Motivating a Resource and Inclusion Fee on Solana

Cavey Cool

August 4, 2026

Summary

SIMD-0553 proposes to transform Solana's base fee into a flat fee plus a fee which scales with the amount of resources a transaction requests. In this document, we delineate how a resource and inclusion fee would accelerate the realization of Solana's vision as the world's fastest global state machine and as the house of all finance. We begin by summarizing the current state of financial activity on Solana, highlighting the greatest growth engines on Solana. We turn to the scale of the opportunity ahead: tokenized equities, perpetual futures and other derivatives, credit, insurance and prediction markets, all of which remain relatively nascent or are absent from the network entirely, and all of which individually have a TAM that eclipse the current developed markets on Solana. We argue that the conditions for Solana to succeed on these fronts are simple — create a platform where users want to trade, makers want to make, and builders want to build — and that a resource fee helps foster the best version of this environment with a flywheel that benefits tokenholders. We discuss the specific implementation details of the resource fee in the SIMD-0553, including how we arrived at the parameters proposed, and what impacts our specific proposals will have. We speculate on the second order effects of the proposal. Finally, we enumerate the various counterarguments against the resource and inclusion fee and make an earnest attempt at steelmanning them for the reader to reference.

Resource Fee Incentive Structure

We will work to convince you, the reader, that with an inclusion fee and burned resource fee in place:

- **Core Devs** are incentivized to maximize resource consumption capacity by optimizing the Solana clients to be able to process as many transactions that consume as many resources as possible.
- **Validators** are incentivized to maximize resource consumption capacity by upgrading to the best hardware and including as many transactions as possible.
- **App Builders** are incentivized to optimize their smart contracts to retain as much of the user fee as possible, leaving more room for other revenue-generating activities on the network. Additionally, as opposed to a fee on volume, they retain all upside from their application and are maximally incentivized to grow their business and keep their revenue flowing onchain versus e.g. a rollup.
- **Toxic Takers** are forced to increase their minimum edge slightly since their per-transaction cost is higher [1], which reduces their toxicity to market makers on Solana.
- **Market Makers** in both spot and derivatives markets have their annual operating costs reduced on both CLOBs and prop AMMs, and (thanks to the aforementioned lower toxicity) can quote slightly tighter.
- **Uninformed Crank Searchers**, who presently pay very low fees while consuming many resources, are now required to pass on some of the value to SOL in exchange for the resources they used in generating the value.
- **SOL Holders** are no longer guaranteed to be perpetually diluted as the tokenomics become more sound, and possibly ultrasound in the most bullish case for Solana.

[1] since swaps consume more resources than quote updates

1 The Landscape of Solana

We will limit our focus to the two largest drivers of revenue on Solana for now, which are memecoins and SOL-stablecoin markets.

1.1 Solana is a Memecoin Chain

On August 3, 2026, the last completed day before this writing, Solana saw $1.54B USD in total volume across all DEXes, with memes making up 34% of that volume. Although the meme-related traffic ebbs and flows, memes have always been a significant portion of onchain volume and app revenue. If you filter out SOL-stablecoin traffic (we will get to this in the next section) to see what else is happening onchain other than people buying and selling SOL[2], you will see that the primary reason users have been on Solana for the past few years is to trade memes (Figure 1). According to DefiLlama, eight of the top ten protocols on Solana by app revenue (last 30D) are protocols that service the meme sector.

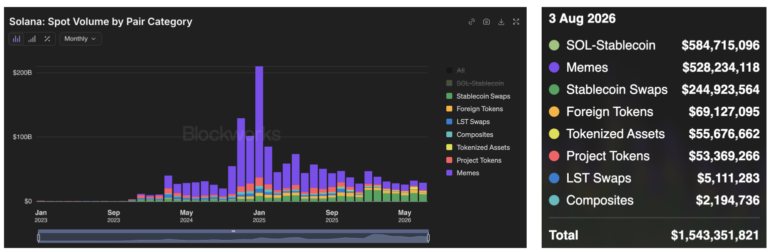


Figure 1: Left: monthly spot volume by pair category (SOL-stablecoin excluded). Right: Solana spot DEX volume by category on August 3, 2026. Source: `https://blockworks.com/analytics/solana/solana-spot-dexs`.

So far, billions of dollars have flowed to these protocols and intermediaries. Entire industries and livelihoods have sprung up from memes on Solana, including but not limited to launchpads, terminals, transaction landing services, DEXes, routers, and data providers.

Analyzing the meme sector begs a few interesting questions. How much of this value flows to SOL? What does flowing to SOL mean (to validators? stakers? burn?)? How much of this value ***should*** flow to SOL? What role does Solana play in these markets? *Let's explore these questions,* starting with the first two.

In the case of PumpFun, the largest protocol in the sector generating about 25% of REV on Solana, about 12% of this value flows to validators[3] in the form of fees and tips. A fraction thereof goes to stakers depending on individual validator commission parameters. Very little of it goes to SOL, the token. In fact, across **ALL** activity on Solana, about 3000 SOL is burned per week[4], and more than half of that is from votes. This is in stark contrast to the days before the activation of

[2] And swapping between stablecoins, which is not interesting in my opinion as non-fungible dollars are a problem created by crypto

[3] Blockworks: `https://x.com/0xcarlosg/status/2083053728894947721`.

[4] https://dune.com/21sharesresearch/solana-key-metrics

SIMD-0096 when 17700 SOL was burned daily [5]. Yet, across the entire chain, on any given recent day, 34.2% of the network's consumed compute units are from PumpFun programs [6].

This brings us to the next two questions: How much value ***should*** flow to SOL? What role does Solana play in these markets?

Solana is not a memecoin exchange. Solana is the world's fastest global state machine — an operating system that enables things like exchanges to exist and compete with one another. In this role, Solana promises builders the reliable storage and availability of state, and access to compute. In other words, it promises access to computational resources. That is Solana's sole job[7]. Yet, Solana sells these resources at zero cost. A zero-priority transaction that does **literally nothing** costs a user just as much as a zero-priority transaction that consumes 1400000 compute units (185M CPU cycles at 4 GHz under the 33 ns/CU that Solana originally targeted) and loads 64 MB of account data. An entire day's worth of resources on Solana is presently sold at the lowest price of 72 SOL[8].

Not only is this failure to price resources the epitome of bad business economics (and a nuisance-level DoS vector), it is a total failure to establish incentives to align participants to realize our vision for Solana.

- **Core Devs**, who are paid in SOL, gain very little from doubling the network's resource consumption capacity. And even if they do, they have no guarantee that the onchain systems will be efficient, undermining their work and leading to a slow Solana.
- **Validators**, who are paid in SOL, gain little from upgrading their hardware to help Solana sell even more of **their** resources at zero cost.
- **App Builders** gain very little from optimizing their smart contracts to use a smaller amount of state and compute.
- **Uninformed Crank Searchers** pay no additional SOL for having bloated and inefficient onchain programs to search, for write locking and serializing transaction processing on Solana, or for being toxic to market makers. (We will elaborate on this point in the next subsection, but is important to mention here)
- **SOL Holders** have no exposure to the use of computational resources on Solana.

This, in my opinion, is enough to motivate the resource fee. But before proceeding in that direction, let's look beyond the meme markets at SOL-stablecoin pairs, which make up the majority of the volume on Solana and have been a resounding success, to motivate it further.

1.2 Spot Markets on Solana

This is an interesting market and it is worth zooming out a bit before discussing the economics. The market makers operating here are highly sophisticated firms that posses tremendous experience in low-latency trading, globally distributed infrastructure and specialized networking—including

[5] `https://x.com/0xcarlosg/status/1892181943069897029`.

[6] https://dune.com/queries/8224366

[7] Note that "promising access" encompasses censorship resistance. Arguably, reasonable sequencing rules nd auction durations are also encompassed by this.

[8] 216000 blocks full (100M CUs) of max cost (≈1.5M CU) zero-priority transactions.

private fiber and microwave links—and advanced quantitative methods for pricing assets. These are the types of market makers (those which competitively quoted sub-bps spreads on crypto majors on globally distributed exchanges atop Solana) that we want to be attracted to trading on Solana, especially now as we are onboarding tokenized stocks and real world assets to Solana.

How can we retain these market makers? What pain points do market makers face on Solana? The principal pain points are fees, toxic takers, and scheduling. We shall leave the discussion for how to deal completely with the final point here for another SIMD, although we will see it come up.

1.2.1 Fees

First, fees. On exchanges with meaningful volume (NASDAQ, HyperLiquid, the full suite of CEXes) market makers are **NOT** charged for updating quotes. They are charged on fills. Exchanges on Solana, such as the Phoenix CLOB, operate this way as well, taking 0.5-5 bps per fill depending on the market. However, because these exchanges are on Solana, makers are subject to the L1 fees for quote updates. Even though makers are currently incentivized to optimize their program for scheduling priority, they receive no reduction in their fee for using less resources. The toxic takers consuming 200x more resources pay the same base fee as these makers.

And these fees are not small. Just one transaction per market per slot costs a maker approximately 394 SOL per year, or almost $30000 at current prices, **BEFORE** accounting for toxic takers or priority fees. We leave it as a fun exercise to the reader to calculate how much it would cost a single market maker to quote all 500 constituents of the S&P500 if SOL were to go to $200 again[9].

This is what motivates transforming the 5000 lamport base fee into a 2-part fee: keeping the static inclusion fee (2500 lamports still, although no longer per signature, that go to the validator) and a dynamic burn component based on resources. The resource fee accomplishes two things here. In addition to the aforementioned incentive for app developers to be more resource efficient, we also now **LOWER** fees for market makers while significantly raising the average burn rate.

[9] $42M. Now try 2 updates per slot!

1.2.2 Toxic Takers

The SOL-stablecoin markets on Solana are cutthroat, and atomic arbitrage between markets is often toxic[10]. Searchers are clever, relentless, and look for any opportunity they can to make pennies (Figure 2).

Figure 2: Although this is not SOL-USDC, this crafty arbitrage transaction demonstrates a searcher working around the path-dependent curve of a prop AMM to arbitrage PUMP four times within a single transaction, extracting cents of edge.

Solana's current architecture, which enables online searching for atomic arbitrage of arbitrarily small size across many venues, paired with a lack of a fee on resources consumed, incentivizes searchers to spam compute intensive transactions that write lock and load many accounts. Arbitrageurs are often said to make markets more efficient by closing price discrepancies between venues. Yet, Solana (the network of validators, and SOL, the token), which enables these markets to exist in the first place, gets very little in this exchange despite providing these computational resources as a service (Figures 3 and 4). A resource fee allows the network to charge these searchers more appropriately for the write locks they take, the accounts they load, and the compute they consume, incentivizing more informed and reactive searching over blind searching.

[10]Sometimes, an atomic arbitrage helps a market maker balance their inventory again

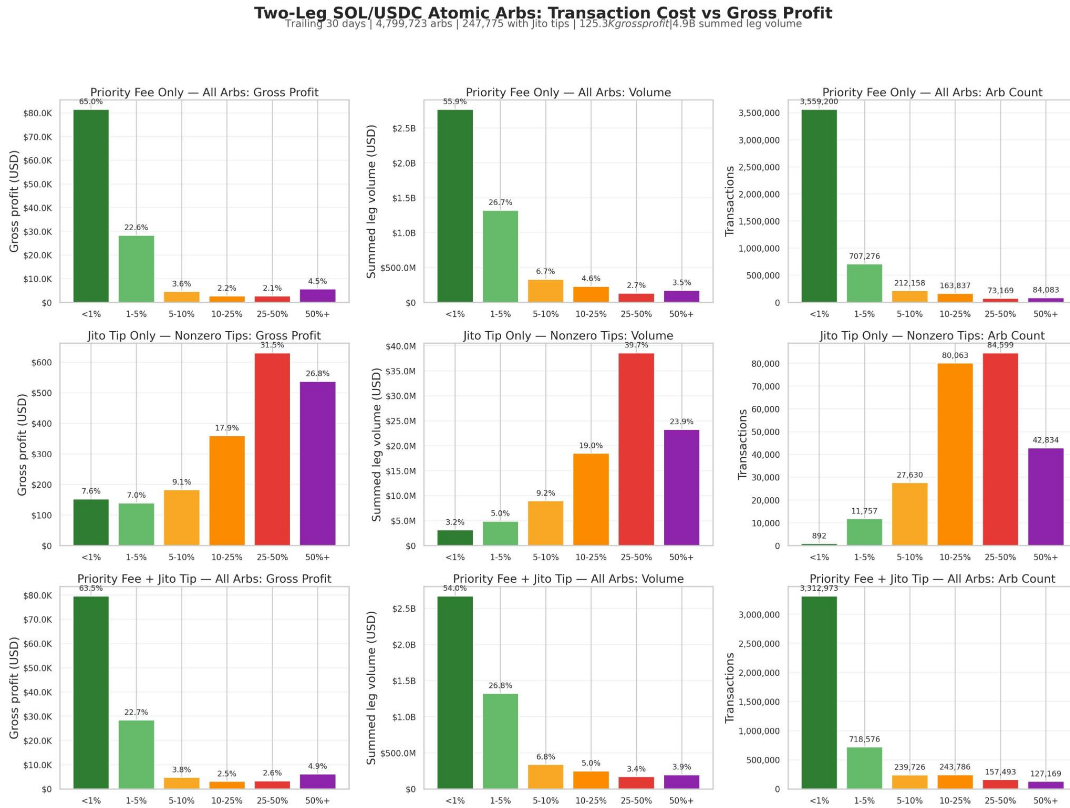


Figure 3: Two-leg SOL/USDC atomic arbs over the trailing 30 days. Row 1: most profit and volume occurs where transaction cost is under 1% of gross profit. Row 2: When the searchers are forced to bid for opportunity against each other, e.g. via the Jito auction, more value is returned to stakers. Row 3: Most arbitrage is outside of such auction mechanisms, with a decent fraction of arbitrage coming from crank searchers.

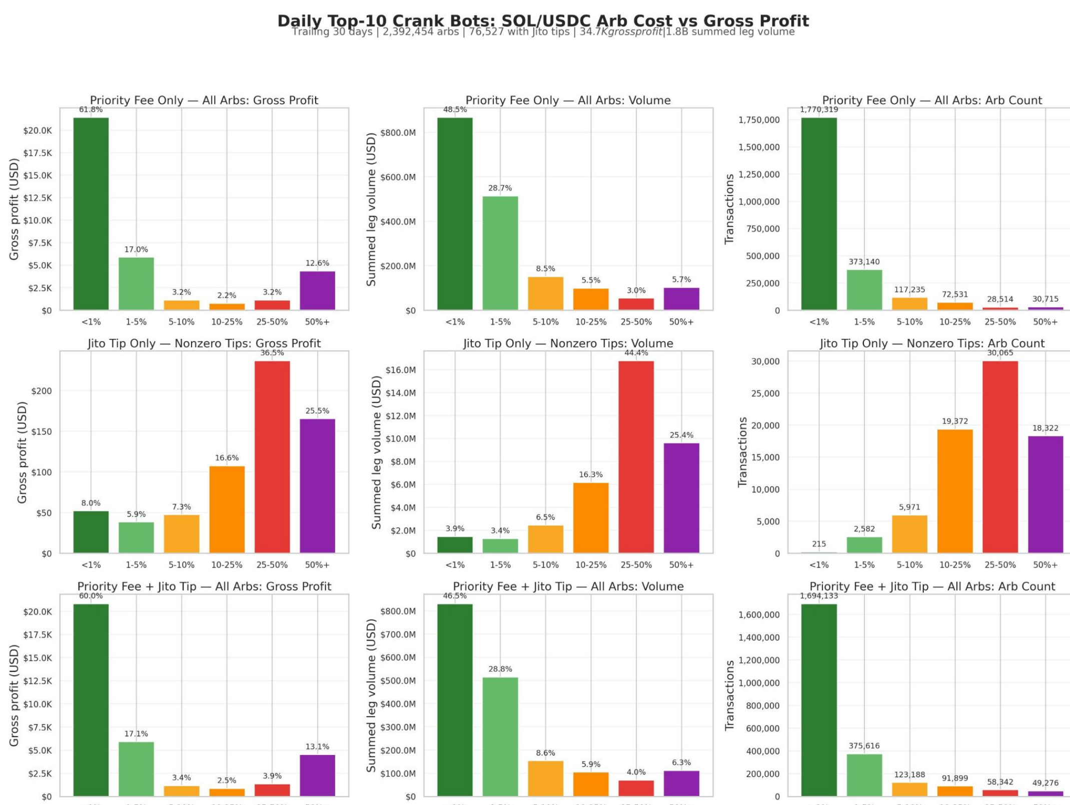


Figure 4: Daily top-10 crank bots on SOL/USDC over the trailing 30 days: cost vs gross profit. Same binning as Figure 3. Crank searchers account for a large share of low-fee arb volume. Just under 1/3 of SOL-USDC volume, and about 25% of profit comes from these crank searchers.

2 Eyes on the Prize

Memes may currently be a significant source of revenue for Solana and its stakeholders, and memes may very well stay around forever, but memetic finance is utterly dwarfed by the rest of finance. According to Dune, the total cumulative volume of all DEXes on Solana since genesis across all venues and assets is $4.3T[11] (and this is double counting across multi-leg hops). On the other hand, on the 26th of June in the year 2026, US equities traded $2.0T in notional volume over 158M trades across all exchanges[12]. The top exchange, the NASDAQ, did $543.8B notional volume on that day.

This is an unbelievably large difference in scale. What would it take for Solana to be able to support such scale? With a budget of 100M cost units per block, 158M trades over 6.5 hours (6752 swaps per second) requires the average swap to take 37025 cost units, assuming nothing else happens on the network (including failing swaps and market makers updating quotes). The largest router, Jupiter, has an average of 51249 compute units per **successful** invocation, and a success rate of 30.5% [13]. Note that compute units are but a subset of cost units, and there are additional cost units that come from signatures, account data loaded, instruction data size, and write locks. Note also, that this does not include other use cases like stablecoin transfers, derivatives, prediction markets, credit markets, memes, and does not account for the bursty nature of trading.

It goes without saying we are quite a ways away from being able to support such scale. Yet, this is not actually otherworldly and quite attainable with a fair bit of work. To be able to **support** this traffic we need a two pronged approach: optimizing the Solana clients and optimizing the onchain systems. To ensure this traffic is **encouraged** to come onboard, we will have other needs we shall discuss when architecting the resource fee in the next section.

2.1 Maximum Performance

2.1.1 Solana Clients

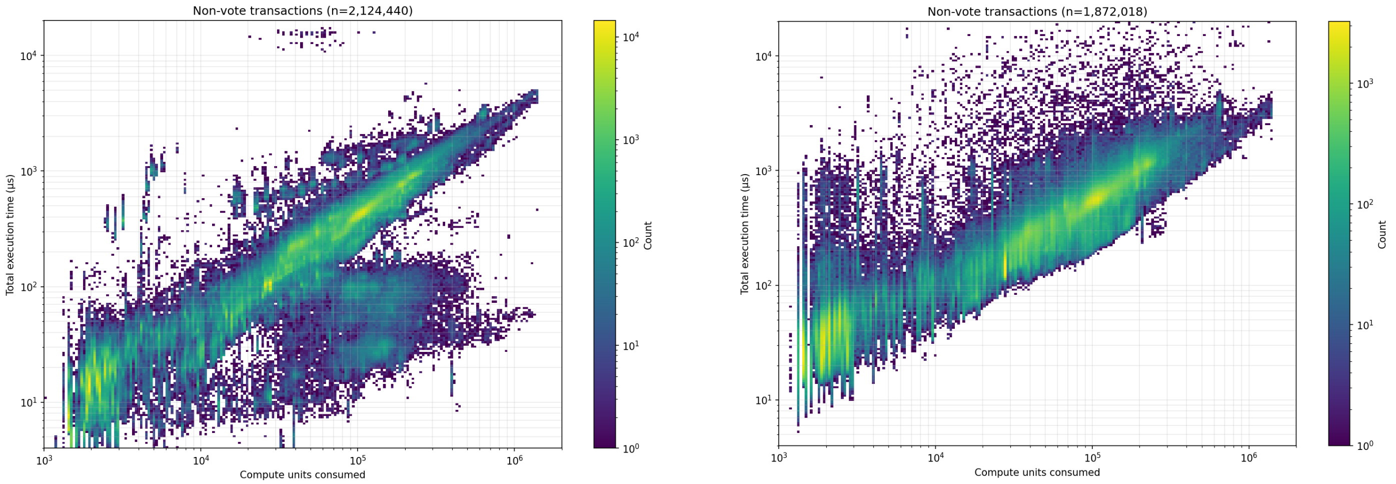
We need the Solana clients to be optimized. Agave has done an incredible amount of work over the course of the last two years to significantly improve performance and we have seen it go from struggling to support 48M cost unit blocks to comfortably supporting 100M cost unit blocks. Firedancer, which had its first mainnet release today (woohoo!), has shown that we can go even further with the runtime (Figure 5 and Table 1). Similarly, Mithril recently discovered a technique that enables 1M ed25519 signature verifications per second on eight Zen5 cores instead of 20[14], which **significantly** reduces the CPU load during the bursts we aim to handle. Having multiple client teams has been a net positive for Solana, as it has led to a clean up of the protocol and multiple discoveries of performance optimizations that improve all clients. Nevertheless, no client has yet reached what could be considered an optimal implementation capable of supporting the world's financial traffic, and we need to support and incentivize further work.

[11] `https://dune.com/queries/8226878/`

[12] `https://cdn.cboe.com/resources/us/equities/market-statistics/historical-market-volume/market_history_2026.csv`

[13] https://dune.com/queries/8224366

[14] https://github.com/Overclock-Validator/narya-ed25519



Firedancer

Frankendancer (Agave)

Figure 5: Runtime cost distributions for Firedancer (left) and Agave via Frankendancer (right) for a large representative sample of mainnet traffic. Firedancer has shown a significantly lower mean and variance of execution times for the same transactions.

	n	$\sum$ ns/ $\sum$ CU	mean(ns/CU)	median(ns/CU)
fd	2,128,034	3.97	5.54	4.40
fdctl (agave)	2,466,233	5.87	12.51	8.28
fdctl (agave) / fd		1.48×	2.26×	1.88×

Table 1: Firedancer vs. Frankendancer (Agave) execution cost in ns/CU. Firedancer is a 1.48–2.26× speedup over Agave depending on the aggregation.

2.1.2 Onchain Programs

Onchain programs are in a similar place as the clients. There has been lots of progress since genesis, yet we are far from what is needed to support all of the world's finance. With the latest p-token upgrade, two token transfers (the most raw form of a swap) costs on the order of 2000 compute units after accounting for the overhead and costs of a cross-program invocation. Anything beyond this is the (necessary) overhead of the DEX itself. A HumidiFi swap alone consumes on the order of 43000 cost units, a bit above what we calculated is required to support a record NASDAQ day today. Great, but not good enough. Other DEXes fare far worse, with several popular ones coming in at over 100000 cost units per swap, and the overhead of splitting a single swap across venues multiplies the issue. **Onchain programs are by far Solana's weakest link when it comes to performance, and an incentive is needed to push the ecosystem to do better.**

Alas, there is yet hope. Recently, Archer Finance launched on mainnet with mind-blowing performance. Take a look at the arbitrage in Figure 6. To be perfectly clear: this is not 60366 compute units for a swap. This is 60366 compute units for an arbitrage program (additional overhead) invoking **two** swaps via Archer. The first swap took 19309 compute units, while the second took 18460. These costs are far below what is required to support the record equity trading day we discussed before. And yet, this program is still far from optimal.

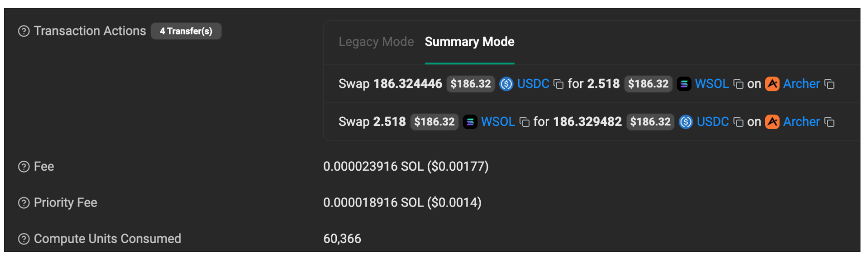


Figure 6: An Archer two-leg SOL/USDC arbitrage for 0.5 cents of edge consuming only 60366 compute units—far leaner than typical DEX swap footprints on Solana today. Source: `https://solscan.io/tx/4He3x3bTw23m3V3YJ5pLt7uLPgnX5AVPRhuqb8V7cU2HLiKXXvRtcvQcE6NZQ8zesiRiYo5Z7ZpoBFM4Ai5tCK6q`.

As written today, SIMD-0553 will reduce costs for makers on all propAMMs, Archer, and Phoenix Eternal by as much as 22%. Additionally, SIMD-0553 would provide a slight incentive to users to use efficient DEXes like HumidiFi and Archer as the resource fees would be lower. Solana needs more DEXes like Archer and a resource fee will help incentivize developers to push Solana forward.

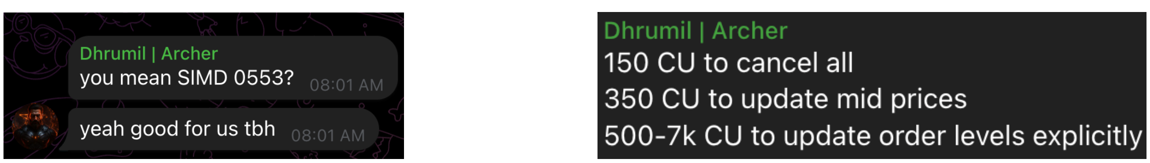


Figure 7: Left: Archer CEO's reaction to SIMD-0553. Right: Archer CEO reporting quote-update and cancel costs in compute units—orders of magnitude below previous central limit order books on Solana.

3 The Architecture of the Inclusion and Resource Fee

We have found motivation for the transformation of the base fee on Solana into a two-part inclusion and resource fee. As primary objectives, and in order of importance, we would like to use the resource fee to

1. Add an incentive for builders to reduce the resource footprint of their applications. **This is indubitably a blocker to bringing all of finance onchain and a large enough reason for the resource fee on its own.**
2. Slightly raise costs for inefficient searchers, to encourage more informed and reactive searching instead of blind spam, and to reduce toxicity on market makers.
3. Slightly lower market maker operating costs.
4. Prevent the perpetual dilution of SOL holders by improving SOL tokenomics.

Importantly, in reference to primary objective #1, introducing a resource fee to fix the incentive structure for builders on Solana does not involve a tax on the revenue of applications. Anyone that wants to build on Solana gets to keep their upside beyond fixed resource costs, without introducing incentives for offchain payments or state rollups. We also identify that the resource fee supports these secondary objectives:

1. Incentivize validators to maximize the resource consumption capacity of the network.
2. Incentivize core devs to maximize the resource consumption capacity of the network.

Why burn the resource fee? If we give any portion of the resource fee directly to validators or their stakers, this changes the incentive structure for blockbuilding. Validators suddenly become incentivized to accept less efficient transactions. The efficacy of prop AMMs, whose grand success is in large part due to the efficiency asymmetry between oracle updates versus swaps [15], suddenly dwindles as validators are incentivized to prioritize toxic swaps over oracle updates. Burning the resource fee in large part keeps the market structure that we know and love intact. Burning the resource fee is a simpler and less disruptive change. Furthermore, a resource fee aims to charge for the cluster's resources, of which the validator is a small part. When a crank arbitrageur lands a 600k CU series of swaps, every validator has to verify the signature(s), parse the accounts, load the accounts, and execute the transaction.

Why does the inclusion fee not scale with signatures? With the resource fee, the user is already being charged for signatures at 720 cost units. Signature verification is just another use of the cluster's computational resources and should never have had a special carveout in the fee calculation to begin with. Why have an inclusion fee at all? For four reasons which I will list in order of importance. Firstly, we must not disrupt validator income too much. Secondly, we must not lower the minimum transaction cost too much to prevent a DoS vector. Unfortunately, today's cost model does not account for the cost of broadcasting the transaction to the network; the inclusion fee could very well be called the broadcast fee if we wish to call it that. Thirdly, with zero priority and zero inclusion fee, a validator would get exactly zero income from including that transaction. Fourthly (which is least important but worth mentioning), all drafts put forth thus far of Multiple

[15] which would still exist even if swaps were perfectly optimal

Concurrent Proposers have an inclusion fee paid directly to the proposer. This inclusion fee is therefore compatible with our current needs and any future MCP design that may or may not come our way.

3.1 SIMD-0553 Fee Parameters

Now that we've established the motivation and objectives for the inclusion and resource fee, we need to discuss additional requirements for the parameters set in SIMD-0553. The two parameters to set are the inclusion fee and the resource rate fee (in lamports per cost unit), which the proposal sets at 2500 and 1/2 respectively, with 1/2 serving as the terminal rate.

There are several ways we can try to attack the problem of setting these parameters, and there are many things to consider. For example, we can try to target a specific burn rate. We can try to cap the largest possible fee. We would like to set a value that does not increase fees for users drastically and abruptly. Note that the priority fees paid on the network remain an order of magnitude larger than the base fee paid, so we have a bit of wiggle room. If this feature gate is activated prior to Alpenglow, when votes still land as onchain transactions, the values of these parameters significantly affect validator operating costs and we must take vote costs into account. If this feature gate is activated after Alpenglow, we need not worry as much except for affecting validator income. Additionally, we must consider second order and time-dependent effects. Ramping up from a lower value to the terminal value gives application developers time to optimize their onchain programs before reaching the terminal value. Additionally, note that low priority transactions have little incentive to request an appropriate amount of resources and that charging for requested resources will likely change that behavior.

Given that most transactions have one signature (about 96.5%), we cannot change the inclusion fee much without significantly affecting validator rewards. The average number of signatures per transaction in the last 24 hours is approximately 1.04, so we are well motivated to keep the inclusion fee at 2500.

Let us first consider the smallest and largest possible transaction costs (1020 and 1573166 cost units, respectively) as well as what the current distribution of requested resources is across a representative sample on Solana Mainnet (Figure 8). Having fixed the inclusion fee, in order to satisfy the requirement that the most efficient transactions see a decrease in cost, we can motivate an upper bound for the resource fee rate at approximately 2.5 lamports per cost unit. With a minimum transaction cost of 1020, the lowest possible fee becomes 5050 lamports. A value of 2.5 lamports per cost unit would cause the median transaction to pay a resource fee of approximately 250000 lamports and the 90th percentile transaction to pay just over one million lamports. The current median is between 5000-100000 lamports, although of course this is very dependent on the current state of the market. Note that this upper bound is motivated by a no-op transaction, and therefore still results in an increase in fees to market makers.

Attempting to match the burn rate to the terminal inflation rate (using the current supply and current levels of traffic), we derive a value for the resource fee rate of about 1.5 lamports per cost unit, motivating a reduced upper bound.

Attempting to give current market makers an approximate 20% discount to their non-priority fees, we can reduce the upper bound further to about 0.5. This leaves Solana inflationary with current levels of traffic, but gives an opportunity to match the burn rate if the blockchain sees further adoption. At this value, validators also see a discount in their operating costs pre-Alpenglow. Tables 2 and 3 show what fees would look like at this terminal rate as well as at the ramp-up rates

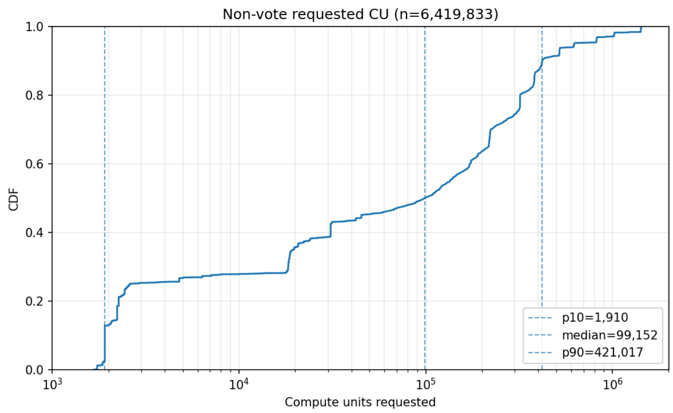


Figure 8: CDF of non-vote requested compute units ($n = 6419833$). Median request is 99152 CUs; p10 and p90 are 1910 and 421017 CUs respectively.

proposed in SIMD-0553 (1/10, 1/4, then 1/2). **At this value, the most computationally intensive transaction that one could craft would cost 0.000789083 SOL, or just over 5 cents USD at current SOL prices (just over 21 cents at all-time highs).** Presently the p90 median priority fee is 17809 lamports, so the p90 resource fee would be over ten times higher. We remind the reader again that this price reflects the terminal rate with requested resources today — before application optimizations, appropriate resource requests, and other second order effects.

Percentile	Cost units	Today	1/10	1/4	1/2
Theoretical min	1020	5000	2602	2755	3010
p10	1910	5000	2691	2978	3455
Median	99152	5000	12416	27288	52076
p90	421017	5000	44602	107755	213009
Theoretical max	1573166	5000	159817	395792	789083

Table 2: Base fees in lamports implied by the non-vote requested-CU distribution in Figure 8, assuming a zero-priority one-signature transaction. Columns compare today's flat 5000-lamport base fee to SIMD-0553 at each resource-fee ramp rate (inclusion fee 2500 + resource fee).

It is important to note that if a user does not specify a limit for the amount of account data loaded, the runtime defaults to a limit of 64 MB, charging the user 16384 cost units. Partitioning the last five full days of mainnet traffic by whether compute and loaded-data budgets were set explicitly or left at defaults, Table 4 shows that users who set both limits request far fewer cost units—and would pay far less under SIMD-0553—than users who leave the defaults in place.

[16]Data from Umberto: `https://dune.com/queries/8228815/12223000`.

Activity	Today	1/10	1/4	1/2	Δ at 1/2
Vote (w/ compute budget)	5000	2877	3442	4383	−12.3%
Zerofi oracle update	7568	5312	5678	6288	−16.9%
DFlow swap (high priority)	2007500	2044544	2103860	2202720	+9.7%
OKX swap (mid priority)	135980	215912	339560	545640	+301%
Pump.fun swap (zero priority)	5000	34482	82455	162410	+3150%

Table 3: Total fees in lamports for example mainnet transactions under today's flat 5000-lamport base fee versus SIMD-0553 at each resource-fee ramp rate (inclusion fee 2500 + resource fee + unchanged priority fee). Low-resource traffic (such as votes, oracle updates) gets cheaper at the terminal 1/2 rate; compute-heavy zero-priority traffic pays more.

Cohort	n	p10 CU	med. CU	p90 CU	p10 fee	med. fee	p90 fee
Explicit CU + loaded	316636923	1957	79091	337629	3479	42046	171315
Explicit CU, default loaded	300602530	18705	185288	471112	11853	95144	238056
Default CU + loaded	77329137	21167	601727	1421155	13084	303364	713078
Default CU, explicit loaded	1	7356	7356	7356	6178	6178	6178

Table 4: Requested cost units and implied SIMD-0553 base fees (lamports) at the terminal 1/2 rate for four budget-setting cohorts over five full mainnet days. Fee = $2500 + \lceil \mathrm{CU}/2 \rceil$ (zero priority). Leaving the 64 MB loaded-data default roughly doubles median cost versus setting both budgets explicitly.[16]

4 The Man of Steel

This section contains the top three counterarguments we've heard for this SIMD. They are here, steelmanned, for the reader's reference. We provide a bit of our own commentary, but leave evaluating the merit of the counterarguments as an exercise to the reader.

4.1 Slight Reduction in Validator Income

As written, SIMD-0553 proposes a flat inclusion fee that goes to the proposer. Unlike the current base fee, this fee does not scale with the number of signatures. The cost model already assigns 720 cost units per top-level ed25519 signature that must be verified from the transaction. As such, the resource fee is already charging the user per signature. Table 5 quantifies how many additional signatures beyond the first appear in recent mainnet traffic. These are the additional signatures for which transactions would no longer pay an incremental inclusion fee under the flat 2500-lamport design.

This income loss amounts to a total of approximately 35.56 SOL per day across all validators. A small validator with 200k SOL stake can expect to see a reduction in their income of about 0.01 SOL per day.

4.1.1 Commentary

We believe this income loss is not substantial.

num_sigs	txn_count	extra_per_txn	extra_sigs
2	9344781	1	9344781
3	610077	2	1220154
4	246001	3	738003
5	19226	4	76904
6	49647	5	248235
7	135211	6	811266
8	246329	7	1724303
9	7481	8	59848
10	228	9	2052
Total			14225546

Table 5: Multi-signature transactions over a recent 24h window. `extra_per_txn` = `num_sigs` − 1 is the number of signatures beyond the first; `extra_sigs` is the implied count of signatures that would have paid an incremental per-signature base fee today but do not under a flat inclusion fee.

4.2 Alternatives (Fees-on-Volume)

Charging for resources is not enough. If the exchanges on Solana succeed, it is because of the network effects of Solana and we should be able to monetize this. Resource usage and transaction value have little correlation, and we should have exposure to the revenue of the applications on Solana. This already has a precedent: Hyperliquid commands several bps on all volume, while Apple commands 30% of all in-app revenue and subscriptions.

4.2.1 Commentary

Although fees-on-volume is certainly another way to get revenue from users and introduce a burn, it does not have all of the other benefits of a resource fee and has a few other unique downsides in addition to being hard to enforce without rearchitecting Solana entirely.

Firstly, it has a different incentive structure for everyone involved. Valiators and core devs do not directly benefit from a faster blockchain. Businesses are taxed and have an incentive to try to move some of their activity offchain, whether it be traditional stablecoin rails or state rollups. Developers would not be incentivized to increase Solana's throughput by optimizing smart contracts as they would with a resource fee.

All proposals thus far have been fairly invasive such as an enshrined toll in the token program. This is not only extremely distruptive in a way that would break many DeFi applications, but is also easily circumventable with a community fork of the token program and is not compatible with encrypted/confidential transfers.

In my opinion, Hyperliquid and Apple are not akin to Solana, as they both have enshrined exchanges. If Solana wishes to take fees on volume, it must too enshrine an exchange (this is not a recommendation to do so).

4.3 Increased Transaction Costs and Complexity

SIMD-0553 adds yet another component to Solana's fee, making fees higher and more complex. A zero-priority Pump.fun swap jumps roughly 32× at the terminal rate; the median non-vote request goes from 5000 to about 52000 lamports; p90 exceeds 200000. Absolute lamports may still be fractions of a cent, but the proposal risks harming the activity it hopes to monetize and potentially pricing out use cases. Because the fee keys off *requested* rather than consumed cost, it also taxes legacy defaults and cautious over-estimates. In many cases, users have no way of known exactly how many resources they need and this may cause more user transactions to fail after attempting to lower their budget.

To elaborate on the complexity, the proposal replaces a two-part fee users can easily reason about with three components that mix compute units (priority fee) and cost units (resource fee). Additionally, core developers were in the midst of planning to remove CUs from the protocol entirely, as metering is both inaccurate and expensive. This proposal holds back Solana from becoming a simpler and faster protocol.

4.3.1 Commentary

The number one way to improve capacity on Solana right now is have optimal onchain programs. This would improve throughput much more than removing metering. But, to get onchain programs, one must incentivize the developers to optimize them. The resource fee accomplishes that and therefore easily justifies making fee slightly more complex. Additionally, the resource fee does not add any more metering or make it any more complex. It simply uses values that are already present in the protocol.

5 Conclusion

Solana's job is to sell computational resources. Today, those computational resources are given away for free. The diffuse benefit core devs and validators capture from increased network activity is not proportional to the increase in resource consumption. App developers aren't incentivized to optimize their programs, and SOL holders have no direct exposure to the network's actual business.

The inclusion fee and resource fee corrects this without imposing a punitive tax on app developers or validators. At the terminal rate of 0.5 lamports per cost unit, the costliest transaction that can be constructed is about $0.05. Developers who have already gone to great lengths to optimize their programs, like market makers, will see transaction costs fall. The traffic which consumes the most and pays the least will see transactions cost rise, in accordance with the resource consumption they exert upon the network.

A record day on the NASDAQ is $543,000,000,000 in volume. Solana could not support that today. Closing the gap requires faster clients, lower quoting costs, and more efficient onchain programs. We believe that SIMD-0553, as proposed, correctly aligns the incentives of all network participants and is a meaningful step towards improving the robustness of SOL's tokenomics.

Acknowledgements

I would like to thank Frank for reviewing my work, and Umberto and Blockworks for the data and analysis that informed several of the figures and tables in this document.