Lens is a prominent alternative to centralized social media. It is built on the Polygon blockchain. At first glance, building on Polygon seems like a practical decision. If you want a decentralized network, just build it on top of an existing blockchain! However, this approach causes significant problems for social media applications. Polygon is a Type 2 network, meaning that a fee must be paid to access the blockspace. While this is not a problem for financial transactions, it is a major issue for social media broadcasts, where willingness-to-pay is very low. The Polygon design is not fit for the purpose of social broadcasts and increasing its throughput does not solve the issues because they are inherent to any fee-based network. Although Semaphore's approach of making a Type 3 network is conceptually new, it greatly simplifies building on top of the protocol, improves the user experience, and maintains sovereign blockspace that does not need to compete with other use cases.

Network fees make it necessary for Lens apps to abstract the fee away from users (or in other words, provide a subsidy for user posts). If this is not done, the result is a terrible user experience: users are required to buy cryptocurrency to use social media and every button press is a taxable event. Although Lens apps can allow users to pay their own fee, this is clearly not a viable path forward for decentralized social media. However, fee abstraction/subsidization dramatically increases the complexity of operating an app and causes incentive compatibility issues. The goal of any decentralized protocol is to make it easy to run. If complexity is necessary, encapsulated complexity is preferable to systemic complexity. SMTP, though simple, is impossible to run without enormous systemic, extra-protocol complexity. This is an issue in Farcaster too. Apps running on Lens, instead of simply relaying messages from users to the network, must decide which messages to broadcast to the network, what fee to pay, and which messages to reject. Furthermore, because costs are paid by the app operator and not the users, there is a significant incentive for malicious users to abuse and sybil attack this system for free spamming/data storage. Semaphore solves this issue because costs need to be abstracted away from the user only once, when an alias is acquired, rather than every time a user posts. Apps also have the option to not abstract alias acquisition at all. Once a user has an alias, there is no complexity for an app operator. All broadcasts can simply be forwarded to the Semaphore network. A real world metaphor: it is much simpler to have a single toll on a bridge into an island than it is to charge every time a car turns at an intersection. Lastly, under a fee based system, apps must compete with each other for blockspace. If Alice's app is able to pay priority fees for users, but Bob's app is not, users will be incentivised to use Alice's app for more reliable access to the chain. Apps that are able to extract the most value from their users (such as by selling user data, etc) will be best positioned to subsidize higher fees. Apps more capable of paying fees could even bully other apps out of reliable access to blockspace. Just because an app is more capable of paying fees does not mean that its users' content provides the most value to the network. Semaphore's karma system means that a user's access to the chain is defined entirely by the user, not the app they choose to use. Karma also prioritizes users who garner the most interactions, which is a good approximation of their content's value.

Polygon is meant to be a high throughput, low fee blockchain. Maybe its throughput will be so great that fees do not cause an issue for Lens? Unfortunately, due to the Jevons paradox, increasing throughput does not trivially result in lower fees: as costs decrease, demand increases. In January of this year, Polygon hard forked in an attempt to reduce fees, which historically have spiked. MEV bots ensure that blocks will never go empty! Recently, as DeFi activity has decreased, so have Polygon fees. This is good for Lens, but means Lens is in the awkward position of betting against the success of the network it is built on! Furthermore, there are limits to how much the throughput of Polygon can be increased due to both decentralization concerns and the physics of RAM/disk due to how EVM state storage works. Worse, because Lens is building on Polygon's expressive blockspace, it must compete for space with DeFi apps, NFT auctions, and MEV bots, which have higher willingness-to-pay. The result is less reliable access to the chain because the chain is primarily a financial instrument. Non-financial use-cases survive only due to (likely temporarily) low activity. This will remain true even if Lens migrates to an application specific rollup because the rollup must compete for expressive blockspace on L1. In Semaphore, all resources are dedicated to processing social broadcasts only. Furthermore the Semaphore avoids the RAM/disk issues of building on the EVM.

Even if fees were consistently low, this would only cause a new problem: cheap spam. In a financial network where users pay a fee, the concept of spam does not really exist: every broadcast was worth it for the individual who paid the fee. But social broadcasts are different. They are for the benefit of not only the sender but also for the other users of the network who want to read interesting things and not see spam. This means that very low Polygon fees result in Lens getting filled with endless spam and high fees result in a bad user experience and a less accessible chain. High fees are no solution to the problem of quality either. There is no reason to expect that those willing to pay higher fees will post better content to the network. Although apps could choose to selectively broadcast messages based on some criteria, this results in a tragedy of the commons. Each app is incentivised to broadcast its own users' posts. Ultimately, the reason that fees are bad as a social blockspace allocation mechanism is that good users and spammers have equivalent access to the chain. Under such a system, all users are required to pay the same fee: you cannot make the chain more accessible to good users and less accessible to spammers. However, this is not the case in Semaphore. When usage exceeds capacity, priority is assigned to aliases based on their karma, derived from the demonstrated value of accrued interactions. Furthermore Semaphore's system means that the chain actually is cheaper for good users than spammers: Semaphore's programmable paradigms enable blocklists, etc that reduce the reach of spammers. This means that good users must pay the cost of an alias only once, but spammers must pay continuously. Lastly, what makes a blockchain a social network? It is not that its creators state that that is its purpose. Any decentralized system will be used in an adversarial way, and making the chain accessible by paying a fee means that the network is simply a data availability layer. There is no way to ensure that the data that is posted to the network is actually interesting social content using fees. Only by prioritizing access for users whose content has garnered interactions can a blockchain remain a viable social network.

Fee issues aside, running Lens on an EVM results in significant complexity, unnecessarily. Turing complete computations require a gas system, necessitating the use of fees. It is not possible to build a Lens node that ignores the rest of the activity on Polygon because modules are designed to be fully composable. Similarly, the scale of Lens is constrained by the requirements of Polygon. Lens may want greater throughput than Polygon would allow. If Lens ever becomes secured by ethereum, such as by becoming a zk-rollup, these constraints would become even more rigid. If Polygon gets more popular, then more pressure will exist for blockspace. Financial activity actually does need turing complete computation, but this is not necessary for social broadcasts. By opting to exist as an EVM contract rather than creating a dedicated chain, Lens is kneecapping itself. Already, these constraints hamper the Lens design. For instance, Lens does not support likes or dislikes, instead requiring apps to add this additional functionality. However, likes/dislikes are important functionality and only supporting at the app level will lead to centralization of apps because like/dislike counts will not be shared between apps. Furthermore, the EVM reduces quality of life significantly. Its tooling is built for financial transactions and results in a clunky user experience for social applications.

The benefits of Lens running on the EVM is that the follow, collect, and reference modules can contain arbitrary logic. However, useful functionality can be replicated in Semaphore without the use of turing complete scripts. Semaphore's programmable paradigms allow users to set policies for how their posts can be interacted with. For instance, a Lens creator might set their follow module so that users can pay $10 to follow and have posts that are follower-only. Similarly, a Semaphore creator can use paradigms to define a set of aliases who have paid $10 out of band and restrict post interactions to aliases in that set. Although the set does not update automatically like it would in Lens, it is easy to make a server automatically update the set upon completion of payment in another system. This also allows payment to be done on any system, even traditional banking, rather than only Polygon. The fact that the set defined on Semaphore does not have the "followers" label is not substantial: a Lens app could trivially show the posts from a creator even if the user is not officially a "follower." Furthermore, because Semaphore's proof of engagement can be checked by validating a single signature, it is easy to submit proofs to another chain. If a creator wants to create NFT's of their posts, for example, a proof of the post can be easily created on another chain. This means Semaphore creators can monetize their content in the same ways as on Lens, without requiring the Semaphore system to support arbitrary computations.

The Lens protocol is better suited to being a social NFT/creator monetization layer than social media itself. Tools for creators to monetize their content in a decentralized way is important, but is a different challenge from decentralized social networks themselves. As stated previously, monetization of Semaphore content can be processed by an external system. That system could be Lens! With proofs submitted to Polygon, Semaphore creators can monetize using Lens easily. Furthermore, because Lens is being used for monetization rather than every individual post, the issues of using Polygon are resolved. Since Polygon is a Type 2 network, using Lens in this way would better utilize the network's strengths and avoid its weaknesses.