LaChain

Overview

Blockchains have the potential to improve people's lives, especially in emerging markets like Latin America, where inflation, low banking, and poor institutions are common. LaChain is here to make this potential a reality by providing high-quality, low-cost blockchain services and infrastructure that enable developers to deploy dApps with dedicated blockspace for use cases important to the region, like settling local stablecoin transactions, DID, local exchanges settlements and others.

What is LaChain?

LaChain is a blockchain that was designed to tackle the urgent economic problems that are common in Latin America. The platform is built to host use cases that are specially crafted to meet the particular difficulties that local businesses and individuals face.

Why LaChain?

The blockchain technology is ready to take off as an internet-like network of interconnected networks. A large quantity of block space is required to achieve scalability, and through an organic process called sharding, blockchains can develop a complex web of interconnected spaces to help with their scalability. With the arrival of LaChain, Latin America, an emerging market, justifies having its own dedicated block space. This will allow the region to utilize the advantages of blockchain technology more effectively and in a way that is tailored to regional demands.

Motivation

We offer our customers a unique value proposition: high-quality, low-cost blockchain services and infrastructure that enable projects to deploy dApps, test products before launching, with good performance, fast speeds, and low fees.

Our network is Latam-oriented, giving projects in the region the opportunity to take advantage of our secure, reliable, and efficient services. As an active validator, you will be rewarded with gas fees and incentives from staking LaChain native currency.

We want to continue to be protagonists of this cultural change by promoting blockchain as the basis of this revolution. We believe that Latin America is fertile ground to be able to carry out this change due to the well-known problems in the region. Inflation, the excessive regulations of the Latin American Central Banks, among other things, make this possible. We want to offer not only a blockchain designed to solve these problems, but also an ecosystem that benefits Latin American users, improving their finances, offering new experiences that put them as the protagonist, owning their assets.

Governance

Blockchain Consortium

LaChain is a blockchain made available by a consortium made up of companies with experience in the crypto industry. The global objective of these companies is to democratize and expand the use of blockchain technologies to users in Latin America and the world. All together keep active and rule LaChain.

https://ripio.com/

Ripio is one of the largest crypto asset platforms in Latin America, with more than 3 million users and more than 200 million in transaction volume per month. It is possible to exchange bitcoin and other cryptocurrencies for national currency.

SenseiNode is focused on generating opportunities through crypto services. Their enable organizations and entrepreneurs to build their products on the blockchain and participate in staking on multiple PoS protocols from a regional source; fostering connection, decentralization and inclusion.

https://senseinode.com/

---

https://num.finance/

Num Finance builds stablecoins of Latin American coins to facilitate access to cryptocurrencies and decentralized finances to millions of people and companies. NUM-S expand cases of use of cryptocurrencies through access to credit, rewards and new means of payment.

Cedalio creates a solution for developers where they only have to define their data model and the business logic they want to decentralize, without having to worry about storage management or smart contract coding.

https://cedalio.com/

https://buenbit.com/

Buenbit is a cryptocurrency exchange and a cross-border payment platform through which users can buy, sell, and invest cryptocurrency. They are based in Argentina and also cater to the Mexican and Peruvian markets.

Foxbit is one of the largest and oldest cryptocurrency exchanges in the world, operating in Brazil with trading in bitcoin, ether, litecoin, TrueUSD and XRP. Since 2014, it seeks to inspire people to gain financial freedom so that they can realize their dreams.

https://foxbit.com.br/

Consensus Mechanism

The Clique is the consensus mechanism in LaChain. In PoA, validators are the ones responsible for creating the blocks and adding them to the blockchain in a series.

All of the validators take turns in proposing the next block (round-robin), and for the block to be validated/inserted in the blockchain, a supermajority (more than 2/3) of the validators must approve the said block.

The plan for the future is to move to Proof of Stake (PoS).

About LaChain

LaChain uses Geth

LaChain is a blockchain based on Geth, the technology behind Ethereum, which makes it secure, fast, and reliable.

The choice of Geth as the technology for LaChain is based on accompanying the latest trends in the Blockchain industry. In this way, LaChain provides its users with an enhanced experience while also allowing them to benefit from future developments applied to major networks such as Ethereum.

Ethereum Virtual Machine

LaChain is a permissionless public community driven blockchain, it boosts smart contract functionality and compatibility with the Ethereum Virtual Machine (EVM). The design goal here was introducing smart contracts into the Latam region ecosystem and building tools to promote dApps integration with blockchain.

Because LaChain is EVM-compatible, it will be launched with support for the rich universe of Ethereum tools and DApps. This makes it easy for developers to port their projects over from Ethereum. For users, it means that applications like MetaMask and many others can be easily configured to work with LaChain. It is just a matter of tweaking a couple of settings.

Nodes

The nodes use Geth v1.13.12.

Connect to LaChain:

Network Name	LaChain
RPC URLs	https://rpc1.mainnet.lachain.network https://rpc2.mainnet.lachain.network https://lachain.rpc-nodes.cedalio.dev
Chain Id	274
Currency Symbol	LAC
Block Explorer URL	https://explorer.lachain.network

Block Explorer

LaChain has an Explorer where you can view blocks, transactions, and blockchain network metrics.
https://explorer.lachain.network

Tokenomics

Native Token

The native token for LaChain is LaCoin (LAC). The user pays the fees related to the requested transactions using LaCoin. This is not an inflationary token, there is a fixed supply living on LaChain.

Name	LaCoin
Symbol	LAC
Total Supply	10,000,000,000 LAC
Icon	

LaCoin can be purchased at:

- Ripio: https://www.ripio.com/
- Ripio Trade: https://trade.ripio.com/market/market-out?pairCode=LACUSDC
- BuenBit: https://buenbit.com/

It will soon be listed on other Exchanges.

GAS Fee

Because LaChain works with the London fork activated (EIP1559), each transaction works with the improved GAS fee scheme. This allows network congestion to be managed more efficiently and operations on LaChain to be affordable in terms of costs.

About LaTestnet

Nodes

Connect to LaTestnet:

Network Name	LaTestnet
RPC URL	https://rpc.testnet.lachain.network
Chain Id	418

Currency Symbol	TLA
Block Explorer URL	https://testexplorer.lachain.network

Native Token

The native token for LaTestnet is LaTest Coin (TLA). This is a test token that can be obtained in a faucet. The total supply was minted at:

0xF637B10647e2D0FD00C6a0b70D9e85Bf6EA0327f

Faucet

The Testnet has a Faucet that grants the possibility of obtaining La Test Coin to be able to pay the transactions in order to test smart contracts and use cases in the development stage.

Faucet is serving from 0x483c99ED4453e2687468d95A176aed45AF7469ba

https://faucet.lachain.network

Development

Deploy a Smart Contract

To deploy a smart contract on LaChain, a custom blockchain network, using Hardhat, an Ethereum development environment. This section will cover setting up Hardhat, creating a simple smart contract, compiling the contract, testing the contract, and deploying the contract on LaChain.

Create an npm project and install Hardhat:

```
npm install --save-dev hardhat
```

To create a simple smart contract for deployment on LaChain, start by creating a Hardhat project by using:

```
npx hardhat
```

This will show you a prompt where you can select your favorite language projects based on your needs. Choose 1 project and go through these steps to compile, test and deploy the sample contract.

In the contracts folder, you'll find Lock.sol, which is a sample contract that consists of a simple digital lock. With this contract, users can only withdraw funds after a set period of time. Feel free to add, modify, or update the contracts according to your needs.

To configure Hardhat for LaChain, update the hardhat.config.js file with the LaChain network credentials and create a .env file to store sensitive information like private keys and API keys.

```
require('dotenv').config();

require("@nomiclabs/hardhat-ethers"); require("@nomiclabs/hardhat-etherscan");

module.exports = {

 defaultNetwork: "lachain",

 networks: {

  hardhat: { },

  lachain: {

   url: "https://rpc1.mainnet.lachain.network",

   accounts: [process.env.PRIVATE_KEY]

  }

 },

 solidity: {

  version: "0.8.9",

  settings: {

   optimizer: {

    enabled: true,

    runs: 200

   }

  }

 },

}
```

To compile the smart contract, first install the Hardhat Toolbox with the command:

```
npm install --save-dev @nomicfoundation/hardhat-toolbox
```

and then run the command:

```
npx hardhat compile
```

To test the smart contract, run the command:

```
npx hardhat test
```

This will execute the tests in the test directory of your Hardhat project. To deploy the smart contract on LaChain, execute the deployment script with the command:

```
npx hardhat run scripts/deploy.js --network <lachain>
```

Replacing <lachain> with the name of the LaChain network you configured in the hardhat.config.js file. After the deployment, you can verify the deployment status on LaChain's block explorer if it's available.

To interact with the deployed smart contract on LaChain, you can use provided tools such as Remix, Truffle or you can create custom scripts using Hardhat and Ethers.js to interact with the contract programmatically. This allows you to read data from the contract, send transactions, and perform other operations as needed.

Deploy with Remix

As with all EVMs, it is possible to implement a smart contract with Remix. Remix Online IDE is a powerful toolset for developing, deploying, debugging, and testing Ethereum and EVM-compatible smart contracts.

It requires no setup and has a flexible, intuitive user interface. There is no need for any local environment settings for deploying solidity smart contracts on LaChain. You only need to have a browser-based Web3 to be able to deploy.

Steps:

1. Go to https://remix.ethereum.org/
2. Select the compiler version in the Solidity compiler page. In this example we use 0.8.18.
3. Write smart contract:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.18;
contract HelloWorld {
   function helloWorld() public pure returns (string memory) {
      return "Hello World!";
   }
}
```

4. Select "HelloWorld" from the contract drop-down menu.
5. Connect your browser-based Web3 wallet to LaChain or LaTestnet. The selected address must have funds in native coin to be able to perform the deploy.
6. Select Injected Provider in the Environment dropdown into Deployment Page. Confirm connection request.
7. Press on the "Deploy" button which would generate another wallet popup that requires transaction confirmation. Done!

Verify Smart Contract

To verify a smart contract in Explorer:

1. After creating a contract, you will receive an address. Go to LaChain Explorer (https://explorer.lachain.network) and enter the contract address in the search bar.

2. Once you find your contract details, select the "Code" tab to view the bytecode. There, you'll see a button called "Verify & Publish." Click on it.

3. Several verification options will be offered. Two common methods are:

- Flattened Source Code (Solidity): Must provide contract address (prefixed with "0x"), contract class name, compiler version, EVM version, if optimization was enabled during compile , the number of optimization runs, the contract code in Solidity, and other optional details as needed. Then click the "Verify and Publish" button.

- Standard JSON Input: In this method, you will provide the name of the contract, the compiler version, a standard input JSON file (following the Solidity format), and other optional details. After filling out the form, click the "Verify & Publish" button.

4. Wait for the response after clicking "Verify & Publish". If all goes well, you will see a check mark next to "Code" in the code tab and an additional tab called "Read Contract" is added to access your verified contract.

Roadmap

- ➔ Testnet launch (March 2023)
- ➔ Explorer and Faucet launch (March 2023)
- ➔ Mainnet launch (June 2023)
- ➔ Bridge (March 2024)
- ➔ DEX (March 2024)
- ➔ Enable PoS (TBD)
- ➔ And more use cases integration (2024)