Input | Output Group

IO Q2 2026 delivery report

This report provides the Cardano community with a clear view of what Input Output (IO) delivered in Q2 2026 (April to June), why each outcome matters, and how it relates to IO's treasury-funded commitments.

July 2026

Prepared by IO

The bottom line

In Q2 2026 (April to June), IO delivered progress across its treasury-funded engineering commitments, spanning core node maintenance, developer tooling, and Cardano's scaling roadmap. Two initiatives, Acropolis and Tiered Pricing, wound down as announced in April, with all funding returned to the treasury. Every outcome in this report is publicly verifiable.

The following outcomes define this quarter's deliverables:

- **Leios' Musashi Dojo testnet went live.** The Musashi Dojo testnet launched on June 23, backed by a newly approved treasury proposal, marking the first time Leios has run on a live network rather than in simulation.
- **Network health and efficiency.** Log-structured merge (LSM) storage completed its beta, with SPO feedback confirming a roughly 65% reduction in memory usage and no stability regressions. The key evolving signature (KES) agent also reached its 1.0 milestone. Ledger-HD's consensus redesign is complete and ready to merge.
- **Node releases stayed on schedule.** Cardano node v.10.7.1 shipped in April with steady performance gains and lower CPU usage; v.11.0.1 followed, under protocol version 11 (PV11), with no regressions. Its cost model changes have already been submitted as a governance action on mainnet, and the legacy tracing system has been fully retired, removing roughly 11,000 lines of dead code from cardano-node.
- **Stake pool incentives went public; nested-transaction work advanced.** The revised stake pool incentive design was published in full and demonstrated to SPOs and delegated representatives (DReps) in June. Progress on nested transactions has continued, and the PlutusContext milestone has shifted to Q3 2026.
- **Plutus got cheaper and safer to build on**. A native Value type removes the need for developers to maintain their own multi-asset libraries, and a faster Plinth compiler cuts execution costs and transaction fees across the board.
- **Cardano High Assurance tooling matured.** Automatic Formal Verification completed its blockchain formalization milestone and the specification of the Universal Annotation Language. Property-Based Testing shipped its tool as a library that includes a testing interface, positive and negative testing, and threat modeling. The Static Analyzer has been finalized with the delivery of the coding rules, a complete analysis of public and private security audits, a set of ready-to-use binaries, and a Visual Studio Code extension.
- **Mithril advanced to zero-knowledge proofs.** A recursive SNARK-based proof system moved from prototype to testnet within the quarter, before responsibility for Mithril's specialist development transitioned to the specialist partner Teragone in June.
- **Hydra's production credentials grew.** Hydra node 2.1.0 shipped in May with improved deposit security, a new SQLite-backed event store, a roughly 7% reduction in snapshot confirmation latency, and compatibility with cardano-node protocol version 12+. Hydra node 2.2.0 followed in June, packaging the partial fanout work, a revamped hydra-tui, and a substantially expanded benchmark suite. 2.2.0 also unified on-chain membership checks

around a single BLS-accumulator hash, reducing the snapshot-signing tuple from seven fields to four and removing redundant hashing.

- **Input Output Research (IOR) treasury proposal ratified.** The IOR proposal for **Cardano Vision 2026** has been officially ratified, securing a 74.96% confidence vote on June 8. From scalability to post-quantum security, the *Yes* vote means that IOR will continue to deliver on foundational research and technology innovation through all Cardano Vision 2026 programs, building on the success of Cardano Vision 2025.

The sections below expand on these outcomes. While the technical delivery of some of this work is complete, IO is engaging independent partners to carry out an assurance review of the output, in line with Intersect's standard administrative process, ahead of formal milestone acceptance.

1. Keeping the core stack reliable, secure, and fair

Cardano's node software is the backbone of the network, and keeping it reliable, lean, and fairly incentivized is a continuous effort. This quarter's work spanned the whole stack: shipping node releases that kept performance steady while reducing technical debt, reducing the node's memory footprint, hardening key management for stake pool operators, evaluating whether staking rewards still fairly serve small operators, and laying the groundwork for nested transactions.

Core maintenance and support

Keeping the node reliable and free of accumulated technical debt is what makes every other initiative in this report possible to ship safely.

What was delivered. Cardano node v.10.7.1 shipped in April, following an iterative process that isolated and resolved several performance regressions, resulting in steady gains in block production, diffusion, and adoption metrics, alongside a substantial drop in CPU usage. Node v.11.0.1 followed, benchmarked specifically under protocol version 11 and confirmed to introduce no regressions relative to 10.7.1. PV11's cost model and execution budget changes have already been submitted as a governance action on mainnet, and the benchmarking infrastructure's PV11 overlay is close to being merged. Separately, the legacy tracing backend was fully removed from cardano-node in June, cutting roughly 11,000 lines of dead code across 15 modules and nine build dependencies, with the experimental RTView dashboard retired in favor of a new Grafana datasource and timeseries HTTP API.

Why it matters. Confirming that each release introduces no regressions and that governance-submitted parameters behave as expected lets SPOs upgrade with confidence rather than caution. Retiring a decade-old tracing system removes maintenance burden that would otherwise accumulate indefinitely, keeping the codebase easier to reason about as new features like Leios are added.

What happens next. With legacy tracing fully removed, the remaining components are being restructured into a self-contained Hermod Tracing System, decoupling their release cycle from cardano-node. The protocol version 11 overlay is expected to merge shortly, completing preparation for the intra-era hard fork.

LSM: beta feedback confirms the memory win

Following the release of cardano-node versions 10.7.1 and 11.0.1 with the on-disk LSM ledger-database backend, IO invited beta users to share feedback through community channels and a structured questionnaire. The response, drawn predominantly from mainnet operators, was strongly positive.

- Resident memory use fell from roughly 15 to 16GB on the in-memory backend to 5.0 to 5.8GB with LSM, a reduction of about 65%
- No regressions in startup time or CPU usage, and no stability issues: no crashes, hangs, corrupted state, or replay loops
- Mean overall satisfaction of 4.8 out of 5.

The main theme in the feedback was documentation rather than any functional defect, so the team reworked the snapshot-converter tool into a clearer command-tree interface and published matching reference documentation for the tool and its configuration values.

KES agent: beta period complete

The KES agent, which separates cryptographic key management from the main node process to reduce the attack surface for stake pool operators, was shipped in Q1. Feedback was gathered through the same SPO channels, and the agent has reached its 1.0 milestone, closing out the beta phase that began with its integration into node 10.6. The KES agent is ready for use on mainnet block producers.

Ledger-HD: the structural blocker is removed

Every Cardano node keeps the entire ledger state in memory, and that state has grown with the network. Earlier UTXO-HD work moved the largest piece, the UTXO set, to disk, but consensus could only manage that one table, leaving other growing components stuck in memory.

What was delivered. A redesign of the consensus foundations that removes the hard-wired single-table assumption, so consensus no longer needs to know the shape or number of on-disk tables. It is implemented across every consensus layer, builds on all four supported GHC versions, and has the full test suite and benchmarks ported to it, while being smaller than what it replaced and far more general.

Why it matters. Lower memory requirements keep the cost of running a node within reach of ordinary hardware, supporting decentralization: the more people who can afford to run a node, the healthier the network. This work is invisible to users, with no behavior changes, but it keeps the door open for SPOs, developers, and individuals to participate without expensive infrastructure.

What happens next. The branch is complete and reviewable. Merge into main is intentionally deferred to a window that does not collide with the higher-priority Ouroboros Leios workstream. Once merged, incremental migrations of tables can begin. Moving the InstantStake dataset to disk, the next planned migration, has slipped and its timeline is to be confirmed.

Revised stake pool incentive scheme: a proposal, in public

Cardano's staking rewards shape who can sustainably run a pool, and there is a clear opportunity to calibrate them so small and independent operators are rewarded fairly for the capital they commit. This quarter, IO published a full evaluation of the current reward formula, along with a sized proposal to strengthen it, aimed at widening participation and reinforcing decentralization.

What was delivered. A full design-evaluation report on Cardano's stake-pool reward mechanism, moving from how the mechanism was intended to work, through five years of mainnet evidence, to a sized proposal for repairing it. The analysis identifies nine structural problems (five affecting individual operators and delegators, four at the system level), centered on a pledge-bonus function that penalizes small pools, rewards under-committing pledges, and leaves about 95.6% of the pledge budget unspent, flowing back to the reserve instead. Four existing community proposals (CIP-0023, CIP-0037, CIP-0050, and CIP-0082) were evaluated against these problems. The report was followed by a live show-and-tell for the community and DReps in June, with a published recording and supporting analysis.

What happens next. The reform is intended to be carried into a CIP, authored and ratified in 2026, and implemented later as a hard fork following the Dijkstra ledger era. In the interim, lowering the minimum pool cost is a governance lever that requires no engineering work.

Nested transactions (Babel fees)

Nested transactions, first introduced through CIP-112 and CIP-118, let a Plutus script be enforced as part of transaction validation, so fees can ultimately be paid in tokens other than ada. This quarter's work focused on the PlutusContext definition needed for scripts to support nested transactions, and on the start of conformance testing for the Dijkstra ledger era.

This milestone's revised target is September 2026, in line with the rest of the Dijkstra-era work, which is delivered together as a single-era upgrade currently targeted for Q3 2026.

2. Plutus

A blockchain is only as useful as the applications built on it, and only as trustworthy as the tools available to secure those applications. This quarter's work made smart contracts cheaper to run, and gave developers and auditors a growing set of ways to verify that code does what it claims.

Plutus: a native Value type, and a faster Plinth compiler

Introduced Value as a built-in type. Value (Cardano's native multi-asset representation) is now a first-class primitive in Plutus, with a suite of operations for efficient, invariant-preserving handling. Smart contract developers and language authors across the ecosystem, including Aiken, Plutarch, and Helios, no longer need to implement and maintain their own Value libraries independently; they gain a single, authoritative representation. The feature will be enabled at the Van Rossem hard fork.

Shipped Plinth compiler and library enhancements. A set of compiler improvements makes Plinth contracts smaller and faster to run, meaning lower transaction fees and tighter execution-unit budgets. These reduce the cost barrier for users and enable higher on-chain throughput, while clearer error messages and better type-checking lower the learning curve for new developers.

Released a new Untyped Plutus Core (UPLC) command-line tool with certifiers for UPLC optimizations. An updated UPLC command-line tool has been released that optimizes Cardano smart contracts with a single command. It cuts execution costs by over 10% on average (up to nearly 50%) for real mainnet scripts, while supporting standard formats like CIP-57 blueprints, reporting before-and-after cost and size metrics, and providing certifiers that verify that optimization doesn't change script behavior.

3. Cardano High Assurance

Automatic formal verification: fully formalized ledger and Universal Annotation Language

Cardano High Assurance is building the tools needed to formally verify smart contracts without requiring every developer to write a formal proof themselves. This quarter's milestones completed the blockchain formalization layer underlying that goal, as well as the Universal Annotation Language that will be used to specify the business logic or security invariants that the different smart contracts of a DApp should satisfy.

What was delivered. PlutusContext V1, V2, and V3 are now formalized, along with the relevant set of ledger rules for smart contract verification, a specified and implemented state transition system, and temporal operators validated against classic model-checking examples. The specification of a Universal Annotation Language was also delivered, along with a set of examples showing how to use it on real smart contracts.

Why it matters. Community members can now use the CardanoLedgerAPI formalization and the Universal Annotation Language to specify smart contract correctness properties, and the Plutus Core formalization to run and prove contracts at the UPLC level, meaning proofs can come from any surface language.

What happens next. These formalizations will be used to prove and disprove real examples together with the Universal Annotation Language. This will let developers annotate smart contracts at the surface-language level, followed by the Plinth Bridge, which connects Plinth directly to Lean 4.

Property-based testing: the library tool ships

Property-based testing lets Cardano developers confirm their smart contracts behave correctly before reaching mainnet, automatically generating and running large numbers of test scenarios in place of manual, case-by-case review. This quarter, that capability shipped as a usable library.

What we delivered. A property-based testing library that is usable via regular CLI calls and can stream test data to a JSON-RPC server for the future Visual Studio Code extension. Everything is integrated directly into the standard Haskell build workflow, which lets developers define initialization states, preconditions, expected transitions, and validation checks, then automatically generates and runs test scenarios against them using QuickCheck.

The team also delivered a Claude skill that helps users define this testing interface and the necessary elements, making it even smoother to use.

It ships with built-in coverage reporting, 11 ready-to-use threat modules covering common attack vectors from datum bloat to token forgery, a framework for writing custom attacks, and five or more worked examples, including Aiken-based contracts and Invariant0 capture-the-flag challenges. All of it is open source, and every commit went through expert peer review.

What happens next. A VSCode extension for visualizing counterexamples and coverage in-editor is planned for September 2026.

Static analyzer: the rulebook is built

Plu-Stan, IO's static analysis tool for Plinth, is built to help developers catch security weaknesses and performance issues before code reaches mainnet. This quarter's milestone delivered the agreed catalog of coding rules and antipatterns that the tool checks against.

What was delivered. Two independently compiled lists of antipatterns and rules, one developed with Anastasia Labs and one with TxPipe, both drawing on public and private audit findings and each rule's own knowledge of common weaknesses. Thirteen of those rules have already been implemented in Plu-Stan to demonstrate suitability for static analysis, and each rule carries a category and severity.

The team also delivered a release pipeline with binaries for popular GHC versions and architectures, along with a Visual Studio Code extension that allows easy navigation to the different findings, each with a textual description.

Why it matters. Cardano smart contracts power real transactions, so consistent, expert-level feedback while code is being written is a real advantage. A shared, audit-informed rulebook gives every Plinth developer that feedback in the editor, before deployment rather than after the fact.

What happens next. Continuous improvement: two new rules identified in the findings, and an AI integration, as additional deliverables not originally planned but that make the tool much better.

Transaction monitoring system: the dashboard ships

The transaction monitoring system (TMS), built in collaboration among IO, AP901, and Politecnico di Milano, is designed to detect attacks and exploits across the Cardano network in near real time. The detection engine that scores transactions across nine attack classes shipped previously; this quarter, the operator-facing layer that makes those scores usable was delivered.

What was delivered. An open source repository covering the full alert-review lifecycle: a real-time view of risk alerts filterable by attack class and severity, a live feed of recent transactions and blocks, a per-alert detail view exposing the full risk score and its underlying signals, false-positive archiving with a required reason and an audit trail, CSV import for externally sourced attack data, a reporting view with configurable time windows and export, and role-based access for admins and reviewers.

Why it matters. A score sitting in a database is not usable on its own. The dashboard turns detection output into a workflow that SPOs, node operators, and security researchers can actually act on, and because it is open source, anyone can deploy it against their own data rather than depend on a hosted service.

What happens next. Development on the TMS continues. The team is now reviewing three components: a machine-learning backend for detecting major fraudulent attacks, a dashboard for transaction-level visibility, and an alerting system for reports and notifications. Full shipment is targeted for Q3 2026.

4. Scaling: Leios, Mithril, and Hydra

Leios, Mithril, and Hydra tackle three different layers of Cardano's scaling challenge: Leios increases the base layer's capacity, Mithril makes it faster and cheaper to verify the chain, and Hydra enables high-speed transactions for use cases that need them. All three advanced this quarter, and Leios reached its most visible milestone yet: a public testnet.

Leios: the testnet is live

Ouroboros Leios is Cardano's protocol-level scaling upgrade. It increases the number of transactions the base layer can process by introducing a second, larger block type alongside the existing Praos block, brought into play only when demand requires the extra capacity. Neither block type replaces the other. Q1 2026 saw a working Leios prototype produce and distribute endorser blocks in a local, multi-node network; this quarter, that work moved from local simulation onto a live, public network.

Musashi Dojo: Leios goes public. On June 23, IO launched the Musashi Dojo public testnet for Ouroboros Leios, in collaboration with the Cardano Foundation, Intersect, SundaeLabs, Tweag and

Blink Labs. It is the first time Leios has run on a public network rather than in simulation. IO operates the network, while independent SPOs run Leios-enabled block producers and participate in consensus, and builders can test DApps, wallets, and infrastructure ahead of the eventual transition to mainnet. The testnet hosts successive node releases, from early prototypes through release candidates, and will run through the mainnet hard fork.

Why it matters. Leios changes how much Cardano can carry, not the contracts already running on it. The design pairs a small, fast block with a larger one brought in for capacity, an approach named for Miyamoto Musashi's two-sword strategy, Niten Ichi-ryu, or 'two heavens as one': two complementary tools, stronger together than either alone. The testnet gives SPOs and builders a live environment to adapt for Leios on their own schedule, and lets builder feedback feed directly into parameter tuning and implementation ahead of launch day.

The program runs through five phases, each named after a chapter of Musashi's Book of Five Rings, and moves through them in sequence, though work on one stage can overlap the next:

- Earth: basic protocol design validation
- Water: exploration of protocol parameters
- Fire: real-world performance testing, the most critical phase for SPOs, spanning varied software, operating systems, architectures, and hardware
- Wind: adversarial testing
- Void: final preparation ahead of the mainnet hard fork.

What happens next. IO is targeting an initial 5 to 20 times increase in Cardano's base-layer throughput capacity, validated across protocol design, parameter choices, and implementation, with a mainnet hard fork targeted for later in 2026. Operator documentation is published and updated through each phase; builder documentation is in development, with CIP-164 and the Ouroboros Leios repository available in the meantime.

Mithril: from succinct proofs to a specialist handover

Zero-knowledge proofs: prototype to testnet in one quarter

Mithril's proof system determines how large and how portable its certificates are. A concatenation-based multi-signature can weigh up to 150KB; a SNARK-based one comes in under 10KB, and a recursive SNARK compresses an entire certificate chain down to tens of kilobytes.

What was delivered. A prototype zero-knowledge proof system for Mithril using zk-SNARKs (MS8.5) shipped alongside tighter Mithril and Cardano integration in releases, documentation, and binaries in mid-April. By late June, that work reached testnet with a recursive proof system compressing the full certificate chain, alongside certification for two new data types, blocks and transactions, with fine-grained control over finality.

Why it matters. Succinct, portable certificates enable proving on another chain, such as Bitcoin, that something happened on Cardano. That is the mechanism the Pogun team is building on to wrap and unwrap Bitcoin on Cardano. It also opens the door to verifying Mithril proofs directly inside Cardano smart contracts, since the underlying Midnight ZK library is expected to gain Plutus support in the future. Certifying blocks and transactions individually, with fine-grained finality control, gives light clients and bridges a way to verify a small transaction quickly or a high-value one with full finality, as the use case demands.

What happens next. Bringing the recursive prover to a production-ready implementation for the Pogun team's Bitcoin use case, and supporting bridges, wallets, and DApps as they build on block and transaction certification.

A new phase: Mithril moves to Teragone

On June 26, IO announced that responsibility for Mithril's specialist development is moving to Teragone Solutions, the team that has worked closely with the protocol since its earliest releases. This is a change in who leads development, not in the protocol or its guarantees.

- Mithril is already in production. Participating SPOs continue to sign and verify the certificate registry without interruption.
- The handover follows IO's standard model for specialist partners: milestones gated through Intersect, development shipped to public repositories, and a transition path agreed before the work begins.
- Planned next steps, now under Teragone, include wider SPO participation, broader client library support, and deeper wallet and infrastructure integration.

Hydra: two releases, partial fanout, and a simpler protocol

Two node releases, on schedule. Hydra-node 2.1.0 shipped in May with improved deposit security, a new SQLite-backed event store, a roughly 7% reduction in snapshot confirmation latency, and compatibility with cardano-node protocol version 12+. Hydra node 2.2.0 followed in June, packaging the partial fanout work, a revamped hydra-tui, and a substantially expanded benchmark suite. 2.2.0 also unified the protocol's on-chain membership checks around a single BLS-accumulator hash, collapsing three separate UTXO-hash fields into one and reducing the snapshot-signing tuple from seven fields to four.

- **Why it matters**. Lower confirmation latency and reduced per-snapshot overhead directly raise the throughput a head can sustain, and protocol version 12+ compatibility keeps Hydra aligned with the current mainnet node so operators can upgrade without being left behind.

Directly open heads: a simpler opening protocol. The 'directly open heads' work landed, removing the separate collect-commitments and abort steps from the head-opening flow so a head can be

opened directly. Leftover handling from the old path was removed from the validators in the process.

- **Why it matters**. Fewer on-chain steps to open a head means a simpler protocol surface, less that can go wrong during setup, and a faster, cheaper path from initialization to a live head, which lowers the friction for anyone standing up a head in production.

Partial fanout: the UTXO-per-head ceiling is removed. After many months of design and review, partial fanout was merged in June. A head holding a large UTXO set can now be closed out in multiple steps rather than a single transaction, with each step verified on-chain through a BLS-accumulator membership proof and its size chosen dynamically so each step packs in as many outputs as will fit.

- **Why it matters**. Previously, a head could only be fanned out if its entire UTXO set fit into a single transaction, thereby capping how many UTXOs a single head could ultimately settle. Removing that limit is what lets high-throughput, long-lived heads, the ones real production use cases need, grow to scale without hitting a hard settlement ceiling.

A revamped terminal interface. The hydra-tui was significantly reworked, adding pending-deposit recovery in both open and closed states, a persistent dark and light theme toggle, an event-history filter, tab navigation, and a fuel verification key option so the funds view can show the head's ada and a separate fuel UTXO side by side.

- **Why it matters**. The TUI is how many operators actually watch and drive a running head. A clearer, more capable interface makes it easier to monitor state, recover deposits, and diagnose multi-party scenarios without resorting to raw API calls, thereby lowering the day-to-day operational burden of running a node.

A benchmark suite that makes throughput claims verifiable. The 2.2.0 release added a real-world transactions-per-second metric, a Mixed UTXO generator, and a matrix command that sweeps across cluster sizes and UTXO shapes, published to a public scenario-benchmarks page with PR-versus-master comparisons on every change.

- **Why it matters**. Hydra's throughput has often been quoted loosely. A reproducible, public benchmark methodology replaces one-off figures with numbers anyone can check and reproduce, which is what a treasury-funded delivery claim should rest on.

What happens next. Continued production hardening, performance improvements, support for live users and workshops.

Notes on evolving initiatives

Building infrastructure that lasts sometimes means changing direction, or handing work to the team best placed to carry it forward. Two initiatives wound down this quarter, and one changed hands.

Acropolis and Tiered Pricing: wound down, funding returned

As announced on April 7, IO halted further development of Acropolis and Tiered Pricing to focus on making Cardano easier to build on and use, returning ₳4.1M in unspent allocation to the Cardano treasury (₳1.4M from Acropolis, ₳2.7M from Tiered Pricing).

Acropolis. Before development ceased, the project delivered on its goal of a modular, Rust-based node architecture: a Data Node offering read-only data access, including a Blockfrost-compatible API implemented directly in the node, that syncs with the blockchain in about an hour rather than the four days a full validating sync can take. Two working demonstrations emerged from the project: a production rewrite of SundaeSwap's order-execution software ('Scooper') and a Midnight Cardano indexer that replaces DB Sync and the Cardano node in Midnight's configuration. The Acropolis codebase remains open source, and the team of engineers has moved on to chain abstraction projects that remove technical hurdles between Cardano and developers and users who haven't arrived yet.

Tiered Pricing. The mechanism was originally designed to manage network traffic, but the upcoming Leios upgrade changes the foundation of how Cardano handles transactions, making the original design obsolete before it shipped. Continuing would have created technical debt rather than value, so the remaining allocation was returned in full.

Mithril: a specialist partner takes the lead

As covered above, Mithril's development moved from IO to Teragone Solutions in June. Unlike Acropolis and Tiered Pricing, this is not a cancellation: Mithril remains in active production and continues to progress, now under the team with the deepest cryptographic expertise in the protocol.

Looking ahead: Q3 2026

Momentum from Q2 carries into several milestones due in the third quarter:

- Leios: the Musashi Dojo testnet progresses through its Water and Fire phases, with a mainnet hard fork targeted for later in 2026.
- Nested transactions (Babel fees): Plutus context and conformance testing, alongside the rest of the Dijkstra-era feature set, targeted for the Q3 2026 hard fork.
- Property-based testing: the Visual Studio Code extension to enable a richer User Experience.
- Blaster: A full bridge from Plinth with Universal Annotation Language to Blaster.
- Transaction Monitoring System: the AI-based detection backend and the alerting system, with full shipment targeted for Q3 2026.
- Automatic formal verification: the Universal Annotation Language and Plinth Bridge.
- Mithril: continued development under Teragone, including wider SPO participation and broader client and wallet integration.
- Hydra: continued production hardening, performance improvements, support for live users and workshops.
- IOR: the mid-year reports will be published soon, providing progress updates on the Cardano Vision workstreams.

This report is compiled from IO's weekly development reports (April to June 2026), milestone acceptance documentation submitted to Intersect, and IO's public announcements. All referenced work is publicly verifiable through IO's GitHub repositories, Essential Cardano's development updates, and Intersect's milestone tracking.