Sundial Babel Fees Implementation

Working Draft - September 16th, 2025

Introduction

Sundial protocol designed to make Bitcoin the universal unit of account for scalable, multi-asset applications. Sundial achieves this by anchoring all settlement in wrapped BTC (wBTC), ensuring that the economic cost of participating in the network is always denominated in Bitcoin terms.

To maximize flexibility for developers and users, Sundial introduces a Babel Fees mechanism. With Babel Fees, transaction fees can be paid in virtually any token bridged into Sundial. Regardless of the token chosen, the network internally converts it to wBTC before finalizing settlement, so the system consistently accounts for fees in Bitcoin.

Sundial implements Babel Fees at the application layer using smart contracts and liquidity pools that perform atomic token-to-wBTC swaps inside the user's transaction. As a result:

- Users see a seamless experience: they can initiate a transaction in their token of choice, without holding wBTC.
- Operators receive fees in wBTC, preserving the Bitcoin-based accounting model.
- Liquidity providers earn returns by supplying wBTC to the fee markets.

This paper describes the architecture, design, and transaction flow of Sundial's Babel Fees implementation.

Architecture Design

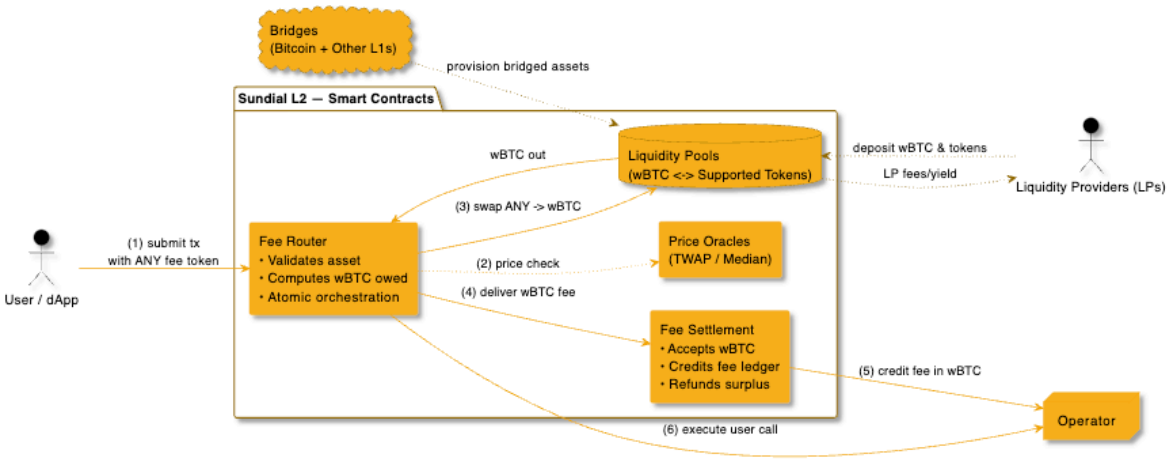
The Babel Fees mechanism is structured around four main components:

1. **Fee Payment Router (Contract):**
   The entry point for any transaction that specifies a non-wBTC token for fee payment. The router validates input, ensures supported assets, and orchestrates the conversion into wBTC.

2. **Liquidity Pools:**
   On-chain pools that hold pairs of wBTC and supported tokens. These pools are the source of immediate liquidity for fee conversion. They may be implemented as AMM-style constant-product pools or simple vaults with fixed-rate quotes.

3. **Fee Settlement (Contract):**
   Collects wBTC from the router after a successful swap and credits it as the transaction fee payment. From the protocol's perspective, all fees arrive in wBTC.

4. **Liquidity Providers (LPs):**
   External participants who deposit wBTC into pools in exchange for yield, funded by users who pay with non-wBTC tokens. LPs ensure the mechanism has sufficient depth for smooth swaps.



Sundial Babel Fees Architecture

The lifecycle of a Babel Fee transaction in Sundial can be broken down into the following steps:

1. **Transaction Construction:**
   - The user initiates an L2 transaction (e.g., transferring tokens or interacting with a dApp).
   - Instead of attaching wBTC for the gas fee, the user specifies their preferred token.
   - The wallet embeds the fee-payment request into the transaction payload, pointing to the Fee Router.

2. **Routing and Validation:**
    - The Fee Router contract verifies that the selected token is supported and that sufficient balance is provided.
    - It calculates the required equivalent of wBTC based on current pool exchange rates and adds any small margin for slippage.
3. **Atomic Conversion:**
    - Within the same transaction, the Router swaps the user's fee token against the relevant liquidity pool.
    - wBTC is withdrawn from the pool, while the user's token is deposited.
    - This occurs atomically: if the swap fails, the entire transaction reverts.
4. **Fee Settlement:**
    - The Router forwards the acquired wBTC to the Fee Settlement contract.
    - The Settlement contract deducts the required wBTC amount and records the transaction as paid.
    - Any surplus (if the user slightly overpays due to slippage buffer) is refunded in the original token.
5. **Fee Credit:**
    - Operators see the fee already settled in wBTC; no additional conversion logic is needed.
6. **Execution and Finalization**
    - The underlying transaction (e.g., token transfer, contract interaction) executes.
    - From the user's perspective, they simply spent their token as gas.

Example Walkthrough

- **Scenario:** Alice wants to send 100 ADA tokens to Bob. She has no wBTC in her wallet.
- **Fee Market:** The network fee is 0.0005 wBTC. Liquidity pools support ADA/wBTC with sufficient depth.

Steps:

1. Alice constructs a transaction, selecting ADA as the fee token.
2. The wallet queries the Router for the current rate: 0.0005 wBTC = 50 ADA.
3. Alice signs and submits the transaction with 50 ADA attached as the fee payment.
4. On-chain, the Router automatically swaps 50 ADA for 0.0005 wBTC from the ADA/wBTC pool.
5. The Settlement contract deducts the 0.0005 wBTC and finalizes the transaction.
6. Validators see the fee paid in wBTC. Alice sees her 100 ADA transfer succeed, and her balance decrease by ~150 ADA total (100 sent + 50 fee).

To Alice, it appears she simply paid gas in ADA; to the system, the accounting is consistent in wBTC.