

# Connecting Smart Devices to Smart Contracts with W3bstream

Xinxin Fan, Zhi Zhong, Dong Guo, Qi Chai and Simone Romano  
IoTeX

Menlo Park, CA 94025, USA

Email: {xinxin, zhi, dong, raulen, simone}@iotex.io

**Abstract**—In this demo, we illustrate the process of creating a machine economy utilizing W3bstream - an emerging open-source framework designed for connecting smart devices to smart contracts. We will emphasize the flexibility and user-friendliness of W3bstream in empowering decentralized IoT applications.

## I. INTRODUCTION

With the increasing number of smart devices connected to the internet and the rapid development of Web 3.0 technologies, the machine economy is emerging as a new frontier in the digital transformation of the Internet of Things (IoT). [1]. Monetizing machine utility and generated data enables the machine economy to transform IoT businesses by creating new business models and customer engagement strategies. W3bstream [3], [4], as shown in Fig. 1, is an open-source framework developed by IoTeX [2] for connecting physical world to Web3. In short, W3bstream collects data from trusted smart devices, generates cryptographic proofs of real-world activities, and feeds them to a decentralized machine economy application to trigger state changes on a blockchain network. In this demo, we provide guidance for realizing decentralized machine economy applications using W3bstream. We expect this to stimulate interesting discussions among researchers and industry practitioners.

## II. BUILDING DECENTRALIZED MACHINE ECONOMY APPLICATIONS

### A. High-Level Workflow

A high-level workflow of building a decentralized machine economy application with W3bstream is shown in Fig. 2.

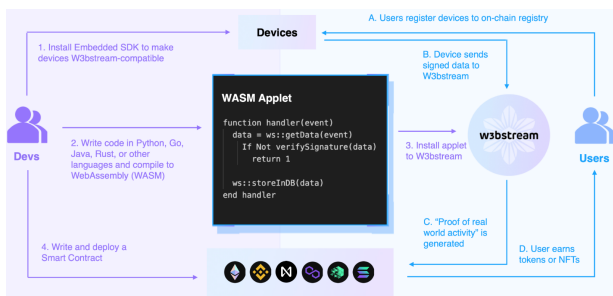


Fig. 2. A high-level workflow for building decentralized machine economy applications using W3bstream

For deploying a machine economy application, a developer needs to complete the following steps:

- 1) A developer installs the W3bstream embedded SDK on smart devices to enable communication with a W3bstream node;
- 2) The developer writes the business logic of a machine economy application in high-level programming languages such as Python, Go, Java, Rust, etc. and compiles it into a WebAssembly (or WASM) [5] applet;
- 3) The developer installs the applet into a provisioned W3bstream node that contains a WASM runtime;
- 4) The developer writes a machine economy dApp and deploys the smart contract(s) on a blockchain.

Once users receive the smart devices, they can join the machine economy application as follows:

- 1) A user registers their smart device(s) on an on-chain device registry;
- 2) The user's smart devices stream signed data to the W3bstream node;
- 3) The W3bstream node processes the data and generates cryptographic proofs of real-world facts based on the predefined business logic;
- 4) The user earns cryptocurrency tokens or NFTs based on the contributions from his/her smart device(s).

### B. Building Components of a W3bstream Node

To support a wide range of applications and business requirements, a W3bstream node includes the following key components, as illustrated in Fig. 3, for interacting with IoT devices, blockchain, and node operators.

- **Service Endpoint:** The service endpoint communicates with IoT devices, blockchain, and node operators using various protocols (e.g. MQTT, HTTP, RPC);
- **Virtual File System:** The virtual file system stores the business logic;
- **Execution Engine:** The execution engine processes the time series data received from IoT devices and executes the pre-defined business logic;
- **Consensus:** The consensus module implements a number of consensus algorithms (e.g., Proof of Authority (PoA), Practical Byzantine Fault Tolerance (PBFT), etc.) for realizing a decentralized W3bstream network;
- **Database:** The database stores raw or encrypted data from IoT devices with a data retention policy;

