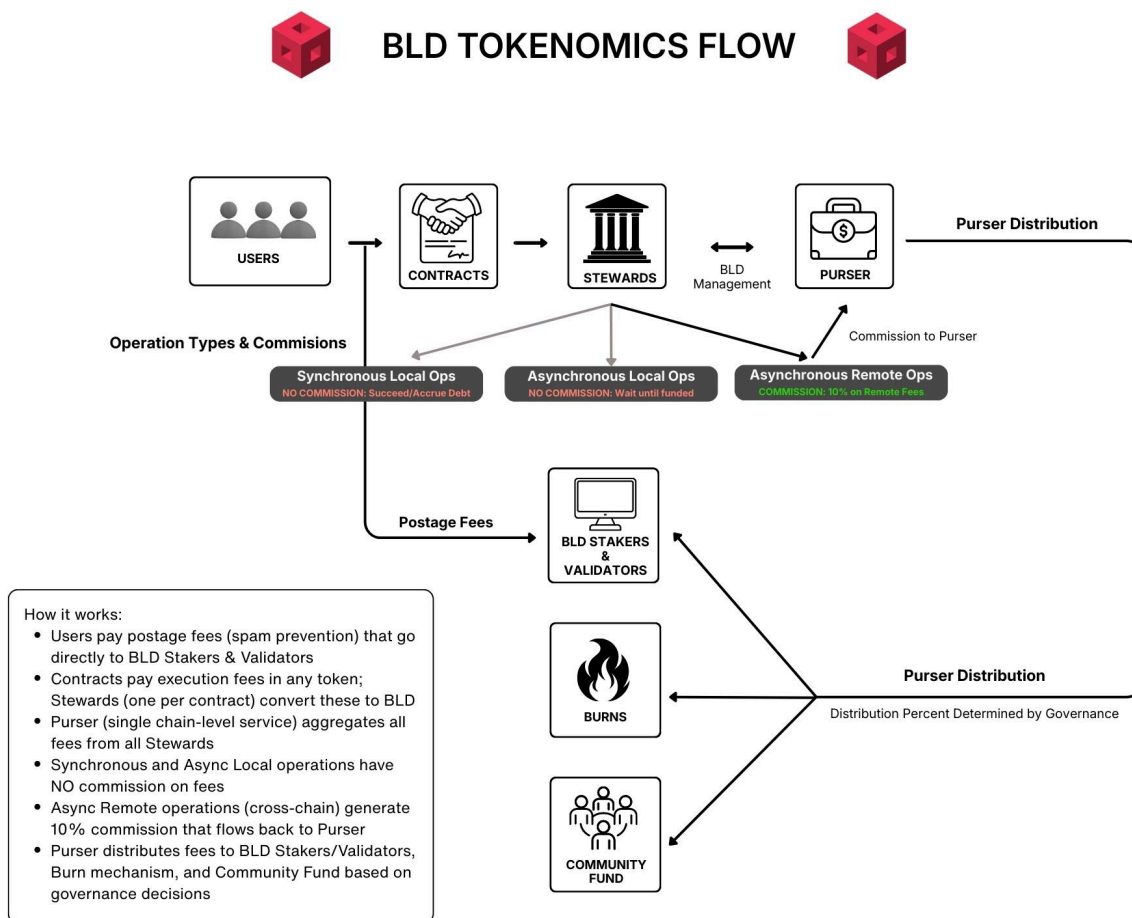
- Steward-managed buybacks or internal swaps may be introduced, which convert collected fees into BLD and can be executed in a market-aware manner.
- Fee burns could be used to offset emissions, achieving effective deflation once burn volume exceeds issuance

By directly linking network usage and orchestration activity to staking yield, the tokenomics structure tightly couples BLD demand and value accrual to the success of applications built on Agoric. Value flows through multiple fee channels (see Figure 2), including:

- Postage fees, paid by users
- Execution & storage cost , paid by applications for computation
- Orchestration fees, paid by applications, including gas fees for remote and cross-chain operations)

(continued below)

Figure 2: Fee Sources & Routing

BLD TOKENOMICS FLOW

This refreshed tokenomics model ensures that BLD value capture scales with real economic activity across the multichain ecosystem. Together with the infrastructure components of Stewards and the Purser, the refreshed tokenomics model is designed to make Agoric simpler for users, more rewarding for stakers, and more powerful for developers and the network as a whole.