Quip Network

The Worldwide Quantum Computer

Whitepaper — Version 1.0

Quip Foundation

Abstract

The Quip Network is a decentralized high-performance computing protocol with a spot compute clearinghouse, an open-source library of composable solvers, and a hybrid quantum-classical router and scheduler that supports virtualization and entanglement. Consumers bid for idle compute and use expert-submitted solvers, while operators compete for the bids with heterogeneous processors, solve jobs, and publish Proofs of Useful Work through Nakamoto-style consensus. Mining pools progressively scale throughput and entanglement, while hybrid workflows route each part of a job to suitable hardware. Because the same hardware can break the cryptography used by most decentralized systems, Quip also provides a quantum-resistant peer-to-peer transaction network and a Byzantine fault-tolerant messaging layer that does not depend on the structure of an underlying transaction network.

Contents

1 Motivation

Doubling the power of a compute cluster with two thousand graphics processing units ("GPUs") requires another two thousand chipsets, while a cluster with two thousand quantum processing units ("QPUs") doubles its computational power by adding one fully connected qubit. The practical limit on quantum computation is therefore the number of qubits that distinct, mutually distrustful operators can entangle, not the number on one die. Reaching that limit requires shared programs that tolerate adversarial actors and accept new processors without permission. Noisy Intermediate-Scale Quantum ("NISQ") systems cannot yet support geographically distributed entanglement, but hybrid quantum-classical solvers already outperform comparable alternatives on several critical tasks.

Quip uses present hardware for valuable hybrid computation while preparing the coordination layer for global entanglement. It addresses both the coordination failures that have slowed progress and quantum computers' ability to break the cryptography that coordinates other machine networks. Those failures determine the protocol's structure, while economics and social choice connect useful computation to post-quantum security.

1.1 Coordination Problems in Quantum Computing

Four coordination failures keep the quantum industry below the point where researchers can discover, validate, publish, and sell genuine quantum advantage:

Siloed and shallow talent pool. Fewer than twenty thousand people work in the pure-play quantum industry worldwide [1]. Most work for government laboratories or private corporations that protect classified or proprietary research. Academic researchers can publish their work, but they rarely receive its economic return. Private knowledge usually reaches outsiders through expensive, custom consulting rather than open code or shared infrastructure. The labor market therefore rewards secrecy, while aspiring quantum software engineers lack a practical way to enter the field.

Unclear return on investment. A customer may identify a problem suited to quantum computation but still lack the information needed to buy a solution. The customer must understand the method's constraints, compare conflicting vendor claims, and measure quantum advantage against a tuned classical baseline. Published demonstrations of quantum supremacy often use contrived workloads, while data pre-processing and post-processing reduce the stated speedup. A CFO therefore needs a transparent benchmark that compares a quantum bid with an HPC quote on the same terms before approving the purchase.

Integration friction. Each quantum vendor provides its own SDK, compiler, calibration regime, queue, and pricing model, and problem formulations rarely move cleanly between architectures due to subtle differences in supported operations and processor topologies. Standard tools such as package managers, container orchestration, IDE-integrated debuggers, and declarative job specifications are missing or tied to one vendor. A team comparing two vendors must usually write its pipeline twice and sign an upfront contract with a five-figure reservation before running a meaningful benchmark.

Under-utilization and uneven queues. Quantum machines are super-cooled and capital-intensive. One superconducting system costs between ten and twenty million dollars, cannot be powered down without expensive recalibration, and carries fixed costs at every utilization level. A few large projects can fill a vendor's calendar before a multi-year laboratory experiment, leaving smaller exploratory jobs waiting indefinitely, only for the queue to drain while the large customer validates its results. Because the other coordination failures make replacement demand unpredictable, vendors lose revenue during idle hours, researchers lose access to hardware needed to prove value, and businesses remain skeptical.

1.2 Post-Quantum Security Risk

The hardware that can address these coordination failures can also break much of the classical cryptography that secures the internet. Recent results have shortened estimates for the arrival of a cryptanalytically relevant quantum computer ("CRQC"): Google improved the running time of Shor's algorithm [2] and cut the qubits required to break elliptic curve cryptography by over a factor of forty [3], [4], down from the twenty million noisy qubits estimated in 2019 [5], while the Pinnacle architecture [6] and Oratomic [7] independently pushed the requirement lower still — to roughly one hundred thousand physical qubits and ten thousand reconfigurable atomic qubits, respectively. Manufacturers have also promised more qubits, better error-correcting codes, longer coherence times, and horizontal scaling. Published estimates now place a break of ECC-256 within roughly half a million superconducting physical qubits and minutes of runtime [4], or as few as ten thousand reconfigurable atomic qubits over a period of days [7]. At the six watts per qubit behind the standard hundred-and-twenty-five-megawatt figure for a twenty-million-qubit machine [8], [9], ten thousand qubits draw about sixty kilowatts; at the 2025 United States average retail price of roughly fourteen cents per kilowatt-hour [10], a full month of continuous operation costs under six thousand dollars. Once a laboratory owns the machine, each additional broken key is therefore an operating expense of a few thousand dollars—far below what a single key protects. The Hudson Institute estimates that the financial system alone has more than three trillion dollars of value at direct risk [11].

Existing transaction networks have responded slowly. ACH and SWIFT have not upgraded to post-quantum certificates, Bitcoin's BIP-360 pay-to-quantum-resistant-hash proposal remains contested [12], and Ethereum's post-quantum task list gives no firm timeline for its open problems [13]. Waiting to rotate keys after the threat appears does not address attackers who collect public keys in advance and leaves unmigrated networks open to chain-wide reorganization attacks. Upgrading these networks in place is itself a multi-year coordination problem.

A protocol that organizes quantum hardware must address the security problem that the same hardware creates.

1.3 Protocol Objectives

The Quip Network directly addresses restricted talent, unclear returns on investment, integration friction, under-utilization, and post-quantum migration.

Paying a shared and open quantum talent pool. Quantum software engineers publish solvers to a permissionless open-source library. Publication reserves the invention, and each consumer job that invokes the solver pays its author a royalty.

Establishing an unbiased benchmark of profitability. CPUs, GPUs, ASICs, and QPUs compete on identical workloads submitted with cleartext bids. The resulting Proofs of Useful Work ("PoUW") compare platforms on the same job and replace vendor claims with measured results.

Accelerating time to the first impactful result. A hardware-agnostic execution layer gives consumers one interface and gives operators a deterministic job specification. The same interface works across the underlying hardware. Consumers use an application, an API, or a quantum codebase without needing a PhD.

Filling idle capacity with paying work. A spot clearinghouse routes pre-emptable jobs to available machines. This increases utilization, protects operator SLAs, and gives smaller exploratory workloads access to otherwise idle capacity. Access sells at cost when demand is low. Its price rises as consumers compete for scarce time.

Upgrading networks to quantum-resistance A hybrid quantum-resistant multisig wraps assets on classical transaction networks with post-quantum signatures. A peer-to-peer wallet swap protocol then lets users on ACH, SCT, SWIFT, Bitcoin, Ethereum, Solana, and other networks interact with Quip. Users keep their assets on the venues where they already run financial strategies.

2 System Goals

2.1 Distributed Quantum Computing

Computing has moved from isolated machines to shared systems, from mainframes and networked workstations to cloud computing and accelerators rented by the second. Quantum computing follows this direction with a different scaling rule because each entangled qubit multiplies the state space instead of adding capacity. A quantum network therefore becomes more valuable as operators entangle more qubits across machines they do not own or trust. The worldwide quantum computer is a literal objective: heterogeneous processors owned by mutually distrustful operators, joined into one coherent system by a protocol.

There is immense additional value enabled by the unique properties of quantum systems, which enable new consensus paradigms made possible by complementarity, no-cloning [14], and no-deletion [15], and which already admit protocol-level Byzantine agreement constructions with no classical analogue [16]. An emerging field of exploration across quantum game theory [17], [18], quantum auctions [19], uncloneable money and single-use contracts [20], [21], [22], and blind quantum computation [23], are poised to radically rework the way we think about distributed systems and multi-party computation. As an example, imagine self-enforcing contracts that mutate the result received by all the parties if any one party defects, or a result that is nothing more than random noise to an adversary,

but becomes crystal clear to any participant who contributed quantum information to the computation.

Building this system is a multi-decade infrastructure project because quantum repeaters [24], memory, and error-corrected links do not yet support geographically distributed entanglement at scale [25]. The coordination layer can operate before those components mature. Quip supplies a compute clearinghouse, an open-source library of composable solvers, and a hybrid quantum-classical orchestrator and virtualizer that routes each job to the processor that can solve it most efficiently. By clearing optimization workloads and generating verifiable quantum randomness on classical and NISQ-era hardware today, the network builds the economic and cryptographic base needed to add physical entanglement as it becomes available.

2.2 Compute and Security Requirements

The network exists to solve otherwise intractable problems through its clearinghouse, solver library, and orchestrator. The remaining components, including the cryptography, let that product operate in an adversarial decentralized setting.

A decentralized quantum network must defend against its own participants because its operators own the machines that can break ECDSA, RSA, and related classical signature and key-exchange schemes. Conventional ledgers depend on those schemes, so Quip miners could eventually forge signatures and rewrite network history if the protocol used classical cryptography. The network therefore requires post-quantum cryptography that its own operators cannot break.

2.3 Legacy Network Integrations

No existing permissionless and distributed protocol can execute quantum gate operations, so the compute core must run its own ledger and virtual machine. Because customers are unlikely to move their liquidity to a new network only to buy compute, Quip connects to the venues they already use.

Quantum-resistant wallets wrap assets in place on Bitcoin, Ethereum, and other host networks, allowing customers to keep their funds and existing financial workflows while transacting with the Quip compute economy. A peer-to-peer request-for-quote protocol moves value into and out of Quip without requiring bridges, oracles, or source networks to execute post-quantum upgrades. Counterparties do not mint and burn assets through custodial escrow, nor do they trust a vulnerable bridge operator or wrapped-asset issuer. They trustlessly swap the keys to single-use wallets, allowing settlement across chains that have not adopted quantum-resistant structures.

2.4 Deployment Model

The protocol develops in three stages, each gated by demonstrated capability rather than a date. The first directs all emissions to two subnets that current quantum computers can serve: one measures quantum advantage on a verifiable optimization problem, while the other provides a verifiable quantum randomness function. The second stage uses token-escrowed governance to open additional subnets that compete for emissions as operators

and researchers measure where hybrid methods succeed or underperform. In the final stage, mature physical interconnects support distributed QPUs to pool and entangle their capacity, moving the network from coordinating independent processors to assembling the worldwide quantum computer.

3 Hardware and Verification Constraints

3.1 Current Hardware Capabilities

Two hardware approaches dominate current quantum computing, and each supports different kinds of useful work:

Quantum annealers are special-purpose devices that physically relax toward the ground state of an Ising spin Hamiltonian. D-Wave's Pegasus-topology machines provide several thousand superconducting flux qubits and express QUBO and Ising problems directly through transverse-field quantum annealing [26], [27], but they cannot execute arbitrary gate sequences. They are therefore well-suited for combinatorial optimization, but lack the flexibility of gate-based platforms.

Gate-based platforms are general-purpose devices built on physical substrates with different trade-offs. Superconducting transmons from IBM, Google, and Rigetti provide fast gates and mature fabrication but have short coherence and mostly nearest-neighbor connectivity, while trapped-ion systems from IonQ and Quantinuum provide long coherence and all-to-all connectivity with slower gate clocks. Photonic and topological systems address other constraints: RIKEN's photonic processors reach megahertz-class clock speeds [28], while Microsoft's Majorana topological array pursues hardware-level error suppression [29].

A device's useful capacity depends on qubit count, qubit distance, coherence time, two-qubit gate fidelity and error rate, topology, and clock speed. Raw qubit count is a poor measure by itself. A deep circuit fails when accumulated gate error overwhelms its signal, regardless of the nominal qubit count.

Recent results have shortened prior timelines, but current machines remain limited. Google's Willow demonstrated *sub-threshold* error correction: adding physical qubits to a logical qubit lowered the logical error rate instead of raising it, providing the first empirical evidence that fault tolerance scales as theory predicts [30]. Reconfigurable neutral-atom arrays [31] and concatenated bosonic qubits [32] reached the same conclusion on other substrates. The photonic and topological work described above attacks the same limit through clock speed and stability.

Current systems remain *Noisy Intermediate-Scale Quantum* (NISQ), with error rates too high and logical-qubit counts too low for deep, fault-tolerant circuits. Genuine advantage therefore comes from shallow circuits and hybrid classical-quantum loops that finish each quantum step before decoherence destroys the result, not from Shor-scale algorithms. Inter-processor quantum networking is less mature because distributing entanglement over distance requires quantum repeaters [24] and quantum memory [33] that preserve state

within strict coherence budgets [25]. Quip therefore defers pooled logical computation across distinct operators to the final stage (§2.4).

3.2 Workload Readiness

The hardware constraints in §3.1 distinguish present workloads from long-horizon workloads. Today, annealers and NISQ devices can solve combinatorial optimizations expressed as QUBO or Ising problems, into which most standard NP-hard problems compile with at most a polynomial blowup in spins [34], the natural workload for annealers and the basis of Quip's flagship Ising subnet (§6). Current devices can also produce samples and verifiable randomness whose resistance to classical simulation is the useful output [35], [36], [37], supporting the Quantum Verifiable Randomness Function subnet (§6). Other present workloads include variational chemistry on small molecules, which runs shallow parameterized circuits inside a classical optimization loop, and selected machine-learning tasks such as kernel methods and feature selection. These workloads tolerate noise, keep the quantum step shallow, and use classical post-processing.

Long-horizon workloads require more error-corrected logical qubits than the field currently has. They include Shor factoring and discrete logarithm at cryptographic scale [2], large-scale Grover search [38], and deep quantum simulation of strongly correlated systems. Each requires circuit depths beyond current coherence and fidelity budgets.

This division sets the three stages in §2.4. The first directs emissions to Ising optimization and the QVRF because current hardware can run and cheaply verify both, providing an embedded benchmark to competing architectures. The second adds subnets through governance as operators identify profitable hybrid methods. Chemistry and machine-learning workloads enter as logical-qubit counts rise, while cryptanalysis activates only after fault-tolerant hardware arrives. Deep simulation and pooled entanglement remain third-stage workloads. The catalog therefore grows only once the hardware can perform the work.

3.3 Verification Constraints

A Proof of Useful Work is admissible only when checking the result costs much less than producing it, allowing the network to reward computation without repeating it. Quip groups useful quantum problems into three verification tiers whose rewards scale with their objectivity:

Cryptographic or deterministic verification. These results are hard to produce and easy to check. Integer factoring and the elliptic-curve discrete-logarithm problem are the standard examples because one multiplication confirms a candidate. Factoring and discrete log are computationally equivalent to the hidden-subgroup problem [2], so an advance applies across a family of applications.

Reproducible benchmarks and quality scoring. These results have no closed-form check but have deterministic scores that anyone can recompute. An Ising result is checked by recomputing $E = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$ and comparing it with an expected ground-state energy. Cross-entropy benchmarking (F_{XEB}) scores how closely random-circuit samples

follow the ideal distribution [35], supporting the QVRF certificate that a producer made a genuine quantum sample before a classical adversary could forge one [36], [37]. In both cases, results yield a quality metric which is comparable between submissions while being independently reproducible

Peer review, arbitration, and prediction-market oracles. These results cover claims with no cheap objective check, which includes much of variational chemistry and the QMA-hard complexity class, where an answer remains hard to verify for classical computers even after it is given. Until the network supports a sufficiently large quorum of quantum oracles, expert judgment, reputation-staked review, and market adjudication establish correctness instead of recomputation or direct validation.

The verification format contributes to a proof of useful work's subnet template and reward mechanism. Tier-one and tier-two problems use the deterministic PoUW templates in §6, where the protocol scores each submission and cheap checking permits trustless block validation. The Ising and QVRF subnets use these tiers, so they are natural candidates for first stage release (§3.2). Tier-three problems cannot settle through recomputation and instead use structured products such as parametric insurance and surprisal markets (§10.3), which reward outcomes according to their information content. These markets use speculative capital to evaluate quantum-advantage claims undecidable by other means, and hinge on escrow of capital using the quantum-resistant interlock protocol (§12.6). A problem's verification tier therefore determines whether it can support useful work and how the network prices and secures it.

4 Related Systems and Prior Art

4.1 Single-Vendor Quantum Platforms

Most quantum computing vendors sell a vertically integrated stack in which one company provides the hardware, SDK, solver catalog, calibration regime, and pricing schedule, while the customer must lock-in to that vendor's topology, ecosystem, and roadmap. IBM, D-Wave, IonQ, Quantinuum, and other cloud-mediated incumbents all use this same model. Often manufacturers choose to be their own cloud providers because data center operators are sensitive to the risk of these low-longevity and high-capex machines, and so resist comitting to a full purchase without some cost sharing or customer commitments.

For vendors to court potential clients, they must also prove advantage over the customer's highly optimized classical pipeline, which takes months of bespoke consulting. It is unhelpful that each architecture supports different tradeoffs in the space of qubit count and quality parameters, and so each vendor publishes their own unique benchmarks that are not obviously comparable.

This model prevents solutions from porting easily between vendors, and the anticipated cost of a high-touch onboarding process—along with sensitive export controls on the technology—discourages vendors from selling on-demand access. As a result, quantum

computer access is sold almost exclusively under reserved-time contracts or co-ownership agreements.

The Quip Network complements these services by enabling pre-emptable spot compute markets that fill in the unused gaps between reserved contract requests. Quip provides an open and neutral solver marketplace, removing the maintenance burden of downstream applications from the hardware providers, and hosts a vetted community of consumers providing zero-knowledge proofs of identity for compliance purposes.

Participants in the network are pseudonymous, and so can mine proofs of useful work to benchmark their machines without revealing their performance profiles unless they see clear advantage. Steady emissions from the network provide an amortizable lifetime value to the machines and so operators can purchase processors with confidence and procure financing from reputable institutions, while augmenting their existing client relationships with the spot bids available on the network.

Quip provides demand aggregation, verification, and settlement across competing vendors, while each vendor provides only its hardware.

4.2 Bitcoin Proof of Work

Bitcoin's SHA-256 proof-of-work converts electricity into security [39], but its hashes have no value outside transaction ordering and provenance. The network accepts duplicated work, full storage replication at every node, and rising energy use because these costs are the security-guarantee of the protocol. Quip spends compute for the same purposes of security, but these drawbacks to the Bitcoin model become assets to the Quip Network and the quantum industry: higher certainty of solution quality, living benchmarks of consumer demand and hardware usage, and much lower costs of energy.

The probabalistic nature of quantum computing changes the role of redundancy. When several operators solve one instance, the protocol gains higher certainty of the correct answer faster, and can compare quality, time-to-solution, and cost across heterogeneous platforms (§6), while stored solutions and circuit descriptions form a public benchmark and instruction-level record for future hardware and algorithm design (§2.1). Due to the nature of quantum decoherence, injecting unneeded energy into the system is destructive to the computation, and so the vast majority of cost of compute is simply the cost of cooling, a fraction of what equivalent clusters of GPUs consume per solution.

The emission models also differ. Bitcoin uses a hard-coded halving schedule that produces abrupt supply shocks on a fixed clock. Quip emits a fixed quantity of QUIP each year. As the number of subnets grows, that quantity is divided among more recipients and issuance per subnet falls (§7.2). Dilution therefore follows the creation of new useful-work categories rather than an external timer.

The chains also have different quantum exposure. A cryptanalytically relevant quantum computer can attack Bitcoin's deployed output set. The main proposed remedy, BIP-360 P2QRH quantum-resistant output scripts [12], requires a contested soft fork of a live multi-trillion-dollar chain, to mitigate on-spend attacks which recent resource estimates place within reach of a fast-clock CRQC [4]. Quip uses quantum-resistant cryptography from

genesis. Its Account wrapper (§12) can increase the security guarantees of value held on Bitcoin before the host chain completes its own migration.

4.3 Bittensor Subnets and Proofs of Useful Work

Bittensor is the closest precedent for Quip's subnet architecture [40]. Competing subnets receive shares of a fixed token emission directed by a vote-escrowed governance token. Bittensor showed at scale that emissions can start a multi-subnet compute economy and that holders will lock tokens to direct emissions toward subnets they value. veQUIP uses the same general form.

However, Quip changes the work, its evaluation, and its token economy. Bittensor subnets mostly produce machine-learning outputs that other miners score, so quality is a consensus among competitors rather than an externally redeemable good. A Quip Proof of Useful Work completes computation that a consumer has paid for, such as a solved optimization or a verified quantum result, and the buyer adds value to the baseline availability rewards. When the verification tier permits, an objective benchmark checks the result instead of a popularity vote (§3.3, §6).

Quip also uses one network-wide token instead of Bittensor's subsidiary token for each subnet, simplifying the tokenomics and removing the ceiling that prevents a subsidiary token from exceeding its backing supranet token. New participants evaluate one asset instead of a portfolio of subnet currencies.

Bittensor launches many subnets and lets them compete. Quip starts with two flagship subnets, Ising optimization and the QVRF, and first proves their mechanics, verification, and operator economics. veQUIP voting opens other problem classes only after that proof has been validated (§2.4). This sequence limits early breadth so the network can establish that its first useful work has genuine value.

4.4 Substrate Runtime and Polkadot Resource Allocation

The Quip Network uses Substrate for specific protocol functions [41]. Substrate's FRAME pallet system provides a modular runtime, while its BABE-plus-GRANDPA pair separates block production from finality. Quip uses this pair to start consensus before the canonical heaviest-useful-work rule takes over (§12.4). Because Substrate also supports forkless on-chain upgrades, Quip can add hash-based account signatures, a post-quantum transaction format, and other quantum-resistant changes to consensus and transactions (§12, §15). A chain such as Bitcoin would need contentious hard forks for the same changes.

Quip borrows economic principles and modularity from Polkadot, but not its parachain architecture. Polkadot's parachain slot auctions and Agile Coretime market price and allocate scarce blockspace through an explicit market. Quip applies that method to scarce quantum and classical *compute capacity*. Job bids auction the capacity, and veQUIP-directed emissions route it.

Quip is *not* a relay chain of parachains. Polkadot parachains are sovereign blockchains with independent state that lease security from a central relay chain. Quip subnets are problem classes, such as Ising optimization, the QVRF, and hidden-subgroup cryptanalysis.

They share one proof-of-useful-work chain, state, token, and validator-miner economy. Useful work anchors the blocks on that chain. Subnets divide the kinds of work the chain pays for, not the ledgers where it settles.

4.5 Hashgraph and DAG-Based Settlement

Hashgraph, Tashi, and related DAG-based ledgers settle value without one linear chain of blocks by recording transactions in a directed acyclic graph [42]. Gossip-about-gossip and virtual voting derive order instead of a leader-elected block sequence. These systems provide high throughput and low-latency finality, showing that strong settlement does not require Nakamoto-style linear ordering.

Quip uses this approach in its DAG-based payment layer (§12.5), and proposes to adopt these principles over time in the compute layer as PoUW volume increases. QUIP transfers settle outside the critical path of main-chain useful-work consensus. A routine payment does not wait for a block whose production depends on solving an Ising instance. Payments confirm through a fast optimistic path, while the Interlock Protocol (§12.2) and the useful-work chain remains the final anchors of record. This separation lets an on-demand quantum result reach its buyer while the Quip Sinnet (§12.4) considers it for a block reward. It also prevents useful-work block latency from stopping time-sensitive jobs.

Pure asynchronous-BFT DAG protocols in the Hashgraph family assume a known, permissioned validator set. They use asynchronous Byzantine fault-tolerance assumptions to bound finality. These membership and synchrony assumptions produce lower latency, but a permissionless useful-work network cannot adopt them in full. Quip therefore uses a timelock-bounded payment fast path instead of adopting virtual voting wholesale. Section 12.5 describes the resulting trade-off among finality, latency, and security.

Stage 3 introduces problems that classical DAG consensus does not solve. Entangled processors must coordinate within coherence-time budgets. The protocol must also bound adversarial state collapse across operators that jointly hold a distributed computation. Classical DAG consensus provides a useful analogy to reason about these problems, but does not offer a complete answer (§8).

4.6 Aerodrome and Vote-Directed Emissions

Aerodrome, its Velodrome predecessor, and the Curve ve(3,3) model provide the direct precedent for veQUIP emission routing. Holders lock the native token to create vote-escrowed voting power, then vote each epoch to direct emissions toward selected liquidity pools. Protocols seeking deeper liquidity bribe voters to attract those emissions. Bribers compete to buy emission flow, while voters route it toward the offers they value, making bribe prices a signal of where the market wants inflation to go.

Quip applies this mechanism at the protocol layer. Holders lock QUIP to receive veQUIP, then vote each epoch to divide fixed emissions among subnets (§10.2). Subnet stakeholders can pay veQUIP voters through an explicit bribe market to direct issuance toward their problem class (§10.4).

The emissions buy a different good. Aerodrome directs inflation toward liquidity depth inside its financial system. Quip directs emissions toward *useful compute*. Votes flow to subnets whose solved problems and verified results the market values most. The bribe-and-vote process therefore selects computational fields for subsidy, not only pools for capital.

Quip also separates the return on committed capital from the governance weight created by commitment. Collateral-backing network liquidity earns native yield from its productive use, while locking earns voting power and bribes. Quip votes price computation rather than liquidity depth.

4.7 Post-Quantum Network Approaches

Projects that defend value against a cryptanalytically relevant quantum computer fall into three classes. The first consists of post-quantum-native layer-one chains: the Quantum Resistant Ledger uses hash-based XMSS [43], [44], Cellframe defaults to Dilithium and supports interchangeable schemes [45], and Quantus Network uses ML-DSA signatures on Substrate [46], [47]. These sovereign chains hold assets natively and compete with Quip Network as quantum-resistant settlement layers, but do not protect value held elsewhere.

The second class retrofits existing chains, although these efforts remain in research or draft form. Ethereum's post-quantum roadmap would replace validator BLS signatures with a hash-based scheme and provide a per-account post-quantum path through account abstraction. Bitcoin's BIP-360 is an unactivated and contested draft for quantum-resistant output scripts [12]. An in-place upgrade takes years of coordination, and legacy key material remains exposed until it finishes.

The third class provides overlays that require no host-chain change. Project Eleven operates a registry that binds existing addresses to post-quantum keypairs [48]. SymmetriQ and similar wallets add post-quantum co-signers. Quip's asset surface belongs to this third class.

Quip adds custody and transfer to the overlay model. Project Eleven records a future-claim proof, and a co-signer adds a second signature. A Quip Account instead holds a wrapped asset behind SHRINCS, a hash-based construction in the WOTS+ [44] and SLH-DSA [49] lineage that commits a large collection of one-time keys under one compact public commitment, and pairs that account with the bridgeless Interlock protocol (§12). A holder can protect value on Bitcoin or Ethereum and move it across chains without a bridge, oracle, or host-chain upgrade.

Quip complements the other two classes. It protects value on chains that native post-quantum projects seek to replace, and retrofits popular projects while they reason about their next upgrade. Once a host completes its own migration, Quip no longer needs to provide that protection, even if there may be compatibility arguments for maintaining support. The wrapper protects ownership and movement of an asset. It does not secure an upgradeable or permissioned underlying contract unless the owner upgrades to one of these wrappers, and the host chain's bridges, oracles, and consensus remain vulnerable until further upgrades are applied. Each retains its own exposure.

5 Protocol Architecture

5.1 Compute and Asset Layers

The protocol has one compute core and one surface for moving value into and out of it. The core is a Nakamoto-style chain whose proof-of-work includes a DAG of several contributory proofs-of-work and uses post-quantum cryptography end to end for the reasons in §2.2. The asset surface consists of quantum-resistant wallets and swap primitives that let users transact with the core from other transaction networks without moving their liquidity.

5.2 Protocol Stack (Five Layers)

The compute core has five layers, each providing an interface that lets its specialists work without understanding the layer below. A job and payment move from the consumer down to the hardware, while the result and validation move back up (§5.4).

Consumer layer. The consumer states the parameters of a desired answer and the amount payable in a home currency, while lower layers handle the remaining work of architecture selection, compilation, and execution. While consumers can request specific platforms, no quantum expertise is required: a trucker can use an application developer's website, while an enterprise can direct large GPU clusters toward better solvers. A human action or API call enters the application layer, and a paid bid promotes a job submission to the solver layer. Individuals, small businesses, and enterprises constitute this consumer layer.

Application layer. Vertical front ends and developer SDKs package raw solvers as industry-specific products. An application makes a solver discoverable and usable, accepts payment in the consumer's home currency, acquires QUIP and any solver-specified tokens, and converts the consumer's data into the solver's input format. A website, application, or no-code API faces the consumer, while a valid job request denominated in network tokens faces the next layer. Application developers operate this layer and sell network compute as a repeatable product.

Smart contract layer. Quip contracts register, escrow, and settle jobs. A contract converts job data into a format that a subnet's miners can serve, escrows its payment during execution, and binds the consumer's escrowed bid to a job specification containing an input type, output type, and verification type. The verification type rejects invalid output. After a valid result, the contract releases payment, routes royalties, and collects the protocol fee. The job specification and escrow face the application layer, while a request to a registered subnet faces the subnet layer and offers a block reward. Algorithm developers write the contracts and job specifications, and protocol developers maintain their runtime and execution VM.

Subnet layer. Each subnet represents one problem class, such as Ising optimization, the QVRF, or hidden-subgroup cryptanalysis, with its own proof-of-useful-work template and verification tier (§6.1, §3.3). The subnet defines problem-statement and solution-submission formats, verification and quality-scoring functions, and a difficulty adjustment that keeps its useful-work rate proportional to network capacity. The contract layer registers

against an abstract subnet template, and matching processors receive a deterministic job specification. Protocol developers supply each proof's parameters and maintain implementations for each hardware architecture.

Compute layer. CPUs, GPUs, ASICs, and QPUs bid on and execute work as first-class peers (§8.1), returning low-energy solutions, verified samples, or other results required by a subnet. They also publish proofs that upper layers can check cheaply. A bid and scored proof connect this layer to the subnet, with the physical processor below it. Compute providers choose the job classes and credentials they accept and can form mining pools across hybrid platforms and multi-QPU clusters (§8.4).

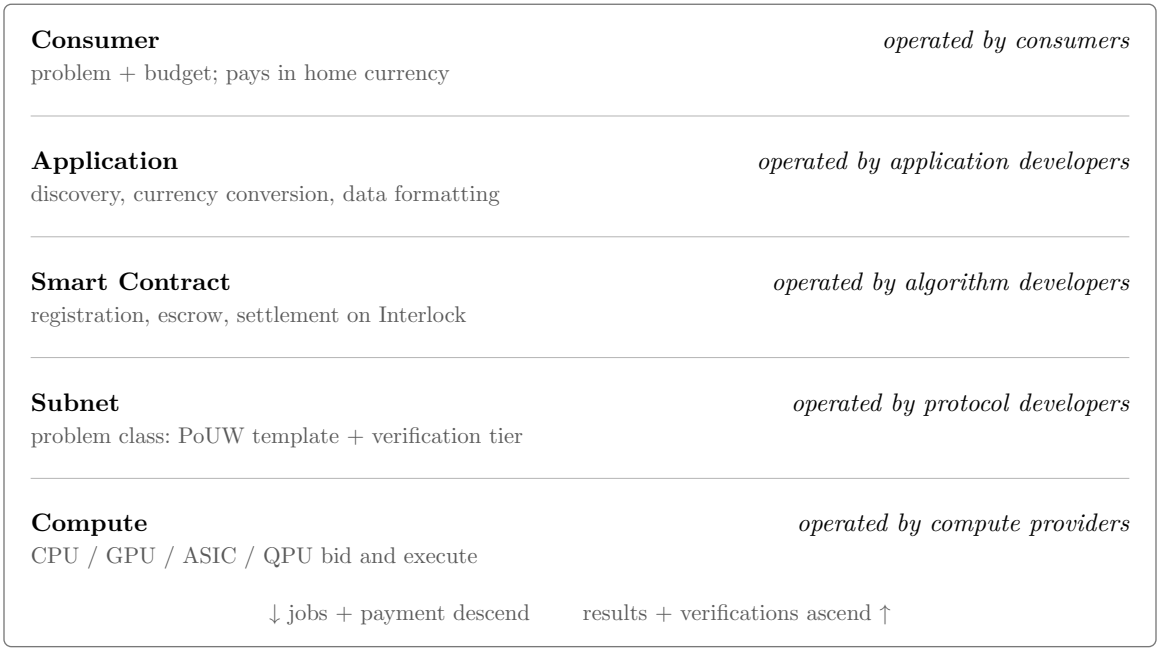

Figure 1: The five-layer protocol stack. A job and its payment descend; a result and its verifications ascend (§5.4).

5.3 Stakeholders

Six stakeholder categories operate the compute core, and their inputs and returns form the payment chain from the consumer buying an answer to the liquidity provider pricing its compute credit.

Consumers. Consumers provide demand by submitting jobs and by bidding for solutions in their home currency. In return, they receive answers that are faster, cheaper, or otherwise unavailable without quantum expertise. Every job that enters the stack begins with a consumer.

Application developers. Application developers package raw solvers in discoverable, industry-specific front ends. They handle currency conversion, data formatting, and token acquisition for consumers. They earn a resale margin on the service and a share of the builder allocation (§7.1).

Algorithm developers. Algorithm developers publish job specifications and solver implementations to the open-source library. Each invocation pays the author a royalty. The royalty compensates the work and reserves authorship of the invention (§6.1).

Protocol developers. Protocol developers maintain Quip's runtime, the execution VM, and the parameters for each Proof of Useful Work. They add implementations as algorithms and hardware improvements. Every solved block and proof pays them a share of fees.

Compute providers. Compute providers operate the CPUs, GPUs, ASICs, and QPUs that bid on and solve jobs. They earn block rewards and job fees for genuine compute. The result's score, not the machine category, determines payment (§8.1).

Liquidity providers. Liquidity providers create price discovery and a liquid market between local currencies and QUIP, the shared compute credit. They earn fees and emissions on supplied liquidity. They also govern emissions by committing QUIP to the Quip Control Asset (§10.2). Control weight rises with lock duration, while the escrowed capital remains liquid as a restaking position (§7.3). Liquidity providers may also participate in prediction markets and provide sureties against misbehaving identities. A consumer that holds compute credits can provide liquidity. Holding, governing, and providing liquidity can therefore be functions of the same position.

5.3.1 Nodes

Three consensus roles run the chain across these economic categories. **Validators** produce blocks and provide the RPC, indexer, and multi-party-computation endpoints that consumers use. They batch and finalize transactions, post a bond at risk, and earn network fees plus emissions. **Miners** are compute providers acting in consensus. They solve Proofs of Useful Work and broadcast winning blocks. They earn block rewards and job fees but may spend compute on a round they lose. **Nominators** stake QUIP behind validators that cannot meet the minimum bond alone. They share the validator's yield and slashing risk.

Role	Stake	Reward	Risk
Validator	Bond + on-chain capacity	Network fees + emissions	Slashing for downtime or fault
Miner	Hardware + energy per round	Block reward + job fees	Compute spent on lost rounds
Nominator	QUIP delegated to a validator	Share of validator yield	Slashing alongside the validator

Table 1: Consensus roles by stake, reward, and risk.

5.4 Job, Payment, and Reward Flow

Value and data move down the stack as the consumer sends a job and payment to the application layer, which formats the data and converts the payment into network tokens. The smart contract layer escrows the bid against a job specification, the subnet selects the problem template, and the compute layer executes it. The verified result follows the same path upward: the subnet scores it, the contract settles it, and the application decodes it into the consumer's required format.

QUIP emissions also move upward. Useful work at the compute layer triggers newly minted block rewards. Most issuance goes to the miner that produced the winning solution. The invoked solver's algorithm developer receives a royalty, and the treasury receives a share. The treasury converts its share into permanent liquidity instead of burning it (§7.4).

The flows have different sources and recipients: one consumer directs a payment down the stack, while the protocol splits an emission subsidy among the parties that completed the work. They meet at settlement, where one verified proof releases the consumer's escrowed payment to the solver and mints the block reward. The same event pays for the job and secures the chain.

A vehicle-routing job shows the full path. A logistics consumer bids a hundred QUIP-equivalent in its home currency through a routing application, which converts the payment, formats the fleet data, compiles the problem into an Ising instance, and sends it to the routing contract. The contract escrows the bid and registers the job with the Ising subnet, which publishes the instance to the compute layer.

A quantum annealer, a GPU running simulated annealing, and an exact classical solver compete on the instance, and the annealer wins by returning the lowest-energy solution set within the timing window. The subnet scores the submission, and the contract verifies the claimed energy in one linear pass. Settlement releases the escrowed bid to the winning miner after deducting the solver author's royalty and protocol fee, while the chain mints the block reward and splits it among the miner, algorithm developer, and treasury. The application decodes the route for the consumer, and the treasury places its fee into the next buyback-into-liquidity cycle (§7.4).

6 Proofs of Useful Work

6.1 Subnet Interface

Every subnet implements the same interface, allowing the chain to process different problem classes from a unified pipeline. A subnet defines problem-input and solution-output formats, a verification function based on its tier (§3.3), and a quality function that ranks valid solutions. A difficulty parameter keeps the work rate steady as hardware enters and leaves.

Settlement uses one reward split. Most of the reward goes to the miner with the winning solution. The invoked solver's algorithm developer receives a royalty, and the protocol treasury receives a share to fund maintenance and development. One proof therefore pays for both solving the problem and building the solver.

A new subnet supplies these functions and an initial difficulty, then governance registers it as a supported problem class (§10.1). The following sections apply the template to Ising optimization, quantum verifiable randomness, long-horizon cryptanalysis, and a set of future classes.

6.2 Ising Model Optimization (QA/QAOA/SA)

The flagship subnet converts industrial optimization into Ising models and equivalent QUBO problems with the objective $E(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$, where each spin s_i is -1 or $+1$. Portfolio selection, vehicle routing, scheduling, and packing can all compile into this form [34].

Finding a low-energy configuration is NP-hard [50], but checking its claimed energy takes one linear pass. The formulation also runs directly on quantum annealers [26] and through QAOA on gate-based hardware [51]. Different hardware can therefore compete on identical inputs (§8.1).

A miner registers for a published problem topology, which is evaluated for difficulty, and submits a nonce, salt, and candidate solution. The chain regenerates the model from the proof's seed, recomputes each energy with fixed-point integer arithmetic, and scores the energy in relation to the topology's difficulty rating.

Energy is a convenient proxy for solution quality, so the round goes to the lowest, most negative energy relative to an expected ground-state estimate. Job proposals reward one or more optima, while faster solutions receive more rewards based on their latency advantage. Difficulty consists of a normalized energy threshold, calculated via a graph reduction suite applied to the proposed topology and a solution frustration evaluator which normalizes submitted solutions across topologies. Difficulty adjusts over a decay period as hardware enters and leaves to keep the useful-work rate steady.

6.3 Quantum Verifiable Random Function (RCS/XEB)

The Quantum Verifiable Random Function (QVRF) turns random-circuit sampling into a public good [35], one of the few tasks where current quantum hardware provably outruns classical machines, and the one near-term application with an end-to-end certified-randomness protocol [36] and a hardware demonstration [37]. A precommitted difficulty parameter fixes the minimum qubit count, circuit depth, and fidelity target. A verifiable-random seed passes through a deterministic pseudorandom generator to produce a specific random circuit, which competing gate-based operators execute before submitting output samples within a limited period of time.

The chain checks each submission with classical cross-entropy benchmarking by computing the samples' linear XEB fidelity against the circuit and requiring the score to exceed the difficulty threshold. The submission must also arrive within a timing window that a classical simulator cannot meet. A randomness extractor converts validated samples into uniform output, and an intertwined hash-chain timing layer anchors it without a trusted clock.

Certified public randomness supports lotteries and gaming, fair ordering, leader election, and key generation. A consumer can also request a private certified-random draw for secure delivery using the Interlock contract on any host chain. Verification is classical and relatively cheap, while honest production requires quantum coherence within the timing window. The subnet therefore belongs to the statistically checkable tier in §3.3. Adversarial classical nodes cannot counterfeit a winning sample within an epoch.

6.4 Hidden Subgroup Problem

The long-horizon cryptanalysis subnet serves as a bug bounty and CRQC canary, under which a cluster of operators earns QUIP by proving that it broke a classical or post-quantum cryptographic primitive. The protocol can check that result at negligible cost, and the proof of work provides a valuable benchmark for forecasting the likelihood that sufficiently sophisticated actors have access to a CRQC.

The main targets, integer factoring and the elliptic-curve discrete-logarithm problem, are both instances of the *hidden-subgroup problem* [2]. This shared gadget also assists in solving problems like the graph isomorphism problem or the shortest vector problem, providing real utility to honest actors, where each of these gadgets is reconvertible into an ECDLP or factorization problem, making it trivially verifiable. This is a tier-one subnet under §3.3 because one multiplication verifies a claimed factorization or discrete logarithm, so no oracle is required.

A break of commercially relevant scale may remain years away, so the subnet pays for progress along a graduated bit-length. A cluster or quantum operator can earn rewards for advancing the boundary before and reaching a full break.

The subnet can break cryptography that protects active financial and communications infrastructure, so two gates control it. The first gate is hardware. Although verification is tier one, Shor-scale factoring needs more error-corrected logical qubits than the field possesses (§3.2) [3], [52], [53]. The subnet waits for the multi-subnet stage in §2.4 and activates as fault-tolerant capacity arrives.

The second gate covers export control and dual use (§9.3). Using zero-knowledge proofs of identity and program provenance, dispatch sends cryptanalytic work only to operators in jurisdictions where that work is permitted, to operators willing to permit such use. Results follow a coordinated disclosure process instead of unrestricted publication. The subnet can then measure the remaining safety margin of deployed key sizes without helping attackers faster than defenders can migrate.

6.5 Candidate Problem Classes

The network keeps a catalog of problem classes beyond the flagship subnets and deferred cryptanalysis class. Each class has a predetermined verification tier from §3.3 and activates when hardware reaches the capability milestone in §3.2.

Unstructured search applies Grover-style quadratic speedup [38] to database search, constraint satisfaction, and inversion of one-way functions. It uses tier-one verification when a witness can be checked cheaply against its predicate. Its advantage is only quadratic, and current coherence cannot support the required logical-qubit depth.

Circuit mapping and compilation includes minor-embedding on an annealer topology, gate synthesis and qubit routing for gate-based devices, and compilation from hardware-agnostic programs to specific backends. A deterministic, repeatable cost metric scores embedding quality because no closed-form check exists. This class therefore uses tier-two verification.

The catalog is demonstrative and does not reflect a commitment to launch these classes. After the flagship-only stage (§2.4), Control Asset governance selects among entries that pair a hardware milestone with the verification method required by its tier. A class becomes eligible for proposal (§10.1) only when hardware can serve it and the protocol can score it, then competes for emissions only after enough holders choose to fund it. The catalog grows with hardware capability, so the network offers only work that it can perform and verify.

7 Token Design

7.1 Supply and Allocation

One billion QUIP will have been emitted when all pre-mine allocations finish vesting. This first billion is divided among subnet emissions (40%), the public and foundation premine (20%), early investors (15%), builders (15%), and community programs (10%). Subnet emissions enter circulation continuously from launch, while allocations for early investors and other team-aligned participants follow a cliff and linear vest.

The first billion is not a maximum supply. After the pre-mine has fully vested, subnet emissions continue at 40 million QUIP per year. The companion document, **Quip Network: Cryptoeconomic Specification**, gives the complete vesting schedules and circulating-supply projections.

7.2 Emission Schedule and Implicit Decay

Quip emits 40 million QUIP each year, regardless of token price, network activity, or the number of subnets.

At launch, the two flagship subnets divide the full emission. Each new problem class adds another subnet that can receive part of the same 40 million tokens, so the amount available to any one subnet falls as the network adds subnets. Quip calls this effect "implicit decay".

Bitcoin creates decay by cutting its total block reward on a fixed schedule. Quip keeps the total reward constant and lets more types of useful work compete for it. As a result, QUIP emitted per unit of compute decreases as the network grows, without explicit halving events.

7.3 Quip Control Asset

The Quip Control Asset (QCA) divides emissions among subnets. In the canonical implementation, veQUIP, holders lock QUIP in an Interlock-enforced escrow (§12.2) to gain voting weight based on lock duration. Longer locks grant more weight, which decreases as the lock approaches expiry.

Each epoch, holders use this weight to direct emissions toward the subnets they want to fund. The escrow position remains available for lending, borrowing, or use as collateral while its QUIP is locked, so long as the collateral is returned to the escrow by the end of

the transaction chain. Holders can therefore participate in governance without leaving the underlying capital idle. The companion Cryptoeconomic Specification and §10.2 through §10.4 describe the control-weight formula, concentration limits, and subnet-vote market.

7.4 Treasury Liquidity Policy

Quip uses protocol fees to buy QUIP and place it in permanent, protocol-owned 50/50 liquidity positions. Each purchase adds market depth, lowering the cost of trading QUIP and increasing the treasury's share of pool fees.

A buyback and burn removes the asset after the protocol pays to acquire it, while holding fees as stablecoins preserves reserves but adds no market liquidity or revenue. A permanent position instead keeps the purchased QUIP in the market and earning fees. As protocol revenue grows, the treasury owns more liquidity and receives more of the market's fees.

The companion Cryptoeconomic Specification defines the counter-asset, position sizing, and any rebalancing rules.

7.5 Hardware Partner Financing

QPU operators pay capital and operating expenses in fiat, while Quip pays mining rewards in QUIP. Two mechanisms help operators manage this mismatch.

The first mechanism is miner sales, in which Quip packages claims on a partner's mining rewards and sells them to supporters for cash up front. The sale brings outside capital into the treasury without new token issuance or immediate sell pressure while distributing future rewards across a broader group of participants.

The second mechanism gives the hardware partner a put option on earned QUIP, allowing the partner to sell those tokens to the treasury at a guaranteed price. If the market price falls below the strike, the put sets a floor under the operator's revenue. If the price rises, the operator keeps the gain. Airlines hedge fuel prices for the same reason: predictable operating costs matter more than speculation on the asset.

8 Heterogeneous Compute

8.1 Hardware-Neutral Competition

Quip treats every processor as a potential miner. Each subnet is agnostic to hardware class so CPUs, GPUs, ASICs, and quantum processors can compete for the same jobs. The subnet pays for result quality rather than machine type, and its quality function (§6.1) awards the reward to the best solution whether a classical or quantum miner produced it.

This competition creates the apples-to-apples benchmark that the quantum industry has lacked (§1.1). A quantum annealer, a GPU running simulated annealing, and an exact classical solver can bid on one Ising instance. The same energy and timing criteria judge every result. The resulting public record shows where quantum hardware has an advantage and where tuned classical systems still perform better.

8.2 Hybrid Computation Pipelines

Useful work passes through a pipeline that places a quantum inner loop between classical pre-processing and post-processing. The pipeline fits a consumer's input to a problem template, simplifies, reduces, gadgetizes, and virtualizes the resulting graph, embeds it onto a target device, selects an algorithm from the open library, routes the job through the clearinghouse, orchestrates any multi-device pipelines, and executes it. On the return path, the pipeline decodes and verifies the result. Classical processors handle the stages they perform best, leaving only a suitable kernel for the quantum device.

This division makes near-term advantage hybrid rather than purely quantum (§3.1). Today's processors cannot sustain deep circuits, but they can handle tightly scoped sub-problems such as a sampling step or embedded optimization within a classical workflow. Because the pipeline selects the algorithm and hardware in separate stages, it can retarget a problem to new hardware or a better solver without a consumer rewrite.

8.3 Distributed QPU Execution

In the final stage, the network pools qubits from distinct operators into one logical computation. Gate teleportation provides the joint operation [54], [55]: two separate processors consume a shared entangled pair and exchange two classical bits, then execute a two-qubit gate as though their qubits were co-located. This allows independent QPUs to run one circuit.

Distance adds two more requirements. Because the no-cloning theorem forbids copying and amplification [14], quantum repeaters must store, swap, and purify entanglement [24], while quantum memory keeps a state coherent during the classical coordination round-trip. Entanglement swapping connects these resources across repeater hops and extends a shared pair beyond one physical link.

Researchers have demonstrated each component, but only separately. Trapped-ion processor modules have used gate teleportation across a roughly two-metre link, where the teleported entangling gate reached near eighty-six percent fidelity and ran a small distributed Grover search [56]. Deployed metropolitan fiber has carried entanglement for tens of kilometres in New York and Berlin, alongside classical traffic [57]. A metropolitan multiplexed repeater node has certified Bell nonlocality across separated memories [58], while solid-state quantum memories have kept states coherent from seconds to, in record demonstrations, an hour [33].

No integrated, multi-hop distributed-QPU system spans untrusted operators today [25]. Circuit cutting can divide a computation among smaller devices, but its sampling overhead grows exponentially with the number of cuts, making it practical only for shallow partitions (§3.1) [59], [60]. Multi-QPU entanglement is therefore a third-stage objective rather than a launch feature. Quip builds the economic and cryptographic coordination layer now so pooled, entangled capacity can join it as the physics matures. Moving from coordinated independent processors to one distributed machine produces the literal worldwide quantum computer described in §2.1. A future protocol-level specification will define how operators advertise, price, and settle shared entanglement.

8.4 Mining Pools and Resource Sharing

Mining pools combine operators for work that is too large or bursty for one machine and divide rewards by contributed effort, as classical pools do to smooth the variance of block discovery. A Quip pool also provides circuit virtualization by using circuit cutting to divide a problem larger than any member's device and reconstructing the result classically [59], [60]. Because sampling overhead grows with the number of cuts, this method pays only for shallow partitions. In the third stage, the pool coordinates entanglement among its members' QPUs (§8.3), assembles one logical machine from separate processors, and divides both work and rewards among them.

This design adds participants who contribute connectivity instead of raw compute. Quantum repeater and memory-node operators supply the entanglement-distribution capacity needed for distributed computation. Their economics resemble Filecoin, Helium, Foam, and other decentralized physical-infrastructure networks because the protocol pays for coverage and capacity rather than direct problem solving, so it must meter and price networking resources as carefully as compute.

A future protocol-level specification will define pool governance, including member admission, sub-problem allocation, and payout disputes when contributions cannot be attributed cleanly. The present design fixes one rule: a pool's reward share settles through the same proof-of-useful-work accounting used for solo operators.

9 Operator Configuration and Dispatch

9.1 Subnet Selection

Hardware dictates economics. An operator first chooses the subnets its hardware can serve. An annealer or other adiabatic device fits the Ising flagship (§6.2), where a QUBO or HOBO maps to its native ground-state search. Gate-based devices, including neutral-atom, trapped-ion, superconducting, and photonic systems, fit circuit-structured subnets such as the QVRF (§6.3), chemistry workloads (§3.2), and cryptanalysis workloads (§6.4). A classical CPU, GPU, or ASIC can bid into any subnet, providing baseline benchmarks that keep quantum claims honest and remaining a direct competitor wherever quantum advantage has not appeared.

At registration, an operator declares its hardware profile, which for a QPU includes its type, qubit count, topology, error rates, and coherence time. The protocol then shows the subnets whose proof-of-useful-work templates match that profile, guiding operators toward work their machines can win.

Operators can revise their subnet choices at any time. Each subnet adjusts its own difficulty (§6.1) and receives a separate emission allocation through Control Asset governance (§10.2), so profitability changes with rewards, competition, and difficulty. An operator can enter when its expected reward rises or its hardware clears the difficulty comfortably, then leave when competition or a lower reward pushes expected returns below operating

cost. Dashboards report real-time earnings and relative performance, making reallocation a routine operating decision rather than a new onboarding process.

9.2 Job-Type Filtering

Policy and capacity, enforced before dispatch. After choosing its subnets, an operator can whitelist jobs by template, problem class, or compliance characteristic, then set bounds on size and duration. These bounds reject models that exceed the machine's memory or embedding capacity and deadlines the operator cannot meet. The operator also sets a minimum profitable bid and limits on power and run time, as well as allowlists or denylists for specific solver requests or those coming from citizens of disallowed jurisdictions (§9.3). Together, the filters define what the operator may run and what it can run profitably.

The bidding and dispatch layer uses these filters to define eligibility (§11.3). An open order reaches every operator whose whitelist and bounds admit it. A bid order starts with a named set of operators, then applies each operator's filters. The matching layer sends a job only to eligible operators. If none accept it, the job expires and returns its escrow. Filtering resolves a mismatch before dispatch, so the network does not spend compute on a job that an operator must later reject.

9.3 Jurisdiction and Compliance Filters

Geography-aware routing. An operator's registration records its jurisdiction and binding regulatory constraints. The matching layer treats both as routing predicates. At dispatch, the network gates sensitive problem classes, including cryptanalysis subnets (§6.4) and classified or dual-use workloads, against export controls using zero-knowledge proof of identity and program provennce. It also applies sanctions rules to both sides of a match before offering the order. These gates can further be applied to arbitrary criteria, such as acceptance of operator terms and conditions. Because dispatch checks on-chain operator attributes, an operator never receives a job that it cannot lawfully run.

Identity as the gating credential. A zero-knowledge identity credential carries the compliance facts needed for routing. When an operator or problem class requires verification, the credential attests the consumer's identity and acceptance of the operator's terms while revealing only the predicate required by the policy. Completing KYC or KYB gives a consumer access to more operators and jurisdictions, while accepting an operator's terms makes that operator eligible to serve the consumer.

Operator entry moves from whitelisting toward permissionless participation. The network opens with hardware attestation, benchmarking, and Know-Your-Business verification where applicable, then makes entry permissionless as operators build reputation and the network matures. Any machine can then run the node software and begin mining, subject to the same per-job compliance filters.

10 Subnet Governance

10.1 Subnet Proposal Requirements

Every new problem class begins with a structured proposal that fully instantiates a subnet template (§6.1), and receives no emission until the proposal defines how the protocol will score its work and settle its claims.

The proposal first defines the computational domain and form of admissible claims. It then selects a verification tier from §3.3 and supplies its verification function: on-chain exact verification for tier one, a reproducible benchmark suite for tier two, or peer review plus veQUIP adjudication for tier three.

The proposal also defines a quality rubric, typically weighting correctness, optimality, diversity, and/or latency. It names the expected hardware class, such as a quantum annealer, gate-model QPU, or classical alternative used as a verification baseline, and sets the initial difficulty for a valid launch submission. A proposal missing any field cannot activate.

Proposal gating follows the stages in §16.3. During Genesis and the early phases, the Quip Foundation reviews every proposal. It deploys subnet infrastructure only after checking market demand, technical feasibility, available hardware partnerships, and resource requirements. This centralized period establishes verification playbooks and trust while the mechanism remains unproven.

After the network meets the activation thresholds, proposals become permissionless and each submission bonds a QUIP stake. A subnet must also attract veQUIP votes (§10.2) before it earns meaningful emission. Together, these requirements control spam and quality: a low-quality proposal with no votes receives no allocation and moves toward retirement (§10.5), leaving the proposer to bear the cost.

10.2 Emission Routing with the Quip Control Asset

The *Quip Control Asset* (QCA) routes emissions among subnets each epoch. In one implementation, the vote-escrowed token veQUIP, a holder commits QUIP to escrow through the Interlock primitive (§12.2) rather than a separate lock contract. The commitment grants control weight that increases with lock duration and decreases as the lock approaches its end, so influence tracks the time that capital is committed.

The Interlock makes the escrowed position a transferable claim that its holder may lend or borrow, allowing holders to govern while retaining other uses for their capital. A holder may use the interlocked assets as collateral for cross-chain intents and earn yield—so long as they retain the collateral at the end of any transaction—or they may leave the tokens idle and receive a baseline, pro-rata share of emissions for participating. In either case, locking earns control weight while the capital remains productive.

Each epoch converts control weight directly into proportional emission. A subnet's share of fixed per-epoch emission equals its share of all weight cast, $\text{allocation}_i =$

$\left(\text{weight}_i / \sum_j \text{weight}_j\right) \times E$. If no one votes, the subnets split emission equally. Any rounding remainder goes to the largest recipient so the allocations remain exact.

The Control Asset binds to a zero-knowledge identity credential to limit capture by one actor. This lets the protocol apply per-identity limits, such as quadratic decay in marginal control weight after a holder's first units of commitment. A token-only system cannot enforce these limits against an actor who splits a balance across wallets.

The QCA forms the demand side of an open incentive market. Subnets bid for allocation through the bribe market in §10.4. The resulting equilibrium reflects the marginal value that each problem class will pay to attract compute, rather than token wealth alone. Miners also receive a small fraction of their rewards as Control Asset, giving the people who perform the work a continuing voice in future emissions.

10.3 Surprisal Markets

Prediction markets can be used to incentivize funding of new subnets with uncertain payoff, and to adjudicate unlikely responses to unverifiable quantum oracles. These prediction markets pay more when the final outcome is less expected: if the market prices an outcome at probability p when it resolves, its *surprisal* is $-\log_2(p)$, the outcome's information content in bits. A near-certain result therefore carries almost no surprisal, while a long shot carries much more. The bigger the upset, the bigger the payout.

Surprisal multiplies the base bounty according to $\text{reward} = \text{base} \times (1 + \alpha \cdot (-\log_2 p))$. The tunable gain α, around 0.3, controls how much the network pays for informative outcomes. Proving a quantum advantage priced at five percent can more than double the base reward. Confirming a ninety-percent consensus barely changes it. The mechanism directs the largest rewards toward claims about which the field was most uncertain.

Each market horizon funds a different type of work. *Sprint* markets resolve in three to six months and offer bounties in the tens of thousands of QUIP for applying existing techniques to new problem instances. *Research* markets resolve over twelve to eighteen months with larger bounties for new algorithms and problem classes, while *Moonshot* markets span years and carry the largest bounties for fundamental breakthroughs, such as factoring a cryptographic-scale key.

Together, these markets evaluate the tier-three claims in §3.3, for which no cheap objective or binary test exists. The network instead aggregates capital-backed beliefs, settles them through staged oracular adjudication, and weights rewards by surprisal. These surprisal markets can also be viewed as a sort of parametric insurance, where deployers on these networks may wish to participate in the surprisal market in case their platform is not competitive—a lackluster performance profile can be recouped by a hedge that pays out when a competitor wins. Together, these dynamics produce the strongest available signal about whether a quantum-advantage claim is real.

10.4 Vote Incentive Market

Quip makes vote-buying an explicit, permissionless market. A subnet or its backer deposits QUIP into a bribe contract for a specified subnet and epoch, and after the epoch closes,

veQUIP holders who voted for that subnet claim the deposit in proportion to their contributed votes. This mechanism comes from the gauge-voting and bribe markets of Curve and Aerodrome (§4.6) and turns emission routing into a transparent auction.

A subnet will buy another vote while the emission it captures is worth more than the bribe, so competition makes bribe prices approach the demand for each problem class. The price paid to attract compute becomes an on-chain measure of how much its buyers value that compute, and emission moves toward the work they will pay most to fund.

Several controls limit capture. The veQUIP time-lock (§10.2) prevents cheap, temporary vote renting, while identity-bound limits (§15.2) reduce the control one actor can gain by splitting funds among wallets. Provisional concentration caps would prevent one subnet from taking all emission regardless of its bribe. Cartel monitoring is also provisional and would flag a subnet that holds a persistently high allocation. These controls let the market price demand without giving one well-capitalized actor control of the network.

10.5 Subnet Retirement

A subnet that stops earning emission winds down instead of receiving an indefinite subsidy. Three signals can flag it for retirement: sustained low demand measured by veQUIP allocation below a viability floor, a persistently high rate of submissions that fail validation or claims that fail oracle adjudication, and work below the quality bar set by the subnet's proposal. Because the allocation market already funds subnets, it also starts retirement: a subnet that receives no votes loses its emission.

A flagged subnet receives a grace period to attract votes, improve verification rates, or restore quality. If it does not recover, the miner-migration path moves operators and open jobs to surviving subnets that accept the same hardware class and job specifications. This prevents a closure from stranding committed compute or work in progress.

Retirement itself runs through veQUIP. Missing votes provide the soft trigger. A hard wind-down, which deregisters the subnet contract and reclaims its slot, requires a governance action. That action uses the same time-weighted vote and the same Foundation-reviewed-to-permissionless progression as subnet creation (§10.1).

11 Subnet Execution Layer

11.1 Subnet Contracts

Each subnet uses two cooperating contracts and three roles: a *builder* authors the problem class, a *proposer* funds an instance, and *solvers* compete to solve it. The long-lived bulletin stores the immutable problem interface, verification function, and transformation into a canonical formulation, which a builder registers once under governance as a blessed template.

For each problem instance, a proposer opens a short-lived order around the template, escrows the reward, and collects competing submissions. The lifecycle is register, fund, verify, settle: the builder registers a spec, the proposer funds an order, and the verification

function checks submissions. On resolution, the order divides the reward among the winning solver or solvers, the builder whose specification was used, and the treasury, or returns the escrow to the proposer if it expires without a valid solution.

Governance upgrades specs instead of editing them in place. A live problem class can therefore improve without changing orders already in progress. Each order also carries an explicit resolution and delivery policy, making settlement deterministic and replay-safe.

11.2 Job Specification Schema

A job specification describes the work without naming hardware. An order contains the problem formulation, initially an Ising model given directly by its coefficients, along with the reward, deadline, access mode, reward-resolution rule, and result-delivery policy. By content hash, it may reference validation and transformation programs that check and canonicalize submissions. Those programs use execution-VM bytecode (§11.4), so each job carries its verification logic instead of hard-coding it into the chain.

Because the descriptor names a problem and verification function but no machine, a CPU, GPU, ASIC, or quantum processor can serve the same order. The protocol adjudicates every result in the same way. It validates and canonicalizes each submission against the spec before marking the order solved, rejecting malformed or out-of-domain results before payment.

11.3 Bidding and Dispatch

An order uses one of two access modes. An open order accepts a solution from any registered solver, while a bid order follows the request-for-quote path and limits eligibility to named operators or hardware classes. Consumers with confidentiality, jurisdiction, or quality requirements (§9) can therefore request work only from operators they have vetted.

The order escrows its reward when it opens and releases it under a stated resolution rule. A single-best rule pays the winning submission. A top-N rule splits the reward among the best N submissions, either equally or by solution quality. The latter makes redundant, high-quality work economical when a consumer values several near-optimal results.

Timing has two phases. Every order has an absolute deadline, but the first valid solution opens a shorter settlement window so the order can resolve without waiting for it. The delivery policy records results on-chain, pushes them to a consumer callback, or makes them available for polling. An unsolved order returns its escrow at expiry. The chain expires and settles orders lazily as solutions and queries arrive, avoiding linear scans of all open orders.

11.4 Execution and Virtual Machine

Two layers absorb hardware differences. The lower is the X-Quadratic Virtual Machine (XQVM), a small deterministic machine for quadratic optimization whose programs describe an XQMX, a sparse model of linear and quadratic coefficients over a binary, spin, or discrete domain, rather than a gate circuit.

Each job contains three independent programs that share an instruction set but no state: an encoder builds the problem from consumer input, a verifier recomputes the energy of a candidate solution, and a decoder converts the solution back into domain terms. The programs compile to compact, content-addressed bytecode that executes with integers and reproduces results to the bit.

The XQVM also supports attestation. A multi-phase static verifier rejects malformed bytecode at the submission boundary, before value is spent. Its checks include bad jump targets, unbalanced loops, ill-typed registers, and stack overflow. Every run can also emit a step-by-step execution trace that witnesses the exact bytecode used to produce a result.

The same bytecode behaves identically in native code, in the browser, and as WebAssembly inside a runtime pallet, with shared conformance vectors enforcing this cross-implementation equivalence. The network can therefore adjudicate work without trusting the operator. The verifier recomputes claimed energy independently and exactly, on-chain where required, while the trace makes the computation auditable. Appendix A lists the machine's concrete parameters.

The upper layer is a hybrid compiler and orchestrator. It carries classical and quantum fragments in one intermediate representation, then lowers the program onto the selected backend. A quantum annealer receives a minor-embedding, a gate-based device receives a synthesized and routed circuit, and a classical solver receives its corresponding target.

This layer runs the pipeline in §8.2, keeps quantum state under a no-cloning discipline, and content-addresses work so operators can deduplicate identical sub-problems. The XQVM gives the network one hardware-independent job and verification standard. The upper layer gives it a hybrid compiler that can retarget those jobs.

12 Asset and Settlement Layer

12.1 QUIP Account Primitive

A Quip Account is a self-custodial account authorized by a post-quantum signature in addition to an elliptic-curve key. The account commits to a quantum-resistant public key—in the current iteration a SHRINCS key—through a public seed and a hypertree root. SHRINCS packages two paths under that single commitment: an everyday *stateful* path spends a one-time leaf per signature from a fixed budget, and the account records spent leaves in an on-chain bitmap namespaced by a monotonic key version. A *break-glass stateless* path—a few-time SPHINCS+-C signature—authorizes recovery and rotation without depending on any record of which leaves were used. Verifying it costs roughly an order of magnitude more gas than the everyday path, which is why it is reserved for recovery rather than routine spending.

Production accounts separate authority by path. The stateful path handles ordinary spends and calls; the stateless path performs recovery and full rotation. Each rotation installs fresh material on both halves. Message signing for third-party protocols uses a dedicated many-time key behind the contract-signature interface, requiring the client to

track usage. Signature verification itself is delegated to a canonical, storage-free, non-upgradeable verifier contract pinned as an immutable [61]: all state — leaf bitmap, action nonce, key version, installed commitments — remains in the account.

Direct calls require the classical owner as caller *and* a valid post-quantum signature; account-abstraction operations, which arrive through a bundler rather than from the owner's address, carry the owner's classical signature inside the operation alongside the post-quantum one, so the same two-key rule holds on both routes. Neither key alone moves funds: a stolen post-quantum key cannot spend through a bundler, and a stolen classical key cannot authorize anything. The two classical surfaces are domain-separated — the operation co-signature and the contract-signature message digest use distinct typed-data domains — so a message signature harvested by a hostile application can never be replayed as authorization for an operation. A protocol administrator controls only fees.

Account abstraction [62] also lets a network paymaster sponsor gas, so users do not need the host chain's native token before transacting; the sponsor authorizes each operation with its own one-time leaf bound to that operation's sender and nonce. The account is scheme-agnostic and upgradeable, under explicit user consent designating a proxy that has been pre-approved by the verifier. The user can move from the current construction to another post-quantum model without moving their assets, and because deployment addresses derive from the factory and a vault identifier rather than from wallet bytecode, the same account address is available on every supported host chain. Appendix A gives the signature parameters and budgets.

12.2 Quantum Interlock Protocol

The Quantum Interlock Protocol settles peer-to-peer cross-chain swaps atomically, without a bridge, oracle, or custodian, extending the hash-locked atomic-swap construction [63] with post-quantum authentication. A separate contract runs the five-step lifecycle: offer, commit, share, countersign, settle.

1. The proposer first makes an offer as a request-for-quote or a maker bid or ask on any shared communication channel.
2. Once they have a counterparty, the proposer commits an asset on one chain under a shared hash whose preimage is a secret known only to the proposer.
3. The proposer then shares the final commitment hashes and call data using a QR code or link that remains secure without both the proposer's private preimage and the committed counterparty's private key.
4. After observing that lock, the committed counterparty locks an asset on any chain under the same hash.
5. The proposer executes against the counterparty's asset by revealing the secret, which the counterparty then ports to the proposer's network to execute and settle their leg of the trade.

If the swap stalls, each party reclaims its own asset after a deadline. The party that must act second receives the longer deadline, preventing an honest participant from timing out after half of the swap has completed.

Each action uses the party's own post-quantum signature and a canonical trade identifier that binds the agreed terms. The swap inherits the post-quantum security of the Quip Account primitive while requiring only ordinary execution from each host chain. Neither chain needs post-quantum consensus for the swap to resist a quantum attack, assuming consensus-level attacks will be infeasible for much longer than account-level attacks.

The canonical construction uses a hash lock and a guaranteed recovery path called reclaim, a trustless, timeout-gated refund of each party's own locked funds governed by the strict deadline order in Appendix A. A machine-checked EasyCrypt program [64] models the protocol and its SHRINCS-authenticated variant, and proves that every committed leg reaches exactly one terminal state — executed or reclaimed — locally and across chains, together with a middleman scenario in which a third party completes the claim after the secret becomes public.

A hash lock cannot prove, without the possibility of spoofing, that a counterparty committed and then abandoned a swap, so the protocol cannot slash a party that grief-locks funds in a denial-of-service attack. Verifiably encrypted signatures and adaptor signatures could close this gap by placing the lock inside the signature, but no well-studied post-quantum construction for either exists today. The interlock therefore uses hash locks and leaves adaptor-based slashing to future work.

12.3 QuipSwap

QuipSwap is the user-facing market above the Interlock. The interlock defines how two committed parties settle, while QuipSwap lets them find one another and agree on terms. In this request-for-quote venue, makers quote prices for moving value between assets and chains, takers accept a quote, and both parties settle directly through the Interlock's five-stage flow.

One deployed contract on each chain serves all swaps on that chain. It escrows assets under hashlocks, identifies each swap by a canonical hash of both parties' terms, and supports native, fungible, and non-fungible assets through one interface. Fees from completed swaps accrue to the treasury and fund the permanent liquidity positions in §7.4.

Peer-to-peer, bridgeless settlement moves liquidity into and out of the compute economy from its existing chain. Users do not mint wrapped assets, trust bridge operators, or wait for a host chain to adopt post-quantum security. This provides the on-ramp in §2.3: consumers fund compute through the swap, and providers carry earnings back to their chosen venue.

If the end user desires, they may also delegate signatures to a third-party who may optimistically execute trades on their behalf, enabling the establishment of broker-dealers and agents on a decentralized limit order book. Further structured products can be layered on top of these commitments with arbitrary composition of commitments and contracts.

12.4 Sinnet Compute Chain

Just like a Sinnet knot weaves together distinct strands, the Quip Sinnet hosts the interleaving proofs of useful work of the compute core as a sovereign layer-one chain built on Substrate. Its runtime includes a useful-work mining pallet that scores and rewards proofs,

a compute-job mempool for the spot market of bids and solutions, an on-chain XQVM execution pallet (§11.4), and the Interlock protocol to facilitate payments for compute across chains. By default, accounts and transactions use hybrid post-quantum signatures pairing sr25519 with FN-DSA-512 for non-replayable consensus commitments (§15.1).

Substrate's BABE and GRANDPA provide block production and finality [41]. Forkless runtime upgrades let Quip change consensus and the transaction model without the hard forks that comparable changes would require on Bitcoin. Appendix A lists the pallets and their parameters. Substrate supplies pluggable consensus and an on-chain execution environment for the VM in §11.4 while Quip retains control of the base layer.

The Sinnet is designed so the miners' useful work also secures the chain. Its subnets produce different proofs, such as a low-energy Ising solution or a validated quantum-random sample, which the protocol weights and combines into one accumulated measure of work. The canonical chain is therefore the one with the most total useful work rather than the most hashing alone.

Heaviest-observed-subtree and uncle-block constructions let work count even when it does not extend the main chain. Honest effort across heterogeneous subnets is not wasted, and an attacker must out-compute the network's full useful output to rewrite history. The Sinnet also indexes QuipSwap proposals and quantum-resistant accounts deployed on host chains. This lets value and identity refer to the Sinnet without requiring those chains to change.

During bootstrap, BABE and GRANDPA produce and finalize blocks while the useful-work layer assigns rewards above them. As the subnet set and miner base mature, that layer moves toward the heaviest-useful-work consensus described here.

12.5 Cross-Chain DAG Payments

The Interlock's hashlocks support DAG-based payments in the manner of gossip-ordered DAG ledgers [42]. Each payment leg locks an asset under a hash and releases it when the secret appears, or returns it after timeout. Within that window, counterparties can treat a leg as settled when it is committed. A chain of these payments forms a directed acyclic graph of conditional transfers, all resolved by revealing the originating secret.

If the secret does not appear before timeout, every leg unwinds and each party reclaims its funds. Optimistic settlement adds no counterparty risk beyond the wait and requires no bridge, custodian, or external attestation. Host chains only hold and release hashlocks.

The cost is latency. A DAG of optimistic transfers can finalize only when its slowest participating chain reaches finality. Until the last leg's timelock can no longer be contested, the full graph remains exposed to reversal. Quip gets bridgeless, custodian-free cross-chain payments by waiting for the most conservative counterparty network, rather than using a permissioned validator set to shorten the wait.

12.6 Escrow, Synthetics, and Structured Products

Further introduction of hashlocked commitments enables support for arbitrary escrow conditions on arbitrary transaction networks in an atomic, consistent, isolated, and durable way. This primitive further facilitates a wide range of structured products, including loans, synthetic options, parametric insurance, and other financial instruments of increasing complexity.

These products are the linch-pin to longer time-horizon validations of quantum-advantage claims that remain undecidable by simulation or direct computation, and unlock novel dispute resolution mechanisms by which bad actors in highly entangled systems can be held accountable. These products hinge on the escrow of speculative capital using the quantum-resistant Interlock protocol.

13 Client Interface and Payments

13.1 Client Onboarding and Key Management

Onboarding pushes post-quantum key generation and management below the attention of the interface. Account creation provisions a Quip Account with preloaded sets of one-time keys so a session does not stop to generate more. The wallet tracks spent keys and rotates the key after every operation.

Account abstraction handles transaction gas and protocol authentication. A network paymaster can sponsor gas, so a user does not first need the host chain's native token. A contract-signature interface lets the post-quantum account authenticate to existing protocols and retain familiar workflows. The same address derives deterministically across host chains. Early deployments target an Ethereum and Solana testnets and a Bitcoin execution layer, meeting users across all their child networks.

For consumers of quantum compute, this layer can be abstracted away completeley, where the money routing and exchange for QUIP tokens is handled in the background by appropriate vendors. They are given a landing page with applications targeted at their industry and compelling demos of the applications of quantum to their business.

13.2 Confidential Job Data

Confidential jobs must let an operator use consumer data and a developer's solver without exposing either dataset. The network uses various privacy mechanisms for each stakeholder at both the classical envelope and the quantum kernel.

For the classical envelope, including inputs, parameters, and the problem statement, the client can encrypt and pre-processes payloads so operators handle decontextualized data on a one-to-one basis. Anonymous graphs and garbled circuits [65] hide the applications that inputs are meant for, so a result is meaningless to an attacker who cannot deanonymize the sender.

Future iterations will support blind quantum computation [23], protecting the quantum kernel by allowing a client with minimal quantum capability to delegate work to a remote

quantum server without revealing the input, computation, or output. Verifiable forms let the client detect a cheating server with overwhelming probability [66], and quantum analogues of garbled circuits extend the same idea to delegated evaluation [67]. These methods are information-theoretically secure in principle but have only small experimental demonstrations [68], so we acknowledge them as aspirational future work.

The same boundaries protect a developer's intellectual property. Developers can choose to run any solver contracts on their own hardware or hosting, protecting their IP from reverse-engineers or open-source scavengers, if that is their preference. The consumer relies on the protocol to hide its data, the developer relies on it to hide the solver, and the operator receives payment for compute without seeing the computation, but knowing the legality of serving that computation.

The verification machinery in §6.1 must work on these concealed payloads. For this reason, the cheap proof required by a subnet (§3.3) certifies a result without revealing the work that produced it.

13.3 Fiat-Denominated Compute Credits

The token is invisible to the buyer. Enterprises buy compute rather than cryptocurrency. A consumer funds an account with fiat or stablecoins and receives cloud credits, a stable currency-denominated balance over QUIP. When a job settles, the system converts between credits and QUIP below the interface using QuipSwap liquidity from the network with the lowest costs.

Consumers see, bid, and receive bills in credits, while the network settles in QUIP. Since conversion occurs only at settlement and the consumer's balance remains stable, the abstraction layer bears price movement between funding and use.

Application developers as the fiat intermediary. The application developer usually holds QUIP, handles conversion, owns the customer relationship, and bills in local currency, much as a software vendor resells cloud capacity at a fiat price. The developer keeps a working QUIP balance, quotes and invoices customers in their own currency, and earns from the spread and solver royalties. Customers avoid token acquisition, custody, and volatility.

The credit model fits familiar enterprise accounting. Customers can prepay blocks of credits that decline per job, receive periodic invoices for metered use, and review per-job costs by department, project, or cost center. Quantum compute appears as an ordinary fiat-denominated line item. QUIP remains in the infrastructure layer, outside procurement and finance workflows.

14 Applications and Participant Workflows

14.1 Initial Application Domains

The first applications come from industries whose optimization problems already fit an Ising/Potts or QUBO/HOBO formulation. In finance, portfolio construction and statistical

arbitrage map directly to the flagship subnet, where a mean-variance objective with risk, budget, cardinality, and turnover constraints becomes a QUBO. The clearinghouse sends it to quantum and classical solvers in parallel, and the first result inside the quality bar wins. Several demos have already exercised this path end to end: portfolio optimization, asset liquidation preference, venue lag forecasting, traveling thief problems, and more.

Logistics adds vehicle routing, bin packing, and scheduling. Materials and drug discovery add molecular ground-state estimation and conformer search. Machine learning adds feature selection, kernel methods, and embeddings. Cryptography is the long-horizon application, where bounties (§6.4) pay for cryptanalysis progress against classical and post-quantum primitives.

Hardware determines when each application arrives. Optimization and sampling problems that current annealers and NISQ devices can handle anchor the early network. Chemistry and cryptanalysis workloads arrive later, under the stages in §2.4, because they require fault tolerance and large logical-qubit counts.

14.2 Participant Workflows

The compute provider. A quantum manufacturer can sell idle capacity, an HPC shop can sell baseline benchmarks, and a classical miner can repurpose its hardware. The operator registers the machine's specifications and completes attestation, benchmarking, and any KYB verification required for whitelisting (§9.3). It then installs the node software, connects the hardware, and selects subnets (§9.1), job filters, and minimum bids (§9.2).

During operation, the provider monitors orders and bids where its hardware is competitive, generates diverse solutions, validates them locally, and submits them. A validated result earns a base reward sized to cover operating costs and a performance bonus based on solution quality and speed, while reputation and uptime multipliers increase rewards for consistent contributions.

The algorithm developer. A researcher publishes a quantum circuit, Monte Carlo method, embedding macro, or problem template in the open library. Academic or cryptographic identity verification binds the researcher to a payment address. Each consumer job that invokes the solver sends its configured royalty to that address. Escrow release settles the royalty as a fraction of the job fee alongside the operator reward and protocol fee (§11.3). A static solver can keep earning without maintenance. New developers find unsolved problems on a bounty board, then earn an initial bounty and continuing revenue share when their solver enters the library.

The enterprise consumer. A firm finds a quantum-amenable problem through an industry funnel (§13.1). It creates an account, which provisions a Quip Account below the interface, then funds the account or buys cloud credits in its own currency (§13.3). Optional KYC or KYB gives the firm access to more operators (§9.3). The firm selects a recommended template, uploads data, and submits a job with a suggested or adjusted bid. The order escrows funds and validates arriving solutions automatically. On release, the firm receives the solution, a performance report, and a cost breakdown. As use matures, price

discovery moves the market clearing price of compute toward its true value: the largest price any firm is willing to pay for the currently available capacity.

The liquidity provider. A return-driven participant supplies liquidity to QuipSwap pools (§12) beside the treasury's permanent positions (§7.4). It earns trading fees and time-limited emissions, with locked positions paying more and granting governance weight. The provider commits QUIP to the Quip Control Asset and routes emissions toward subnets it considers valuable and ready (§10.2). It also votes on subnet proposals and retirements and joins the surprisal and incentive markets (§10.3, §10.4). An enterprising provider may even do offline price discovery at reserved-time venues to arbitrage the difference in price offered by quantum clouds and that offered by the network. One role therefore covers liquidity provision, emission governance, and positions on quantum-advantage claims.

The provider manages impermanent-loss exposure across chains and venues, and helps start new pools. This supports the two-sided market in which consumers acquire QUIP and operators realize rewards without slippage. A consumer with unused compute credits can take the same role and supply those credits as liquidity.

15 Security and Risk

15.1 Signature Schemes

The network uses two post-quantum signature regimes for two threat profiles. On the asset surface, each account acts on host chains and manages its own key state through the hash-based SHRINCS construction [44], [49], whose security reduces to that of its hash function and assumes nothing that a quantum computer is known to break.

A stateful leaf must sign exactly once, creating the main operating constraint described in §12.1. The on-chain leaf bitmap and the lifetime spent-tree registries enforce that rule cryptographically rather than trusting signer bookkeeping. Accounts support many signatures by drawing successive leaves from one committed tree and rotating to a fresh tree well before the budget is exhausted, and the one exception — the dedicated message-verification key, which answers view-only queries and therefore cannot consume anything on-chain — is bounded by the client instead.

Sinnet validators sign repeatedly with long-lived keys and are subject to chain reorganizations that may expose a previously revealed key, so they cannot use a one-time scheme. A single post-quantum scheme would not be alive to the differing storage and liveness requirements of different use-cases. The Sinnet instead uses hybrid signatures for accounts, transactions, and, as consensus matures, block production and finality. Each combines a classical scheme, sr25519 or ed25519, with the lattice-based FN-DSA-512 [69], and remains unforgeable while either component remains secure.

FN-DSA-512 is the pending FIPS 206 standard [69]. It is deterministic, stateless, and can be implemented in constant time, as demonstrated in the Quip library `pqhybridsign`. Domain-separated concatenation binds the classical and post-quantum halves under the emerging IETF standard for composite signatures, and a verifier checks both, so a forger

must break both. For repeated consensus signing, the protocol derives signatures deterministically from the message and chain state, then caches each signature. This prevents the differential-fault attacks that affect naive randomized signing. Appendix A gives the concrete sizes.

Quip does not use ML-DSA [47] for validator security as the signature sizes balloon the chain state too aggressively: the sr25519 + ML-DSA-44 envelope carried roughly 2,484 signature bytes against roughly 730 for FN-DSA-512. The Gaussian sampling and floating-point arithmetic of a stock FN-DSA implementation have shown side-channel and fault weaknesses in long-lived, co-resident signers, so Quip admits only the deterministic, constant-time signing profile described above. Quip also excludes newer multivariate schemes such as SNOVA because recent cryptanalysis leaves their security margins unsettled. The hybrid construction follows the requirement in §2.2. Quip's quantum miners could eventually forge a single classical scheme, while an unforeseen weakness in young post-quantum cryptography should not compromise the chain by itself.

15.2 Sybil and Mining Concentration Risks

Proof of Useful Work makes identities cheap but influence expensive because rewards and consensus weight come from useful compute, not the number of registered nodes. An adversary gains nothing by creating a thousand wallets unless they perform work credited by the subnet's verification function (§6.1), and fabricating that work costs as much as doing it. Consensus weight therefore follows delivered solutions and validated samples rather than node count.

Quantum hardware creates a separate centralization risk. A superconducting system costs tens of millions and cannot be powered down without recalibration (§1.1), so few operators own frontier QPUs. Stage 3 entanglement also requires mining pools (§8), which concentrate capacity by design.

Several protocol mechanisms push in the other direction. Competition among CPUs, GPUs, ASICs, and QPUs keeps classical miners economically viable. Topology-aware difficulty adjustments (§8.1) give each class a target suited to its capability, so a larger or differently-shaped class does not price out the rest.

The Control Asset also binds to a zero-knowledge identity credential (§10.2). It can impose per-identity limits, including quadratic decay in marginal control weight, even if an actor divides its balance among wallets. In Stage 3, entanglement adds geographic distribution as another counterweight. Connecting QPUs across jurisdictions spreads capacity instead of placing it in one location.

15.3 Verification and Oracle Attacks

Results without cryptographic checks expose attacks that tier-one subnets do not face. Tier-two claims use reproducible benchmarks, while tier-three claims use oracle and prediction-market adjudication (§3.3). Operators may collude on false benchmark reports. Oracles may accept bribes or make errors. A miner may also game a naive quality threshold with low-diversity or pre-computed solutions that do not represent genuine work.

Surprisal markets add another attack (§10.3). An actor with private knowledge may move a market's probability before resolution. This inflates the $-\log_2(p)$ multiplier and turns manipulation into reward.

The protocol combines several defenses. Verifiers and oracles post collateral that is slashed if a community appeal overturns their decision, while independent operators re-check a random fraction of accepted results so collusion must survive an unpredictable audit. Quality scoring rewards solution diversity and low latency (§6.2), preventing simple attacks based on reporting one optimum many times or replaying a pre-computed answer.

Oracle adjudication also changes with the type of claim. A Foundation-operated oracle begins with simple objective claims. The system then moves toward reputation-weighted community oracles and uses cryptographic verification by default. Human judgment handles novel claims and an insurance fund covers oracle errors. Surprisal weighting includes prior probability, so likely outcomes pay little and offer less incentive to fabricate a consensus claim.

Together, these mechanisms set an economic-security bound. Each role is staked, accepted results face random re-checks, and the markets price their uncertainty. The design aims to make capture cost more than the reward available from a false outcome.

15.4 Reorganization Attacks

A reorganization attack has a distinct form on Quip because the operators whose useful work secures the chain may eventually own enough hardware to build an alternative history faster than the honest majority. Two defenses limit the depth of such a reorganization.

First, the Sinnet chooses the chain with the most total useful work (§12.4). Under the heaviest-observed-subtree and uncle-block construction, work still adds weight when it does not extend the main chain, so an attacker must out-produce the full useful output of every subnet rather than win one hashing race to rewrite settled history. Second, GRANDPA provides validator-voted finality during bootstrap (§12.4). Under the honest-validator assumption, a finalized block cannot be reorganized, so finality votes cap reorganization depth independently of accumulated work.

Validators add a commitment-checkpoint defense above these mechanisms by rejecting a proposed chain state that omits recent QUIP commitments they have already observed. A deep reorganization that silently drops settled transfers is therefore invalid rather than merely a shorter fork. An earlier design assigned checkpointing to dedicated nodes, but the current design gives it to validators (§5.3, §12.4), which already produce blocks and provide RPC and indexer infrastructure. These roles make validators the natural observers of recent commitments.

The trust assumptions are explicit. Deep-reorganization resistance requires honest operators to control a majority of total useful work and an honest validator supermajority to participate in finality. Commitment checks also require each validator to keep an honest local view of recent commitments. The validator's stake and heaviest-useful-work weighting are intended to align that view with the network.

16 Deployment Stages

16.1 Genesis

The genesis block launches the Sinnet, allocates QUIP under §7.1, and deploys Accounts plus the Interlock swap protocol on the first host chains. During this phase, the Foundation reviews subnets and seeds the treasury's first permanent liquidity positions (§7.4), while operator entry is permissioned through hardware attestation and KYB where applicable (§9.3). Industry funnels bring the first consumers in at the steep end of the subsidy ladder (§13.1).

16.2 Flagship Subnets

The Ising optimization (§6.2) and QVRF (§6.3) flagship subnets receive the full emission at launch. Both produce workloads that anyone can verify, while the Ising subnet measures whether quantum hardware has an advantage over classical alternatives. From the first block, CPU, GPU, ASIC, and QPU operators compete directly (§8.1), and their results create the apples-to-apples benchmark described in §1.1. The verticals in §14.1 then bring in more consumers.

16.3 veQUIP Governance

Vote-escrowed governance starts after the flagship phase proves the mechanism. The Quip Control Asset begins routing emission among subnets (§10.2), the surprisal and incentive markets open (§10.3, §10.4), and subnet proposals move from Foundation review to permissionless entry (§10.1). Activation depends on sustained flagship uptime, fee revenue, unique-user milestones, and clean-security milestones rather than a calendar date, so the network decentralizes allocation only after showing that the flagship system works.

16.4 Multi-Subnet Operation

Stage 2 expands the network to many subnets competing for emission (§2.4). It adds future proofs of useful work as additional subnets (§6.5) and registers the hidden-subgroup and ECDLP cryptanalysis subnet (§6.4) for activation. Under-performing subnets retire (§10.5). Miner sales and put options bring external capital into operator capacity (§7.5), while the vertical applications in §14.1 move from pilots to production. As the fixed emission spreads across more subnets, implicit decay takes effect (§7.2).

16.5 Distributed QPU Network

The final stage begins when repeaters, quantum memory, and error-corrected links mature (§3.1). Mining pools then coordinate entanglement among distributed QPUs (§8.3, §8.4), moving the network from independent processors to pooled, globally entangled capacity. Its heaviest-useful-work chain then secures the literal worldwide quantum computer described in §2.1.

This stage has a multi-decade horizon and depends on physics outside the protocol's control. Quip builds the coordination layer first so new capacity can join as the hardware arrives.

17 Conclusion

Quip organizes quantum and classical compute into one market, where its spot clearinghouse matches work with capacity, its open-source library supplies composable solvers, and its hybrid orchestrator routes each job to the processor that handles it best.

That compute market requires post-quantum security because a network that pays quantum miners cannot rely on cryptography those miners are built to break. The Sinnet therefore uses quantum-resistant consensus, accounts, and messaging, while Accounts and bridgeless QuipSwap let users enter from the chains where their liquidity already resides.

Quip decentralizes in stages. The Foundation gates Genesis, permissionless subnets and the Control Asset take over emission routing after the mechanism proves itself, and mining pools connect the network's compute into globally entangled capacity as quantum networking matures.

This last stage is the hardest to predict, in that the advent of globally entangled quantum computers enable new paradigms in distributed protocols—quantum game theoretic equilibria are subtly different than cooperative game theoretic cores [17], [18], enabling new types of auctions [19], insurance contracts, detection of malfeasance, potentially even new structured products. The very idea of uncloneable quantum money [20], [21] and single-use contracts [22] storeable on portable quantum media could potentially obsolete the very technology of blockchains. If the protocol succeeds in becoming the largest collection of entangled quantum computers, the network will literally be able to solve problems that no other platform on earth can solve, all while remaining verifiable to classical observers under blind quantum computing [66].

The full buildout spans decades, but the coordination layer can serve real workloads on current hardware—today—because its markets, proofs of useful work, and post-quantum settlement do not depend on the physical interconnects required by the final stage. Researchers receive an economic path that the existing industry has not provided, operators gain demand for idle capacity and a benchmark for genuine advantage, consumers gain access to compute they could not otherwise buy, and capital gains a claim on the useful work performed. Quip builds this coordination layer first, then uses it to assemble the worldwide quantum computer.

Appendix A: Parameters

Supply, emission, and treasury

Supply after pre-mine vesting	1,000,000,000 QUIP
Allocation	40% emissions / 20% premine / 15% insiders / 15% builders / 10% community
Annual emission	40,000,000 QUIP (fixed, flat schedule)
Decay	implicit — fixed emission divided across a growing subnet set
Base unit	1 QUIP = 10^{12} (12 decimals)
Treasury policy	protocol fees buy back QUIP into permanent 50/50 liquidity (no burn)

Governance — Quip Control Asset

Primitive	Quip Control Asset (QCA); veQUIP is the vote-escrow implementation
Escrow	enforced via Interlock; position is a liquid restaking token (lendable, borrowable)
Control weight	scales with lock duration, decays toward expiry (curve provisional)
Allocation	per epoch, proportional to weight cast; equal-split fallback
Concentration	ZK-KYC identity-bound; per-identity quadratic decay (provisional)
Monitoring	single-miner alert above $\sim 75\%$ of a subnet's work; subnet viability floor $\sim 1\%$ of emission

Sinnet — consensus and runtime

Base	Substrate solochain (polkadot-sdk fork)
Block production	BABE, 6-second slots, $c = (1, 4)$
Finality	GRANDPA
Canonical chain	heaviest by total useful work; BABE + GRANDPA bootstrap
Account ID	BLAKE2-256 over domain `quip-account-v1` and the hybrid public key
Core pallets (index)	Babe (2), Grandpa (3), Balances (4), Xqvm (8), QuantumComputeMempool (9), QuantumPow (10), Session (12)

Signatures and Accounts

L1 signature	hybrid sr25519 + FN-DSA-512 (FIPS 206, FALCON); both verify, secure while either component holds
Composite size	~ 730 bytes signature, ~ 929 bytes public key (sr25519 + FN-DSA-512)

Rejected for validators	ML-DSA-44 (FIPS 204 — 2,484-byte envelope), SNOVA (cryptanalysis)
Account signature	SHRINCS — stateful hypertree (64 WOTS-C chains/signature) + stateless SPHINCS+-C break-glass; Keccak-256 or SHA2-256 suite
Stateful budget	one leaf per signature, budget fixed at keygen and encoded in the 68-byte stateful public key; spent leaves tracked in an epoch-namespaced on-chain bitmap
Stateless budget	SPHINCS+-C 256s profile, 2^{20} signatures; main-key stateless half used at most once per bundle by construction; the ERC-1271 key's budget is client-enforced
One-time enforcement	per-leaf bitmap + monotonic `keyVersion` + lifetime spent-tree registries over keccak(pkSeed ‖ root)
Replay ordering	every signed context binds the account's live action nonce; one outstanding authorization at a time
Verification cost	ERC-7913 storage-free verifier reached by staticcall: stateful (everyday) signature $\leq\sim$ 167k gas, stateless (break-glass) $\sim$ 1.66M gas; envelope decode $\sim$ 15k; SDK userOp verification-gas ceiling 2,500,000 covers both
Deployment cost	wallet proxy $\sim$ 215k gas through the factory; implementation $\sim$ 4.5M gas at 21.1 KB against the 24.6 KB contract-size limit; shared verifier singleton $\sim$ 780k gas at 3.4 KB
Account abstraction	ERC-4337 (EntryPoint v0.7, `0x0000000071727De22E5E9d8BAf0edAc6f37da032`) + sponsoring paymaster; ERC-1271 view-only message verification
Authorization rule	AND-gate: owner ECDSA (direct caller or in-userOp co-signature) + SHRINCS signature; separate typed-data domains per surface
Deployment	UUPS factory behind an ERC-1967 proxy; sender-guarded CREATE3, so addresses match across chains

Proofs of useful work

Ising objective	$E(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$, with $s_i \in \{-1, +1\}$
Problem bounds	Sherrington-Kirkpatrick expected ground-state energy of the topology [70]
Ising scoring	energy (primary), spin-flip frustration, latency
Fixed point	MILLI_SCALE = 1000 (energy `i64`, coefficients `i32`)
Model RNG	ChaCha8Rng (deterministic, `no_std`)
Ising reward	per-block winner-take-all (lowest energy); epoch 100 blocks; $\leq$ 8 proofs/block

QVRF workload	random-circuit sampling
QVRF verification	linear XEB fidelity $\geq$ threshold within a timing window
QVRF extraction	Toeplitz / Trevisan over validated samples

Execution VM and job market

Problem model	XQMX sparse matrix (linear + quadratic) over binary $[0, 1]$ / spin $[-1, 1]$ / discrete domains
Program form	Encoder / Verifier / Decoder triple; shared instruction set, no shared state
Machine	256 registers; value stack depth $\leq$ 8,192; integer-only (`i64`)
Instruction set	93 opcodes
Bytecode	`.xqb` with 15-byte XQBC header, CRC-32/ISO-HDLC
Step limit	10,000,000 default (on-chain: derived from block weight)
On-chain	Rust `xqvm` crate compiled to WebAssembly in `pallet-xqvm`
Job roles	Builder / Proposer / Solver
Hardware classes	CPU, GPU, ASIC, QPU (D-Wave, IBM, IonQ, Pasqal)
Access modes	Open / Bid (RFQ)
Reward resolution	SingleBest / TopNWeighted / TopNEqual
Result delivery	OnChainOnly / Callback / CallbackWithPoll

Markets and Interlock

Surprisal	$-\log_2(p)$; reward multiplier $1 + \alpha(-\log_2 p)$, $\alpha \approx 0.3$ (range 0.1–0.5)
Market horizons	sprint 3–6 months · research 12–18 months · moonshot multi-year
Interlock construction	hash-time-lock (HTLC) + WOTS+ authentication
Deadline ordering	commitA < commitB < executeA < executeB < reclaimA < reclaimB
Asset types	native, ERC-20, ERC-721, ERC-1155
Recovery	reclaim (timeout refund, formally verified); slashing is future work (adaptor / VES)
Formal verification	EasyCrypt state-separating proofs — passive safety and fairness proven; full active-adversary proof open

Appendix B: Glossary

QUIP (Quantum Internet Protocol) The network's native token and shared unit of compute credit; also the name of the self-custodial account primitive whose spending authority is a hash-based one-time signature (§12.1).

Account The on-chain QUIP wallet deployed on a host chain, authorized by a SHRINCS signature plus the owner's classical signature, consuming one one-time leaf per operation; the unit of the asset surface (§12.1).

SHRINCS The account's hash-based signature construction: one public commitment over a stateful hypertree of one-time leaves (everyday path, budgeted) plus a stateless SPHINCS+-C break-glass path for recovery and rotation (§12.1).

Sinnet The network's sovereign Substrate layer-one, whose canonical chain is the heaviest by interleaved total useful work; it hosts the subnets, the job market, the DAG of swaps, and settlement (§12.4).

QuipSwap The peer-to-peer, bridgeless, request-for-quote swap market layered over the Quantum Interlock Protocol (§12.3).

Quantum Interlock Protocol The atomic cross-chain swap settled by a shared hash lock and each party's SHRINCS signature, with a commit–execute–reclaim lifecycle (§12.2).

hash lock (HTLC) A contract that releases a locked asset only on revelation of a secret preimage, or returns it after a timeout; the canonical Interlock construction (§12.2).

VES / adaptor signature A verifiably encrypted signature that folds the lock into the signature itself; a future Interlock upgrade that would enable slashing. No well-studied post-quantum construction yet exists (§12.2).

WOTS+ Winternitz One-Time Signature Plus — a hash-based signature whose security rests only on its hash function; the primitive underlying SHRINCS leaves and the sunset first-generation Account wallets (§15.1).

hybrid signature A composite of a classical scheme (sr25519 or ed25519) and a post-quantum scheme (FN-DSA-512), unforgeable as long as either component holds; secures Sinnet (§15.1).

FN-DSA-512 The FIPS 206 lattice-based (FALCON) signature standard used as the post-quantum half of Sinnet's hybrid signatures (§15.1).

XQVM (X-Quadratic Virtual Machine) The deterministic, hardware-agnostic execution VM whose programs describe a quadratic optimization model and independently recompute a solution's energy (§11.4).

QVRF (Quantum Verifiable Randomness Function) The flagship subnet that produces certified public randomness via random-circuit sampling verified by cross-entropy benchmarking (§6.3).

XEB (cross-entropy benchmarking) The classical statistical test scoring how closely sampled outputs track an ideal quantum circuit; the QVRF's verification metric (§3.3, §6.3).

Ising model An optimization over ± 1 spins minimizing $E(s) = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j$; the flagship subnet's problem class, equivalent to QUBO (§6.2).

surprisal The information content $-\log_2(p)$ of an outcome the market priced at probability p; the reward multiplier in the prediction markets (§10.3).

QCA (Quip Control Asset) The governance-and-allocation primitive that routes emission across subnets each epoch; veQUIP is its canonical implementation (§10.2).

veQUIP The vote-escrowed implementation of the Quip Control Asset — QUIP committed into Interlock escrow to earn time-weighted control weight while remaining liquid (§10.2).

Validator A consensus node that produces blocks and serves as RPC, indexer, and multi-party-computation infrastructure; it enforces the commitment-omission reorg defense (§5.3, §15.4).

Miner A compute provider in its consensus guise — it solves a proof of useful work and broadcasts the winning block (§5.3).

Nominator A staker who delegates QUIP behind a validator that lacks the minimum bond, sharing its yield and its slashing risk (§5.3).

Quantum Cluster A coordinating cluster of operators that earns a bounty by demonstrating a classical or post-quantum cryptographic primitive is broken; the cryptanalysis subnet of §6.4.

Appendix C: References

[1] Quantum Economic Development Consortium, "State of the Global Quantum Industry 2026," technical report, 2026.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," vol. 26, no. 5. pp. 1484–1509, 1997. doi: 10.1137/S0097539795293172.

[3] C. Gidney, "How to factor 2048 bit RSA integers with less than a million noisy qubits." 2025.

[4] R. Babbush *et al.*, "Securing elliptic curve cryptocurrencies against quantum vulnerabilities: resource estimates and mitigations." 2026.

[5] C. Gidney and M. Ekerå, "How to factor 2048-bit RSA integers in 8 hours using 20 million noisy qubits," *Quantum*, vol. 5, p. 433, 2021, doi: 10.22331/q-2021-04-15-433.

[6] P. Webster *et al.*, "The Pinnacle architecture: reducing the cost of breaking RSA-2048 to 100,000 physical qubits using quantum LDPC codes." 2026.

[7] M. Cain *et al.*, "Shor's algorithm is possible with as few as 10,000 reconfigurable atomic qubits." 2026.

[8] D. J. Bernstein, "Cryptographic maintenance: a Boeing 747 in flight." 2025.

[9] E. Parker and M. J. D. Vermeer, "Estimating the Energy Requirements to Operate a Cryptanalytically Relevant Quantum Computer," technical report Working Paper WR-A2427-1, 2023.

[10] U.S. Energy Information Administration, "Electric Power Monthly: average price of electricity to ultimate customers by end-use sector." 2026.

[11] A. W. Butler and A. Herman, "Prosperity at Risk: The Quantum Computer Threat to the U.S. Financial System," technical report, 2023.

[12] H. Beast, E. Heilman, and I. Foxen Duke, "BIP-360: pay to quantum resistant hash / pay-to-Merkle-root (P2MR)." 2024.

[13] Ethereum Foundation, "Post-quantum cryptography on Ethereum." 2026.

[14] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, pp. 802–803, 1982, doi: 10.1038/299802a0.

[15] A. K. Pati and S. L. Braunstein, "Impossibility of deleting an unknown quantum state," *Nature*, vol. 404, pp. 164–165, 2000, doi: 10.1038/404130b0.

[16] M. Fitzi, N. Gisin, and U. Maurer, "Quantum solution to the Byzantine agreement problem," *Physical Review Letters*, vol. 87, p. 217901, 2001, doi: 10.1103/PhysRevLett.87.217901.

[17] J. Eisert, M. Wilkens, and M. Lewenstein, "Quantum games and quantum strategies," *Physical Review Letters*, vol. 83, pp. 3077–3080, 1999, doi: 10.1103/PhysRevLett.83.3077.

[18] D. A. Meyer, "Quantum strategies," *Physical Review Letters*, vol. 82, pp. 1052–1055, 1999, doi: 10.1103/PhysRevLett.82.1052.

[19] T. Hogg, P. Harsha, and K.-Y. Chen, "Quantum auctions," *International Journal of Quantum Information*, vol. 5, no. 5, pp. 751–780, 2007, doi: 10.1142/S0219749907003183.

[20] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983, doi: 10.1145/1008908.1008920.

[21] S. Aaronson and P. Christiano, "Quantum money from hidden subspaces," in *Proc. 44th ACM Symposium on Theory of Computing (STOC)*, 2012, pp. 41–60. doi: 10.1145/2213977.2213983.

[22] R. Amos, M. Georgiou, A. Kiayias, and M. Zhandry, "One-shot signatures and applications to hybrid quantum/classical authentication," in *Proc. 52nd ACM Symposium on Theory of Computing (STOC)*, 2020, pp. 255–268. doi: 10.1145/3357713.3384304.

[23] A. Broadbent, J. Fitzsimons, and E. Kashefi, "Universal blind quantum computation," in *Proc. 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009, pp. 517–526.

[24] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, "Quantum repeaters: the role of imperfect local operations in quantum communication," *Physical Review Letters*, vol. 81, pp. 5932–5935, 1998, doi: 10.1103/PhysRevLett.81.5932.

[25] D. Barral *et al.*, "Review of distributed quantum computing: from single QPU to high-performance quantum computing," *Computer Science Review*, vol. 57, p. 100747, 2025, doi: 10.1016/j.cosrev.2025.100747.

[26] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Physical Review E*, vol. 58, pp. 5355–5363, 1998, doi: 10.1103/PhysRevE.58.5355.

[27] A. D. King and others, "Beyond-classical computation in quantum simulation," *Science*, vol. 388, pp. 199–204, 2025, doi: 10.1126/science.ado6285.

[28] A. Kawasaki and others, "Real-time observation of picosecond-timescale optical quantum entanglement towards ultrafast quantum information processing," *Nature Photonics*, vol. 19, pp. 271–276, 2025, doi: 10.1038/s41566-024-01589-7.

[29] M. Aghaee and others, "Interferometric single-shot parity measurement in InAs–Al hybrid devices," *Nature*, vol. 638, pp. 651–655, 2025, doi: 10.1038/s41586-024-08445-2.

[30] Google Quantum AI and Collaborators, "Quantum error correction below the surface code threshold," *Nature*, vol. 638, pp. 920–926, 2025, doi: 10.1038/s41586-024-08449-y.

[31] D. Bluvstein, S. J. Evered, A. A. Geim, and others, "Logical quantum processor based on reconfigurable atom arrays," *Nature*, vol. 626, pp. 58–65, 2024, doi: 10.1038/s41586-023-06927-3.

[32] H. Putterman, K. Noh, C. T. Hann, and others, "Hardware-efficient quantum error correction via concatenated bosonic qubits," *Nature*, vol. 638, pp. 927–934, 2025, doi: 10.1038/s41586-025-08642-7.

[33] Y. Ma, Y.-Z. Ma, Z.-Q. Zhou, C.-F. Li, and G.-C. Guo, "One-hour coherent optical storage in an atomic frequency comb memory," *Nature Communications*, vol. 12, p. 2381, 2021, doi: 10.1038/s41467-021-22706-y.

[34] A. Lucas, "Ising formulations of many NP problems," *Frontiers in Physics*, vol. 2, p. 5, 2014, doi: 10.3389/fphy.2014.00005.

[35] F. Arute, K. Arya, R. Babbush, and others, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, 2019, doi: 10.1038/s41586-019-1666-5.

[36] S. Aaronson and S.-H. Hung, "Certified randomness from quantum supremacy," in *Proc. 55th ACM Symposium on Theory of Computing (STOC)*, 2023, pp. 933–944. doi: 10.1145/3564246.3585145.

[37] M. Liu, R. Shaydulin, P. Niroula, and others, "Certified randomness using a trapped-ion quantum processor," *Nature*, vol. 640, pp. 343–348, 2025, doi: 10.1038/s41586-025-08737-1.

[38] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th ACM Symposium on Theory of Computing (STOC)*, 1996, pp. 212–219. doi: 10.1145/237814.237866.

[39] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system." 2008.

[40] Opentensor Foundation, "Bittensor: a peer-to-peer intelligence market." 2021.

[41] G. Wood, "Polkadot: vision for a heterogeneous multi-chain framework." 2016.

[42] L. Baird, "The Swirlds hashgraph consensus algorithm: fair, fast, Byzantine fault tolerance," technical report SWIRLDS-TR-2016-01, 2016.

[43] The Quantum Resistant Ledger, "QRL: an XMSS-based post-quantum blockchain."

[44] A. Hülsing, "W-OTS+ — shorter signatures for hash-based signature schemes," in *AFRICACRYPT 2013, LNCS 7918*, 2013, pp. 173–188. doi: 10.1007/978-3-642-38553-7_10.

[45] Cellframe, "ChipmunkRing: a practical post-quantum ring signature scheme for blockchain applications." 2025.

[46] Quantus Network, "Technical foundations of quantum security." 2026.

[47] National Institute of Standards and Technology, "FIPS 204: Module-Lattice-Based Digital Signature Standard (ML-DSA)." 2024. doi: 10.6028/NIST.FIPS.204.

[48] Project Eleven, "yellowpages: binding existing addresses to post-quantum keypairs." 2025.

[49] National Institute of Standards and Technology, "FIPS 205: Stateless Hash-Based Digital Signature Standard (SLH-DSA)." 2024. doi: 10.6028/NIST.FIPS.205.

[50] F. Barahona, "On the computational complexity of Ising spin glass models," *Journal of Physics A: Mathematical and General*, vol. 15, no. 10, pp. 3241–3253, 1982, doi: 10.1088/0305-4470/15/10/028.

[51] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm." 2014.

[52] H. Kim and S. Hong, "New space-efficient quantum algorithm for binary elliptic curves using the optimized division algorithm," *Quantum Information Processing*, vol. 22, p. 237, 2023, doi: 10.1007/s11128-023-03991-6.

[53] M. Garn and A. Kan, "Quantum resource estimates for computing binary elliptic curve discrete logarithms," *IEEE Transactions on Quantum Engineering*, vol. 6, pp. 1–23, 2025, doi: 10.1109/TQE.2025.3586541.

[54] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels," *Physical Review Letters*, vol. 70, pp. 1895–1899, 1993, doi: 10.1103/PhysRevLett.70.1895.

[55] D. Gottesman and I. L. Chuang, "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations," *Nature*, vol. 402, pp. 390–393, 1999.

[56] D. Main *et al.*, "Distributed quantum computing across an optical network link," *Nature*, vol. 638, pp. 383–388, 2025, doi: 10.1038/s41586-024-08404-x.

[57] M. Sena *et al.*, "High-fidelity quantum entanglement distribution in metropolitan fiber networks with co-propagating classical traffic," *Journal of Optical Communications and Networking*, vol. 17, no. 12, pp. 1072–1081, 2025.

[58] T.-X. Zhu *et al.*, "A metropolitan-scale multiplexed quantum repeater with Bell nonlocality," *Nature Photonics*, vol. 20, p. 812, 2026, doi: 10.1038/s41566-026-01911-5.

[59] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, "Simulating large quantum circuits on a small quantum computer," *Physical Review Letters*, vol. 125, p. 150504, 2020, doi: 10.1103/PhysRevLett.125.150504.

[60] W. Tang, T. Tomesh, M. Suchara, J. Larson, and M. Martonosi, "CutQC: using small quantum computers for large quantum circuit evaluations," in *Proc. 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2021, pp. 473–486. doi: 10.1145/3445814.3446758.

[61] Ethereum community, "ERC-7913: signature verifiers." 2025.

[62] V. Buterin, Y. Weiss, D. Tirosh, S. Gazso, T. Nomura, and others, "ERC-4337: account abstraction using alt mempool; ERC-1271: standard signature validation method for contracts." 2021.

[63] J. Gugger, "Bitcoin–Monero cross-chain atomic swap." 2020.

[64] G. Barthe, B. Grégoire, S. Heraud, and S. Z. Béguelin, "Computer-aided security proofs for the working cryptographer," in *CRYPTO 2011, LNCS 6841*, 2011, pp. 71–90. doi: 10.1007/978-3-642-22792-9_5.

[65] A. C. Yao, "How to generate and exchange secrets," in *Proc. 27th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1986, pp. 162–167. doi: 10.1109/SFCS.1986.25.

[66] J. F. Fitzsimons and E. Kashefi, "Unconditionally verifiable blind quantum computation," *Physical Review A*, vol. 96, p. 12303, 2017.

[67] Z. Brakerski and H. Yuen, "Quantum garbled circuits." 2020.

[68] P. Drmota, D. P. Nadlinger, D. Main, and others, "Verifiable blind quantum computing with trapped ions and single photons," *Physical Review Letters*, vol. 132, p. 150604, 2024.

[69] National Institute of Standards and Technology, "FIPS 206 (draft): FN-DSA — FFT over NTRU-Lattice-Based Digital Signature Standard (FALCON)." 2026.

[70] D. Sherrington and S. Kirkpatrick, "Solvable model of a spin-glass," *Physical Review Letters*, vol. 35, pp. 1792–1796, 1975, doi: 10.1103/PhysRevLett.35.1792.