



ResilientDB Fullstack

Ecosystem Tools and ResAI



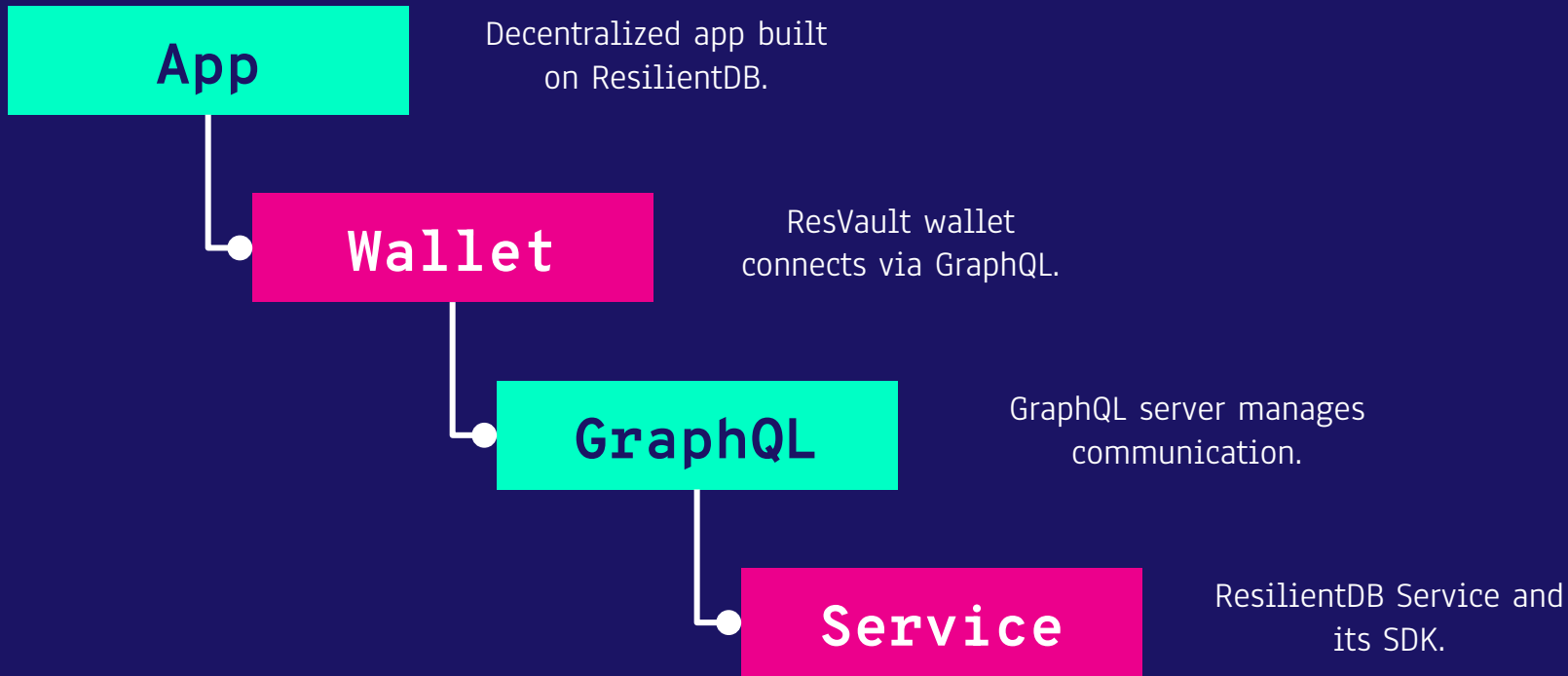
ExpoLab
Creativity Unfolded

What is ResilientDB Fullstack?

ResilientDB Fullstack is a unified ecosystem of applications and AI-powered tools built on ResilientDB, designed to simplify interaction, accelerate development, and improve observability. At its core, ResVault serves as the secure wallet, seamlessly connecting users and applications with the ResilientDB GraphQL server. With the ResVault Web SDK, web applications can easily enable wallet-based authentication and transactions.

ResilientDB further extends flexibility with SDKs in Python, Rust, and TypeScript, allowing smooth integration across diverse environments. On top of these foundations, the ResAI Suite, including Beacon, Nexus, and ContractForge, brings intelligent support for query exploration, distributed orchestration, and AI-generated smart contracts.

ResilientDB Fullstack Components



ResilientDB Service and SDK

Crow HTTP server

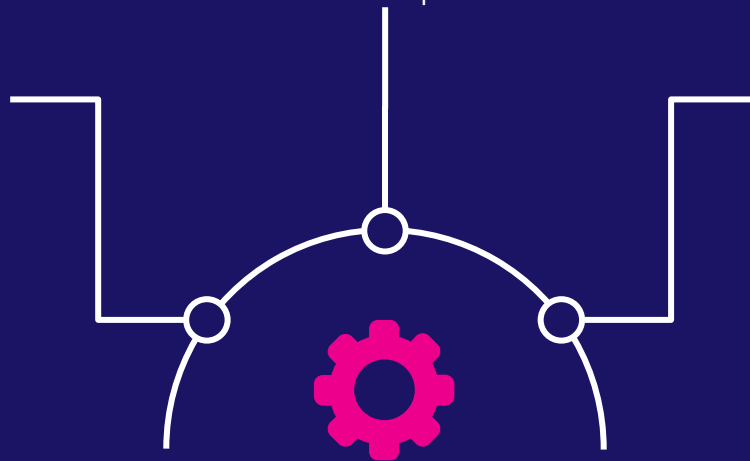
HTTP server exposes ResilientDB's key-value service endpoints.

KV service

ResilientDB Key-Value service offers efficient data management.

Python SDK

Python SDK utilizes HTTP server to connect with key-value service.



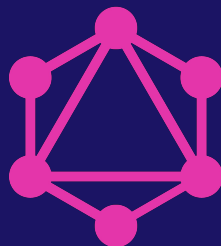
ResilientDB SDK Support

	Python	Rust	Typescript
Status	Stable	Alpha	Alpha
Usage	<u>GitHub</u>	<u>Cargo</u>	<u>NPM</u>



ResilientDB GraphQL Server

ResilientDB GraphQL server offers efficient data retrieval by allowing specific data requests, reducing over-fetching. Its single endpoint simplifies API structure compared to REST's multiple endpoints. The GraphQL Playground further enhances development with an interactive interface for testing and exploring APIs.



Repo: <https://github.com/apache/incubator-resilientdb-graphql>

Query vs Mutation

Query

GraphQL

Fetches data without changing it, allowing precise and efficient data retrieval.

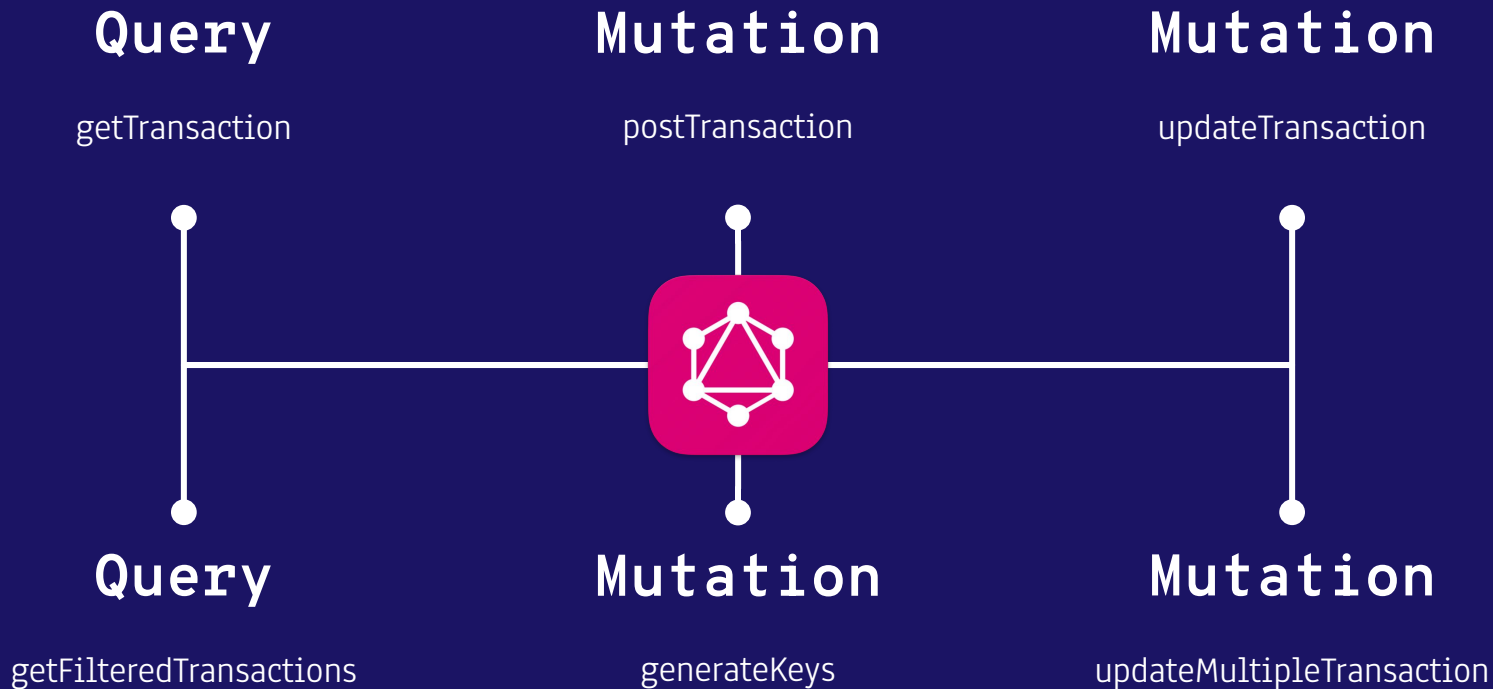


Mutation

GraphQL

Modifies data, handling create, update, and delete operations, altering data state.

ResilientDB GraphQL Queries & Mutations



What is MetaMask?

MetaMask is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension, which can then be used to interact with decentralized applications.



URL: <https://metamask.io>

What is ResVault?

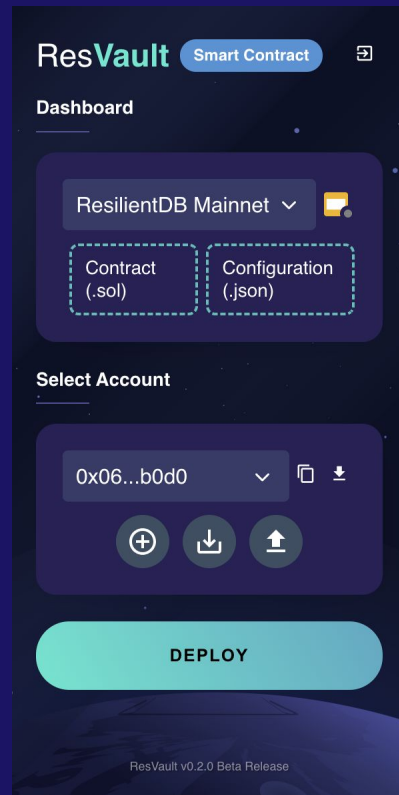
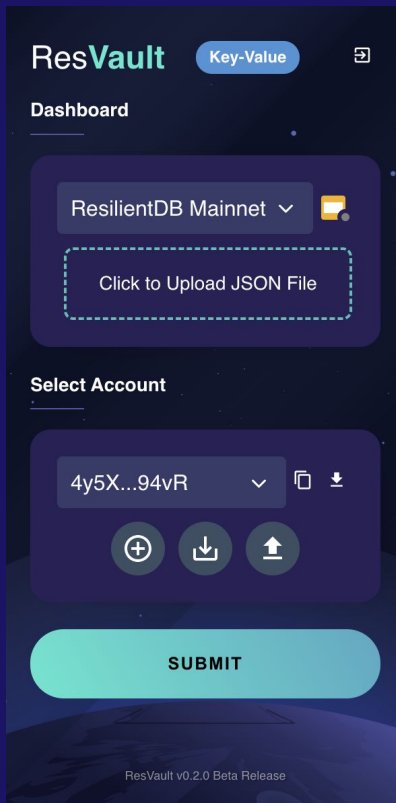
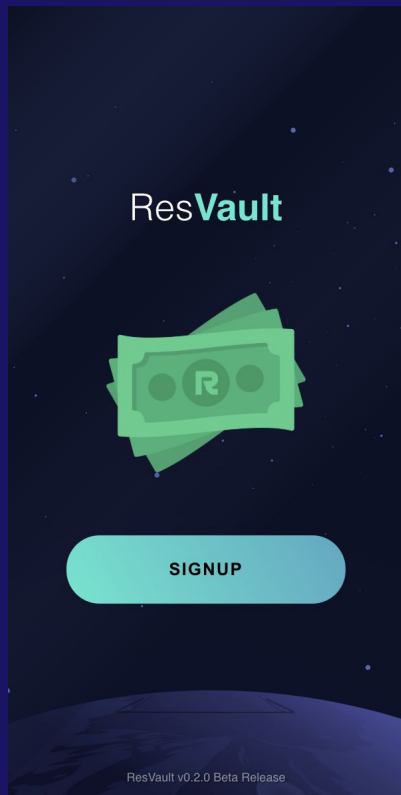
ResVault is a chrome extension that serves as a wallet for ResilientDB. The wallet allows you to commit and retrieve data from ResilientDB and provides user account creation and deletion. It communicates with ResilientDB using the ResilientDB GraphQL server and enables transaction logging. It also allows you to deploy Smart Contracts on ResilientDB.

So, ResVault is quite similar to Metamask in terms of functionality.



Install from Chrome store : [Link](#)

What is ResVault?



Install from Chrome store : [Link](#)

ResVault Features

Txns



Submit transactions
via ResVault.

History



Access history to view
transactions.

Account



Create and log into
accounts easily.

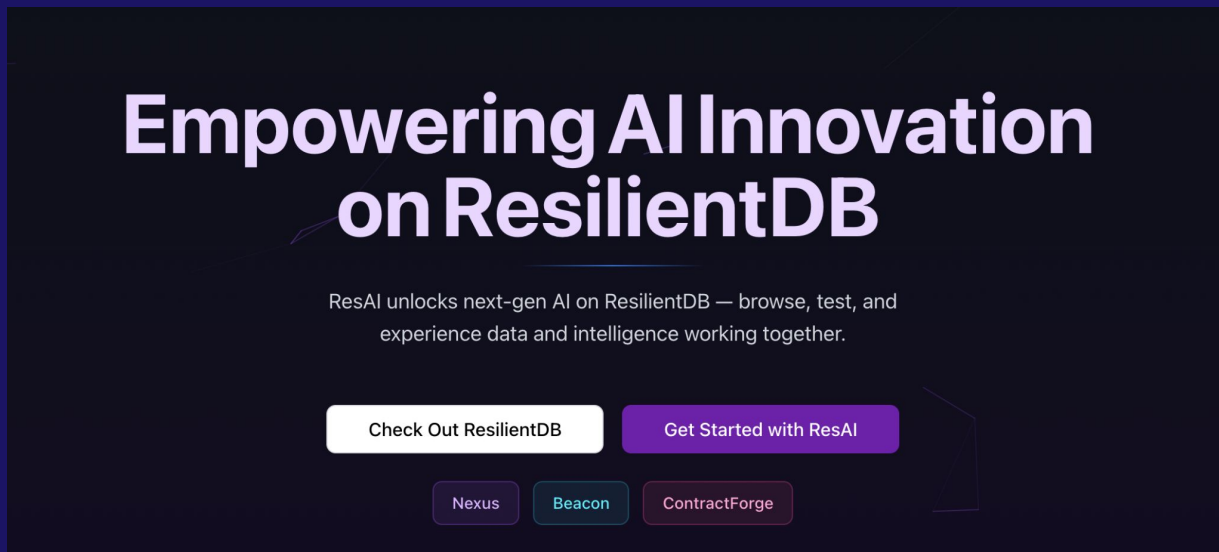
Remove



Effortlessly delete your
account.

Introducing ResAI

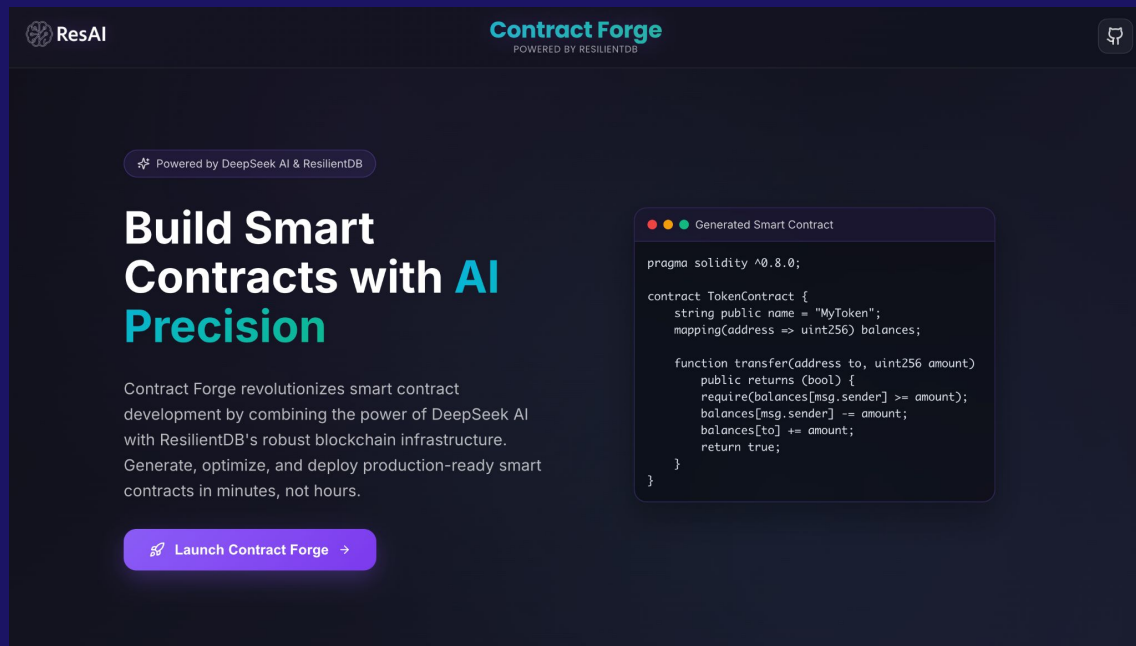
ResAI is an AI-powered suite in ResilientDB that assists developers and researchers with intelligent interaction. It enables query exploration, contract generation, and orchestration of distributed workloads, making it easier to build and optimize applications on ResilientDB.



<https://resai.resilientdb.com/>

ContractForge

ContractForge is an AI-assisted smart contract generator for ResilientDB. It enables developers to create, verify, and deploy contracts quickly, lowering the barrier to entry for decentralized application development while ensuring reliability and correctness.



ResAI

Contract Forge
POWERED BY RESILIENTDB

Powered by DeepSeek AI & ResilientDB

Build Smart Contracts with AI Precision

Contract Forge revolutionizes smart contract development by combining the power of DeepSeek AI with ResilientDB's robust blockchain infrastructure. Generate, optimize, and deploy production-ready smart contracts in minutes, not hours.

[Launch Contract Forge →](#)

```
Generated Smart Contract

pragma solidity ^0.8.0;

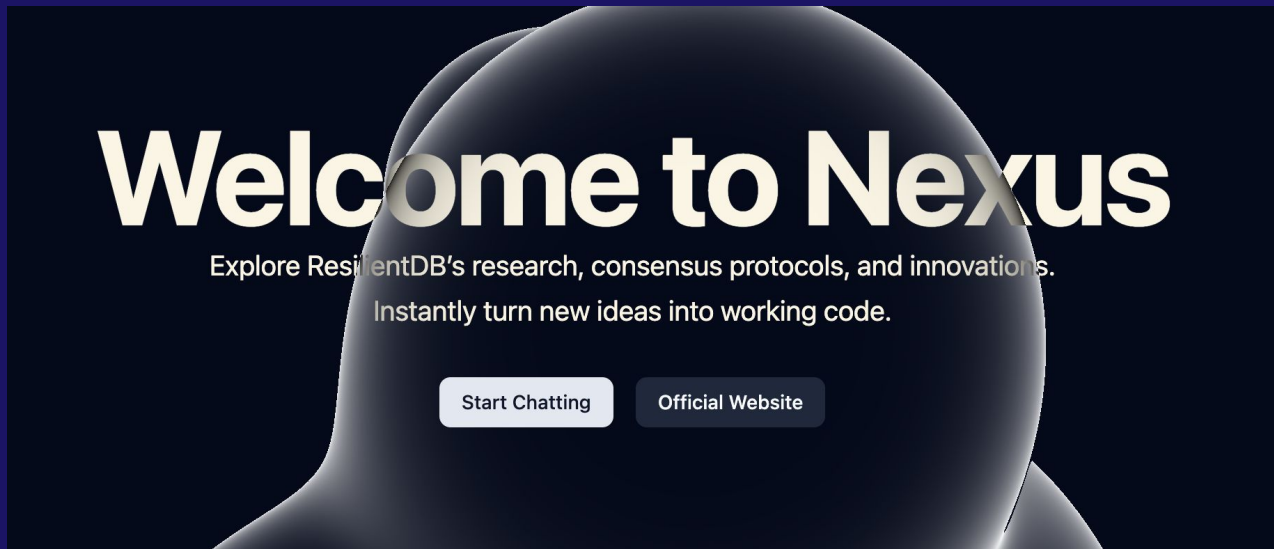
contract TokenContract {
    string public name = "MyToken";
    mapping(address => uint256) balances;

    function transfer(address to, uint256 amount)
        public returns (bool) {
        require(balances[msg.sender] >= amount);
        balances[msg.sender] -= amount;
        balances[to] += amount;
        return true;
    }
}
```

<https://contractforge.resilientdb.com/>

Nexus

Nexus is an AI-powered RAG tutor built on ResilientDB. It serves as a knowledge hub for the lab, storing research papers, books, and learning materials, while enabling students to query and interact with this content through natural language.




<https://nexus.resilientdb.com/>


Nexus


Research Library

 New Chat

AVAILABLE DOCUMENTS 1 of 3 selected


 Search documents...

 Select All

☒  Blockchain Transaction Processing

bchain-transac... 0.8 MB

Selected

☐  Resilient Concurrent Consensus for High-...


rcc.pdf 0.3 MB

☐  ResilientDB: Global Scale Resilient Blockchain Fabric

resilientdb.pdf 1.1 MB

Nexus

Ask questions about Blockchain Transaction Processing...

Research 

 Search

PDF Preview

 Blockchain Transacti...

Blockchain Transaction Processing

Sreyah Gupta, Mohammad Sadoghi

Synonyms

- Blockchain Data Management
- Blockchain Consensus
- Cryptocurrency

Definitions

A blockchain is an append-only linked-list of blocks, which is maintained at each participating node. Each block records a set of transactions and their associated metadata. Blockchain transactions act on the identical ledger data stored at each node. Blockchain was first perceived by Satoshi Nakamoto (Satoshi 2008) as a peer-to-peer digital-commodity (also known as crypto-currency) exchange system. Blockchains received traction due to their inherent property of immutability—once a block is accepted, it cannot be reverted.

Overview

In 2008, Satoshi Nakamoto (Satoshi 2008) introduced the design of an anticipated technology that revolutionized the research across the distributed systems community. Nakamoto presented the design of a peer-to-peer digital-commodity exchange system, which although employed by several participants, prevents the use of a centralized design. Nakamoto envisioned a system where the participants exchange commodities among themselves in a democratic, decentralized and transparent manner while upholding their right to privacy. Nakamoto visualized this digital-commodity as a *monetary* token that could be used by participants to provide or receive services. This led to the birth of *Bitcoin*—a cryptocurrency—and introduction of a new design paradigm *Blockchain*. A blockchain in its simplest form is an *append-only* linked-list of blocks.

arXiv:2107.11592v2 [cs.DB] 4 Aug 2021

2



Fig. 1: Basic Blockchain Representation

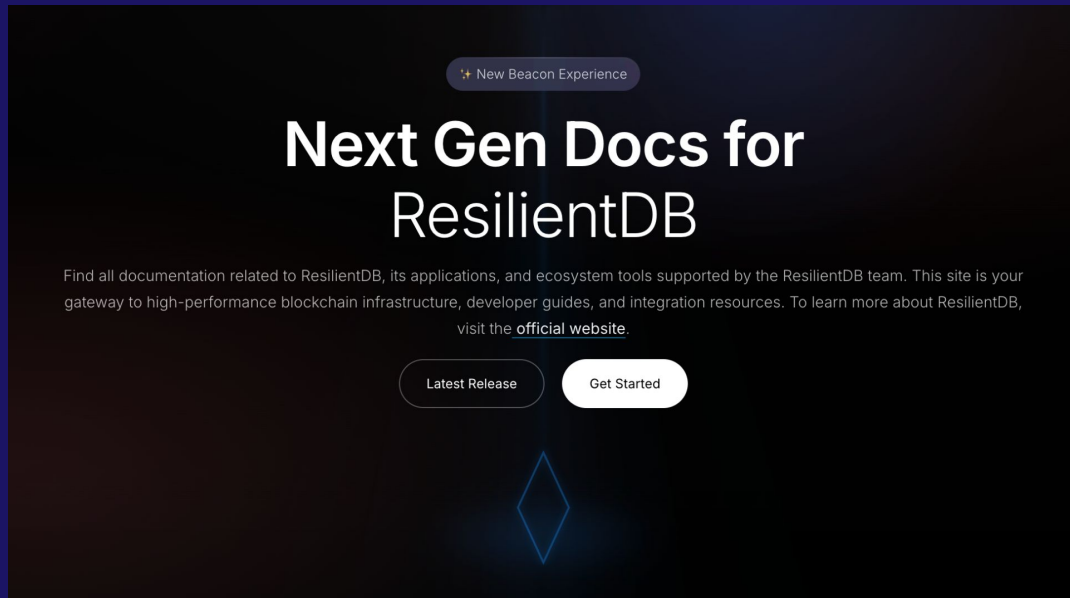
Sreyah Gupta, Mohammad Sadoghi

at the execution layer and persisted in the immutable ledger at the storage layer. Clients and servers also employ necessary cryptographic constructs to securely exchange messages among each other.

The preceding discussion illustrates

Beacon: NextGen Docs

Beacon is the next-generation documentation platform for ResilientDB. It unifies tutorials, technical references, and AI-powered guidance into a single hub, making it easier for developers and researchers to learn, explore, and build on ResilientDB.



<https://beacon.resilientdb.com/>

Beacon: NextGen Docs

Chapters

1. [Client Interaction](#)
2. [Network Communication](#)
3. [Consensus Management](#)
4. [Message/Transaction Collection](#)
5. [Transaction Execution](#)
6. [Storage Layer](#)
7. [Checkpointing & Recovery](#)
8. [ResilientDB Configuration](#)

Documentation > Resilient DB > Chapters > Chapter 1: Client Interaction

Chapter 1: Client Interaction

Welcome to your first step in understanding ResilientDB! Imagine ResilientDB is a powerful, secure, and reliable database system run by a team of computers working together. How do you, as a user or an application developer, talk to this system? You can't just shout commands into the void! You need a defined way to communicate, like using a customer service counter at a bank or an API (Application Programming Interface) for a website.

This chapter introduces the **Client Interaction** layer of ResilientDB. It's the set of tools and libraries that act as that "customer service counter" or API, allowing you to:

- Send requests (we call these **transactions**) to the ResilientDB network.
- Receive results back from the network.

Think of it as the official "front door" for interacting with ResilientDB, hiding all the complex machinery working behind the scenes.

Beacon: NextGen Docs

Set Transaction by Key-Value Pair

```
# Set a transaction by a id (key) and its value
curl --location 'https://crow.resilientdb.com/v1/transactions/commit' \
--header 'Content-Type: application/json' \
--data '{
  "id": "key_test",
  "value": "value_test"
}'
```

① This curl command can be directly imported into Postman. In Postman, you can:

1. Click "Import" and paste the curl command
2. Use the "Code" button to generate equivalent code in TypeScript, Python, or other language
3. Save the request to a collection for future use

GraphQL Usage examples

GraphQL Responses explained

Response Examples

Success Response (201 Created):

```
id: key_test
```

① Error Response (400 Bad Request):

```
Invalid transaction format
```

Unavailable Service (503 Service Unavailable):

```
Service Unavailable
```

Beacon: NextGen Docs

Interactive Playground

Try out Python code directly in your browser! The playground below lets you experiment with Python code and see the results in real-time.

Python Playground

Load example

```
1  """
2  Welcome to the ResilientDB Python Playground!
3
4  This interactive environment allows you to test and experiment with ResilientDB
5  using our Python SDK. The code here matches the patterns used in the official
6
7  Available Features:
8  1. Basic Transaction Operations
9     - Create simple transactions
10    - Create transactions with metadata
11    - Retrieve transactions by ID
12
13  Choose an example from the dropdown menu above to get started!
14
15  Note: Each transaction needs a unique ID. Our examples use timestamps to ensure
16  """
```

Run

Clear Output

Send Transaction with Metadata

Welcome - Start Here

Send Simple Transaction

✓ Send Transaction with Metadata

Get Transaction

Complete Workflow

Pre-loaded templates

▶ Running code...

Creating transaction with ID: test_1759138744

Transaction data:

```
{
  "id": "test_1759138744",
  "value": "Hello from ResilientDB!",
  "type": "kv"
}
```

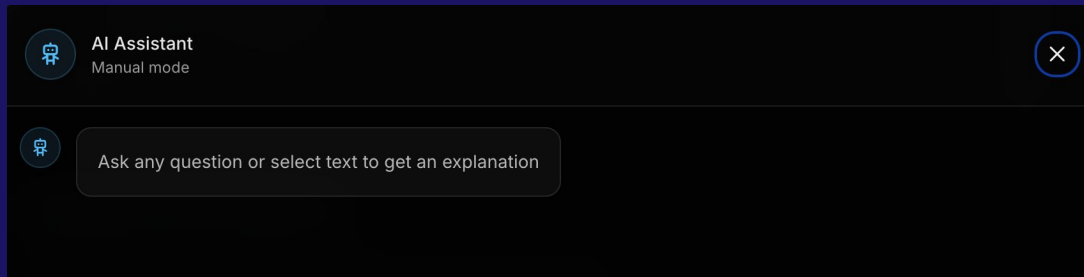
Sending transaction...

Response:

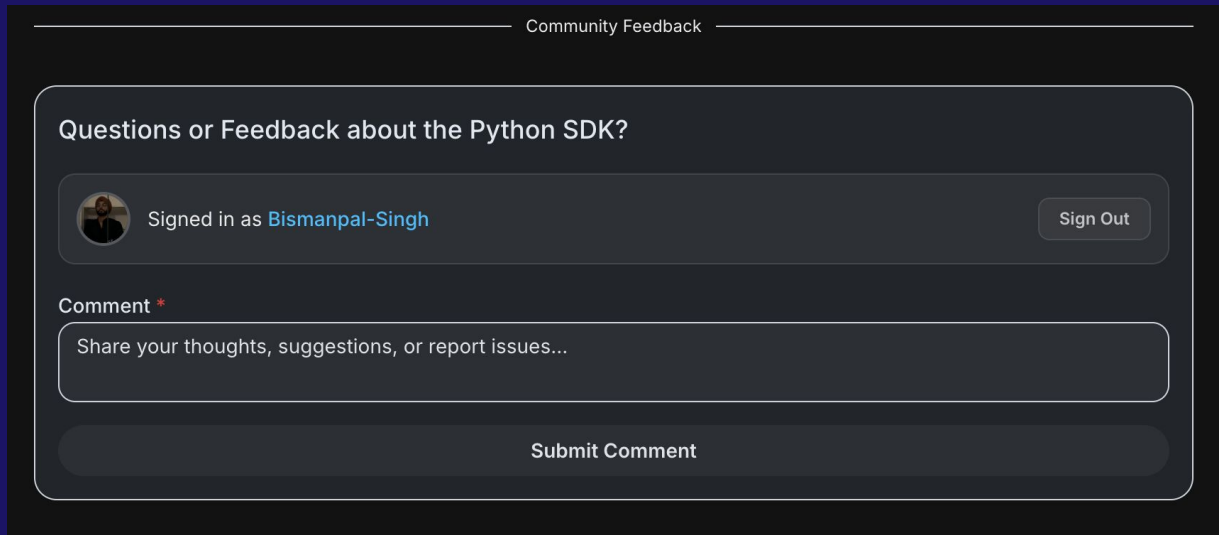
Interactive Playgrounds

Instant outputs

Beacon: NextGen Docs



AI Assistance
in Beacon

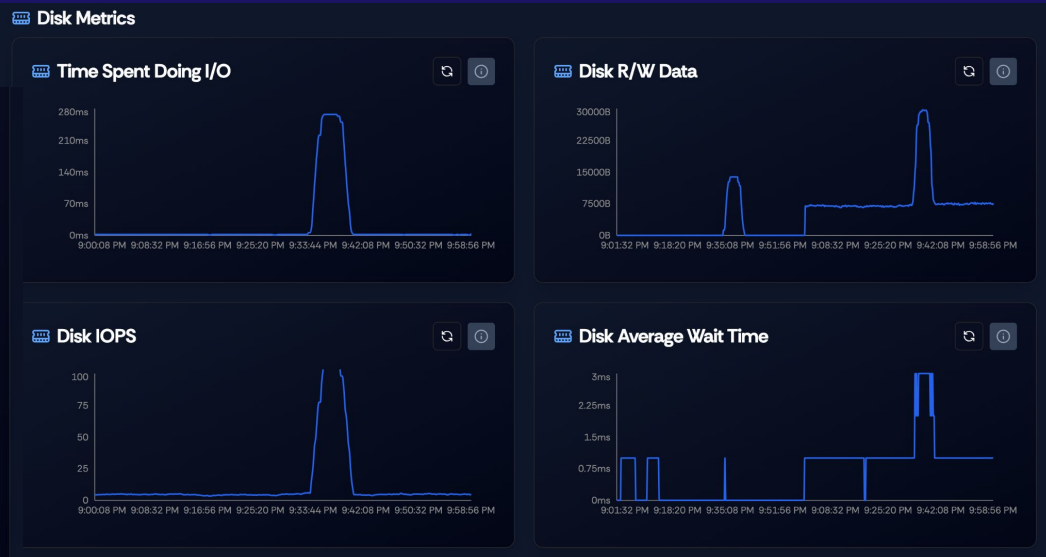


Raise issues from
Beacon itself



ResLens: Continuous Profiling

ResLens is a Profiling and Observability tool built on top of Prometheus to provide code, data, memory and protocol level insights. ResLens provides a unique flamegraph viewer and AI profile analyzer to turn profiling data into actionable insights to perform optimizations, explain code, aid in feature development and more.

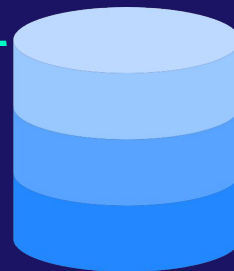


<https://reslens.resilientdb.com/>

Some more...

Resilient Applications

Powered by ResVault and GraphQL



What is ResContract CLI?

The ResContract CLI is a command-line tool for creating, deploying, and managing smart contracts within the ResilientDB ecosystem. It provides a streamlined interface for developers and students to interact with smart contracts efficiently.



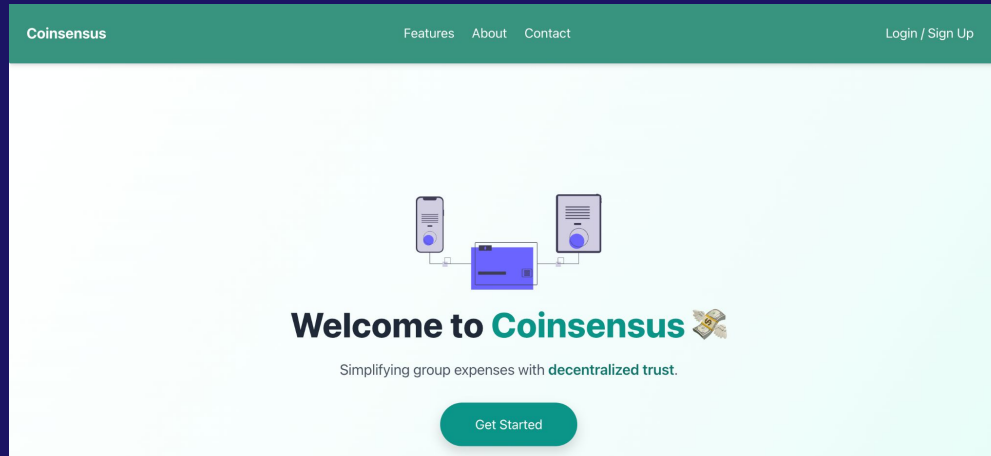
Using the ResContract CLI

Command Overview

- `create` : Create a new account.
- `compile` : Compile a Solidity contract.
- `deploy` : Deploy a smart contract.
- `execute` : Execute a function within a deployed smart contract.

What is CoinSensus

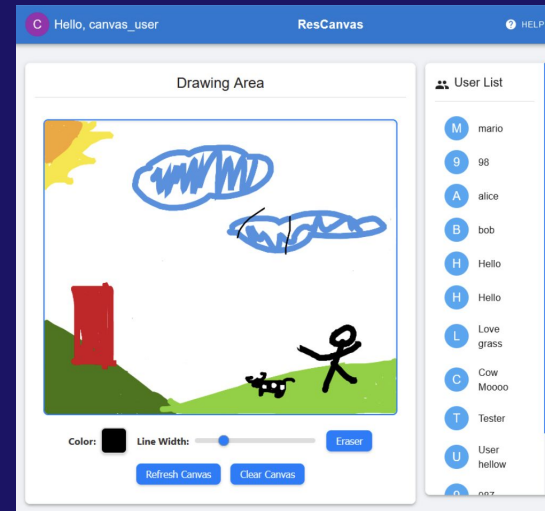
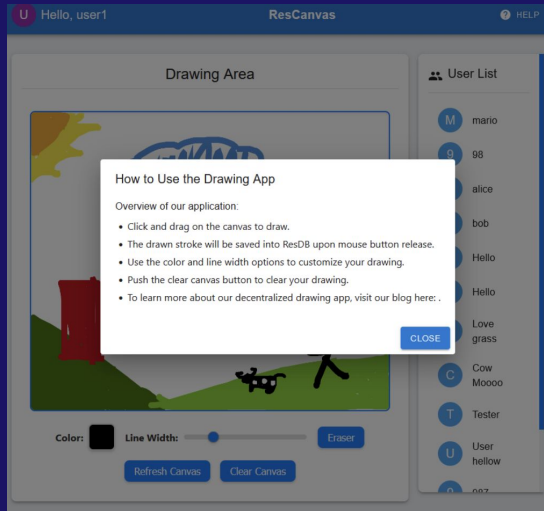
Coinsensus, a blockchain-based bill management platform powered by ResilientDB, is designed to create a trustless and transparent system for expense tracking among friends and groups. By leveraging blockchain technology's decentralized and tamper-resistant nature, the project ensures secure and reliable monitoring of debts and lending without relying on a central authority.



Live URL: <https://coinsensus.resilientdb.com/>

What is ResCanvas?

ResCanvas, a breakthrough in web-based drawing platforms that utilizes ResilientDB to ensure that user's drawings are securely stored, allowing for multiple users to collaborate and create new works of art and express ideas freely without any limits, tracking, or censorship. The canvas drawing board is the core feature of ResCanvas, designed to allow users to perform drawings using their mouse or touchscreen interface.





Everything else

Check out more Resilient Applications on

<https://github.com/ResilientApp>

Check out Ecosystem Tools at

<https://github.com/ResilientEcosystem>



Potential Project Ideas

- Core Engine Development

- [Protocol] Support Lightweight Read-only Transaction Design
- [Protocol] Support Raft - Crash-tolerant Consensus Protocol [ATC 2014]
- [Storage] Enhancements to the In Memory Blockchain State. *
- [Execution] Gas Fee Implementation on Read/Write Transactions. *
- [Storage] Extending secondary indexing features / Query Layer Integration. *

- Ecosystem Tooling

- [SDK] OpenAPI Compatible API layer for Automated SDK Generation.
- [MCP] MCP Server For API / GraphQL and Smart Contract Interaction.
- [Infra] Extending existing bootstrapping tools like Orbit and Integrating AI Features into it.



Thank you!

