DECISION FOR THE DEVELOPMENT TEAM

Should Auriq hire Halborn?

An independent assessment of Halborn’s public work, the incidents at peer networks, and the evidence Auriq should require before mainnet.

> **RECOMMENDATION**
> Invite Halborn to quote a Layer 1 protocol and implementation audit. Make the choice contingent on the assigned specialists, written scope, and independent verification of fixes.

What Auriq would be buying

An outside team would challenge the design assumptions, inspect the node and wallet at a recorded code revision, and report where behavior departs from the rules. Auriq would own the fixes. Halborn should retest them on the release candidate that miners, stakers and wallet users will download. Halborn sells a Layer 1 service covering consensus, validators, and network resilience. [5]

Why the fit is credible

Halborn has reviewed an independent proof-of-work chain with post-quantum signing, assessed staking-related logic for Babylon, and helped Dogecoin verify repairs in a shipped node release. The remaining hiring question is who will review Auriq’s unique transition from two mining algorithms to stake-led chain selection. [6], [9], [10]

What could still go wrong

Auditors can miss a defect. A correct implementation can also depend on assumptions that fail in production, such as an eligible staker staying online or operators installing the correct update. Reserve custody, software distribution and live monitoring each need a named owner.

Status on November 20, 2026

Auriq’s public outline describes development and a conditional testnet path. Its v0.4 litepaper promises independent findings before mainnet. The public materials reviewed did not identify Halborn as a signed auditor or show a completed Auriq audit. The project links its node and wallet on GitLab, but anonymous access did not provide a release candidate for this assessment. The team’s secondary developer identifies January 2027 as a working target. Auriq’s public outline makes launch conditional on published test results. [1], [2], [22]

What holders and exchanges could verify

Publish a report naming the audited release, the defects found, the fixes independently checked and the work excluded. That gives a pool, wallet operator or exchange evidence for its own due diligence. A completed audit cannot promise a listing, adoption or a market price.

PROVIDER EVIDENCE

Where Halborn has earned credibility

These examples show what individual reviewers examined and what teams could verify afterward. Auriq should request comparable detail in its own proposal.

Published work	What it establishes for Auriq
Soqucoin node February–March 2026	Halborn reviewed block validation, wallet encryption and post-quantum code on a proof-of-work chain. Its report recorded 30 findings, including three critical and one high, and marked all 30 solved. They included potential chain disagreement and wallet recovery failures. Soqucoin used ML-DSA-44. Auriq specifies ML-DSA-65. [6]
Soqucoin follow-on March–April 2026	A separate review of another cryptographic subsystem found four critical issues, one high and one low. All six were recorded as solved. The first report's closure counts applied only to its original scope. [7]
Quantum Resistant Ledger	The review covered a JavaScript ML-DSA-87 package and an older Dilithium variant. It documented the exact commit and examined signing and verification behavior. This is relevant experience within the algorithm family. Auriq's library, parameter set and signing rules need their own review. [8], [20]
Babylon Genesis October 2025 fieldwork	A client-hosted report describes one engineer, four weeks, selected repositories and identified commits. Its tests addressed staking and validation logic, and the report recorded partially addressed recommendations. Ask for named specialists, time by workstream and remaining findings. [10]
Dogecoin Core Maintainer confirmation	Dogecoin maintainers confirmed that Halborn reported critical peer-network defects and verified fixes in v1.14.6. Older nodes still needed to upgrade. This is client-side evidence that Halborn can find a defect in a Bitcoin-derived node and verify the repair. [9]

The gap to close in procurement

Request an example from the proposed lead covering fork choice, low-participation staking, difficulty changes and supply accounting. Commission a second specialist for any area the proposed team cannot substantiate.

How to compare bids

Seek one comparable proposal against the same written scope. Give greater weight to named reviewers, examples of relevant findings and evidence of fix verification than to a provider's client count. If most of the budget buys preliminary design review, reserve a separate amount for hands-on review of the final software.

INCIDENT EVIDENCE

Would an audit have prevented these?

A relevant review could have caught a defect earlier. Historical reports cannot establish that any specific firm would have done so.

Salvium | Unauthorized minting

Cypher Stack's July 2024 review evaluated the described transaction protocol and explicitly limited conclusions about any implementation. After fixes at the end of December, Salvium reported an omitted cryptographic check. Its March 2025 supply reconciliation found that approximately 9,459,087 SAL entered circulation early and were sold through exchanges. An audit of Auriq's actual issuance code and a separate supply calculation address the corresponding risk. [11], [12], [13]

Salvium | June 2026 network and release disruption

Release notes document spam, inconsistent local node databases, temporarily unspendable funds and wallet scanning issues. During recovery, a build issue distributed an incorrect hard-fork height. MEXC suspended deposits and withdrawals on June 25 at the project's request. Auriq should verify the delivered binary, rehearse recovery with an outside operator and compare fresh and existing node state. [14], [15]

XELIS | June 2026 accounting defect

The v1.22.1 changelog documents an incorrect refund in scheduled virtual-machine executions that could create excess XEL. The project reported exploitation and said the counterfeit coins were removed from the accepted chain before reaching the market. The changelog verifies the code issue. The market and containment account comes from the team. For Auriq, test every path that issues, pays or burns coins, including reversal of a block. [16], [17]

Zcash | Repeat review found a critical issue

QEDIT's earlier NU5 review reported zero critical findings. In May 2026, a researcher carrying out an ongoing protocol audit identified a critical Orchard circuit issue, leading to a coordinated emergency change. These reviews concerned evolving software. The lesson for Auriq is to repeat reviews after material changes and keep a private reporting path open. [18], [19]

> **IMPLICATION**
>
> The closest practical defenses are independent accounting, testnet recovery drills, and release verification. Auriq's transparent ledger makes it feasible for an outside operator to reconcile supply from genesis.

PROPOSED AUDIT SCOPE

What the reviewers should prove

Use the v0.4 litepaper and the promised consensus specification to define expected results. These are proposed checks for the shipping implementation. [1], [2]

Workstream	Evidence to require from Halborn and the team
Issuance and fee burning	Independently calculate every new coin and burn through genesis, halvings, the last pre-tail block and the four-coin tail. Confirm the 80M reserve plus 688M in block rewards at the cap, then ongoing tail issuance. Exercise the burn ceiling, tips, surplus base fees, rounding and reversal of blocks. Below the ceiling, fees of 4 AURIQ must offset a 4-AURIQ subsidy exactly. Fees of 0.5 AURIQ must leave 3.5 newly added. Tips only transfer coins.
Chain selection and staking	Test disagreement and reconnection across both mining algorithms, block 129,600 activation, the five-work-block limit, separate difficulty changes, the 1,440-block reorganization rule and the final handoff to stake-only blocks. Rehearse few, zero and returning eligible stakers. Easier difficulty still requires a signer when a stake block is mandatory.
ML-DSA-65 and wallets	Check the exact library and version against known signing and verification results. Verify the wallet and node authorize identical transaction data and that spending keys stay offline during cold staking. Test restoration on a fresh machine, including mining rewards and staked balances. Measure validation time and storage under realistic block load. [20], [21]
Checkpoints and archives	Validate the process for approving and distributing checkpoints. Have an independent archive supply all historical signature data for verification from genesis. Compare the resulting unspent outputs and supply with an archival node, the default witness-pruned node and a node restored after an interruption.
Reserve and key custody	Verify that reserve addresses and their change cannot stake, while paid-out coins follow the normal rules. Rehearse authorization, recovery and eventual multisignature migration with test keys. The founder initially controls one published address. Record who can detect and respond to a compromised key. [3]
Release and upgrade integrity	Identify the Bitcoin Core baseline, local changes, dependencies and activation parameters. Have an outside operator reproduce the signed release, verify published checksums, and run the upgrade from public instructions. Confirm miners, wallets and independently run nodes agree after the upgrade.

One fork-choice rule needs reconciliation

Page 3 describes a geometric mean of two work types before staking, followed by stake-led chain trust. A November 19 activity entry describes a geometric mean of three work types. Ask the developers and reviewers to reconcile the wording, the release code and example competing chains in each phase. The brief activity entry alone cannot establish a defect. [1], [23]

PROCUREMENT AND LAUNCH

What should be in the agreement?

Get a written quote for each phase and identify its owner. Reviewer time, scope and retests determine the value of the engagement.

Decision	Request or acceptance condition
Before signing	Name the consensus and cryptography reviewers. Obtain comparable work, availability, person-days per workstream, price per phase, scope exclusions, number of retests, cost of design changes and emergency availability. Ask who can review a fix independently of its author.
Design and implementation	Begin with the consensus design, supply rules and trust assumptions. Freeze the candidate and list exact repository revisions, dependencies, build settings, and custody or operational work covered. Require each finding to name impact, owner, status and evidence from retesting.
Before mainnet	Publish scope, exclusions, version and the status of findings after coordinated fixes. Require closure of all critical and high findings. A medium finding threatening funds, supply, chain agreement or recovery should also hold launch. Record owners and dates for any other accepted issue. Link the report to the shipped binaries.
After launch	Assign operators to compare independent supply totals, node agreement, stale blocks and wallet sync. Monitor staking participation around activation and the availability of independent archives. Provide a private reporting channel, severity-based response times, release-signing procedures and contact paths to pools and exchanges.
When code changes	Review the changes between the retested commit and the production build. Reaudit material changes to consensus, signatures, the wallet, fees, checkpoint policy or custody. Before staking activates, repeat its rehearsal against the actual release and current operator participation.

Budget and independence

The public outline says the founder funds early work. The first-year reserve policy allows AURIQ-denominated payments and forbids sales or conversion of the reserve. Confirm funding before commissioning the audit. Ask for a cash quote and retest costs. If the auditor accepts AURIQ, disclose the relationship and agree on a fee independent of the findings. Keep capacity for remediation and a second opinion on consensus. [2], [3], [4]

SOURCE NOTES

Sources and reading guide

Each numbered title opens the original publication. Project incident statements are identified separately from software release notes.

01 **Auriq** · Litepaper v0.4, pp. 2–8

02 **Auriq** · Project outline and testnet exit criteria

03 **Auriq** · Strategic Reserve policy

04 **Auriq** · Charter and emergency authority

05 **Halborn** · Blockchain Layer 1 assessment

06 **Halborn** · Soqucoin blockchain node report

07 **Halborn** · Soqucoin additional review

08 **Halborn** · Quantum Resistant Ledger library audit

09 **Dogecoin Core** · Maintainer disclosure on release 1.14.6

10 **Babylon Labs** · Halborn report on Babylon Genesis

11 **Cypher Stack** · Salvium transaction protocol review (paper mirror)

12 **Salvium** · Initial inflation vulnerability findings

13 **Salvium** · Supply reconciliation, March 2025

14 **Salvium** · June 2026 recovery releases

15 **MEXC** · June 25 deposit and withdrawal notice

16 **XELIS** · Core changelog, release v1.22.1

17 **XELIS** · Project's public incident statement

18 **QEDIT** · Zcash NU5 / Orchard audit

19 **Zcash Foundation** · Emergency Orchard upgrade report

20 **NIST** · FIPS 204, ML-DSA standard

21 **NIST** · ML-DSA algorithm test specifications

22 **Auriq** · Node, wallet, and release links

23 **Auriq** · Public repository activity feed, November 19 entry

This assessment draws on Auriq's public design and project status, auditor reports, maintainer releases and incident statements available on the review date. The scope and launch conditions on pages 4–5 are recommendations for the development team.