# Part 1: How We Got Here

Ethereum was initially designed with the idea that a single party would handle the entire process of block creation. This entailed aggregating transactions from the mempool, crafting the block header, and either finding the golden nonce in proof of work or simply signing the block header in proof of stake. For its initial years, block building was straightforward: mining nodes pulled transactions from their mempool, ordered them based on gas price, which signifies the computational work of each transaction, and stayed within the gas limit per block. However, with the rise of decentralized finance (DeFi), this approach to block building has undergone significant changes.

## The Centralizing Threat of MEV

In DeFi, the sequence in which transactions are ordered can make a big difference. Let's say there's a transaction waiting in the mempool that aims to swap 1 ETH for BITCOIN (I'm talking about HarryPotterObamaSonic10Inu, of course) on UniSwap. The amount of BITCOIN you'd receive is based on the current ETH to BITCOIN ratio in the UniSwap pool. If someone else’s transaction, swapping 2 ETH for BITCOIN, gets processed right before yours does, you'll end up with fewer BITCOIN because the ETH to BITCOIN ratio has changed. Given the significance of transaction order, and miners control this ordering, it led to the emergence of what we call MEV, or Miner/Maximal Extractable Value. MEV represents the potential profit a miner can make by choosing which transactions to include, exclude, or rearrange.

MEV might seem harmless at first. Heck, it could even appear as a booster for network security, more incentives for mining or validating, right? Besides the usual block rewards and transaction fees, now there's MEV up for grabs. But the reality is far from harmless. If left uncontrolled, MEV could become a potent centralizing force. Here's a story to illustrate this: Imagine you've just got wind of this MEV game and hear that validators are raking in more than 10% APR because of it. Tempting, right? You're all in! So, you send your 32 ETH to the staking contract and kick off a validator node. But wait a minute... You're only seeing a 4% return. When your turn to propose a block comes around, the transactions simply line up by their gas price. No MEV magic. You're not armed with the intricate algorithms and tactics to mine that MEV gold. Without the know-how, you're stuck with the default: ordering transactions by gas price.

Here's where the centralizing pull kicks in. Even if you're a quant, your simple computer, maybe a raspberry pi, doesn't hold a candle to their supercomputers running next level MEV extraction plays. The obvious end game here is to turn off your validator and instead send your ETH to these super powered setups that promise a share of the MEV pie. Fast forward, and you might see a handful of these behemoths essentially running the network, a truly unsettling centralized outcome. If this is where things head, then Ethereum's foundational goals have failed. A network dominated by a select few might as well be a centralized database at that point.

## The Birth of Flashbots

Phil Daian, a Ph.D. student at Cornell with a specialization in smart contract security, was among the early identifiers of the MEV issue. In August 2017, he published a blog titled "The Cost of Decentralization in 0x and EtherDelta."

This blog was inspired by the front-running vulnerabilities he identified during the 0x ICO.

Front-running involves spotting a transaction in the mempool that aims to swap Token A for Token B. A front-runner then programmatically initiates a similar transaction that offers a higher gas price. This strategy ensures that the front-runner's transaction is processed before the original one. After the initial transaction is processed, the front-runner can immediately trade Token B back for Token A, ending up with a larger quantity of Token A than they began with. This tactic is sometimes called a sandwich attack

, as the user's transaction gets sandwiched between two transactions initiated by the front-runner. As a result, while the front-runner gains profit, the individual behind the original transaction receives less of Token B. While sandwich attacks are common, there are various strategies individuals can use to extract MEV effectively.

Visualizing the Sandwich Attack: How Front-runners Profit at the Expense of Others

During the ICO boom, Phil and a team deployed a bot that earned about a million dollars annually. After sharing their methodology in the blog post I mentioned earlier, several similar bots emerged, creating a competitive landscape where bots would outbid each other's gas prices to achieve transaction priority. This prompted Phil to deploy nodes globally, capturing real time transactional data. This research culminated in his famous "Flash Boys 2.0" paper, which delved deep into the MEV challenges caused by decentralized exchanges.

Here's a fun related story Phil shared when he was a guest on theChopping Block. When Hayden Adams, the founder of UniSwap, shared his design for what's now the most popular decentralized exchange on ethresear.ch. Phil immediately messaged his concerns to both Vitalik and Hayden. Phil believed that UniSwap's design would cause a significant amount of MEV, making it a prime target for exploitation and putting users at risk of transaction reordering and sandwich attacks. Vitalik responded suggesting that these could just be looked at as an additional fee mechanism to use the blockchain. Phil was so pissed at this response and he thought powerful financial entities like Goldman Sachs would come in and eat retails lunch just like the current financial system. However, with time, Phil has come around to Vitalik's perspective (all praise lord Vitalik).

Recognizing the importance and challenges of the MEV space, Phil co-founded Flashbots, a company focused on research and solutions in the MEV arena. Flashbots realized that MEV is going to exist and Flashbots it's mission to make sure that the existence of MEV doesn't lead to a system where being a bad person or creating negative externalities is both better for you individually and is more profitable than being a good actor. An example of this is in TradFi, the most profitable strategies often involves exploiting the system's edges. Also Flashbots thought that there might be a way to harness the energy of MEV for the users and to subsidize people who secure the network and also to subsidize transactions on the network to get people better prices to give people the execution they want in these systems. If designed right, MEV could be part of what makes crypto outperform traditional systems.

## Harnessing MEV: The Role of Auctions and Proposer-Builder Separation

Flashbots recognized that miners' monopoly over transaction ordering was a valuable resource. Their first step to democratizing MEV involved creating an auction system for transaction ordering rights. This led to the creation of MEV GETH, which first introduced the concept of proposer-builder separation (PBS). Ethereum Foundation's Barnabé Monnot describes PBS as: "A design philosophy where protocol participants might use third-party services during their consensus duties." Up until this point, miners had complete control: they decided on the transaction order, did the hashing, and then added the block. But MEV GETH switched things up. It introduced outside actors, called searchers, who paid for the right to have their bundle of transactions included in the miners block.

With MEV GETH, miners had a new endpoint. They could receive transaction bundles optimized for MEV from searchers. Each bundle would also contain a transaction that provided miners a fee, incentivizing them to select this particular bundle. Naturally, miners chose the bundle offering the highest fee. When searchers compete for MEV opportunities in the public mempool, their bids naturally escalate due to this competition. This competition ensures that miners receive the lion's share of MEV benefits.

Let's break this down with an example: Imagine Alice is a searcher and spots an arbitrage opportunity between two decentralized exchanges. She could profit 0.07 ETH by buying Token X on UniSwap and then immediately selling it on SushiSwap for a higher price. So she creates a transaction bundle optimized for the 0.07 ETH MEV opportunity and is willing to pay a miner 0.05 ETH to prioritize her transactions in the next block. Bob, another searcher, identifies the same opportunity. He constructs a similar bundle but offers a 0.06 ETH payment to miners for the same privilege. Alice and Bob both send their MEV optimized transaction bundles to miners. On the other end, a miner receives these bundles and has to decide which one to include in the next block. Naturally, the miner chooses Bob's bundle due to the higher fee offered, ensuring the miner reaps maximum benefit. It's a win-win situation. The miner captures the majority of the MEV opportunity, receiving 0.06 ETH out of the 0.07 ETH opportunity. Meanwhile, the searcher secures a net profit of 0.01 ETH, which they otherwise wouldn't have been able to obtain. The essence of the MEV GETH mechanism is this competitive bidding. Alice and Bob compete against each other, offering incentives to the miner, thus ensuring the miner captures a significant portion of the MEV benefits.

However, if they simply opened up an endpoint for anyone to send the miners bundles, malicious actors might exploit this openness to overload their system, potentially launching a DOS attack. To address this vulnerability, Flashbots introduced the Flashbots Relay. This relay serves a crucial filtering role: it assesses incoming transaction bundles based on their potential profitability for miners, the validity of the transactions, and the offered fee. Only the optimal bundles are then forwarded to the miners. This method does introduce a level of centralization, as the process is dependent on the Flashbots Relay to screen out undesirable or potentially harmful traffic. Interestingly enough, a level of PBS already existed between the mining pool operator and their workers. Typically, the operator constructed the block body, including the bundles sent from the relays, hashed the block header once, and dispatched it to workers to continue hashing and find the golden nonce.

Overview of MEV GETH: The journey from search to transaction bundle inclusion in the miner's block

## Part Two: The Current Landscape

When Ethereum transitioned from Proof of Work (PoW) to Proof of Stake (PoS), the landscape of transaction validation and block proposal significantly changed. While PoW relied on miners and computational power (hash rate) to validate and add new blocks to the blockchain, PoS shifted this responsibility to validators who would stake

their ETH to become block proposers.

MEV GETH was being used by nearly all the mining pools, but with Ethereum transitioning to PoS, the system required modifications. PoS was designed to accommodate solo stakers: individual validators operating on low resource devices like a Raspberry Pi. PoS was designed with the goal of ensuring a balanced landscape: whether you're a solo staker or part of a substantial staking pool, there would be no inherent advantage in the validation process for any participant. Prior to the PoS transition, a few mining pools dominated the hash rate. This allowed for a trusted relationship between these pools and the Flashbots Relay. Any dishonest actions, such as a mining pool stealing MEV from a searcher, could jeopardize this relationship. Say a miner received a bundle with a sandwich attack from a searcher. If the miner further sandwiched the searcher with their own transactions, it would bring short term gain, but it would sever ties with Flashbots, costing them future MEV earnings because they would lose access to the Flashbots Relay.

### Introducing MEV Boost

Solo stakers, unlike large mining pools, might not have long term motivations to maintain trust. In certain scenarios, they could find it more profitable to exploit the MEV from a builder and subsequently disappear from the network. This action would result in them being fully slashed, losing all 32 ETH, but in some cases, the potential profit from stealing the MEV can outweigh this loss. This indeed occurred in April, when a rogue validator swiped $20M from a sandwich bot before shutting down their validator. Further reading on this incident.

In response to this new attack vector, Flashbots rolled out MEV Boost, a system designed for an environment with solo stakers.

The Mechanics of MEV Boost:

1. Relays:

Unlike the previous system where only Flashbots acted as the relay, MEV Boost democratizes this. Now, anyone can serve as a relay, broadening participation and security. Flashbots has also open-sourced their relayer code.

1. Builders:

A new role emerges - the Builder. These entities collect transaction bundles from searchers and combine them into complete blocks.

1. Auction System:

Builders make bids to include their full block and submit them to the relays. The relays perform a crucial verification step to ensure block validity.

1. Validator Interaction:

The relays forward the highest bid, along with the respective block header, they receive from the competing builders to the validator who’s turn it is to propose a block to the Ethereum network.

1. Block Commitment:

The designated validator signs the block header, which is a commitment. Once signed, they are bound to that block. If they attempt to sign a another block that would be seen as a malicious act and they would be slashed.

1. Final Proposal:

With a commitment in place, the relay sends the full block details to the validator, and it's formally proposal to the network.

Relays:

Unlike the previous system where only Flashbots acted as the relay, MEV Boost democratizes this. Now, anyone can serve as a relay, broadening participation and security. Flashbots has also open-sourced their relayer code.

Builders:

A new role emerges - the Builder. These entities collect transaction bundles from searchers and combine them into complete blocks.

Auction System:

Builders make bids to include their full block and submit them to the relays. The relays perform a crucial verification step to ensure block validity.

Validator Interaction:

The relays forward the highest bid, along with the respective block header, they receive from the competing builders to the validator who's turn it is to propose a block to the Ethereum network.

Block Commitment:

The designated validator signs the block header, which is a commitment. Once signed, they are bound to that block. If they attempt to sign a another block that would be seen as a malicious act and they would be slashed.

Final Proposal:

With a commitment in place, the relay sends the full block details to the validator, and it's formally proposal to the network.

The MEV Boost process

This setup introduces trust issues:

- Builder-Relay Trust:

Builders need to trust that relays won't steal their MEV. Consider a scenario where a relay, after receiving a block from a builder, swaps out the builder's address in a sandwich transaction for its own. Then they pass manipulated header on to the proposer.

- Proposer-Relay Trust:

Proposers, on the other hand, must trust that the block headers they sign are valid. Proposing an invalid block would result in the forfeiting the block rewards since the network would reject such a block.

Builder-Relay Trust:

Builders need to trust that relays won't steal their MEV. Consider a scenario where a relay, after receiving a block from a builder, swaps out the builder's address in a sandwich transaction for its own. Then they pass manipulated header on to the proposer.

Proposer-Relay Trust:

Proposers, on the other hand, must trust that the block headers they sign are valid. Proposing an invalid block would result in the forfeiting the block rewards since the network would reject such a block.

PBS designs see a recurring challenge: while interactions between the proposer and transaction ordering actors are a given, there's a clear need for a mechanism where:

1. Proposers can commit to a builder's block without knowing its contents but remain assured of the block's validity.
2. Builders can safely send their block to the proposer, confident that their MEV won't be stolen.

Proposers can commit to a builder's block without knowing its contents but remain assured of the block's validity.

Builders can safely send their block to the proposer, confident that their MEV won't be stolen.

MEV Boost trust assumptions

Before diving deeper into MEV Boost, it's essential to understand the default way Ethereum creates blocks without the use of MEV Boost. This setup hinges on the collaboration between a validators Execution Client and Consensus Client. When a transaction is received by the Execution Client, it checks the format, adds it to its mempool, but doesn't process it. Simultaneously, the Consensus Client handles the PoS consensus, picking a validator to create the next block. The selected validator's Execution Client then arranges transactions by gas price into a new block, which is then forwarded to the Consensus Client and put forth to the network. Other validators attest to the block's accuracy, and once verified, it becomes the chain's newest link.

This process changes if the validator opts to use MEV Boost. Validators integrating MEV Boost do so with their consensus client. When they're up to propose a block, they don't rely on their Execution Client anymore and instead connect to a network of relays. Validators can choose which relays to connect to.

MEV Boost is optional, but 95% of validators are utilizing it. Essentially almost every validator, except those run by Vitalik, are delegating block building to a third party. This delegation indicates that a core function of the Ethereum protocol, block building, is now primarily conducted outside of the Ethereum system itself. One key player in this setup is the relay and their role somewhat contrasts with Ethereum's foundational principles. Presently, there are around 9 active relays, but only 6 of them have >9% share of blocks relayed.

Breakdown of Top Relays and Builders by Market Share in the past 7d. Source: https://www.relayscan.io/

Trust becomes an issue since the relationship between the relays and the builder and the relay and the validator isn't trustless. There's also a concern about censorship resistance. Relays, during their auctions, have the discretion to determine the validity of blocks. This discretion allows them to exclude blocks with transactions linked to sanctioned addresses. A case in point is when the OFAC Tornado Cash sanctions happened some relays exercised this power. Recent data shows that 38% of blocks in the past week adhered to OFAC guidelines due to such relay imposed censorship.

## Part Three: Looking Ahead

Ethereum is devising a strategy to reincorporate the processes currently operating outside its core protocol. The goal is to mandate that proposers obtain blocks from builders, essentially letting the protocol handle the relay's current duties. The relay system as it stands has its vulnerabilities. For instance, a relay might not properly validate a block, misverify the builder's bid in relation to the payment intended for the proposer, or even delay or fail in block delivery. On top of this, running a relay doesn't come cheap. As of now, there's a lack of a sustainable funding model for them. The Ultrasound Relay, the most utilized relay, says its operating costs are estimated to be between 70k-80k euros annually, and that's excluding other expenses like software maintenance. Relays currently operate as public utilities.

It's also worth noting that since MEV Boost is external software developed by a company (Flashbots) it isn't as rigorously tested as software within the protocol. This was evident with the Prism client post the Shapella upgrade: an integration bug with MEV Boost caused issues with the proposer's signature, leading to missed slots and slashing. The goal of integrating this process into the Ethereum protocol is to address these challenges, ensuring that even if an agreement between the proposer and builder falls apart, the proposer remains reimbursed. So, if a builder provides a faulty block, the proposer still gets the full bid, leaving the builder to bear the consequences. While the specifics of this integration, referred to as ePBS (enshrined proposer-builder separation), are still being researched, and possibly a couple of years from realization, there are already many different ideas of how it could possibly look.

### How to Enshrine Proposer-Builder Separation

To understand potential ePBS implementations, it's essential to first understand some basic components of Ethereum’s PoS algorithm. In Ethereum time is segmented into 12 second intervals called slots. 32 of these slots come together to form an epoch. In each slot, a validator is randomly selected to propose a block. Simultaneously, a committee is designated to attest to the validity of the block they deem compliant with Ethereum's PoS fork choice rules, ideally attesting to the most recently proposed block as the blockchain's head. If a block isn't proposed in the given slot, then 4 seconds in, the attesting validators attest to the previous block.

Now, onto ePBS designs. The most favored model spans two slots. First there’s a bidding phase, where builders send their bids to the validators. Then Slot 1 begins with the proposer picking a bid and committing to it by publishing a block that commits to that builder’s bid. A group of attestors then cast their vote in favor of this block, ensuring its place in the chain. In Slot 2, the builders see the bid that was committed in the proposer's committed block and the attestations on it. Recognizing the proposer's irreversible commitment, the builder whose bid was selected releases their block and is assured that their MEV can’t be stolen. Finally, attestors validate this new block.

"Two-slot" ePBS design

A newly released model is similar to the two slot approach but introduces a payload-timeliness committee. First, a builder's bid is selected and committed to by the proposer, and then the committee of validators gives its attestation. Subsequently, the builder reveals the block's payload (its transactions), and the payload-timeliness committee confirms that the payload was provided on time and its validity. The other differences between these two methods lie in the specifics of Ethereum's Proof of Stake operations, but that's beyond the scope of this post.

ePBS design with a Payload-Timeliness Committee

Another design revolves around the concept of a slot auction. Here builders, during their bid, commit to a slot in the epoch without specifying the block. They essentially pledge to create a block during their allocated slot, offering a certain price to do so. This offers adaptability, especially for cross domain MEV which requires real time action.

So far all the ePBS designs grant the builder full control over the block's transactions. A potential workaround is the use of an inclusion list. This list, sent by the proposer to the builder, ideally all the transactions currently in the mempool or doesn’t have to be, contains transactions that must be part of the builders block if there’s space. If the builder's block is full, they must indicate so, affirming they acknowledged the list. Such a method strengthens the network censorship resistance. If a builder wants to censor a transaction it will become difficult and costly to do so over time. Due to EIP 1559, consecutively filled blocks will cause the base fee to exponentially rise. Therefore if a builder continually censors a transaction by filling up a block with dummy transactions, the escalating costs make this infeasible overtime.

There might be instances where the proposer wants to have some influence on block creation. Another ePBS feature might involve the proposer making a section of the block (either the start or end) and delegating the remainder to a builder. All of these designs and features aren't mutually exclusive, it's more about balancing their benefits and drawbacks.

## The Optimistic Relaying Approach

Another take on ePBS leverages our existing trusted relays. The idea is to incrementally reduce the responsibilities of the relay until it primarily serves as an optimizer, rather than a crucial component. In its first phase, we shed the relay's duty of verifying block validity. This greatly reduces the cost of running a relay as there’s no longer a need for block simulation to ensure its validity. Plus, it streamlines the relay’s role, shaving off about 100 to 200 milliseconds of latency in their communications with the proposers and builders. So, how do we ensure the proposer gets their payment if a block turns out to be invalid? The builders would be mandated to post collateral, equal to their bid, when they bid. If the block is invalid, the collateral covers the payment the proposer would've received. This concept is termed Optimistic Relaying V1.

Optimistic Relaying V1

Pushing optimistic relaying a step further to V2, we can eliminate the relay’s need to download the block, reducing another 50 to 100 milliseconds of latency. The same assurances apply: if a block never downloads, the builder's collateral pays up.

Optimistic Relaying V2

Ultimately, the end game for Optimistic Relaying starts looking a lot like the payload-timeliness committee model I touched on earlier. Here's the sequence: Builders submit their bids on a peer to peer layer. The proposer accepts a bid and follows up with a signed header. Then, the builder rolls out the block. At this stage, the relay's sole job is overseeing the peer to peer layer mempool, basically clocking when different activities occur. The relay's role becomes super lightweight, it just needs to keep tabs on the mempool. It makes the relay operate a lot like the payload-timeliness committee. All these steps build towards a future where the relay is replaced by the payload-timeliness committee, streamlining the entire protocol.

## Leveraging Builders for Additional Protocol Enhancements

PBS emerged as a response by Flashbots to the centralizing effects of MEV, aiming to attempt to harness MEV for positive outcomes. Given the new role in Ethereum specializing in block building, there's an opportunity for these entities to act like supercomputers, contrasting the lightweight validators. Protocol designs are surfacing that capitalize on these powerful builders. The idea is to keep validators uncomplicated and straightforward (some might even say cucks), while builders, having no such restrictions, can function at a much higher computational level. A primary application for these enhanced builders is for scaling.

Ethereum Researcher Dankrad Feist's Danksharding design suggests that these high resource intensive builders can construct one big block that contains all the data. That data is then segmented and committed to by multiple KZG commitments. Constructing this block requires considerable resources, but validating them is relatively inexpensive. Lightweight validators can then apply Data Availability Sampling to inspect a small section of the block and be nearly certain of the entire dataset's accessibility, yielding an added ~16x boost in data throughput from Proto-Danksharding. The intricacies of Danksharding are complicated and not covered here, but the key point is that these advanced builders can be delegated additional roles to further enhance the network.

Another idea to take advantage of the builders is the potential realization of based rollups. A few years back, Vitalik discussed rollup sequencing models, coining the term for one of them Total Anarchy

, in which anyone can publish a rollup block and the first sequence that hits the chain is the official block. This approach had many drawbacks, like onchain spam and ambiguity over the winning sequence. However, Justin Drake's recent article on based rollups highlighted a more efficient strategy taking advantage of the builders. In this model, the builder on layer one functions as the rollup sequencer, cherry picking the optimal sequence to include onchain. This ensures only the optimal sequences are incorporated.

Beyond rollups, the introduction of powerful builders can spur other innovative structures, like stateless clients. They empower nodes to operate as usual but without the baggage of preserving outdated states. These allow the nodes to be more lightweight and hinge on the builder's ability to generate specific cryptographic proofs.

## PEPC: Protocol-Enforced Proposer Commitments

Protocol-enforced proposer commitments (PEPC, pronounced pepsi) is a concept introduced by Ethereum Foundation researcher Barnabé Monnot in October 2022. You can delve deeper into Barnabé's original post here. At its core, PEPC aims to grant proposers a greater say in block building, that they've lost by selling the entire task to specialized builders. In PEPC, proposers can add extra conditions for a block to be deemed valid, apart from the usual Ethereum requirements. If a block fails to meet any of these extra conditions, it's considered invalid, and attestors should reject it. This is different from EigenLayer commitments where validators with extra commitments either lose some staked ETH for noncompliance or get rewarded for fulfilling them. However, the block remains valid regardless of these commitments.

Imagine Alice is a proposer in the Ethereum network. With PEPC, Alice has the flexibility to make a specific commitment for the upcoming block. She could commit that her block will contain at least three transactions pertaining to a particular smart contract, and she's willing to allocate 70% of the block's gas for these. She communicates this commitment, and it becomes part of her block's validity conditions. Now, Bob, a Builder, sees Alice's commitment. He packages together a bundle of transactions fitting Alice's criteria and sends it her way. If Alice's block, after being constructed, aligns with her commitment (i.e., contains the specified transactions consuming the designated gas), then the block is deemed valid and can be added to the blockchain. If not, Alice's block won't be accepted, ensuring that she adheres to her declared commitments.

One key difference between ePBS and PEPC lies in the commitments nature. In ePBS, proposers and Builders follow a fixed, uniform procedure. It's a kind of one size fits all approach. A proposer commits to a specific block hash, and the builder then produces a matching payload. However, PEPC introduces variety. Each proposer can set unique conditions, offering a lot more flexibility. It's crucial to note that PEPC's existence depends on ePBS, they complement each other. The exact workings of PEPC are still under discussion and research.

### Conclusion

PBS isn't a specific implementation, it's a design philosophy. It says that there are incentives for the division of labor and that protocol actors will delegate some responsibilities to more specialized external entities. The protocol's aim is to offer a reliable, as trustless as possible interface to ensure this delegation is smooth, fair, and inclusive. Without this, some actors might have an edge, leading to the centralization issues first observed with MEV before the PBS era. At its core, PBS emphasizes fairness and decentralization. While the exact elements to be integrated into the protocol will be seen in future Ethereum updates, Ethereum’s overarching goal remains clear: accessible, open, trustworthy stateful computing, overseen by a decentralized group of lightweight validators.

## Resources for Deeper Understanding

.css-1je14yt{max-width:48rem;}

.css-3uyxxe{max-width:100%;width:100%;}

.css-11j27fm{display:block;background:linear-gradient( to right, rgba(var(--groupBackground), var(--shadeGroupBackground)), rgba(var(--groupBackground), var(--shadeGroupBackground)) ),linear-gradient(to right, rgb(var(--background)), rgb(var(--background)));color:rgba(var(--foreground), var(--shadeTextSecondary));font-size:1rem;position:relative;border-radius:1.5rem;margin-left:auto;margin-right:auto;width:100%;--relativeBackground:var(--groupBackground);pointer-events:none;}.css-11j27fm::after{content:"";border-radius:1.5rem;position:absolute;top:0px;left:0px;right:0px;bottom:0px;pointer-events:none;border:1px solid rgba(var(--foreground), var(--shadeForegroundSecondary));-webkit-transition:125ms ease-in-out border;transition:125ms ease-in-out border;}.css-11j27fm::before{content:"";border-radius:1.5rem;position:absolute;top:0px;left:0px;right:0px;bottom:0px;pointer-events:none;-webkit-transition:125ms ease-in-out box-shadow,125ms ease-in-out background;transition:125ms ease-in-out box-shadow,125ms ease-in-out background;z-index:-1;box-shadow:0 0 0.5rem rgba(var(--groupBorder), var(--shadeGroupBorder));}

.css-16no7tu{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-box-pack:center;-ms-flex-pack:center;-webkit-justify-content:center;justify-content:center;padding-left:2rem;padding-right:2rem;padding-top:4rem;padding-bottom:4rem;width:100%;}

.css-1wpjqbx{height:24px;width:24px;display:block;}

.css-11j27fm{display:block;background:linear-gradient( to right, rgba(var(--groupBackground), var(--shadeGroupBackground)), rgba(var(--groupBackground), var(--shadeGroupBackground)) ),linear-gradient(to right, rgb(var(--background)), rgb(var(--background)));color:rgba(var(--foreground), var(--shadeTextSecondary));font-size:1rem;position:relative;border-radius:1.5rem;margin-left:auto;margin-right:auto;width:100%;--relativeBackground:var(--groupBackground);pointer-events:none;}.css-11j27fm::after{content:"";border-radius:1.5rem;position:absolute;top:0px;left:0px;right:0px;bottom:0px;pointer-events:none;border:1px solid rgba(var(--foreground), var(--shadeForegroundSecondary));-webkit-transition:125ms ease-in-out border;transition:125ms ease-in-out border;}.css-11j27fm::before{content:"";border-radius:1.5rem;position:absolute;top:0px;left:0px;right:0px;bottom:0px;pointer-events:none;-webkit-transition:125ms ease-in-out box-shadow,125ms ease-in-out background;transition:125ms ease-in-out box-shadow,125ms ease-in-out background;z-index:-1;box-shadow:0 0 0.5rem rgba(var(--groupBorder), var(--shadeGroupBorder));}

.css-16no7tu{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-box-pack:center;-ms-flex-pack:center;-webkit-justify-content:center;justify-content:center;padding-left:2rem;padding-right:2rem;padding-top:4rem;padding-bottom:4rem;width:100%;}

.css-1wpjqbx{height:24px;width:24px;display:block;}

.css-16no7tu{display:-webkit-box;display:-webkit-flex;display:-ms-flexbox;display:flex;-webkit-box-pack:center;-ms-flex-pack:center;-webkit-justify-content:center;justify-content:center;padding-left:2rem;padding-right:2rem;padding-

```
top:4rem;padding-bottom:4rem;width:100%;}

.css-1wpjqbx{height:24px;width:24px;display:block;}

.css-1wpjqbx{height:24px;width:24px;display:block;}
```