SAGA

LITEPAPER
MARCH 2022

WHERE YOUR JOURNEY BEGINS

SAGA

CONTENTS PAGE

Saga is for developers.

Developers are the unsung heroes of web3 and deserve far better platforms and support on their way to building their applications and chains.

Many in the crypto community believe that sovereign chains communicating freely with one another is the future for crypto infrastructure, and that now is the right time to move in that direction. However, most native crypto developers, not to mention web2 developers coming into web3, forgo being a part of this vision because they find the upfront investment and effort too high, only to see their projects constrained once they grow.

Saga makes this vision accessible.

We are a protocol for automatically provisioning application-specific blockchains in the multichain metaverse, or multiverse. We are building this by rearchitecting developer and end user interactions with blockchains. Using shared security, innovations in validator orchestration, and an automated CI/CD deployment pipeline standardized across apps in single tenant VMs, Saga makes launching a dedicated blockchain, or chainlet, as easy as deploying a smart contract.

Our network is best leveraged as a scalability layer for web3 infrastructure. Saga offers:

- No upfront cost to launch a chain
- Predictable developer pricing for provisioning block space
- Automated deployment of a single tenant application onto its own chain
- A dedicated shared security chain for a developer's applications, ensuring high throughput, no dependencies on other applications using Saga, easy upgradability and congestion relief

In addition, Saga's unique token mechanism design enables interesting and flexible developer business models and end user experiences that were not supported before in crypto:

- Saga does not charge network fees directly to the end user, leaving monetization instead in the hands of developers.
- Saga enables transaction fees to be denominated in any IBC-compatible token, not just the Saga token. Because our chainlets are meant to be used by developers for new and established projects alike, our token model uniquely enables the tokens of other protocols to accrue value even though the Saga Chain serves as the backend.

Saga's initial focus will be on gaming and entertainment chains. Not only do these applications have the most urgent need for dedicated runtime environments and close to 100% SLA, they are also the fastest growing sector in web3 with the most immediate potential for mass appeal. As the metaverse expands, performance requirements and end user expectations are climbing higher and higher, and crypto-based applications have to deliver.

While the metaverse is a burgeoning use case, DeFi will always be at the heart of web3. Saga believes that its innovations in blockchain, which ultimately open the door to processing low value, high volume transactions at scale, will enable new classes of DeFi applications that were not previously possible.

ARRIVING AT THE MULTIVERSE

As demand for smart contract technology continues to grow, developers and end users are running into three main challenges: high and unpredictable gas fees, congestion and lack of flexibility and upgradability as applications have to share blockspace with more and more applications on the same chain.

In the category of high gas fees, Ethereum receives the lion share of attention. In an CNBC article[1], MacKenzie Sigalos writes: "It has always been a

tough go for Ethereum users. The blockchain has a long-standing problem with scaling, and its highly unpredictable and sometimes exorbitant transaction fees can annoy even its biggest fans." February 5th, 2022, was a particularly notable trading day since nearly $20 million was spent on Ethereum gas fees, representing an 800% increase over the $2.4 million spent on gas fees the day before. The increase was mostly due to congestion.[2]

Congestion and high gas fees do not always go hand in hand. Many Layer 1s since Ethereum have been able to offer lower gas fees but still suffer from congestion and significant performance issues on the chain. In an effort to mitigate congestion, these base layer protocols are trying to develop blockchains with "fatter pipes" to try to cram more transactions into limited blockspace. While faster individual blockchains are helpful, congestion issues invariably resurface as end users compete to use a variety of applications. Fees on Avalanche's C-chain rivaled[3] that of Ethereum during peak congestion while excessive transactions[4] on Solana have contributed to several network halts. The fundamental problem is that blockspace is scarce in monolithic blockchains where resources are shared across myriad projects.

Finally, anytime a developer builds on a chain other than their own, they are dependent on the project that launches the chain, the community and other stakeholders for upgrades and decisions about their development cycles.

It is no wonder, then, that applications and protocols are outgrowing their chains and need to scale horizontally or migrate onto a different chain. Axie Infinity, Compound and Flow are some of the most recent examples that have chosen this path.

From its inception, the Cosmos tech stack has sought to solve these problems by emphasizing self-sovereignty. Each developer was meant to have their own dedicated chain, and therefore blockspace, to build on. However, deploying secure applications on Cosmos has also been difficult, slow and technically

complex. Every application has to recruit, maintain and coordinate between its own validator set, and so developers have to consider the multiparty dynamics of secure consensus that are not a consideration for builders on Ethereum, Solana or any other monolithic Layer 1. In addition, Cosmos aimed for maximum composability, which meant that while developers could customize applications as they liked, they also had to deal with the complexity of working with many modules.

Since the Cosmos SDK was first released in 2018, several advancements in the ecosystem have made the developer and end user experiences better. In 2021, Inter Blockchain Communication (IBC) was released, which fulfilled the Cosmos promise of enabling sovereign chains to communicate with one another and transfer assets securely. Over the last year, projects such as Juno, Evmos, Archway and Agoric have given developers a controlled and simplified VM environment in which to develop their Cosmos-based applications. In 2022, the Cosmos Hub will release its first version of Interchain Security, also commonly known as shared security, which paves the way for a validator set on one chain to validate subsidiary chains.

Together, these three key innovations in Cosmos allow for a uniquely scalable ecosystem of chains, also known as the multichain world.

We do not believe it is an accident that the Cosmos moment has arrived just in time for the Cambrian explosion in the metaverse. As web3 leads growth in the gaming, entertainment and fintech markets, greater demands will be placed on the robustness and scalability of the underlying chains and the end user experience.

Saga aims to be the protocol for this multiverse.

MARKETS AT A GLANCE

GAMING

Over the last few years, over $3 billion of gaming assets has been tokenized into NFTs. Yet this is just a fraction of overall gaming industry, which is predicted[5] to be a $269 billion industry by 2025 with close to 3 billion gamers world-wide.

ART AND ENTERTAINMENT

Art NFTs are less[6] than 1% of the $1.7 trillion dollar art industry. In 2021, musicians only captured 12% of the $42 billion music industry[7]. Crypto protocols like Royal, Async and Audius are addressing ownership issues and pioneering new value-capture methodologies, resulting in millions of dollars being passed directly to musicians.

DEFI

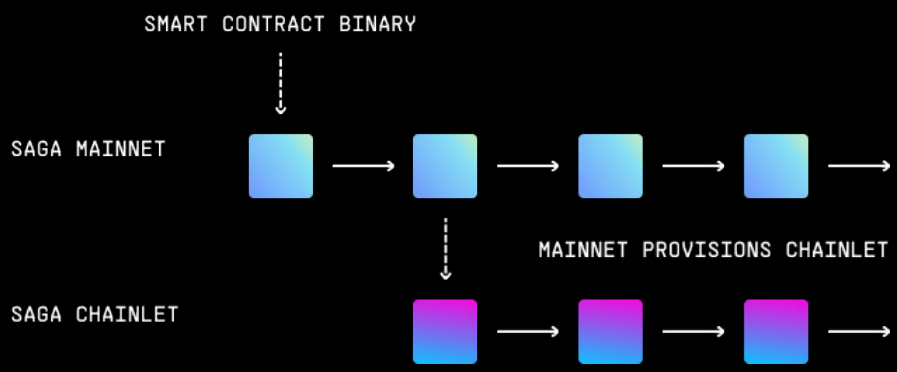
The market-cap of DeFi sits at below $100 billion and Matthew Roszak believes that it could grow 1000% percent in 2022[8]. While DeFi has gone from securing $1 billion dollars in 2020 to securing over $200 billion dollars in 2022, the market cap of global equities is estimated to be over $100 trillion[9].The generational opportunity to secure real-world assets on-chain has yet to be fully explored. Onboarding the next generation of web3 use cases will only be possible via infrastructure that can scale.

SAGA MAINNET

The Saga mainnet is the developer's portal into the Saga ecosystem.

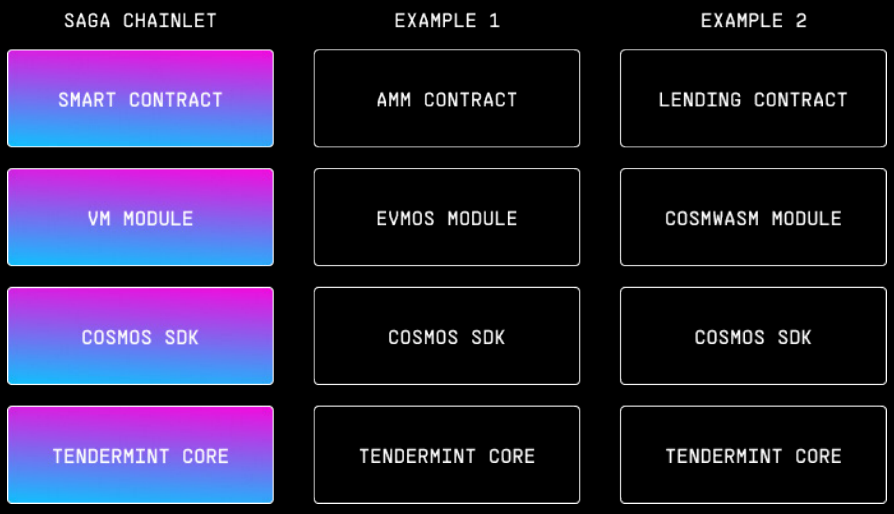
DEPLOYING A SMART CONTRACT

Like most smart contracting platforms, the developer deploys their compiled smart contract binary onto the Saga mainnet. Once initialized, the Saga validators automatically provision a Saga dedicated chain, or chainlet, containing the smart contract.



SAGA CHAINLET

In essence, a Saga chainlet is a sovereign blockchain with a VM module running a single smart contract.



Because each chainlet contains only the smart contracts a developer or team would like to deploy on a single chain, the developer has a flexible and usable development environment:

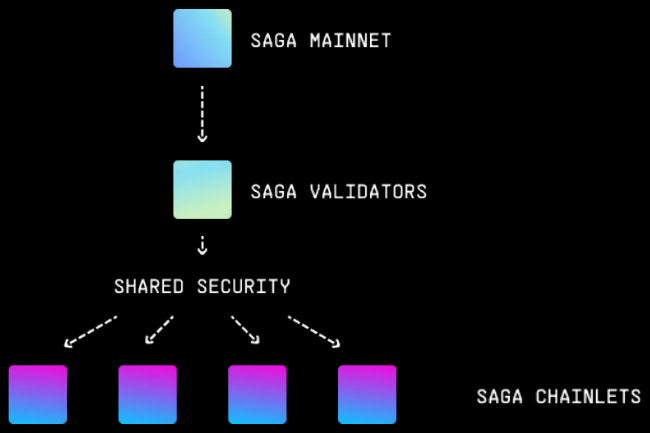
- For the developer, the deployment of chainlet is as easy as deploying a smart contract
- Chainlet block space is not shared with any other

application, leading to predictable fees
- Saga is VM agnostic, which means developers can bring their own VM module to plug into their chainlet
- Chainlet components such as VM module, SDK and Tendermint Core can be upgraded independently from other applications and Saga mainnet
- Ultimately, Saga could also support other technology stacks down the line besides Cosmos SDK and Tendermint

Saga enables developers to manually “shard” various workflows into multiple chainlets to leverage horizontal scaling. Deploying a Uniswap-like AMM smart contract into its own application specific chainlet will already offer significant scalability improvements. However, once the demand for the smart contract starts exceeding the limitations of current blockchain technology, the developer could deploy multiple instances of the same smart contract focused on particular subsets of activity. In the AMM example, the developer could deploy a smart contract instance for each asset pair. The chainlet architecture enables near limitless scalability, as long as these chainlets can be secured properly.

SAGA SHARED SECURITY

One of the challenges of deploying a Cosmos-based application specific chain is the complexity around securing the chain. Each application chain requires gathering validators, distributing staking tokens and designing a token mechanism that helps secure the chain. Saga uses shared security to remove this barrier to entry. Every Saga chainlet is secured by the Saga mainnet validators using shared security.



There are many flavors of shared security. Saga uses a model similar to the Cosmos Hub's version 1 interchain staking called Optimistic Coordination to ensure security of each of the chainlets.

- Every validator on the Saga mainnet is required to validate every chainlet provisioned
- Validators agree on a set of SLA for chainlet provisioning and maintenance such as timely deployments, guaranteed compute capacity, minimum uptime, honest consensus participation, and inter-blockchain communication relaying
- An auditor monitors the chainlet and creates a governance case in the event that a validator fails to meet service obligations
- Saga mainnet enforces any consequences needed for validators violating SLA

With Optimistic Coordination, chainlets automatically inherit the security of the Saga mainnet. However, to make provisioning chainlets as simple as possible, Saga requires a suite of tools to facilitate chainlet orchestration for validators.

VALIDATOR ORCHESTRATION TOOLS

Deploying a smart contract and provisioning a chainlet on Saga is intended to be a fully automated process. We can expect thousands of independent chainlets to be running at the same time. This means that Saga validators will need to be able to automatically manage thousands of independent binaries on their servers and keep up with resources needed to maintain them. To facilitate this process, Saga will develop validator orchestration tools that

will simplify the process of placement, scheduling, releasing, installing and upgrading:

- Automatically spin up new binaries and server environments for each chainlet
- Automatically schedule binaries to be executed based on chainlet subscription SLAs
- Facilitate deployment of new chainlets on multiple compute resources
- Monitoring tools for validators to ensure sufficient resources are available for provisioning new chainlets

One of the more difficult tasks for validators will be predicting the required hardware necessary to service the utilization needs of all chainlet applications. On the one hand, having too much excess compute resources is wasteful. However, not having enough resources may result in slashing due to computer SLA violations. Therefore, Saga's validator orchestration tools will need to help validators predict when new hardware resources will be necessary.

Splitting out applications into one or more chainlets means that composability will be more difficult to achieve on Saga. Validators will be required to relay IBC messages between any two pairs of chainlets, including the Saga mainnet. Our IBC will additionally be more performant, by virtue of all our chainlets being co-located in each validators' datacenter. However, more tools and development will be needed to make the IBC user experience as seamless as possible.

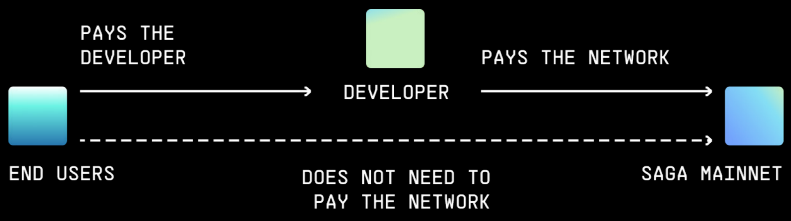
Saga's token mechanisms rely heavily on seamless cross-chain composability. Manually transferring assets between chainlets and other Cosmos chains through IBC is a poor user experience for the end user. Ultimately, we want assets to seamlessly transfer between blockchains without the end user initiating IBC transactions. In addition to creating validator orchestration tools around IBC, Saga will work with the broader Cosmos ecosystem to augment features to automate and abstract the IBC experience away from end users.

TOKEN MECHANISM AND ECONOMICS

We split the token economics into two conceptual components: the front end and back end. The front end is the portion of the token flow between the chainlet end users and the chainlet developers. The back end is the portion of the token flow between the chainlet developers and the Saga chain. Splitting up the token mechanism into two independent parts allows for interesting usability benefits and value accrual for developers and our partner chains.

FRONT END: END USER TO DEVELOPER

Imagine a scenario where Uber hosts their mobile apps on Amazon's servers. The expected user interaction flow is as follows: end users pay Uber then Uber pays Amazon for hosting their applications. It is unusual to expect end users to pay Amazon directly. However, this is the norm in most smart contract platforms today. The fees from the underlying network are paid by end users, not application developers. This results in end users needing to custody enough tokens to pay for the fees and prevents developers from exploring interesting business models such as ad-based or freemium models.



On Saga, end users of each application interact only with the chainlet, and all fees for provisioning chainlets are paid by the app developers. We will discuss the token flows for how this chainlet provisioning happens in the back-end section.

This front end model enables developers to explore interesting monetization models that were previously not possible. End users are abstracted away from network fees and the developers can implement more flexible monetization methods:

- The developer can still implement fees per transaction. However, this model enables those

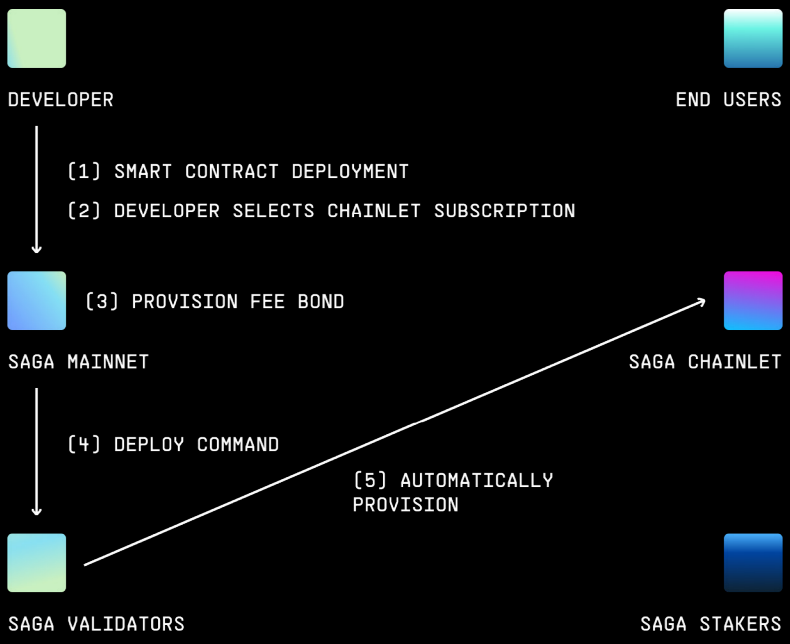
fees to be paid in whatever tokens the developer wants. The fee tokens gathered by the developer can be any external tokens (like stablecoins), Saga tokens, or tokens created by the developer. This fee will accumulate in a wallet controlled by the developer

- The developer may implement a subscription service where only whitelisted users may use the smart contracts
- The developer can also choose to make their product free for end users with monetization coming from other sources like advertisements. In this scenario, the developer will need to set up some spam prevention mechanisms to prevent malicious users from utilizing the chainlet capacity purchased by the developer. However, this can also be solved via whitelisted accounts

BACK END: DEVELOPER TO SAGA MAINNET

In Saga, the developer pays for provisioning the chainlet. The interaction flow is designed to be similar to how an Amazon EC2 instance is provisioned for web developers. In order for a developer to deploy a chainlet, a developer must subscribe to a fee deposit, which guarantees a specified level of compute capacity for a specified amount of time for the provided chainlet. The cost increases for higher capacity subscriptions, and there will be discounts offered based on the duration of commitment.

Once the subscription is selected, the developer needs to post the necessary bond in Saga tokens to provision the chainlet. In essence, the fee bond is a pre-pay account to provision chainlets and will be drawn down over time. At the outset, the network will provision “free credits” akin to trial accounts offered by services like AWS and Google Cloud to allow developers to freely set up testnets or experimental chains, but these free credits will only allow for a limited amount of chainlet capacity for a limited time.



Once the chainlet is provisioned, the end user can directly interact with the chainlet. The maximum utilization of the chainlet will be restricted based on the subscription tier selected by the developer.

The subscription model prevents idle, unused chainlets from existing forever. The only requirement for maintaining the chainlet is the payment of the subscription fees. As the protocol depletes the fee bond balance, the developer must maintain a sufficient balance to pay for the subscription.

We may also allow developers to stake the fee

bond tokens. Provided the developer has enough Saga tokens staked in the fee bond, this could allow developers to automatically pay for chainlet subscriptions from the staking rewards.

Low deposit balances can be replenished by developers or broader community incentives, but once depleted, validators may halt the chain and after a predetermined grace period, remove the unused binary and underlying data.

The fee generated from developers is intended to pay for the validators' infrastructure cost of maintaining chainlets. Saga incentivizes validators to offer the most competitive rates to developers through our validator selection mechanism.

VALIDATOR SELECTION

Traditionally in Cosmos-based projects, the validator active set is determined solely by the delegation count. With Saga, we will use the delegation process to incentivize validators to lower prices for provisioning chainlets. Saga will use a form of auction mechanism to have the validators bid with the lowest price they are willing to offer developers to provision a new chainlet. Only the cheapest validators will be selected as the active set. The price to developers will be set as the rate offered by the last (and most expensive) validator on the active set.

Over time, this validator selection mechanism will ensure that the price offered to developers is as low as possible while ensuring that every validator is sufficiently compensated for their infrastructure costs. Only the best validators who can minimize costs will continue to be included in the active validator set.

To be clear, however, all Saga chainlets will be validated by the full set of Saga mainnet validators even though the identity of individual validators may change through this auction mechanism. This condition is necessary to maintain the security of the "optimistic coordination" mechanism described above.

STAKING REWARDS

Since back end fees are intended to be distributed directly to validators, the token stakers are compensated by inflation. Saga uses a simple inflation model where the inflation rate is determined by the total percent of locked Saga tokens. The protocol will target a specific staking rate (70% for example) by adjusting the inflation rate.

Similar to other Cosmos chains, validators will set a commission rate of the inflation they want from delegators. This is in place to compensate validators during the network's early days when the fees may not fully cover their operating costs. In the long run, we are targeting an economic system where the entirety of the validator's costs and margins are fully covered by the fees. When that time comes, we expect the network to eventually target zero or even negative commission rates where validators pay delegators from their fee revenues to attract delegation.

PROTOCOL CONTROLLED RESERVE

Saga has a native protocol controlled reserve - a pool of Saga tokens that the protocol controls. The protocol controlled reserve is intended to be used for Saga mainnet-critical activities including:

- Funding the "trial credits" to developers to get started with provisioning a chainlet for free
- Funding relayers for IBC activities
- Funding protocol-level partnerships where key technology partners, such as virtual machine developers, are paid directly
- Funding other common goods activities and features in the future that may be introduced on the mainnet
- Burning to reduce total token supply to target a specific supply

The protocol controlled reserve will be initially funded at the Token Generation Event, but will continue to be funded through a few potential types of taxes imposed on the token system, such as:

- % of end user fees
- % of developer fees

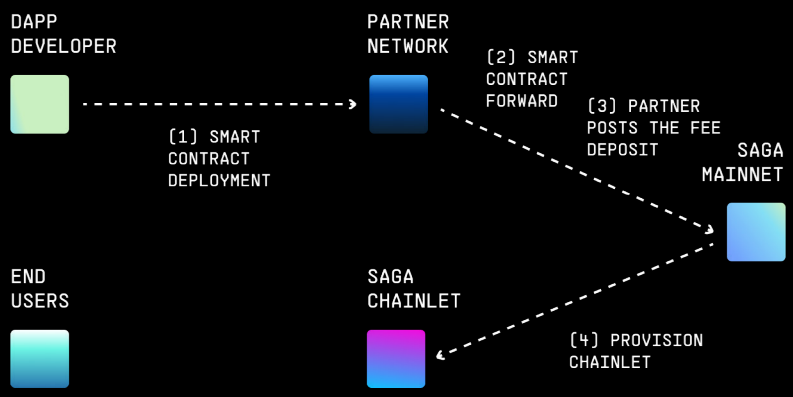
› % of inflation

By controlling where the PCR taxes token activity, the protocol can incentivize certain behaviors while having fine-grain control over how economic accrual happens in the system.

PARTNERSHIP MODELS

Saga's unique and flexible product design enables partnership models with other ecosystems that allow for accrual of value to these ecosystems' tokens and communities even when Saga serves as the base chain.

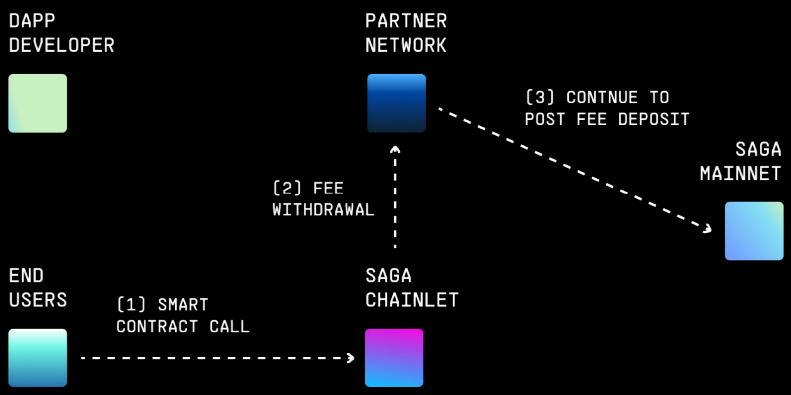
First, Saga could partner with various other layer one blockchain ecosystems to offer horizontal scalability solutions while keeping the economics of the respective tokens intact.



Imagine a scenario where a partner network is expecting a smart contract that will generate too much traffic in the partner network's mainnet. To prevent their blockspace from being too congested, the partner network could deploy the smart contract on Saga instead. The partner network can post the chainlet provisioning fee in Saga tokens to get the smart contract provisioned.

The chainlet can be configured to demand end users to pay fees in the partner network tokens. This way, the end user's interaction flow is identical to the case in which the smart contract is deployed on the partner mainnet. The fees paid by the end users follow our front-end token model and get distributed to an account owned by the partner

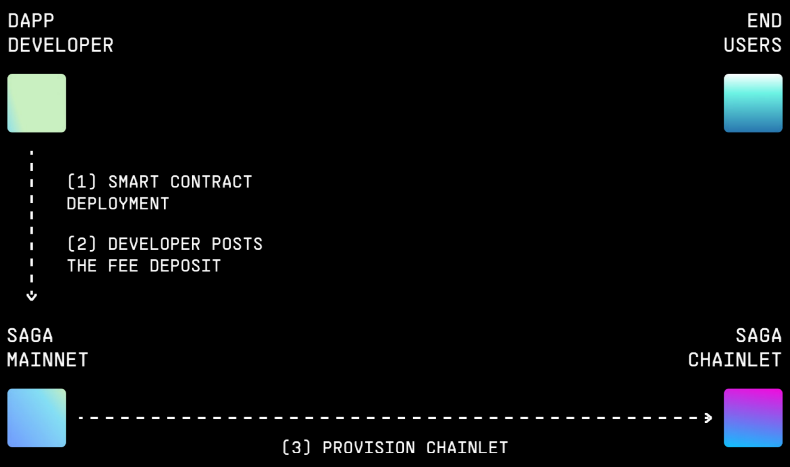
network. A portion of these fees can be converted to Saga tokens to pay for the chainlet subscription. The remaining balance can be configured to be distributed back to the partner's validators and stakers in the same method as smart contracts deployed on the partner's mainnet.



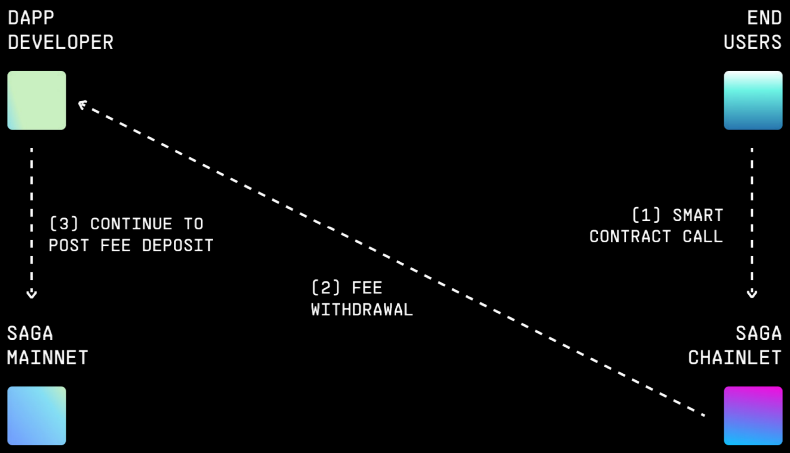
Saga's token mechanism allows for accumulated transaction fees (net of Saga chainlet fees) to accrue value to partner stakeholders. The partner also keeps their developer and end user pipeline, maintaining brand loyalty while scaling through Saga.

A second method of partnership is a revenue share between the Saga mainnet and technology teams across the ecosystem. This structure covers cases in which developers are coming directly to Saga to build their applications. Saga is meant to be a VM agnostic chain, and numerous VM options will be available to developers who come to Saga looking for an environment to build their application in. Therefore, Saga will want strong technology partners to support and develop VM technologies on Saga. Saga can always use its Protocol Controlled Reserve to pay and incentivize these developers to continue to maintain and support their respective VM options for Saga.

More specifically, the developer chooses a VM environment and deploys a smart contract on Saga. They then post the fee deposit that is needed to provision and maintain the chainlet. Saga will provision a chainlet for the developer that uses the native token of the VM for transaction fees.



When an end user uses the application, they call the smart contract directly on the chainlet and pay transaction fees in the native token of the VM. These fees accumulate in a wallet controlled by the developer. The developer can convert part of the fees to pay for the required bond needed to provision the chainlet or however they see fit. Saga can also take a portion of the fees and send them to the network affiliated with the VM technology partner as a revenue share, incentivizing value accrual for the partners' tokens.



FOUNDING TEAM

Rebecca Liao, Chief Executive Officer - Saga is Rebecca's third startup. She was most recently Co-Founder and COO at Skuchain, alongside Zaki Manian, one of the original inventors of Cosmos. Under her leadership, Skuchain's platform grew to $5 billion+ in annual volume. She is currently Advisor to Sommelier Protocol, where she is designing their DAO to optimize for governance, platform growth & regulatory compliance. Prior to crypto, Rebecca was an early employee and Director of Business Development and Head of Asia at Globality, a Softbank-backed B2B unicorn for AI-powered procurement of professional services. She was foreign policy and tech policy advisor for President Joe Biden's 2020 presidential campaign and Secretary Clinton's 2016 presidential campaign. She began her career as an international corporate attorney at Skadden Arps Slate Meagher & Flom LLP and Fenwick & West LLP.

Jacob McDorman, Chief Technology Officer - Jacob is an entrepreneur, product builder, and researcher. He has endeavored in growing product ventures of both his own and others for over a decade, as a startup founder, development studio owner, and advisor. He is vision oriented with a healthy portion of practical approach.

Jin Kwon, Chief Strategy Officer - Jin has contributed to the Cosmos ecosystem since 2018, when he joined Tendermint as Chief of Staff to the CEO. More recently, Jin served as VP of Corporate Development as well as managing director of Tendermint Ventures. Jin has a multidisciplinary background in engineering, sales, marketing and finance and wants web3 proliferated in everyday use cases.

Bogdan Alexandrescu, Vice President, Engineering - Bogdan is a technology expert and industry leader. A successful entrepreneur with previous start-ups in technology, fintech, and business consulting. He's founding partner at Blocktech Ventures, a venture capital firm with a focus on frontier technology and blockchain, member of the Bitcoin Foundation and active investor and advisor in the blockchain ecosystem. Before joining Saga he was

Engineering Leader at Apple Inc, focusing on large-scale Distributed Computing, Machine Learning, AI & infrastructure projects. Previously Tech Lead for Twitter Ads and Research Specialist for University of California.

CONCLUSION

Saga's mission is to give developers what they need to build in the multiverse. Web3 has already evolved to the point where new infrastructure configurations and token mechanisms are necessary to keep up with the growth of the industry and incredible influx of end user interest, the potential of which is still untapped. Saga's unique network and token design meet this moment. For new developers, developers at established projects looking to grow and the end users they all support, a new protocol needs to bridge to the multiverse.

Your journey starts here

CITATIONS

[1] Sigalos, MacKenzie. "Ethereum Just Activated a Major Change Called the 'London Hard Fork' — Here's Why It's a Big Deal." CNBC, 5 Aug. 2021, www.cnbc.com/2021/08/05/ethereum-just-activated-its-london-hard-fork-and-its-a-big-deal.html. Accessed 24 Mar. 2022.

[2] "Ethereum Average Transaction Fee." Ycharts.com, ycharts.com/indicators/ethereum_average_transaction_fee.

[3] "Rising Fees on Avalanche Trigger Scramble for Solutions." The Defiant, 24 Nov. 2021, thedefiant.io/avalanche-ethereum-gas-fees-superpruning/. Accessed 23 Mar. 2022.

[4] Goschenko, Sergio. "Solana Formally Acknowledges Problems with 'High Compute' Transactions Clogging the Network." Bitcoin News, 8 Jan. 2022, news.bitcoin.com/solana-formally-acknowledges-problems-with-high-compute-transactions-clogging-network/.

[5] Nystrom, Mason. "Crypto Research, Data, and Tools." Messari.io, 10 Feb. 2022, messari.io/article/a-glimpse-into-the-leading-categories-within-the-deca-billion-nft-industry?referrer=category:web3. Accessed 23 Mar. 2022.

[6] Beasant, Alexander. "Crypto Research, Data, and Tools." Messari.io, 22 Jan. 2021, messari.io/article/crypto-and-music-jamming-into-web-3-0?referrer=category:web3. Accessed 23 Mar. 2022.

[7] Woetzel, Jonathan, et al. “The Rapid Growth in Global Wealth | McKinsey.” Www.mckinsey.com, www.mckinsey.com/industries/financial-services/our-insights/the-rise-and-rise-of-the-global-balance-sheet-how-productively-are-we-using-our-wealth#:~:text=The%20world%20has%20never%20been. Accessed 23 Mar. 2022.

[8] Wu, Ethan. “Why DeFi Could Be an $800 Billion Industry next Year, according to a Crypto Expert.” Markets Insider, markets.businessinsider.com/news/currencies/defi-crypto-800-billion-industry-billionaire-decentralized-finance-vesper-2021-08. Accessed 24 Mar. 2022.

[9] Tolkchev, Artem. “DeFi Can Be 100 Times Larger than Today in 5 Years.” Cointelegraph, 6 Nov. 2021, cointelegraph.com/news/defi-can-be-100-times-larger-than-today-in-5-years.