Specter: Dark Pool Liquidity on Solana

The Specter Foundation

Abstract

Public Solana DEXs expose the full transaction graph, making large trades easy to front-run and reverse-engineer. Specter is a dark-pool-oriented routing layer built around a single hub asset, $SPEC, deployed on Meteora DAMM v2. The protocol models the Solana liquidity graph as a weighted, directed multigraph and applies a cost functional that incorporates spread, depth, transfer taxation and venue entropy. Traders see a unified interface; under the hood, Specter decomposes orders across a set of proprietary AMM venues while applying a privacy-weighted score to each route. The implementation is intentionally simple, relying on existing Solana routing infrastructure for execution, while the economic surface exposed to users is governed by a 3.5% transfer fee on $SPEC itself and a 0.25% swap fee on routed trades. This document formalizes the cost model, fee mechanics, and privacy scoring used by the interface, providing a precise description of what the system claims to optimize, without constraining how the underlying execution client is implemented.

1 Introduction

Solana offers sub-second finality and deep on-chain liquidity, but most routing layers expose a completely transparent execution surface. A sufficiently motivated observer can reconstruct who traded, in which pools, and with what price impact. For large directional orders, this visibility increases execution cost via adverse selection and sandwich attacks. Dark pools on Solana, often implemented as proprietary AMMs with private or semi-private front-ends, attempt to mitigate this by hiding order flow, but fragment liquidity and add integration complexity for ordinary users.

Specter takes the opposite approach: instead of building a new matching engine or chain, it wraps existing Solana liquidity into a single, hub-centric view. The $SPEC asset acts as the routing nucleus. Pairs of the form SPEC–X live on Meteora DAMM v2; Specter aggregates quotes from proprietary AMM venues and public pools and projects them into a unified cost model. The goal is not to beat every specialized router on raw price, but to provide a visually and mathematically coherent picture of dark-pool execution around a single token.

This paper introduces three main components: (1) a dark-pool liquidity graph and execution cost functional, (2) a deterministic fee model built around a fixed total supply of 100 million $SPEC, and (3) a privacy-score heuristic that assigns each trade a scalar score between 0 and 10. The emphasis is on algebraic clarity rather than implementation minutiae: any execution client that can realize the described routing decisions on Solana is considered conformant.

2 Liquidity Graph and Dark Pools

We model the accessible Solana liquidity as a directed multigraph $G = (V, E)$, where each vertex v in V corresponds to an SPL token mint and each edge e in E corresponds to a tradeable pool between two tokens. For Specter, we distinguish a subset of vertices that contain $SPEC as one leg (for example SPEC–SOL, SPEC–USDC, SPEC–wETH) and a subset of edges that represent dark-pool or proprietary AMM venues, such as HumidiFi-style dark pools.

Each edge e is annotated with a tuple of execution parameters:

$$e = (u, v, D, s, \lambda, \xi)$$

Here u and v are the endpoint tokens, D is an effective depth denominated in the quote asset, s is the quoted half-spread in basis points, λ is an effective latency term in milliseconds, and ξ is an opacity score in the interval [0, 1] describing how dark the venue is ($\xi = 1$ for fully private, $\xi = 0$ for fully transparent order books).

Given a path π that routes a notional size Q through a sequence of pools, we approximate the execution cost as:

$$C(\pi, Q) = Q \cdot (1 + S(\pi, Q) + I(\pi, Q)) + L(\pi)$$

The term $S(\pi, Q)$ collects spread contributions, $I(\pi, Q)$ captures impact from finite depth, and $L(\pi)$ is a latency penalty. For a single edge, we use the rough approximations:

$$S^{edge}(Q) \approx s$$

$$I^{edge}(Q) \approx (Q / D)^2$$

For a path π that traverses a set of edges, we sum these contributions:

$$S(\pi, Q) = \Sigma_{e \in \pi} S^{edge}(Q)$$

$$I(\pi, Q) = \Sigma_{e \in \pi} I^{edge}(Q)$$

The latency term is modeled as a path-dependent penalty:

$$L(\pi) = \sigma \cdot \sqrt{\Delta t} \cdot L_{agg}(\pi)$$

Here σ is an exogenous volatility scale, Δt is an effective latency window for the path, and $L_{agg}(\pi)$ is a dimensionless aggregate latency across venues. Dark-pool edges typically have higher latency but are offset by lower observed spreads and reduced information leakage.

3 Fee Model and $SPEC Supply

Specter assumes a fixed total supply of the hub asset:

$$S_{total} = 100{,}000{,}000\ SPEC$$

At any time t, the circulating supply $S_c(t)$ satisfies $0 < S_c(t) \leq S_{total}$. The remainder is held in program-visible accounts subject to external locking rules, such as vesting or multisig control, which are not specified here. The only invariant assumed by the model is conservation of total supply.

Two deterministic fees interact with this supply: (1) a 3.5% transfer fee on movements of $SPEC itself, and (2) a 0.25% swap fee on notional routed through the Specter interface. For a transfer of q units of $SPEC, the fee and net outflow are:

$$q_{fee} = 0.035 \cdot q$$

$$q_{out} = q - q_{fee} = 0.965 \cdot q$$

If B_t denotes the cumulative balance of all fee-collecting accounts at time t, then:

$$B_{t+1} = B_t + \Sigma_{transfers\ in\ (t,\ t+1]}\ q_{fee}$$

Similarly, the circulating supply evolves according to:

$$S_c(t + 1) = S_c(t) - \Sigma_{transfers}\ q_{fee} + \Delta_{unlock}(t)$$

Here $\Delta_{unlock}(t)$ encodes any one-way unlocks from locked allocations. If no unlocks occur, transfer activity induces a strictly decreasing circulating supply.

For routed swaps, we define $V_{swap}(t)$ as the total notional volume passing through the Specter interface over a horizon. The swap fee collected is:

$$F_{swap}(t) = 0.0025 \cdot V_{swap}(t)$$

How these fees are disposed of, for example forwarded to a treasury, used off-chain, or recycled into incentives, is left as an implementation detail. The only invariant assumed in this document is that the fee rates themselves are mechanically enforced by the contracts or by a tightly coupled execution client.

4 Specter Routing Surface

From a routing perspective, Specter presents users with a unified surface over the Solana DEX graph. Given a candidate path π and trade size Q, the router evaluates a cost functional that trades off execution price against privacy-oriented preferences. We write:

$$J(\pi; Q, \theta) = C(\pi, Q) - \theta \cdot \Omega(\pi)$$

Here $C(\pi, Q)$ is the execution cost defined in Section 2 and $\theta \in [0, 1]$ is a user- or interface-level preference parameter. The term $\Omega(\pi)$ is a privacy-oriented benefit functional. A simple choice is:

$$\Omega(\pi) = w_H \cdot H(\pi) + w_D \cdot D(\pi)$$

The quantity $H(\pi)$ is a route-entropy term computed over the venues actually used along the path, and $D(\pi)$ is the fraction of notional executed in dark or proprietary venues. The weights w_H and w_D control how aggressively Specter sacrifices raw price for venue diversity and dark-pool usage. For $\theta = 0$, the router reduces to a pure cost minimizer; for θ close to 1, it favors high-entropy, high-opacity paths.

In practice, Specter samples a set of candidate routes from the Solana DEX graph, evaluates $J(\pi; Q, \theta)$ for each, and selects a path or mixture of paths with minimal adjusted cost. The exact sampling strategy is left open so that integrators can plug in their own routing engines as long as they honor the functional J defined here.

5 Dark Pool Routing Architecture

Figure 1 illustrates the high-level architecture of Specter's dark pool routing mechanism. When a user initiates a trade, the request flows through the Specter Router, which evaluates all available liquidity venues through the $SPEC hub asset. The router applies the privacy-weighted cost functional to select optimal paths, preferring dark pool venues (shown with thicker connections) when the privacy benefit outweighs additional execution costs.

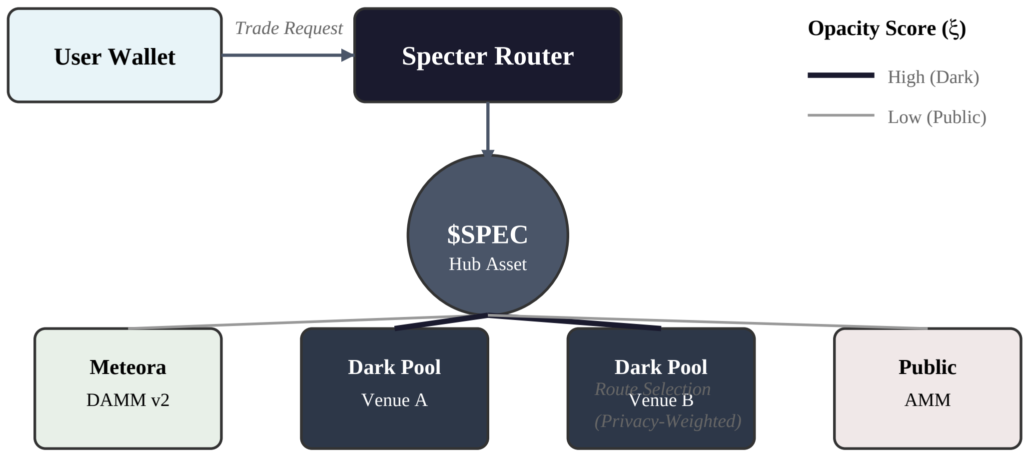

Figure 1: Specter Dark Pool Routing Architecture. Trade requests flow through the $SPEC hub, with route selection weighted by venue opacity scores (ξ). Dark pool venues receive preferential routing when privacy benefits exceed cost differentials.

6 Privacy Score

Beyond per-trade routing, Specter assigns each wallet address a scalar Privacy Score *PS* in the interval [0, 10], intended purely as a user-facing metric. The score is derived from four dimensions measured over a rolling 30-day window:

(1) venue diversity — how many distinct venues are touched; (2) trade size variance — how unpredictable the notional sizes are; (3) wallet freshness — how recently the wallet first appeared; and (4) temporal entropy — how irregular the trade timestamps are.

We map these dimensions into a single score via:

$$PS = 3 \cdot \log(N + 1) + 2.5 \cdot V + 2 \cdot W + 2.5 \cdot T$$

Here N is the number of unique venues visited in the window, V is a normalized variance score based on the ratio of standard deviation to mean trade size, W is a decaying function of wallet age (newer wallets score higher), and T is a normalized time-entropy score based on the distribution of trades across the day. Each component is clipped so that $PS \leq 10$. The coefficients are chosen to keep typical active users in the midrange while making it possible to approach 10 only with highly diverse, irregular, and fresh activity patterns.

For example, a wallet that touches five different venues, uses uneven trade sizes, was created recently, and trades at irregular times might achieve *PS* around 8 on this scale.

7 Meteora DAMM v2 Embedding

Pairs of the form SPEC–X live on Meteora DAMM v2 pools. From Specter's perspective, each such pool provides a price and depth surface that can be treated as another edge in the liquidity graph. Let ζ denote the DAMM configuration, including reserves and curve parameters:

$$\zeta = (R_{SPEC}, R_X, \kappa, \phi, \ldots)$$

Given ζ, Meteora defines a pricing function *f* that maps an input trade size into an output price for a SPEC→X swap:

$$P_{SPEC \to X} = f(\zeta)$$

For small trades, the marginal price impact can be linearized around current reserves, but Specter does not alter these dynamics. Instead, it treats each DAMM pool as an edge *e* with depth *D* derived from the current reserves (R_{SPEC}, R_{X}) and an observed half-spread *s* estimated from recent trades. Dark-pool behavior is achieved by rendering these pools, together with external proprietary AMM venues, as part of a unified mesh around SPEC, rather than by modifying Meteora itself.

8 Risk Surface (Quantitative View)

From a quantitative perspective, each routed trade induces three primary risk components: execution slippage from finite depth, adverse selection from latency, and fee drag from the $SPEC transfer tax when routes traverse SPEC pairs internally.

Let $Q_{\mathrm{SPEC}}(\pi)$ denote the total SPEC notional moved along a path π, counting all internal SPEC legs. The associated fee drag from the transfer tax is:

$$F_{SPEC}(\pi) = 0.035 \cdot Q_{SPEC}(\pi)$$

If these transfers are symmetrically reversed within a short horizon (for example, entering and exiting the SPEC hub in a single composed transaction), the net SPEC position for the user is unchanged, but the fee drag manifests as an effective additional spread on the SPEC legs. When quoting prices, Specter can absorb this by inflating the effective half-spread on SPEC pools by an amount proportional to 3.5% of the side actually used.

Execution uncertainty can be summarized by a variance proxy for the realized execution price along a path:

$$Var[P_{exec}(\pi)] \approx \sigma^2 \cdot \Delta t(\pi) + \kappa \cdot I(\pi, Q)^2$$

Here σ^2 is the underlying asset variance per unit time, $\Delta t(\pi)$ is the latency window as in Section 2, $I(\pi, Q)$ is the impact term, and κ is a pool-specific curvature constant. While Specter does not expose this formula directly to users, it informs how risk can be visualized on the interface, for example by shading routes by estimated variance.

9 Conclusion

Specter is intentionally narrow in scope: it does not introduce a new consensus mechanism, a new L1, or a new AMM primitive. Instead, it constructs a mathematically explicit view of dark-pool-style routing around a single hub token on Solana. The liquidity graph, cost functional, fee rules and privacy-score heuristic defined in this document fully describe the surface that the Specter interface presents to users.

Any execution client capable of sampling candidate paths on the Solana DEX graph, evaluating the functions $C(\pi, Q)$ and $\Omega(\pi)$ for those paths, and enforcing the fixed fee rates of 3.5% on $SPEC transfers and 0.25% on routed swaps can legitimately be described as an implementation of Specter. The underlying machinery may consist of direct calls to proprietary AMMs, integration with existing Solana-wide routers, or a combination thereof; the mathematics presented here is deliberately agnostic to those engineering choices.

By separating the narrative layer (dark pools, proprietary AMMs, and privacy) from the execution substrate, Specter aims to provide a clean target for both front-end design and future protocol extensions while remaining compatible with the rapidly evolving Solana DeFi stack.

References

[1] Lostin. "Solana's Proprietary AMM Revolution." *Helius Blog*, 2025.

[2] OKX Learn. "What is HumidiFi (HFI)? Solana's Dark Pool DEX Explained." *OKX*, 2024.

[3] Emma Rose. "HumidiFi: A Deep Dive into Solana's Leading Dark Pool Protocol." *Markets.com*, 2025.

[4] Arcium Labs. "Redefining DeFi with On-Chain Dark Pools." *Arcium*, 2024.

[5] HumidiFi. "HumidiFi Litepaper." *Jupiter / HumidiFi Docs*, 2024.

[6] David Liang. "Proprietary or Dark Automated Market Makers on Solana." *Figment*, 2025.

[7] Jupiter Aggregator. "Metis v1.6 Router Documentation." *Jupiter Docs*, 2025.

[8] Meteora Finance. "Dynamic Liquidity Markets (DLMM) and DAMM v2 Documentation." *Meteora Docs*, 2025.