Documentation Index

Fetch the complete documentation index at: /llms.txt

Use this file to discover all available pages before exploring further.

Learn

- Introduction
- Towns Smart Contracts
- Towns Messaging Protocol
- Token Governance
- TOWNS Token
- Grants

CRYP2NOVA

Build

- Messaging Data Structures
- Delegate Key Signing
- Towns Encryption Protocol
- React SDK
- Bots
- Mini Apps

Node Operator

Archived copy · cryp2nova.com

White Paper

- Towns Technical Overview
- Towns Protocol Technical Whitepaper
- MiCA Whitepaper

Introduction

Copy pageCopy page
Copy pageCopy page

What is Towns Protocol?

Towns Protocol is an open source protocol for building decentralized real-time messaging apps. It consists of an EVM-compatible L2 chain, decentralized off-chain stream nodes, and smart contracts that are deployed on Base. Towns allows people to create programmable communication use cases, referred to as “Spaces,” in a permissionless manner. These Spaces are ownable, feature on-chain subscriptions (”Memberships”), an extendable reputation system, and end-to-end message encryption.

Purpose and Vision

At its core, the ecosystem seeks to empower people to create, manage, and participate in digital communities in a secure and permissionless manner. The primary goal is to provide a robust, secure, and decentralized platform where people have complete control over their data, privacy, and engagement in these digital spaces, while protecting their reputation. The vision extends to fostering an era of digital social that is both user-centric and community-driven, leveraging blockchain technology to ensure trust and security. Towns’ goal is ambitious. To secure freedom of communication for a sustainable permissionless future.

Archived copy · cryp2nova.com

Key features

An app chain purposefully built for social networking – Towns secures Read/write entitlements separately on Base, allowing our app chain to make liveliness tradeoffs and send messages to thousands of participants as fast as any existing centralized social network. **Ownable communication** – Space creators truly own the space they created as an onchain asset. **Programmable spaces** – Spaces are deployed onchain with programmable interfaces allowing the rules, like who can read and write, to be customized to integrate with any other external EVM-compatible contract. **On-chain memberships** with built-in protocol fees – Users are required to hold a valid membership token to send and receive messages in a Space. Paid memberships enforce a protocol fee used to pay for the operation of the network, while free Spaces do not incur protocol fees. **On-chain social graph** – Membership tokens and Spaces are discoverable on chain. **Extendable reputation system** – Towns's communication spaces can be programmed to allow members peer-based, Space-specific ratings discoverable on chain. **End-to-end encrypted messaging** - Ensures secure, private communication with advanced encryption, protecting messages between sender and entitled users.

High-Level Component Summary

1. **Towns Smart Contracts**: Deployed on the Base Mainnet Chain, an Ethereum Layer 2 platform, these contracts enable users to create and manage their programmable communication space. Each space is minted as a unique entity, with the minter receiving an owner token, representing their controlling interest in the space. The smart contracts facilitate various functionalities, including membership pricing, gated-access, role-based entitlements, and moderation customizations for each space.
2. **Towns Protocol**: The backbone for social messaging within the ecosystem. It comprises the Towns Chain and Stream Nodes, collectively managing encrypted communications across the network. The protocol ensures secure and private group messaging, essential for maintaining the integrity of communications within each space.
3. **River Eridanus Association**: Initially established in the first quarter of 2024 as a Swiss Association – the **River Eridanus Association** supports the further growth and development of the **Towns Protocol**, through operating its grant programs and disbursement of its initial grant of tokens. The interaction between the **River Eridanus Association** and the **Towns Protocol** operates through a series of smart contracts that ensures transparency and alignment. Notably, the funding for the **River Eridanus Association** is decided by the holders of the TOWNS Token holders through the governance of the Towns Lodge.
4. **Towns Lodge**: Responsible for governance and stewarding of the 3 billion TOWNS Tokens within its Treasury, with the holders of the TOWNS tokenholders ultimately deciding how to deploy the Treasury in support of the Towns Protocol. The Towns Lodge provides essential balance to the Towns ecosystem as it provides the TOWNS tokenholders the ability to direct funding decisions through token governance.
5. **Clients**: Towns is a permissionless protocol meaning anyone can build a client to interact with it. Here is a list of popular clients already building on Towns: Towns, LateCheckout

Was this page helpful?

Yes No

Overview

Next ›

Ask a question... ⌘I

x github website

Archived copy · cryp2nova.com

Powered by

mintlify

This documentation is built and hosted on Mintlify, a developer documentation platform

On this page

- What is Towns Protocol?
- Purpose and Vision
- Key features
- High-Level Component Summary

CRYP2NOVA

Archived copy · cryp2nova.com