Chenilles Glow Whitepaper

April 2023
(Latest update May 16th 2023)

https://bit.ly/ChenillesGlowWhitepaper2023

François-René Rideau
fare@mukn.io

Mutual Knowledge Systems, Inc.
https://mukn.com

218 NW 24th Street, 2nd Floor
Miami FL 33127

Abstract

We present our new *Chenilles* Network of generalized state channels across blockchains, which works in synergy with our existing *Glow* language for safe DApps.

Like the Bitcoin Lightning Network, *Chenilles* enables fast and private self-custodial payments along paths in a network of state channels. But improving on these first generation state channels, *Chenilles* enables both payments *across multiple blockchains*, and private smart contracts in addition to private payments.

Previous solutions for programming generalized state channels required unmanageable complexity: hand-writing many exactly matching variants of on-chain and off-chain code for each smart contract, where the slightest discrepancy could result in losing funds. The *Glow* language makes the task manageable: *Glow* automates the generation of the many exactly matching variants of on-chain and off-chain code involved in such smart contracts, across multiple blockchains if needed. *Glow* can target public or private contracts, with or without state channels, with or without zk-SNARKs, on any smart contract platform (Ethereum, Bitcoin, Cardano, Filecoin, etc.), and is also designed around formal methods to make correctness proofs tractable.

Together, *Chenilles* and *Glow* will unlock a new world of private, scalable and safe self-custodial decentralized commerce and finance, eventually spanning all blockchains. Users will be able to perform atomic swaps of exotic financial instruments between two side-chains, to participate in auctions on the Ethereum network while paying in Bitcoin on the Lightning Network, or to fill the orders of decentralized exchanges without funds leaving self-custody.

Mutual Knowledge Systems, Inc. is developing *Chenilles* and *Glow* as Open Source platforms, with an ambitious business model to become the company that connects all decentralized applications together.

Table of Contents

Motivation: Unlocking Cross-Chain DeFi & DeCo

Cryptocurrencies and Decentralized Ledgers, as introduced with the Bitcoin blockchain by Satoshi Nakamoto in 2009, offer a potential solution to the problem of runaway inflation of the money supply by government-mandated central banks—a problem that has already hit “first world” countries such as Cyprus in the Eurozone, and is now looming over the USA as well as all of Europe. Smart Contracts, introduced with Ethereum in 2015, enabled the creation of Decentralized Finance (DeFi), a way to reproduce traditional finance and more on top of blockchains, without centralized middlemen capable of censoring political opponents, freezing or confiscating their funds, etc., for authoritarian governments—a problem also recently experienced even in Canada, a “Western Democracy”, and that only seems to get worse everywhere.

The full potential of decentralization technology will only be realized when every human (and soon, AI) can use it daily as a way to conduct commerce without all the current centralized middlemen, in a way that can embrace and extend legacy systems rather than exist apart. But the advent of this Decentralized Commerce (DeCo) will require some fundamental innovations to improve the scalability and usability of blockchains and DeFi:

- Current blockchains offer limited transaction volumes that could not presently satisfy the demands of the world population.
- Using blockchain technology is difficult and unforgiving, with potential unrecoverable loss of funds in case of mistake by the user or of successful attack by criminals.
- Blockchains form a constellation of isolated ecosystems that do not interoperate, except by going through a few centralized or semi-centralized “exchanges” and “bridges” that often get attacked by criminals, private or government-backed.

For these reasons, blockchain technology today is only available to professionals and to enthusiastic amateurs with enough time, skills or capital to be early adopters despite all the difficulties.

We at Mutual Knowledge Systems (MuKn) propose the *Chenilles* Network, a next-generation State Channel Network. Existing State Channel networks that only help with scaling payments within one blockchain (and possibly its bridged extensions to other blockchains and side-chains); but *Chenilles*, thanks to its synergy with our language *Glow*, can also help scale smart contracts in addition to payments, and can help with payments and smart contracts across multiple Blockchains, not just within one blockchain, and can even work without having to trust a third-party bridge. *Chenilles* will become a key platform to enable usable DeFi and DeCo at scale across all blockchains.

Existing Issues with Decentralized Networks

This section explains the advantages and limitations of existing blockchain networks. You can skip to the next section Chenilles: Next Generation State Channels with Unique Capabilities if you are already familiar with them.

Layer 1 Blockchains

The main "Layer 1" (L1) Blockchains (such as Bitcoin, Ethereum, Cardano or Filecoin) create decentralized currencies that can be used for safe censorship-resistant payments between participants without third party risk. They provide

- **Security**: It would take a huge amount of capital to disrupt the network, at which point this capital becomes wasted.

BUT

- **Little Scalability**: L1s don't scale—for important security reasons. A blockchain can only have as many transactions as all the validators can see, only as fast as they can all know they all saw.

Layer 2 Side-Chains

Some "Layer 2" (L2) side-chains (such as Polygon, Optimism or Metis) provide:

- **Increased Scalability**: L2 side-chains can increase scalability, but each side-chain can only increase it so much, with the same ultimate limit on transactions.

BUT

- **Reduced Security**: L2 side-chains tend to be less decentralized, with less security.

Depending on their validation strategy, L2 side-chains have additional limitations:

- *Centralization*: Some L2 side-chains are plain centralized, and can scale better than others, but lose most of the security properties of blockchains, reintroducing third-party risk.
- *Economic Validation*: Some L2 side-chains (such as Polygon) use the very same economic validation technique as L1 blockchains such as Proof-of-Stake, except with less capital involved, which makes them a bit less secure. Such side-chains can be viewed as just another L1 blockchain with a builtin bridge to another L1. The bridge itself is often the weakest link.
- *Interactive Validation*: Some L2 side-chains (such as Optimism) use interactive proofs to ensure that transactions are valid. We introduced this technique in a previous whitepaper but no current implementation seems to have fully implemented it so far. The technique also requires publishing data either on a "data availability engine" such as Celestia which introduces additional trust issues, or on the L1 chain itself (in what is then called a "rollup") with associated scalability limitations.

- *Non-Interactive Validation*: Some L2 side-chains (such as Metis) instead use non-interactive proofs (zk-SNARKs) to ensure that transactions are valid. These chains are much easier and faster to use, but due to the current cost of building those proofs, the proof-building itself is often centralized, which reintroduces third-party risk. Moreover, these side-chains have the same data availability issues as those using interactive proofs.

Layer 2 State Channels

State Channels, as introduced by the Bitcoin Lightning Network and its many copycats, are a completely different "L2" approach. They provide a safer solution with different limitations:

- **Scalability**: State Channels allow L2 payments to scale indefinitely. L2 payments can be confirmed in a fraction of a second with no fees and no limits on the number or size of payments. State channels themselves are created and settled through L1 transactions with the usual scaling limitations, but each state channel would typically only need to be settled once per month, significantly reducing the demand for L1 transactions compared to direct blockchain payments without state channels.
- **Privacy**: State Channel payments are fully private—only participants know about individual transactions, and the public can only see channel creations and settlements.
- **Security**: State Channel funds are self-custodial, so each participant's funds are safe at all times with no third party risk and no reduction in security compared to using the L1 Blockchain. The worst that can happen is a participant's funds may be temporarily locked during some challenge period (typically one week) if another participant stops cooperating, requiring the participants to force an "exit" from the state channel. The uncooperative participant is then blacklisted from future interactions unless they pay some reparation.
- **Network Payments**: State Channels allow for atomic cascaded payments along a path of state channels (A pays B who pays C who pays D, etc.). Paths can be established between participants who already know each other, or can be automatically discovered along a Decentralized Network. Payments remain scalable and private.

BUT,

- **Only Funds under Custody**: Each State Channel can only manipulate funds under its custody, as deposited using regular L1 transactions. That's what makes them safe.
- **One Way or the Server Way**: Simply sending payments requires only a one-way state channel and little to no tech expertise. But a participant who wants to receive payments or interact with a two-way State Channel, must entrust a server with a "hot wallet" and keep the server online, monitoring the L1 blockchain regularly (typically at least once a week) or else funds may be lost. This is fine for many businesses and for crypto enthusiasts, but more effort than most users want to commit. Another safe attitude is to always make an L1 settlement before you go offline for a prolonged period of time after receiving funds.
- **Only One Chain**: Existing State Channels only function on a single blockchain.

- **Only One Token**: Existing State Channels only support exchanging a single (kind of) token on a given state channel—only bitcoins, only ethers, only a specific ERC20, etc.
- **Payments Only**: Existing State Channels do not support smart contracts. A few systems support them in theory, but they are unusable in practice because they require manually writing every contract multiple times in multiple languages wherein any discrepancy can cause a loss of assets.
- **Two Participants Only**: Existing State Channels only support two participants in a given state channel. Given that these existing state channels are already limited to "Payments Only", this is not a big additional limitation, as adding more participants may allow somewhat more efficient utilization of capital but still wouldn't otherwise enable new applications.

Chenilles: Next Generation State Channels with Unique Capabilities

We propose a next generation State Channel Network, *Chenilles*, that can lift all the limitations about State Channels, except the one that makes them safe: each channel is limited to the assets locked in the channel and cannot be used to spend tokens not in its trust.

Chenilles has all the upsides of existing State Channel networks, starting with the Bitcoin Lightning Network, as described above: **Scalability** (and therefore **Affordability**), **Privacy**, **Security** and **Network Payments**. It also shares two essential limitations of State Channels: **Only Funds under Custody**, and **One Way or the Server Way**. However, it lifts away the limitations of *Only One Chain*, *Only One Token*, *Payments Only* and *Two Participants Only*. Instead, *Chenilles* is **Multi-Chain**, **Multi-Token**, and **Multi-Participant** all thanks to enabling **Smart Contracts** in addition to Payments. It also supports **Improved Privacy with Hidden Channels**, and **Further Improved Privacy with zk-SNARKs**. Finally, it supports **Nestability**, and some more **Advanced Smart Contracts**.

The rest of this section explains in more detail these superior features of *Chenilles* over existing State Channel networks. You can skip to the next section Applications enabled by Chenilles if you're only interested in the applications that these features make possible.

Smart Contracts using *Glow*

Unlike existing State Channel networks, *Chenilles* supports arbitrary smart contracts.

Some existing State Channel networks allow for some form of smart contracts between participants in theory, but they have limitations that make this feature unusable in practice: writing a smart contract on these systems requires the author to write the exact same contract in two or more completely different languages, such as Solidity for the on-chain code on the one hand, and Go and/or JavaScript for off-chain code on the other hand. Any single discrepancy between those versions, as well as any bug in any of them, puts all assets at risk. This places an extremely low ceiling on how complex applications can be.

Our *Chenilles* system lifts this limitation thanks to its integration with our programming language *Glow*: *Glow* allows programmers to write a decentralized application (DApp) as a single specification in a high-level language, from which the compiler can generate both on-chain and off-chain code. *Glow* is also uniquely designed to facilitate the use of formal methods and eliminate entire classes of complexity and vulnerabilities usually required in smart contracts.

This unique smart contract capability enables *Chenilles* to unlock a host of other capabilities—making them either possible at all, or useful when they wouldn't have been.

Multi-Token

Unlike existing State Channel networks, *Chenilles* supports multiple token classes in a single state channel, both fungible and non-fungible: Ethers, ERC20, ERC721, ERC1155, including wrapped BTCs, stable coins, and NFTs.

Because other systems only support payments, they would not be able to do anything with multiple tokens in a single state channel that they could not do with multiple simpler state channels each holding a single token. But because *Chenilles* supports smart contracts, it can be used to write atomic swaps, futures, exchanges using rates provided by oracles, and all kinds of transactions that make it useful for a state channel to hold multiple tokens at the same time.

Thanks to such contracts, *Chenilles* can also handle Network Payments where one party pays in one kind of token, the other party receives another kind of token. A private contract enables a participant to lock an exchange rate and protect themselves against the “implicit option” issue with cross-currency payments, wherein the other participant will have deposited suitable collateral.

Multi-Chain

Unlike existing State Channel networks, *Chenilles* supports multiple blockchains at once. Not only that, it allows for cross-chain transactions.

Chenilles allows for Network Payments between Ethereum and Bitcoin, and potentially between any two blockchains capable of supporting State Channels. *Chenilles* can also be made to interoperate with the Bitcoin Lightning Network itself, leveraging all its existing liquidity, as well as with any existing State Channel Network of interest.

Chenilles can transfer tokens that are wrapped on another blockchain by a Bridge, such as trading WBTC on Ethereum, at one end of a Network Payment involving BTC on the other end.

But *Chenilles* can also transfer tokens without a bridge, using private swap contracts, so users don't have to trust the security of a Bridge to enable payments across networks.

Multi-Participant

Unlike existing State Channel networks, *Chenilles* allows more than two participants in a single State Channel. This feature not only allows for more efficient use of capital, but also allows for many new kinds of private contracts between more than two participants.

Existing state channel networks, that only support payments, would only be able to support multiple participants in a state channel as a way to ensure slightly more efficient use of capital, at the cost of extra complexity. But since *Chenilles* allows for smart contracts, it is also possible for multiple participants to conduct more elaborate interactions: private auctions, multi-party swaps with many tokens, oracles and bridges, direct NFT trades, etc.

Note that multi-participant state channels should only be used by professionals or dedicated amateurs, who can trust themselves and each other to maintain servers that are always available online, so the capital doesn't end up locked during an exit challenge period. Casual users should stick to one-way two-participant channels with professionals on the other side.

Improved Privacy with Hidden Channels

Unlike existing State Channel networks, *Chenilles* supports hiding channels behind a regular UTXO, such that no outside party needs to even know that a State Channel is being used.

Chenilles by default will secure the assets of a state channel into a regular address or UTXO, controlled by a regular ECDSA signature, indistinguishable from any other regular address. The participants will use an interactive multisig protocol to control this address, and will pre-sign transactions using this address as part of the *Chenilles* protocol. Only in cases of non-cooperative exit and challenge will assets be put under the control of a conspicuous state channel contract, by using one of those pre-signed transactions. As long as the participants cooperate in settling the state channel, the result will look like a regular transaction.

On blockchains like Ethereum with an Account Model, *Chenilles* will provide a contract implementing the UTXO model. Outsiders can then see that a transaction is using the *Chenilles* contract, but can't tell if it is being used for a state channel, or merely for the efficiency or privacy advantages of the UTXO model.

Further Improved Privacy with zk-SNARKs

Unlike existing State Channel networks, *Chenilles* supports hiding payment conditions behind zk-SNARKs. Information is shared on a need-to-know basis, and both parties involved in a transaction must consent to the rules of the interaction.

Chenilles can use *Glow*'s zk-SNARK backend to compile any contract to a condition that is completely opaque to third parties. In case one participant becomes uncooperative, the other participants can then exit the state channel after a challenge period without revealing the details of their dealings. There are legitimate reasons why a participant may become uncooperative: technical difficulties, successful attack on their infrastructure, disease, death, war, etc.

Since these are private contracts, the cheaper kind of zk-SNARKs that rely on trusted setup can be used—wherein the participants run the setup off-chain between each other before they start the on-chain interaction. Even these zk-SNARKs remain relatively expensive on Ethereum, so such enhanced privacy will come at a premium compared to revealing payment conditions. Note that either way, the price paid in either extra gas or loss of privacy is only paid if one party stops cooperating: the normal scenario is for all participants to keep cooperating until the state channel is fully settled and no assets are left in it, at which point no payment condition is revealed, zk-SNARK or not.

Nestability

Unlike existing State Channel Networks, *Chenilles* allows for *nested* state channels: a given state channel can be split into multiple sub-channels, such that multiple conditional payments or contract transactions can take place at the same time.

Existing State Channel Networks can do only one thing at a time. If one participant along a payment path stops cooperating, all the state channels along the path are blocked until complete an exit including a challenge period (typically week-long). *Chenilles* can split a state channel into multiple sub-state-channels, such that failure to cooperate along a payment path only affects the liquidity involved in the transactions lacking cooperation, but do not affect the rest of the state channel. In particular, splitting a sub-state channel from a state channel doesn't require any L1 transaction unless a participant stops cooperating.

Thus, a *Chenilles* state channel participant can be waiting for another participant to act on one transaction, yet still make progress on further ongoing transactions with the same participant or a different one. Participants may rely on conditions to be fulfilled, and still accept transactions on other topics with the same participant or other participants. In particular, one channel can be blocked during a payment along a path, yet the other channels along that path won't be blocked from making progress while waiting for that channel to make progress.

Advanced Smart Contracts

Unlike existing State Channel networks, *Chenilles* supports smart contracts that interact with the blockchain and with other state channel contracts.

Existing state channel networks, when they allow for smart contracts at all, do not allow for those smart contracts to interact with other state channels, with other smart contracts on the blockchain, including oracles, bridges, automated market makers, exchanges, auctions, futures, insurances, etc.

Chenilles allows smart contracts to use information published on the underlying blockchain, or on other suitably bridged systems:

- A payment can be committed based on publishing some secret with a much shorter window than the exit challenge period of the state channel—making it possible to confirm a transaction within hours even though it will take a week to fully settle, even when a participant stops cooperating (if he does cooperate, confirmation can be achieved within a second or so).
- A public auction can be conducted on the basis of conditional state channel transactions based on publishing a code identifying the winner, wherein only participants can see the auction, participants cannot see each other, and on-chain fees are much reduced.
- A futures contract can use a historical price oracle provided either off-chain (via a signed message) or on-chain (via consulting an on-chain contract). The futures contract could even use a complex formula with integrals and derivatives that is checked by a decentralized execution platform such as TrueBit, Fluence, Dfinity, etc.—or verified using a zk-SNARK.

Chenilles contracts can take place along a path of state channels:

- Each participant leaves enough collateral on each state channel along a path to each other participant, such that whichever way the contract interaction goes the other participant is safe. Collateral amounts are negotiated by participants off-chain to make sure everyone is suitably incentivized to complete their part of the interaction even if exchange rates change—as long as volatility doesn't explode beyond mutually anticipated limits during the execution window.
- A master contract on a blockchain that supports smart contracts (e.g. Ethereum) computes the amounts owed by each participant to each other participant.
- Participants then either complete their side of the transaction or lose their collateral that is then used to compensate the other participants.
- Once all participants completed their side of the deal, the master contract releases the collaterals back to the respective participants. If some participants fail to do their part, then after a suitable timeout, the other participants can claim the collateral of the former.
- In a normal execution, all transactions happen via state channels in a matter of seconds. If one participant fails, the other participants may find themselves holding the failing participant's collateral on a different blockchain, and have to use another *Chenilles* transfer or an exchange to receive the value in the token they care about.
- By using *Chenilles* as an intermediary on Ethereum (or another blockchain with smart contracts), one could thus connect state channels on Bitcoin and Monero (both of which lack suitable smart contracts) and execute contracts that involve the exchange of BTC and XMR according to any smart contract conditions that can be specified using *Glow*.

Applications enabled by Chenilles

Chenilles enables all the applications that State Channels do. But Chenilles also enables many kinds of applications that are not affordably possible with existing technologies, State Channels or not. This vastly increases the potential value of the network compared to existing state channel networks.

Payments with automatic currency conversion

Chenilles enables payment of transaction wherein

- a price is agreed upon in any generally accepted reference currency, yet
- the customer pays in whichever currency they want to hold (whether crypto-native token or fiat-indexed stablecoin), and
- the merchant receives whichever currency they want to hold, and
- some intermediaries are automatically found who will offer the best rate after accounting for fees and the customer's and merchant's assessment of execution risk (limited to the usual state channel failure and delay, that never puts the main assets at stake).

By decoupling these three currencies, Chenilles enables individuals and businesses to each keep their accounting in whichever currencies they must legally or contractually use, yet each hold assets in the currencies they believe will best hold value.

Micropayments for metered decentralized utilities

State Channels make it possible for users to automatically pay small amounts for small services based on a metering of how much services they use. Decentralized services can then be provided wherein users pay by volume, get serviced on a commodity market by interchangeable providers who watch each other. Users and providers don't have to build mutual trust relationships, instead only having to trust the overall network. This automated decentralized intermediation can be used to pay for any service that can be verifiably provided over the Internet, including but not limited to: storage, search, computation, advertisement, gaming, etc. Companies and individuals can then sell the "spare cycles" of their hardware, optimizing global resource utilization. In the future, even physical utility providers could also be paid this way for electricity, water, telecommunications, etc., to enjoy the improved efficiency.

Indeed "pay as you go" pricing can be more economically efficient than either "pay before" or "pay after": As compared to users pre-paying for services, micropayments ensure that users don't end up with an unused balance that they forfeit. As compared with users paying for services after the fact, micropayments ensure that providers are not left with unpaid bills by indelicate users that are expensive to go after. Metering also allows for a fair payment of resources between a lot of users. Finally, the lack of any ongoing balance either way means that users and providers do not have to build and maintain a long-term trust relationship; thus there is a low barrier to entry to become either user or provider; no need for burdensome identification forms or vast capitals to immobilize as collateral, or expensive reputation to maintain; just put down in a state channel some amount sufficient to cover service usage, and use a smart contract to ensure payment goes with service rendered.

Of course, users and providers do not have to manually approve each micropayment for each fraction of a cent—the expense in time and mental attention would cost more than the transactions are worth. Instead, users would set soft and hard limits in their user interfaces to be aware of their utility bills and be able to control their expenses without getting bad surprises, but would otherwise let their software handle the payments automatically without user interaction.

Decentralized Banking

A salaryman or contractor can get paid once a month in a L1 transaction, then make lots of small private payments over the State Channel Network over the month. The monthly transaction can also be used for the bank to make a partial settlement with reduced fees.

Chenilles makes this scenario much more likely than the Bitcoin Lightning Network, because it supports holding currencies other than Bitcoin (including fiat-indexed or fiat-backed stablecoins), and payments to be made in any currency rather than only in Bitcoin. Therefore Chenilles does not depend on the world having adopted Bitcoin as a single currency to be useful, unlike the Bitcoin Lightning Network.

Note that this use case, where the bulk of L2 payments are one-way from depositor to bank (and from there to other people), while any significant deposits to the depositor are on L1, remains safe for depositors whose computer fails to connect to the network every week: indeed, since deposits are on L1, the only intermediate L2 transactions the bank may successfully roll back if it tries are those in its favor. In the odd event that a depositor receives money via L2, they have the choice of confirming the payment on L1 before they go offline, if they are concerned about the bank trying to steal that amount.

Private Smart Contracts

Chenilles enables network participants to bind each other through arbitrary smart contracts, in a way that is not only scalable, but completely private: only participants can see that the transactions happen, and non-participants cannot even see that any contract or state channel exists at all and between whom. Even in the fallback case when a participant stops cooperating with the other(s), the conditions attached to ongoing contracts can be hidden behind a zk-SNARK: whether it’s an atomic swap, a futures contract, a poker game, etc.

Private smart contracts can also rely on publicly available data by checking that said data was signed by a suitable oracle committee, or that the data has been included on the blockchain. For instance, a public auction can be run through a series of private conditional payments where the condition is that the bidder is declared as the winner by the seller publishing their mutually agreed random nonce as the winner. No one but the seller needs to know who was the winner for how much—though it is also possible to have the public contract enforce revelation of that information, if desired, e.g. by a clearinghouse listing the auction for a fee.

Business Model For MuKn

There are many ways for MuKn to monetize the *Chenilles* Network.

Big Picture: A layer above blockchains

First, let us make it clear that using or operating *Chenilles* does *not* require introducing a new token, and that this is a great advantage.

Chenilles is blockchain-agnostic—it can work with any blockchain that has scripts or contracts. We are building a layer *above* existing and future blockchains and side-chains, rather than yet another blockchain or side-chain. We can thus connect these blockchains without being impeded as a competitor, rival or enemy, instead being welcome as a facilitator, a friend, a bringer of traffic.

Furthermore, by not requiring users to hold yet another coin and instead letting them pay principal and fees in the coin of their own choosing, *Chenilles* removes friction rather than increases it. Thus, we are drastically reducing the barrier to transferring from one blockchain to the next, and increasing competition between those chains based on their financial fundamentals, effectively commoditizing them.

We are building an over-net that can extract the value from the underlying blockchains, just like the early Microsoft built a layer of languages and operating systems above PC hardware, capturing the value of the PC market while commoditizing the hardware manufacturers.

Now that this big picture has been drawn, here are specific revenue streams that it enables.

Fees on Channel Creation

The *Chenilles* Network is Open Source, and anyone could in theory create their own fork of the *Chenilles* contract. Yet by using the same contract as everyone else:

1. Users can cheaply interact with the liquidity in the existing contract, enjoying network effects.
2. Users don't have to pay the high gas cost of creating their own copy of the contract.
3. Users benefit from improved privacy by putting their assets in a common pool of UTXO, which makes it harder for third parties to identify who is transacting with whom.

Therefore, there is a strong incentive for users to use the existing contract rather than create their own fork. We at MuKn can thus enjoy a first mover advantage and levy a small fee, much smaller than the cost of creating a contract, each time a user creates or settles a state channel, and make money as the network grows in number of users. On existing UTXO networks, such a contract isn't needed, but the default clients could still pay a small fee to a known address upon channel creation, as long as this fee is smaller than the cost of negotiating use of an alternative client with the other state channel participant.

Commissions on Network Payments

Chenilles will be the heart of a worldwide payment protocol from any currency to any currency, including both crypto and fiat currencies, with very low fees. There can even be state channels for the "On Ramp" and "Off Ramp" parts of a payment—though these parts will probably be centralized and require KYC.

Each of the liquidity providers along the path to a payment can earn a small fee, and MuKn can earn a commission on those fees by connecting users to liquidity providers. Even with those fees, the network can be much cheaper than existing centralized payment networks. And even though MuKn only gets a small commission on those small fees—we can disrupt the market, displace the big players of today and expand the already trillion dollar yearly transaction volume by serving populations currently unbanked in the developing world. Existing centralized systems can't compete with our decentralized system in terms of capital, trust and time required to process and settle transactions. They can use regulatory capture to exclude competition in the USA, but there are plenty of countries where the legacy systems are so bad that nothing will stop people from adopting a better alternative.

A DAO for Network Maintenance

The fees collected by the network can themselves be distributed to users according to a DAO, with a corresponding governance token. That governance token can be sold to fund development. Incentives towards early adoption can be provided through yield farming to bootstrap the network effects.

Note that this token is not a utility token and users of the network are not required at any point to purchase or hold the token. It is a pure security token, a way to electronically transact shares of the MuKn subsidiary directly responsible for the maintenance of the network.

Controlling the Default Client

MuKn will control the default client with which most users interact with *Chenilles*. MuKn will then be able to connect developers and users to liquidity providers and value-added services, and earn commissions.

Cryptocurrency users, who value privacy, will presumably be willing to pay fees rather than be served ads; but as we start reaching a broader audience, ads may become an option. And even before then, we can place privileged partners higher on the list of all potential providers offered to our users.

Decentralized Application Development

We at MuKn are uniquely positioned to help others write decentralized applications on top of *Chenilles*, thanks to our expertise and our *Glow* language:

- Until *Glow* is widely adopted, mastering it is a *moat* that gives us a clear advantage over our competition.
- Once *Glow* is widely adopted, being in effective control of its development will give MuKn a lot of opportunities for monetization as we provide access to all the services available through *Glow*.
- Either way, our team can leverage its unique combination of expertise in formal methods, crypto-economics, languages, distributed systems, security and business, to build what other teams cannot because they lack the synergy of simultaneously having experts in each of these many domains.

Thus, MuKn will be able to weave partnerships with all the first movers by helping them develop their contracts, and there again, earn money in fixed amount contracts, or profit shares, etc.

Steering an Ecosystem

As *Chenilles* and *Glow* get widespread adoption, we will not only enable a new class of applications that were extremely costly to build before—we will allow these applications to interoperate, extend and complement each other. MuKn will find itself at the helm of a thriving ecosystem that brings together all decentralized applications on all blockchains.

Unlike existing projects, MuKn does not depend on the success of a new token. We are not trying to create yet another blockchain that would somehow become standard. Quite the contrary, we are building a marketplace that spans existing and future blockchains, abstracts them away, and commoditizes them—just like Microsoft abstracted away the PC hardware with its operating systems and commoditized the hardware vendors. The possibilities are endless, and MuKn will be the Microsoft of Decentralization, the King of anti-Monopoly.

Conclusion: Invest in the Future of DeFi and DeCo

Mutual Knowledge Systems (MuKn) is building the *Chenilles* Network, a next generation State Channels platform that will bring blockchain-based decentralized applications to the masses—not just Decentralized Finance (DeFi), but full Decentralized Commerce (DeCo). The future of blockchain is not just speculative trade between professionals and enthusiasts—it is an everyday payment and contracting tool for everyone.

MuKn is seeking investment to bring *Chenilles* to market and grow it to its full potential in the shortest possible delays. Contact us at contact@mukn.com for further information.

References and Bibliography

Glow website: https://glow-lang.org/
Glow whitepaper: https://bit.ly/GlowWhitepaper2020
Glow GitHub repository: https://github.com/Glow-Lang/glow

Here is a list of relevant scientific publications by past and present direct contributors to our language (in bold), many of them peer-reviewed, demonstrating our expertise in the domain at hand: the design and implementation of programming languages that enable developers to build safe decentralized applications, amenable to formal verification.

François-René Rideau, "AVOUM: Account-View-on-UTXO-Model", whitepaper, 2022.

François-René Rideau, "Durabo: Unstoppable Message Feeds, 2021", whitepaper, 2021.

François-René Rideau, "Simple Formally Verified DApps—and not just Smart Contracts", *EthCC[3], 2020*.

François-René Rideau, "Glow Whitepaper", *2020*.

Jay McCarthy and **François-René Rideau**, "Alacrity: A DSL for Simple, Formally-Verified DApps", *DevCon5*, *2019.*

François-René Rideau, Language Abstraction for [V]erifiable Blockchain Distributed Applications, *IOHK Summit, 2019*.

François-René Rideau, "Composing Contracts without Special Provisions—using Blockchain History". *Hackernoon*, 2019.

François-René Rideau, Binding Blockchains Together With Accountability Through Computability Logic, *LambdaConf 2018*.

François-René Rideau, "Why Developing on Blockchain is Hard? - Part 2: Computing Proper Collateral". *Hackernoon, 2018.*

François-René Rideau, “Why Developing on Blockchain is Hard? - Part 1: Posting Transactions”. *Hackernoon, 2018*.

François-René Rideau, “Legicash: Binding Blockchains Together through Smart Law”. Legicash Whitepaper, 2018.

François-René Rideau, “Climbing Up the Semantic Tower — at Runtime”. *Off the Beaten Track Workshop at POPL, 2018.*