

Overview:

Loom Network is a Platform as a Service (PaaS) that allows Ethereum based Solidity applications to be efficiently deployed and run on public, private and side-chains. The goal is to allow application developers to have smart contracts with enhanced features such as access to more computing power when necessary (or to access the same computing power at lower costs) and free trials for attracting new users. The system also includes the ability to interact with APIs from third parties that are not on chain. For example, Two-Factor Authentication can be achieved by connecting to a 2FA endpoint before calling a contract. Loom's vision is to be the all in one platform that will enable developers to create applications without being forced to adapt to new programming languages and without having to manually manage their servers. Adding to that, deploying their applications to the outside world shall be seamless and simplified to few commands. Like a blockchain version of Heroku and AWS.

The Loom Network core software runs our own semi-public blockchain. Instances of our core software, Tapestries, can be launched by actors who possess our specially issued Licensees. The Loom Network platform is created by the combination of the Tapestries we have launched, in harmony with user hosted Tapestries. Each of these Tapestries will be secured by a variant of Proof of Stake which we call Insurance Policy, more on that later on. Providers who wish to compete on prices are able to create cheaper Tapestries that may have more relaxed insurance policies depending on the use case or for the creation of test environments, proof of concepts, and open source applications. Loom allows smart contract developers to weave together their Decentralized Application (DApp) using a combination of Tapestries to get the desired security and cost, based on their threat model and budget. For example, an Accounting Decentralized Application must require the highest security when interacting with their accounting ledgers, but changing a user's profile picture should not be treated as rigorously. Currently, every blockchain implementation handles all transactions in the same manner, without any kind of prioritization. The Loom Network aims to solve all these issues.

Motivation

As Ethereum scales beyond simple payments between users and evolves to a blockchain that enables DApps which perform complex and timely functions, there is a real lack of infrastructure that addresses the exponentially growing needs of the market. Current solutions and tools include:

1. Scaling solutions that are still in research phase and are not going to be available in the immediate future for commercial use (Sharding, Plasma, Raiden)
2. Infrastructure that provides a limited and/or siloed service that doesn't cover the needs of a commercial-scale user: ChainLink, BlockApps, Infura, etc. There is a need for infrastructure that fulfills the need for services that provide speed, confidentiality and governance on demand while also maintaining the security provided by the Ethereum main network.

Value proposition

When we started Loom Network, we started with a simple question: What kind of infrastructure needs to be in place in order for developers to seamlessly and easily migrate and adapt to blockchain development?

We identified the core issues and built the Loom Network to accomplish these specific goals:

1. **Enable free trials for blockchain applications.** Every transaction on the blockchain includes transaction fees which get translated to extra costs. Users must be allowed to experience free trials by using a DApp's features for a predetermined amount of time.

2. **Coalesce writes to blockchain to save money.** Combine multiple state changes into one. Group transaction and contract calls, effectively reducing transaction fees and improving speed due to the reduced count of transactions.
3. **Integrate 3rd party APIs as Oracles** Blockchains are only aware of data existing inside of their perimeter and currently there is no efficient method to access external world data. Oracles are a way to import needed data from APIs to a blockchain and use them later on in smart contracts, but current solutions are extremely costly. Linking APIs to a contract's function is one of the most sought after features at the moment and we are looking on tackling that problem.
4. **Inexpensive CPU power and Storage for smart contracts.** A developer should be able to expand to vast amounts of compute and storage for affordable prices.
5. **DApp Software Development Kits (SDK)** There is a need for development environments that encompass all the basic services, allowing a developer to get started with development as soon as possible.
6. **Push notifications, SMS.** Push notifications when using blockchain data is another important feature that is missing currently. More than 50% of the internet's traffic stems from mobile devices, the same can be expected for the usage of blockchains (currently mobile applications that interact with blockchains are mainly wallets with no smart contract functionality). We will devote ourselves to bringing you push notifications about blockchain related events to your mobile clients.
7. **Expose rest apis for legacy apps to hit smart contracts.** There is a multi-billion dollar industry based around integrating legacy systems into the existing web: Elemica Network, SAP supplychain, IBM supplychain, etc. The same need exists to create flexible and legacy-aware API endpoints so that legacy apps (the manufacturers, shipping services, suppliers of the world) can integrate seamlessly with blockchain services (decentralized marketplaces, decentralized supply chain).
8. **Host static websites, both on HTTP and on Swarm.** Allow a domain name to be associated Private blockchains, secured by public chain. In order for apps to be serverless and run only on blockchain technology, we need the ability to host static websites seamlessly on swarm and on HTTP.

The Competitive Environment

This year alone, multiple teams have come forth to the blockchain space drumming up support for their Initial Coin Offerings (ICO). However, only a small portion of them are projects related to blockchain infrastructure, and out of these, almost none have a product. There are two broad reasons why this is the case:

1. Many projects' goals focus on a narrow scope for demonstration purposes. This results in them being unusable in practice due to the lack of supporting infrastructure.
2. A team might come up with great ideas but might not have the technical experience to implement their ideas in a reasonable timeframe.

The Market Opportunity

Prior to Loom the founders were part of Ethereum related projects and have first hand encountered significant barriers towards commercial scale deployments. They were encountering issues that were complex and made it impossible for their projects to scale at the desired level.

For example, every transaction on the blockchain costs the user money, and at today's Ethereum prices, prohibitive amounts of money. Even for B2B software, where the client is relatively cost-insensitive, it still makes no sense to start charging before there was ever a chance at a trial. The few projects with working code usually end up deploying it on Ropsten (a Ethereum Test Network), which was never built to handle this purpose nor is it possible to have any reasonable user experience from this. At the time of writing, a transaction on Ropsten takes anywhere between 5

minutes to hours.

Of course, the developers involved in Ethereum are aware of this and proposed Raiden and Plasma and various other scaling solutions. But these solutions are all far away and do nothing to help quench the immediate thirst.

Furthermore, as users get over the excitement of new technology, it is slowly becoming clear that, at least for commercial app developers, the goals solutions like Plasma do not necessarily align with their own. After all, there is always going to be a trade-off between decentralization and cost. So a real opportunity exists for a chain that integrates and seamlessly bridges the best of both blockchain and Traditional application development.

How Loom Network Fits Into the Ecosystem

Loom Network aims to provide the same level of seamless experience that developers are used to from traditional Platforms as a Service such as Heroku and DigitalOcean. Applications which include Solidity code are deployed as a package to a Loom Tapestry of the client's choice. The code can leverage the full set of platform features such as push notifications, caching, cheap compute, cron scheduling and many other tools to be offered in the Loom Marketplace. It's easy to imagine a future blockchain-based social protocol to run on multiple Loom Tapestries while storing data in a seamlessly integrated Graph Database from the marketplace.

Barriers to entry

Ethereum does not yet support WebAssembly and other blockchains that feature smart contracts, such as EOS, are still years away from production. Anyone who wishes to build a PaaS for Ethereum needs to be able to extend Solidity. For a competitor that means that they would have to go over our intellectual property and also tackle the several technical issues and barriers.

Existential threats to the industry

The main competition for Ethereum is not other smart contract enabled blockchains, but rather the colossal enterprises from the traditional world. It is easy to foresee big companies such as Paypal or Stripe rolling out their own private services that out-perform and out-price blockchain-based services. As the concepts and benefits of blockchains reach mainstream adoption, this moment is swiftly approaching. Therefore, the market cannot wait for the existing scaling solutions to be fully implemented, because in all likelihood it will be too late. The mission of Loom Network, is to provide all the infrastructure-level services necessary for DApps to start rolling out to mainstream audiences in the immediate future.

Paving the way to mainstream scale

When Heroku first launched as a way to deploy Ruby-based applications without any worry or knowledge of the underlying infrastructure, the result was nothing like ever seen before. Suddenly developers could deploy production-quality applications orders of magnitude faster than before. If we draw a similar path forward for Decentralized Applications, a platform like Loom is much needed.

The Loom Ecosystem

The Loom Network is not a single isolated service but a living and growing ecosystem. Bottom up, there is the Loom Core that enables all the compute, storage and off-chain interactions. There is a marketplace for first and third party add-on services which we call Loom Weaves. Finally, all of this is encapsulated in a sidechain that can be hosted by Loom or other resellers.

Core Services

The core of Loom Network is a Solidity compatible Virtual Machine that extends the basic Solidity features with native integrations to Loom Core Services. These services(caching, storage, compute) can handle the essential computing and storage needs of an application in order for it to run at commercial speeds and scale.

Loom Marketplace

Loom Marketplace will host Loom Weaves and allow them to be accessible by users and developers. It will also enable users to improve their workflow and their development process. Integrating a Loom Weave into your DApp hosted in a Loom Network side chain will be frictionless and will be similar to modern software marketplaces that improve usability and code integration.

Loom Weaves — Loom Compatible 3rd Party Add-ons

These add-ons are a way to add convenience and significantly improve time-to-market for DApp developers. Some of our planned first-party add-ons are: api2block, cronblock, blockavatar ()

Loom Tapestries — Sidechains that encapsulate the Loom Core with a selection of Weaves

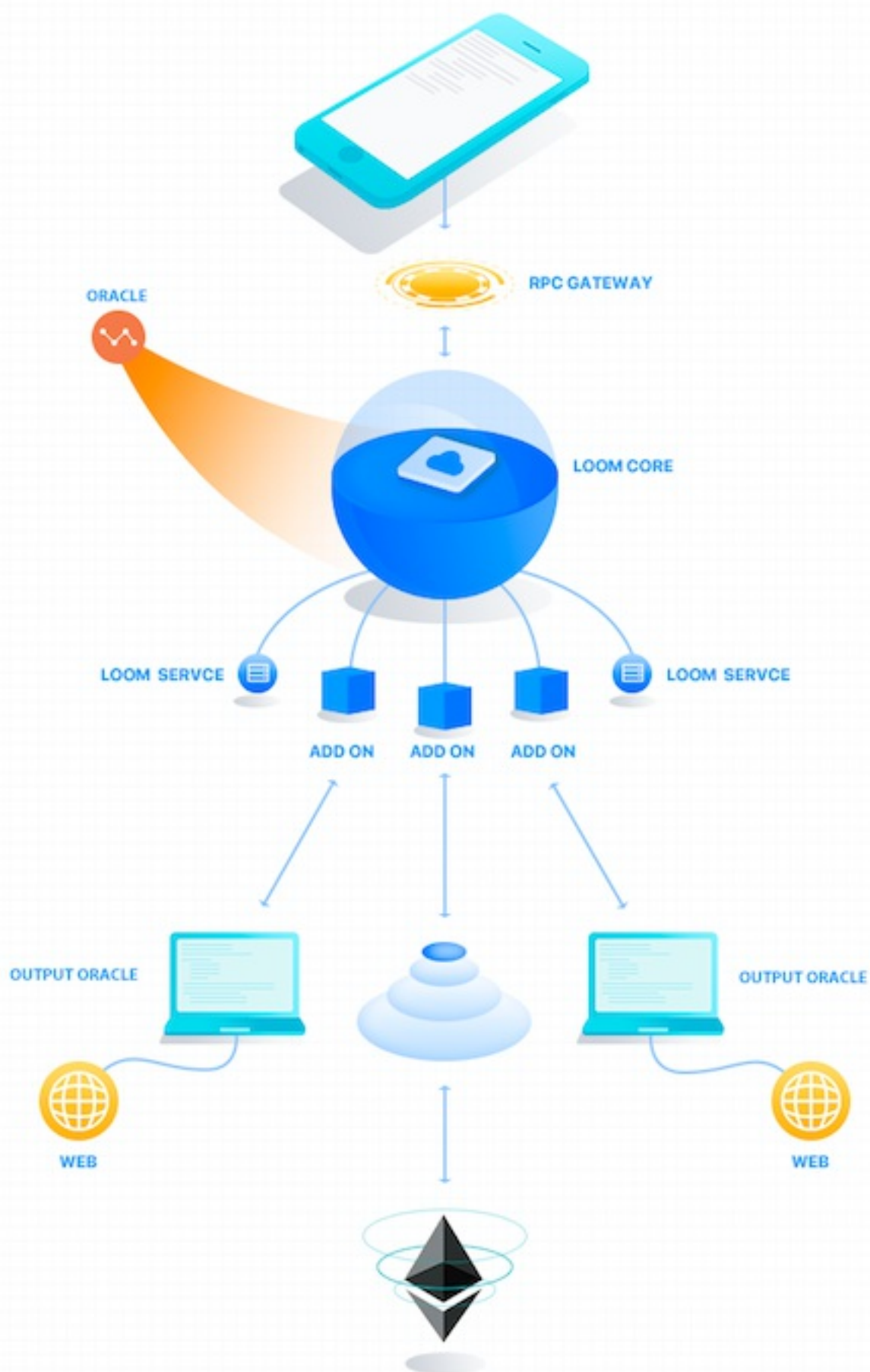
All of the core Loom Services and Weaves can be encapsulated in self-contained side-chain instances. These instances can be hosted by us or anyone with reseller rights. These Tapestry side-chains have service guarantees that are staked by funds, resulting in a set of co-existing sidechains that form the Loom Network. Side-chain providers have full flexibility over their chain's specifications, whether it is extremely secure and redundant with high insurance amount but also high usage costs, or a less secured blockchain that is only used for development testing or continuous integration.

Core Platform Overview

The Correct Technical Trade-off

At the moment, Ethereum's use cases range from decentralized exchanges and marketplaces to decentralized organizations, even to decentralized identity storage. To an application developer, it acts as a distributed state machine, as well a way to do validated compute in a secure and transparent manner. Purely financial applications are a small portion of the use cases the get enabled by Ethereum's features. However, the fact that all transactions are handled with the same utmost level of security restricts the throughput of transactions it is able to process. This extreme level of security can be often redundant and unneeded by applications with more relaxed threat models.

Loom Network runs a side-chain that makes the assumption that all users are simply storing and updating state. Due to reducing the amount of information that is changed, consensus (and thus transaction settling) can be done much faster.



Loom Web Gateway

The Loom Web Gateway is a service that exists at the border between Loom and traditional web APIs. It lets a user to serve data stored on Loom to the Web and provides a simple way for external APIs to send data to Loom via HTTP.

Loom Web3 RPC Gateway

Transparently allows existing web3 html/javascript applications to transition to the Loom ecosystem without changing any code. This can even work without the assistance of Metamask.

Loom Discovery Smart Contract

At any time there are many Loom Network side chains running. This smart contract allows discovery of major Loom networks. It also allows 3rd parties running their own Loom networks to advertise their network.

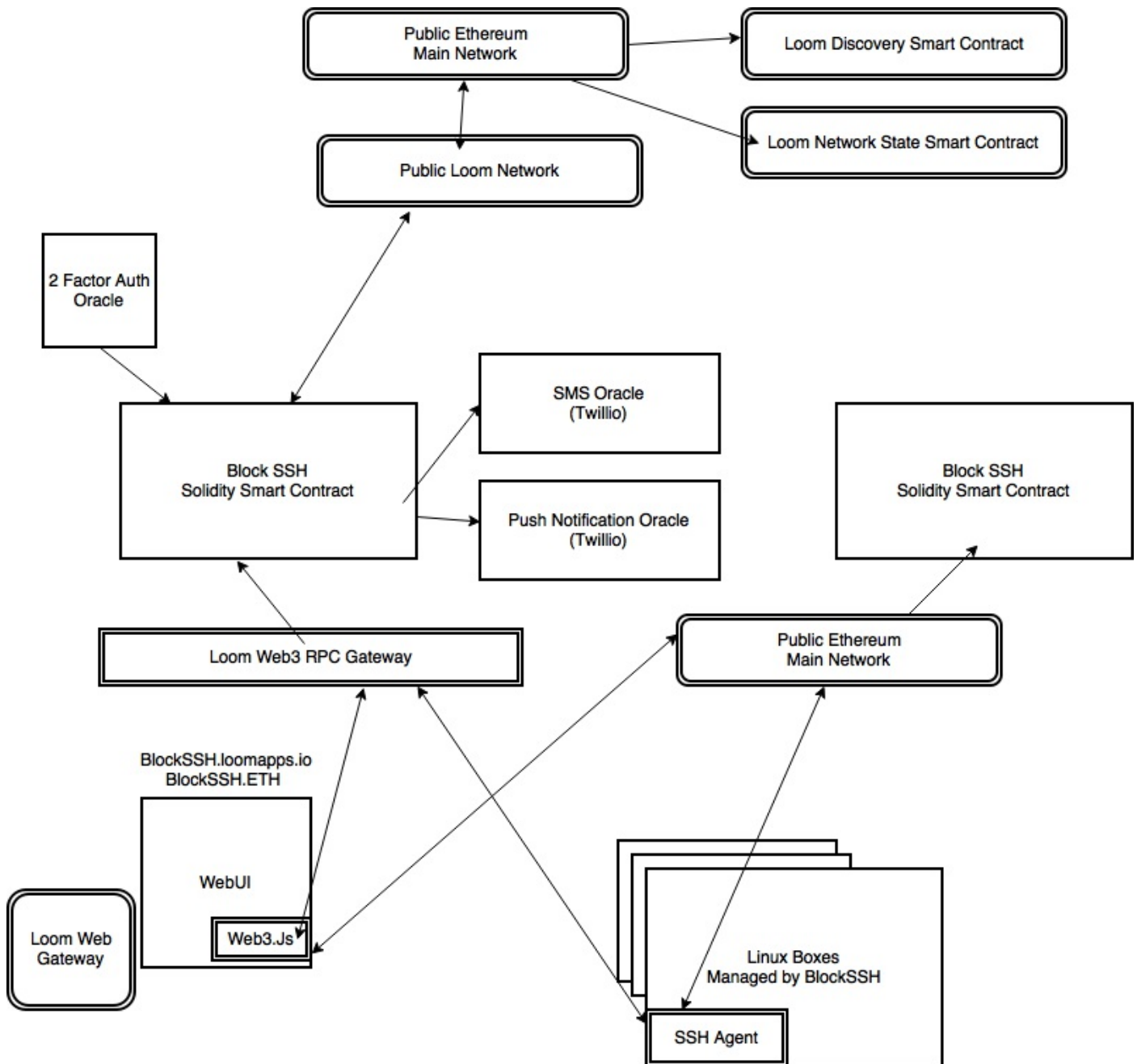
Loom Network State Smart Contract

We secure our side chains by storing hashes of each application's state into the Ethereum blockchain at our Network State Smart Contract. That way if sometime there is a dispute about the state of an application, one can verify that the hash of the application's state at that time is valid.

Public & Private Loom Network

Loom will run two instances of the Loom Network open to the public. One will be a free to use test network. Another one will be a highly reliable production network that app owners pay to access and run apps. Users of applications don't have to pay anything. We will have varying price tiers depending on how much needs to be stored back into the Ethereum chain, and frequency of updates to the public Ethereum chain.

Sample Deployment



Building for Loom and on Top of Loom

Loom will be designed to be intuitive to use, centered around high-quality User Experience. Application data, including Solidity code is deployed via Git. Loom then goes through the build-steps and deploys the application in sandboxed containers. Users pay proportionally with their usage of compute and storage (as well as for any Loom Weaves that they've added). A unified administrator panel allows users to deploy their container on side-chains of their choice. The final cost factors in exactly each chain's contribution.

Platform Marketplace

An easy to use marketplace is available for Loom Weaves. Each Weave is vetted and fully compatible with Loom Core services and should be easily integrated to your application with minimal code changes.

First Party Weaves

We present our first batch of Weaves which we have been building on top of Loom in order to demonstrate the full power and its ease of use:

api2block

This allows end users to pipe together APIs into existing smart contracts with neither side needing to know about the other's state. api2block comes with custom adapters for common APIs like Twillio, Stocks and Salesforce. These API pipes work in both directions, either as inputs or as outputs. That way you can feed Ethereum data into your AI trading bot as easily as storing your Twillio data on chain.

BlockSSH: ssh key management

Store your encrypted SSH keys into the blockchain, so you can easily manage your machines independent of location while also making sure the keys are unable to be lost or stolen. Depending on your governance model, you will be able to manage permissions on your servers when developers join or depart from your organization.

Blockchain 2-Factor Authentication (2FA)

A google-authenticator compatible 2FA generator that authenticates using blockchain Credentials.

Scaling Loom in a Decentralized World

Loom Network is architected in such a way that it can be functional at its initial release while leaving room for continuous improvements over time. We believe blockchain technology must fulfil the increasing demand that projects have for usable infrastructure. We wanted Loom Network to be pragmatic and usable from the beginning.

Weaves

We split our services to Loom Core and Loom Weaves. We believe it is fundamental to our ability to scale, not just in terms of functionality but in terms of the number of use-cases Loom can be applied to. As we can see from existing PaaS providers, applications have vastly different and diverse sets of needs. It isn't practical nor performant for all of these to be included by default into the Loom Core. Also, this incentivizes infrastructure-building teams to build on top of Loom instead of building basic infrastructure by themselves from the ground up.

Tapestries

By choosing to design Loom services as a web of containers that can be encapsulated into a single side-chain, Loom was built for the decentralized and distributed world. By offloading the decentralization to the sidechain level, the Loom Network as an ecosystem can incorporate a significant amount of resilience and security without compromising key performance and cost metrics.

Secured Proof of Uptime

A Proof of Stake variant for insuring sidechain uptime

Smart-Contract enforced Proof of Stake is built into the core of Loom's architecture. It transforms the standard service level agreements (SLA) common in the industry into an automatically enforceable insurance policy. More than that, variations in staked providers makes it possible to use multiple sidechains in parallel to increase redundancy and security at a lower cost than possible through a single provider. Furthermore, it applies the correct economic incentives to *lower* the price of transactions while simultaneously giving users the choice between security and cost. (On the Ethereum blockchain, all things are equally secure, the tradeoff is just between time and cost, which is not a good tradeoff for vast majorities of DApps).

The Naive Proof of Stake Implementation

A naive implementation of staking for sidechains is to hold a certain amount of Ethereum in escrow on a smart contract which is to be distributed to customers in the event of degradation of service or a full-blown failure. However, this incentivizes malicious actors to enlist themselves as customers and then immediately attack the host chain. Furthermore, event-based insurance policies puts the provider at a natural disadvantage because of black-swan events — which is to say, a provider could maintain 99.99% uptime while still have to pay out due to some external event not under its control.

Loom's Staking algorithm tackles these issues in two ways:

1. Customers are subjected to a vesting period. With each additional day of service used, the compensation will increase until it hits a rolling one month window. At which point the customer will receive the full payout proportioned to their invoice if the guaranteed uptime percentage is lower than the promise within the rolling window. (ie. 99.95% uptime in one month, or a 10% payout)
2. Sidechain providers do not only stake Ethereum, they also stake computing and storage resources from other Loom Tapestries. Similar to how banks hold cash reserves but are also free to borrow from the Federal Reserve at the federal lending rate, sidechain providers can choose to stake computing resources from elsewhere on the Loom Network to secure their chain from peak demand.

Multi-layered Staking and the Securitization of SLAs

As good as staking Ethereum is, it's still just the payout that happens after an undesired event occurs. Whether this is data-loss, or service degradation when capacity is exceeded, the payout is still a worst case scenario. This is why Loom tries to structure staking so that it incentivizes securing sidechains first and foremost by purchasing spare capacity from other sidechains.

Once you start to think of computing and storage as a kind of leverage resource, where providers bet on how much customers use the full extent of their allotted capacity, then the concept of staking with capacity really makes sense. This only becomes achievable because of the existence of two things: the shared Loom API and platform, and the immutable Ethereum BlockChain. The stored merkle trees in the blockchain become the arbiter of validity, and the shared API platform becomes the fabric to make code portable.

Loom as a Underwriter of Uptime

Because Loom itself has a natural competitive advantage over Tapestry providers (control of platform, no licensing fees, etc), it becomes unfair and unattractive to Tapestry providers if Loom is directly competing against them. That's why Loom's Services mainly serve two goals: to help onboard customers onto the Loom Network itself (free deployments, small deployments), and to also help *underwrite* sidechain providers.

Loom will sell emergency compute and storage to sidechain providers. This will give sidechain providers access to a

neutral third-party whose interests do not collide and is instead motivated to help them get out of trouble.

Licensing Structure

Unlike many projects built around the sole aim of raising funds without a clear business model and monetization strategy, Loom Network is built around a per-second billing system very much the same as traditional PaaS providers. The core difference between the business structure of a platform like Heroku and Loom Network is that Loom does not run a monopoly on Loom Tapestry instances — instead we encourage participation by spinning up your own Loom Tapestry on your infrastructure.

Our role model is ARM, whose processors powered and enabled the mobile revolution. Instead of fabricating their own chipsets, ARM licensed their designs and instruction set to many chip fabricators. Their varying market positions resulted in the current plethora of ARM chips available — from low-end vendors with little to no documentation and support, all the way to vertically integrated powerhouses like Apple.

The reasoning for this business structure is two fold:

1. We leverage network effects (Metcalfe's law) by empowering partners to spin up their own sidechains. We believe that this way we can incentivize individuals to participate, thus expanding our network.
2. We want to restrict our core-competency around platform innovation and architecture instead of physical server infrastructure. This is because running server farms is a competitive market in an already mature space, so it is far wiser to focus all of our efforts on improving the platform capabilities.

Reseller License Token

With those goals in mind, we've decided to publish the license of our platform software, which comes with the right to spin up sidechains for commercial profit as well as the ability to integrate those sidechains with the Loom Network, as ERC-20 compatible tokens on Ethereum.

Each token represents the lifetime right to run a commercially profitable Loom Tapestry for 1 second per hour (Loom runs on a per second billing schedule for both platform end users and sidechain resellers).

In the future when the demands regularly exceed the capacity of the sidechains, the number of tokens will increase. However, newly minted tokens will always be distributed proportionally to current token owners.

Spinning up side-chains

Loom Tapestries are free to spin up locally for development, and also free for open-source projects (up to certain capacity). Furthermore, the first-party Loom Tapestry is also available for developers to try for free and spin up small test versions of their app. The goal is to encourage blockchain development and accelerate the adoption of Loom Network. Developers will start to get charged when their needs reach commercial scale levels.

There are substantial financial incentives to being a Loom Tapestry operator. By leveraging Loom's features, hosts can offer their services to customers without needing to build their own platform. The system revolves around us releasing open source packages which will need special Licenses for using them to their full extent. These packages will be bought with our Loom token. They will also allow hosts to setup a blockchain that matches their needs, and let them charge their customers in fiat money, Ether or Loom tokens. Loom will maintain a portion of the reseller license tokens issued.

Future Scalability

The entire architecture of the Loom Network is not just built to be truly competitive in the blockchain of the future. Loom Network is scalable on two fronts:

1. By allowing multiple third-parties host their own network, we can work together with them to further expand our capacity
2. Through our Proof of Stake variant, mining is deprecated and thus we avoid potential bottlenecks (miners frontrunning transactions, demanding high transaction fees to validate blocks, 51% attacks)

Funding Goals

TODO

Financial Viability

We expect to run most of the company's expenses by operating the Loom Public Network. We will generate profit by maintaining the public chain, and charging app developers to host there. Alternatively, we can do Proof of stake and validation services for 3rd party Loom networks. There are also ample opportunities for deals regarding enterprise or in-house networks.

Marketing to the Mainstream

We want to be the invisible platform that allows Distributed Applications to be accessible even to the non-technical consumer. That will be achieved by eliminating the troubling points that we experienced during the development of our own previous blockchain applications and by encouraging users to get involved and be part of the network with as little pre-required knowledge as possible.

Team

Matthew Campbell

Matthew just finished up being a Microservices scalability expert at DigitalOcean where he build a distributed Time-series database onto of Kubernetes, Cassandra, and Go. He is writing a book called "Microservices in Go". He has spoke at over 20 international conferences, including GothamGO, Hashicorp Conf, JS Conf, GO India, UK GOlang, MicroXchng, Prometheus Conf. You can see some videos of Matt's previous talks [here](#). Matthew was a founder of Errplane and Langfight. In the past he worked at Thomson Reuters, Bloomberg, Gucci, and Cartoon network.

Luke Zhang

Luke just finished his role as a core software engineer at BlockMason which successfully finished their credit protocol token sale. During his tenure, he noticed just how many critical infrastructure was missing in order for DApps to gain critical mass. His unique background, which includes having worked on a global spanning logistics network that competed with SAP, and his proficiency with cutting edge functional programming gave him the technical basis to join Loom.

James Duffy

James is a serial entrepreneur with a mixed background in online marketing, brand-building, and full-stack web development. His most recent project was an ecommerce brand he started and grew to 7 figures. Prior to that, he wrote a bot that profitably traded cryptocurrencies, he was lead developer at a Korean ecommerce startup that automated overseas ecommerce through mass web scraping and purchase bots, and he created and ran one of the largest job-portal sites in Korea.

Conclusion

Why blockchain Alone Can Never Successfully Scale (Not short term at least)

If one were to sum up the design goals of a blockchain, it is that it given the traditional computing's tradeoffs between security, validity of result, speed and cost, a blockchain will choose to guarantee security and validity no matter how slow things are and how costly it becomes. This has proven to be a wildly successful formula and liberating to developers when there exist trustable and trustworthy storage and compute.

This is also why blockchains' goals currently do not truly align with that of the application developer's. Yet when we see projects that choose to bridge blockchain and off-chain solutions, the vast majority simply choose the naive implementation: to write their code as Smart Contracts on a blockchain and slow their applications to a crawl. The middle ground between on-chain and off-chain shouldn't be pipes that send information back and forth — that just forces everyone to the lowest denominator of blockchain speed and costs.

The blockchain ecosystem can be viewed as a spectrum between absolute security and validity on one end (the current state of Ethereum) and untrustworthy computing on the other (code running on centralized servers not under one's control, but extremely fast. Once one views the blockchain in this way, it becomes obvious that a true bridge should be a platform that can have blockchain-type security and validity at narrower scopes, but which gains reasonable computing speeds.

The Loom Network provides exactly that — modern web service speeds staked by on-chain insurance, all bound together by a single development toolset and environment.

A Tantalizing Future

Imagine it's a few years later, the local car dealer just closed a deal with a new supplier for winter tires. The current ordering system run software built in Web Assembly and deployed on the Public Loom Tapestry. But with the owner of the dealership not fully trusting the new supplier, he decides to randomly take 10% of orders and process them securely through Ethereum at the cost of 50 dollars a month in gas.

So he goes to his app admin panel, one click to deploy a parallel instance on to Ethereum using WebAssembly and EVM, and then goes to the Loom Shuttle router and routes 10% of the data flow to the linked Ethereum Smart Contract. Voila, with a few simple clicks, the entire data pipeline still preserved, the car dealership owner increased the security, validity, and visibility of his transactions in a seamless manner.

FAQ

1. When is public beta We are going to offer a semi private beta, starting November 1. Then opening up to full public with limited capacity by November 30th

Glossary

TODO