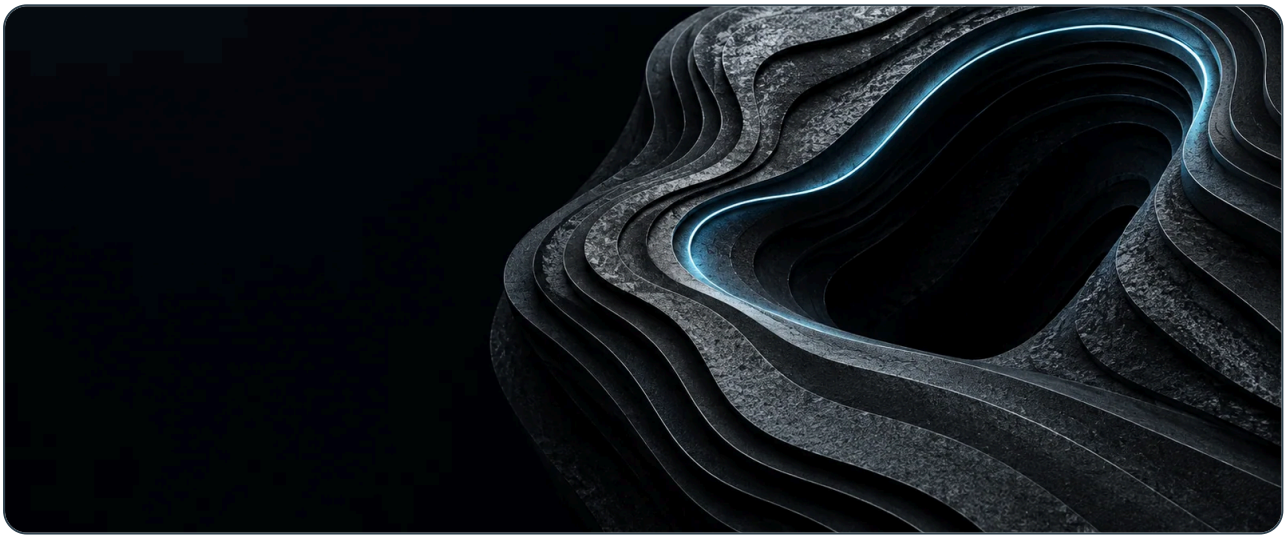
THE DARKPOOL WHITEPAPER

A private space.
A public chain.

Architecture, privacy boundaries and a direction for a more private Ethereum.

Version 0.1 | 12 October 2026 | Prelaunch working paper



Working paper · public activation pending

This paper describes the current design and future direction. No independent security audit has been completed.

Check verified availability →

01

Abstract

Darkpool is an Ethereum privacy-wallet project built around a shielded balance, locally generated proofs and recoverable wallet state. Its initial purpose is to reduce the direct onchain link between the source of funds and subsequent activity. Its longer-term direction is a broader set of privacy tools for payments, application interactions and developers.

The current implementation includes shielded-note transfers, public deposits and withdrawals, seed-derived fresh wallets, and payment and disclosure modules at different stages of testing. Public utility activation is pending. This paper describes the design and its limits; it is neither evidence of a completed deployment nor a security audit.

02

The problem: public infrastructure, personal activity

Ethereum makes settlement independently verifiable. That transparency also exposes financial relationships: addresses, asset movements, balances, counterparties and the timing of public transactions. Reusing one address can make otherwise unrelated activity easy to associate.

Darkpool approaches privacy as a set of explicit boundaries. Users should be able to understand what is hidden, what remains observable and what a service provider can learn. The objective is useful privacy with verifiable execution and recovery, rather than a claim of universal anonymity.

03

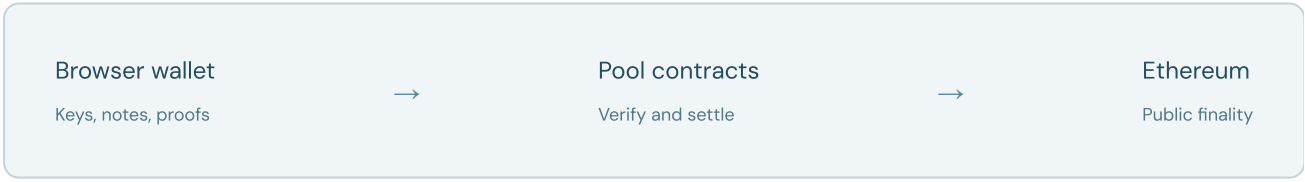
Design principles

- **User-controlled keys.** Spending authority derives from the wallet seed; recovery must not depend solely on a hosted interface.
- **Local proof generation.** The wallet constructs proofs on the user's device. The design does not require sending the spending seed to a relayer.
- **Visible privacy boundaries.** Public transaction legs, fees, timing and service metadata must be described before an action.
- **Verifiable releases.** Contract configuration, proof artifacts and the published deployment must agree.
- **Progress through evidence.** A built feature, a tested feature and a publicly available feature are different states.

04

System architecture

The system separates the browser wallet, the shielded-pool contracts and supporting services. The wallet derives keys, decrypts notes, prepares witnesses and generates proofs. A pool records note commitments and spent nullifiers, verifies proofs and enforces its public operation parameters. An indexer and RPC connection supply chain data; a relayer can submit a constrained operation and pay network gas.



The repository's current core uses a Circom JoinSplit circuit with Groth16 proofs and Poseidon-based commitments. The pool operation has four nullifier slots and four output commitment slots. These are implementation properties, not a promise that the complete system has been independently verified. A final production release must identify its precise circuit and artifact set.

05

Notes, commitments and spending

A note represents value owned by a spending identity. Its commitment is recorded in the pool's Merkle tree; encrypted note data allows the intended wallet to discover and reconstruct its balance. A valid spending proof demonstrates the required ownership and inclusion relationships while satisfying the circuit's value constraints.

A nullifier marks an input as spent without publishing its note opening. The pool rejects reused nullifiers. Outputs create new commitments, allowing value to be split, transferred or retained as change. Public input and output amounts, fees and operation bindings remain part of the visible transaction context.

Note encryption and spending use distinct key material. The implementation uses X25519, HKDF and XChaCha20-Poly1305 for note encryption. Cryptographic key separation does not make the browser or device immune to compromise.

06

The transaction lifecycle

1. **Shield.** A public deposit brings supported value into a pool and creates encrypted notes. The depositing address, asset, amount and time remain visible.
2. **Transfer privately.** A proof spends existing notes and creates notes for a Darkpool receive address. Note ownership and private transfer amounts are not published as plaintext fields; commitments, nullifiers, fees and timing remain observable.
3. **Withdraw or interact.** A withdrawal pays a public Ethereum address. A trading integration adds public market activity. The funding note is not directly exposed, but correlations may remain possible.

Quotes, recipient bindings, deadlines and the intended operation must be checked before signing. The wallet's transaction journal supports consistent retries and reconciliation; the chain determines whether an operation actually settled.

07

Privacy model and remaining visibility

SURFACE	PROTECTED BY THE DESIGN	OBSERVABLE OR POTENTIALLY LINKABLE
Inside a pool	Note openings, ownership and private transfer relationships	Commitments, nullifiers, fees, pool activity and transaction timing
Deposits and withdrawals	The spent note is not directly identified by its opening	Public addresses, amounts, asset and timing at the boundary
Fresh-wallet trading	The direct connection to a funding note	Fresh wallet, trade, balances, route and execution details
Network and device	Spending secrets need not be given to the relay	IP or request metadata may reach RPC or relay providers; a compromised device can expose secrets

Timing, distinctive amounts, address reuse and a small set of plausible counterparties can weaken privacy. A relayer-paid transaction does not remove all network metadata. Darkpool does not promise anonymity, invisibility or protection from every form of chain analysis.

08

Fresh wallets and application interactions

Ghost Buy is designed to fund a Stockereum purchase from shielded ETH into a fresh seed-derived wallet. That wallet and its trade are public. Ghost Sell uses delegated account execution with sponsored gas and re-shields the guaranteed proceeds; any recoverable surplus and public market activity must be explained to the user.

EIP-7702 provides account-code delegation that can support sponsored execution. It is an execution mechanism, not a privacy guarantee. Darkpool's integration depends on the exact delegated account implementation, signed constraints and relay behavior. Read EIP-7702 ↗

Shielded swaps and future venue integrations must disclose their public legs. Market, pool, amount, fee and timing visibility should never be described as a confidential trade merely because its input came from a shielded balance.

09

Payments and selective disclosure

Payment requests, local invoice records, private batch payouts and scheduled withdrawals extend the wallet into payment workflows. These modules are under development or testing and have separate deployment dependencies. A link or QR code may reveal payment details to its holder. Publicly funded batches reveal funding information; schedules also depend on reliable submission and current chain state.

Selective disclosure lets a user intentionally share information. An all-pools viewing key can reveal existing and future notes across pools and networks using that receive key. A pool snapshot exposes selected observed notes, while a one-payment receipt targets a narrower payment. Shared information cannot be revoked by deleting the original file.

Viewing exports do not confer spending authority, but they are sensitive capabilities. A receipt or local ledger is not automatically a complete, independently certified account history. The recipient of a disclosure must understand its scope and verification limits.

10

Security assumptions and recovery

Security depends on the cryptographic primitives, circuit constraints, proof ceremony, contract implementation, wallet software and operating environment. An independent audit has not been completed. Tests and fork rehearsals are valuable development evidence, but they are not substitutes for an independent security assessment.

The proof ceremony and artifact verification must be finalized before production use. The published wallet must use the exact artifacts expected by the deployed verifier. Changes to a circuit can require new artifacts and a separately reviewed deployment or migration.

The core pool’s deposit guardian can adjust the acceptance of new deposits; the role is not a general authority to redirect note owners’ withdrawals. Optional screening applies at documented deposit boundaries. A production release must publish the actual configuration and role assignments.

The 24-word seed restores spending keys and derived wallets. Chain data, correct artifacts and a compatible recovery client are still needed. Encrypted backups preserve additional local state. Direct submission offers a path when a relay is unavailable, but requires network access and gas and reveals the submitting wallet.

11

Fees and the role of DARK

Users must distinguish Ethereum gas from application and service charges. Cost varies by operation, quote and network conditions. The wallet should show the exact supported quote before signing; the fees guide describes the current design.

The core ETH privacy utility does not require the DARK token to exist. DARK-related fees, staking and other token-dependent modules require a real, verified token deployment and their own release configuration. The public token address has not been published as a verified live deployment at this paper's revision.

Future economic policies are design decisions, not guaranteed rewards, returns or price support. The token plan and verified status page are separate from the utility's technical specification.

12

Delivery and release discipline

The immediate priority is a bounded public release with verifiable contracts, finalized proof artifacts, reliable hosted services and demonstrated recovery. Later phases improve everyday usability, complete payment workflows, expand carefully reviewed application integrations and make the building blocks more accessible to developers.

The project roadmap sets out six phases with deliverables, validation gates and evidence to collect. Phases can overlap. They express priorities and intended work, not dated guarantees. A feature becomes public only after its actual dependencies and release checks are satisfied.

13

Research toward broader Ethereum privacy

The longer-term ambition is to improve privacy across more of the Ethereum experience: transaction relationships, receiving addresses, application permissions, intentional disclosure and network requests. No single pool or wallet solves all of these layers.

Research areas include metadata minimization, alternative submission paths, stealth-address interoperability, scoped viewing capabilities and proving efficiency. ERC-5564 is a reference point for stealth-address research; Darkpool does not currently claim to implement that standard. Read ERC-5564 ↗

The repository also contains an association-set constraint experiment. It is a research prototype: it does not implement a complete payment protocol, prove deposit ancestry or establish a production provider policy. A production design would require separate protocol work, circuit and ceremony decisions, security review and a migration strategy. It cannot be enabled by relabelling the existing proofs.

14

References and revision scope

This working paper is grounded in the current Darkpool source, protocol implementation and project delivery notes. It describes design intent and documented implementation behavior as of 12 October 2026. Deployment facts can change; consult the status page for current availability.

- Darkpool corresponding source and license notices
- Darkpool privacy model and security model
- Ethereum.org: zero-knowledge proofs — background on proving without revealing private inputs
- EIP-7702: account-code delegation
- ERC-5564: stealth addresses

Revision 0.1 establishes the architecture, privacy boundaries, release discipline and six-phase project direction. Future revisions should document material protocol changes and their compatibility implications.