



HOLD FOR RELEASE UNTIL MARCH 15th 2018

Blockchain startup Loom Network releases developer platform for building large-scale decentralized apps on Ethereum

Bangkok, Thailand – (March 13th, 2018) – Loom Network, a Techstars-backed blockchain startup that just raised \$25M in funding, has released the equivalent of Amazon Web Services for blockchain apps — a next-generation blockchain platform that allows developers to run scalable, decentralized, censorship-free apps and games directly on the blockchain.

Blockchain technology has received a lot of attention in recent years, but most of this has been focused on its usage with decentralized currencies, such as Bitcoin. But Loom Network has been focusing on a different use-case: running entire applications on their own blockchains.

James Duffy, Co-Founder of Loom Network explains: “With Bitcoin came the world’s first Blockchain, a decentralized ledger of financial transactions that cannot be counterfeited, forged, or modified. Ethereum took it a step further by allowing developers to upload indelible pieces of code into the Blockchain, called ‘smart contracts’, that anyone could execute by paying a small fee. Loom Network is the next logical step in this evolution — it allows developers to deploy their entire application on a blockchain. We’re allowing developers to build unstoppable applications that can never be shut down or censored, because they’re hosted and controlled by the same communities that use them.”

Background: Data Ownership & Web 2.0

With traditional Web 2.0 applications, the data generated by users is owned by the company that runs the service. This has led to a Web where the majority of user-created content is owned by a few giant corporations who control access to it. Facebook and Twitter sell your data to advertisers, and there’s only one way you can interact with your news feed, decided by the latest algorithm released by their developers (which is designed to maximize their revenue, not necessarily user experience). If you don’t like a change to the service, you’re often locked into the platform because your data is held hostage and the cost of switching is so high — you have no choice but to deal with it, or quit and lose the years worth of posts and interactions you’ve built up.

And while it’s hard to imagine with a company the scale of Facebook or Twitter, free services can also shut down. Years ago, a microblogging service called Posterous was used by millions of users. However, when it was acquired by Twitter and subsequently shut down a year later, these users were forced to migrate their years of post history elsewhere and find a new blogging service. Luckily the company provided an export function that made it easy for the users to port their data, but this isn’t always the case.

The Future: Web 3.0 & Decentralized Applications

Applications built on Loom Network promise something different. Loom runs Decentralized Applications (DApps) on their own blockchains, called DAppChains. And on a DAppChain, there is no one centralized server storing all the data.

Instead data is replicated to hundreds or thousands of servers, called “nodes”, all running the same software. As with Bitcoin, these nodes use special consensus algorithms to come to agreement over changes to data, where the majority rules — whatever 50% or more of the nodes agree on becomes the new state of the DAppChain. This means users can’t be censored or banned from these services unless more than 50% of the nodes agree to do so, in a kind of digital democracy for Internet applications.

Since DAppChain developers can choose to allow anyone to run a node on the network, that means users can directly support their favorite games, services, and applications by running a node to help host them. With user-run nodes, a service can never be shut down as long as at least one user is still willing to run a node, similar to users on Bittorrent seeding a file. And like Bittorrent, Loom Network makes it easy to run these nodes, even for non-technical users, turning these DApps into what Duffy calls “unstoppable applications.”

Since nodes must come to a majority consensus on the data, this also gives nodes the ability to “vote” on changes to the software running the service. If the developers release a software update that drastically changes the user experience and users don’t like it, the nodes have the option to keep running the old version of the software. If more than 50% of the nodes reject the update, the network will continue running the old version as if nothing happened. And even if only a small percentage of nodes and users don’t like the update, they have the option to “hard fork” — which means they split off from the network and keep running the old version, keeping an exact copy of the data from the beginning of time. The community would split into two functioning networks, each with its users running the service exactly the way they like it.

Duffy explains, “I foresee a future where the Internet is built up of large-scale decentralized applications that are community-owned, forkable, transparent, and immune to censorship or control from a single party. Changes to the code are voted on by community members, so once a developer releases an application to the world, it takes on a life of its own and he no longer has total control over it. Applications become like living, breathing entities. There’s a kind of organic freedom there that’s almost poetic.”

This data being openly available also means that third-party developers can build their own DApps that interact with the data in unique ways. Instead of only interacting with your news feed in the way Facebook decides you should, in a DAppChain-based social network, there could be dozens of interfaces to choose from that curate and display this data differently. Your experience could be completely customized and tailored to your needs. This actually used to be the case with Twitter years ago, until they shut down their public API in 2015, turning it into a paid service only for enterprise companies. With a DAppChain, users would never have to worry about this happening, since the data is open by nature of the blockchain protocol.

You could even have DAppChains that read and interact with the data from other DAppChains. Rather than the Internet consisting of silos of data owned by disparate corporations, a DAppChain-based Internet would consist of data that is fully transparent, interconnected, and accessible by all parties. This would be a shift so massive that some are calling it ‘Internet 3.0’.

A Network of DAppChains Connected by Ethereum

There have already been a few examples of successful blockchain-based applications. One of the most well-known is [Steemit](#), a blogging and content curation website that has built a community of 450,000 people. But Steemit is built on its own blockchain technology, which makes it hard for ordinary developers to replicate — it would be a lot of work for developers to code their own blockchain from scratch, and the nascent blockchain would be vulnerable to attacks by malicious users until it had acquired a sufficient number of nodes to support it.

Loom Network makes it simple for developers to build their own Steemit-like DApps by providing an SDK that acts as a “DAppChain generator”. Developers can simply focus on coding the application logic, while Loom’s SDK automatically generates the underlying Blockchain logic for them.

These DAppChains are also secured by Ethereum, the world’s most decentralized public blockchain. DAppChains run in parallel to Ethereum smart contracts, which means developers can utilize Ethereum for the parts of the DApp requiring the highest security, thus offering these DAppChains better protections against malicious users than is possible with standalone blockchains.

Running these DApps as their own DAppChains also provides far greater scalability than running the entire DApp on Ethereum itself. Currently DApps on Ethereum suffer from high costs and low speeds, as they are competing with all other DApps on the network for the same limited computing resources. This results in high costs and slow speeds for even the most basic DApps.

CryptoKitties, the wildly popular game that runs on the Ethereum blockchain, brought Ethereum to its knees at its peak — leaving many normal transactions stuck for days due to unexpected network congestion. Loom Network’s DAppChains allow for much greater scalability. With Loom Network, you could run a million CryptoKitties simultaneously without having to worry about network congestion or interrupting other important transactions, all while still providing the security benefits of letting users store their digital cats permanently on Ethereum.

Loom’s SDK gives developers the option to choose the consensus algorithm underlying their DAppChain, which provides DApps the flexibility to customize their ruleset to optimize for performance and scalability or higher security. At the beginning, they plan to offer a version DPoS (Delegated Proof of Stake), a consensus algorithm that provides a high level of scalability but at the cost of higher centralization, and later offer POS (Proof of Stake) algorithms from Tendermint and Ethereum’s Casper as those projects become available.

The SDK then generates a Ruby-on-Rails-like scaffolding for the application, which the developer can build their app on top of. Developers can simply focus on building their apps, without needing to build their own blockchains. The idea is to make it as simple for developers to build Web 3.0 blockchain apps as it currently is for them to build vanilla Web 2.0 apps.

The First DApps Available on Loom Network

To demonstrate the type of DApps that can be built on their platform, Loom Network recently released [DelegateCall](#), a social question-and-answer site for developers that runs entirely on a DAppChain.

On DelegateCall, users can ask and answer questions about blockchains and DApp development. When questions and answers are upvoted by other users, rather than simply earning useless karma points, they can exchange their karma for an ERC20 token on Ethereum, which can be sold for real money. Thus DelegateCall financially incentivizes users to contribute to the site and provide the highest quality answers.

Since DelegateCall runs on its own DAppChain, users can use Loom Network's Block Explorer to watch the underlying transactions as they occur on the blockchain. All of the posts, comments, and upvotes that happen within the system are openly readable by anyone, providing a level of open data transparency not available on traditional applications.

To convert their karma points into tradable and spendable ERC20 tokens, users must first have a LOOM token — purchasable on the Loom Network website for \$1.99, or for cheaper on third-party exchanges. The LOOM token acts as a membership license to all DApps running on the Loom Network, so once you have one you can use it for life on any DApps running on Loom in the future. The token allows you to transfer your assets from a Loom DAppChain to Ethereum, or between DAppChains, providing users with true ownership of digital assets.

The Loom Network team is also building 3 games in-house that will run on their own DAppChains, to be released over the next few months. Themed around their popular educational platform [CryptoZombies.io](https://cryptozombies.io) (an interactive academy that teaches users to program their own Ethereum DApps), these games are meant to showcase some of the functionality of blockchain-based games that aren't possible in traditional games.

CryptoZombies Battlegrounds is a tradable card game similar to Hearthstone or Magic The Gathering, but where digital ownership of cards is tracked on the blockchain. This means players will be free to buy, sell, and trade their cards as they please without needing to go through the game's server. Like any other blockchain-based asset, your cards 100% belong to you, and no third party can take them away by banning you from the service, or prevent you from selling them to another user.

CryptoZombies Rancher is a digital collectible game like Pokemon, where you collect, train, upgrade, and battle your zombies against other players' zombies. Just like *CryptoZombies Battlegrounds*, these zombies exist as digital assets on the blockchain that 100% belong to you even outside the game world, and are tradable and resellable between players.

Loom is also working with multiple third-party developers who are building out additional games and apps on their platform, and are planning to launch more educational resources around DApp development to further increase the number of developers building blockchain apps in the future.

About Loom Network

Loom Network is a next-generation blockchain platform that allows developers to easily build and deploy decentralized, scalable, and censorship-free games and social applications on the blockchain. (Think World of Warcraft or Twitter on the blockchain).

Loom Network was founded by Matthew Campbell, Luke Zhang, and James Duffy. in 2017. Matthew Campbell was a systems scalability expert at Digital Ocean, Thomson Reuters, Bloomberg, and has spoken at over 20 developer conferences such as microXchg & Gophercon. Luke Zhang was the lead developer at Blockmason (another blockchain startup focused on decentralized credit), and previously built a global supply-chain network. James Duffy is a marketer and entrepreneur who built a 7-figure ecommerce company and was previously lead developer a Korean ecommerce startup.

Press Contact:

Dilanka McLovin
dilanka@loomx.io

Media Files:

v.0.2.1 Alpha-Bui
0 Updates Availa

DASHBOARD

BLOCKEXPLORER

Blocks

Transactions

LOCAL DEVELOPMENT

DAPPCHAINS

Templates

Make Your Own

DELEGATES

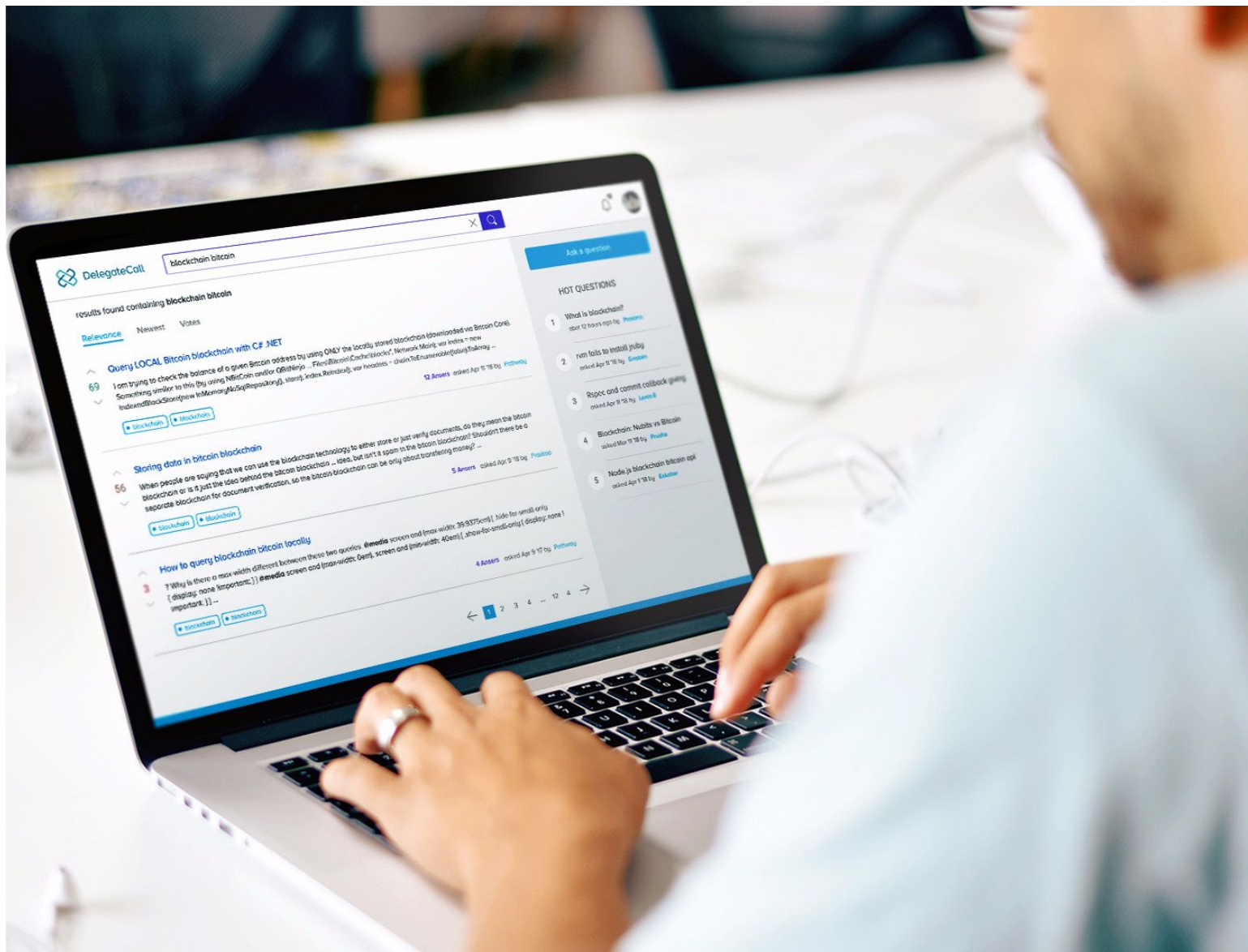
SETTINGS

DOCUMENTATION

Blocks

Filter by Most Recent

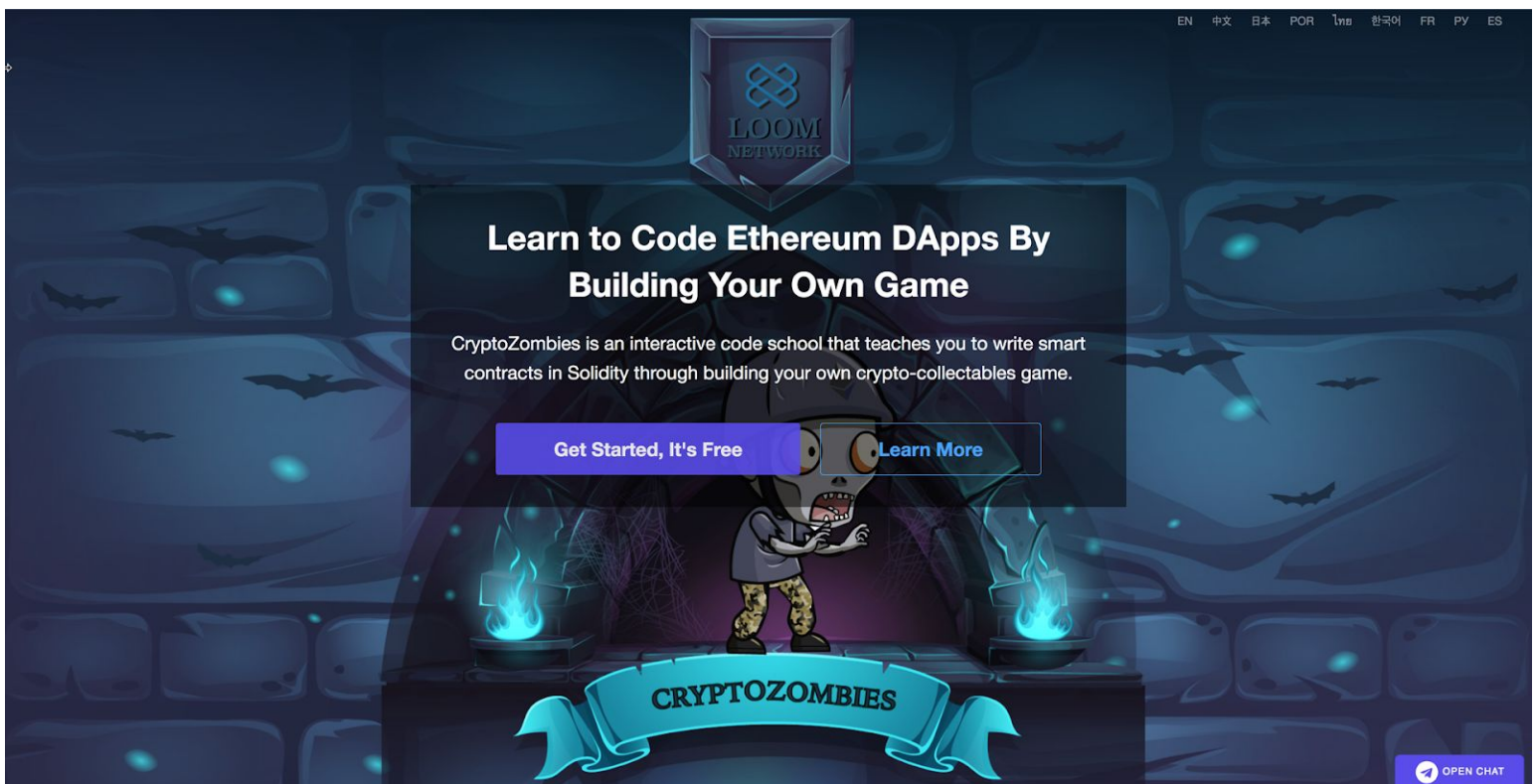
Block	Txn	Age	Speed	
#968	477 txs processed	> 1 minute ago	35ms Via Node #33	
#967	21 txs processed	> 1 minute ago	21ms Via Node #33	
#966	2 txs processed	> 1 minute ago	1ms Via Node #33	
#965	31 txs procssed	> 1 minute ago	2ms Via Node #33	
#964	36 txs procssed	> 1 minute ago	15ms Via Node #33	
#963	15 txs processed	> 1 minute ago	21ms Via Node #5	
#962	366 txs processed	> 1 minute ago	37ms Via Node #3	
#968	477 txs processed	> 1 minute ago	35ms Via Node #33	



DelegateCall.com, the first DApp launched on Loom Network



Early demo screen from “CryptoZombies Rancher”



Screenshot of CryptoZombies.io homepage

CryptoZombies

Resume Progress

EN 中文 POR 日本 ไทย 한국어 FR RU ES Sign Out

Chapter 13: Events

Our contract is almost finished! Now let's add an **event**.

Events are a way for your contract to communicate that something happened on the blockchain to your app front-end, which can be 'listening' for certain events and take action when they happen.

Example:

```
// declare the event
event IntegersAdded(uint x, uint y, uint result);

function add(uint _x, uint _y) public {
  uint result = _x + _y;
  // fire an event to let the app know the function was called:
  IntegersAdded(_x, _y, result);
  return result;
}
```

Your app front-end could then listen for the event. A javascript implementation would look something like:

```
YourContract.IntegersAdded(function(error, result) {
  // do something with result
})
```

Put it to the test

We want an event to let our front-end know every time a new zombie was created, so the app can display it.

1. Declare an **event** called **NewZombie**. It should pass **zombieId** (a **uint**), **name** (a **string**), and **dna** (a **uint**).
2. Modify the **_createZombie** function to fire the **NewZombie** event after adding the new **Zombie** to our **zombies** array.
3. You're going to need the zombie's **id**. **array.push()** returns a **uint** of the new length of the array and since the first item in an array has index **0**, **array.push()** will be the index of the zombie.

Check Answer

Contract.sol

```
1 pragma solidity ^0.4.19;
2
3 contract ZombieFactory {
4
5   // declare our event here
6
7   uint dnaDigits = 16;
8   uint dnaModulus = 10 ** dnaDigits;
9
10  struct Zombie {
11    string name;
12    uint dna;
13  }
14
15  Zombie[] public zombies;
16
17  function _createZombie(string _name, uint _dna) private {
18    zombies.push(Zombie(_name, _dna));
19    // and fire it here
20  }
21
22  function _generateRandomDna(string _str) private view returns (uint) {
23    uint rand = uint(keccak256(_str));
24    return rand % dnaModulus;
25  }
26
27  function createRandomZombie(string _name) public {
```

Hints

OPEN CHAT

Screenshot of CryptoZombies.io lessons