

Blockchain for Transparent Ad Revenue Sharing

Team ID: 25FSG1BC310

Team Members: Atharv Shiva Hiremath (23BCE1460), Abeer Gupta (23BCE1599), Rithish (23BCE5125)

School of Computer Science [CSE core]

VIT Chennai



Presentation Outline

Navigating the Future of Ad Revenue



Introduction & Problem Statement



Motivation & Objectives



Related Work & Architecture



Methodology & Core Algorithm



Technology Stack & Implementation



Deployment, Testing & Results



Conclusion & Future Work

Blockchain for Transparent Ad Revenue Sharing

The Problem: Opacity in Digital Ad Revenue

The digital advertising ecosystem, involving Advertisers, Platforms (e.g., YouTube), and Content Creators, currently operates with significant transparency gaps.

Traditional Process

- Linear, centralized flow: Advertiser pays Platform; Platform takes a cut, then pays Creator.
- Creators lack visibility into initial payment amounts, platform commissions, or payment timelines.

Key Limitations

- Lack of transparency
- Potential for unfair revenue splits
- Payment delays
- High dependency on the platform as a trusted intermediary

Motivation: Building Trust & Efficiency

Challenges in Current System

- Centralized points of failure
- Data manipulation risks
- Hidden fees and slow cross-border settlements

The Blockchain Solution

- An immutable ledger provides a single source of truth.
- Smart contracts automate revenue split logic, removing manual intervention and bias.



Key Beneficiaries

- **Creators:** Gain full transparency into earnings.
- **Advertisers:** Verify funds are distributed as intended.
- **Platforms:** Build trust and credibility with an open model.

Blockchain for Transparent Ad Revenue Sharing

Project Objectives

Decentralized Transparency

Design a system to eliminate hidden commissions and provide a transparent ledger for all ad revenue transactions.

Tamper-Proof Records

Implement a tamper-proof record of revenue splits using blockchain technology.

Automated Distribution

Automate the revenue distribution process using a self-executing Solidity smart contract.

Cost-Effective & Scalable

Develop and deploy a cost-effective and scalable solution on the Polygon (Amoy Testnet) blockchain.

Blockchain for Transparent Ad Revenue Sharing

Related Work: Learning from the Ecosystem

Brave Browser (BAT)

- **Strength:** Successful token-based attention economy.
- **Gap:** Not focused on Advertiser-Platform-Creator revenue split.

AdEx Network

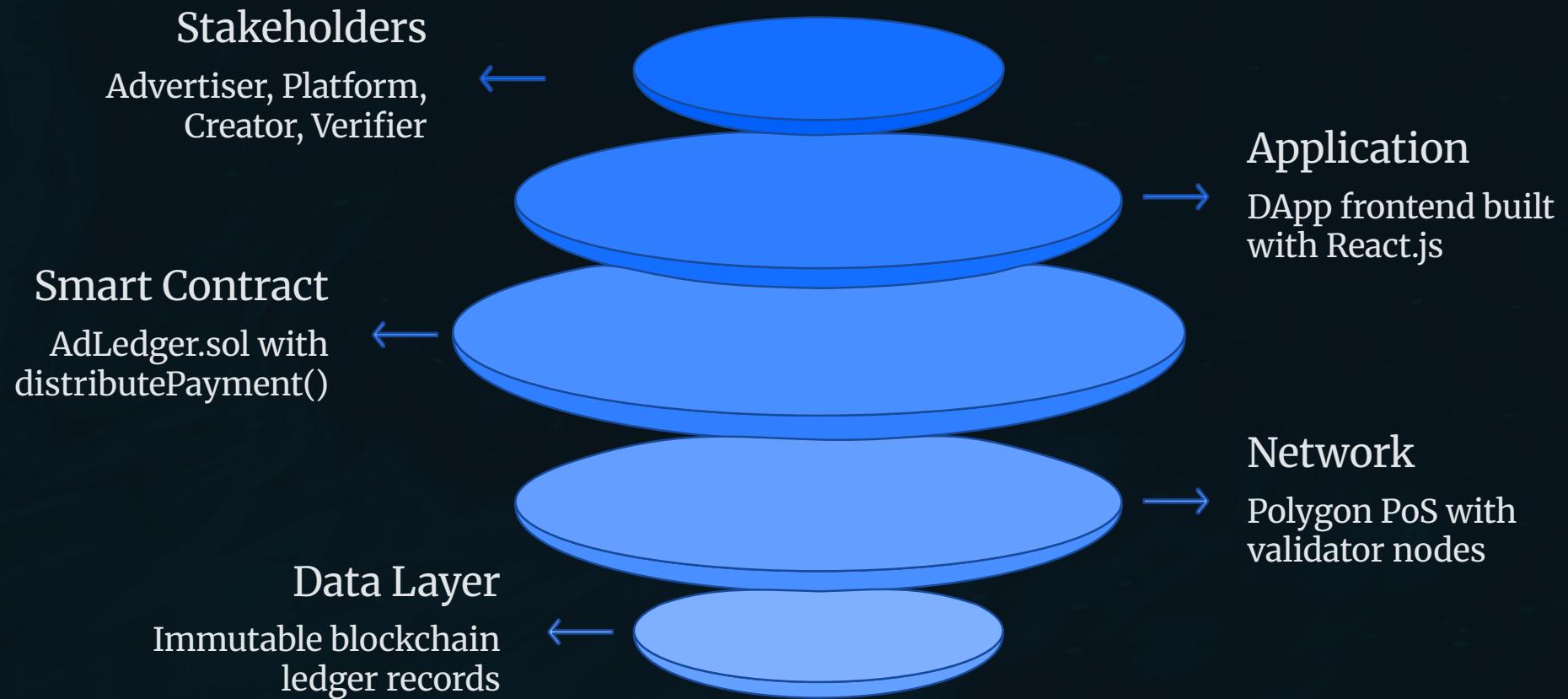
- **Strength:** Focuses on transparency and fraud reduction in ad exchanges.
- **Gap:** Complex for simple revenue-sharing agreements.

Steemit

- **Strength:** Direct crypto rewards for content creators on a social media platform.
- **Gap:** Reward pool mechanism differs from direct ad revenue splitting.

Blockchain for Transparent Ad Revenue Sharing

System Architecture: A Multi-Layered Approach



Methodology & Consensus: Transaction Flow



Initiation

Advertiser uses DApp to define payment terms (amount, creator, splits) and sends funds (MATIC).



Smart Contract Trigger

DApp calls `distributePayment` function on the smart contract.



Consensus (Polygon PoS)

Transaction broadcast. Validators, staking MATIC, validate and include in a new block.



Execution & Settlement

If valid, smart contract executes, splitting and transferring funds to Creator/Platform wallets.



Record Keeping

Successful transaction permanently recorded on blockchain, visible to all parties.

Core Algorithm: distributePayment

Pseudocode

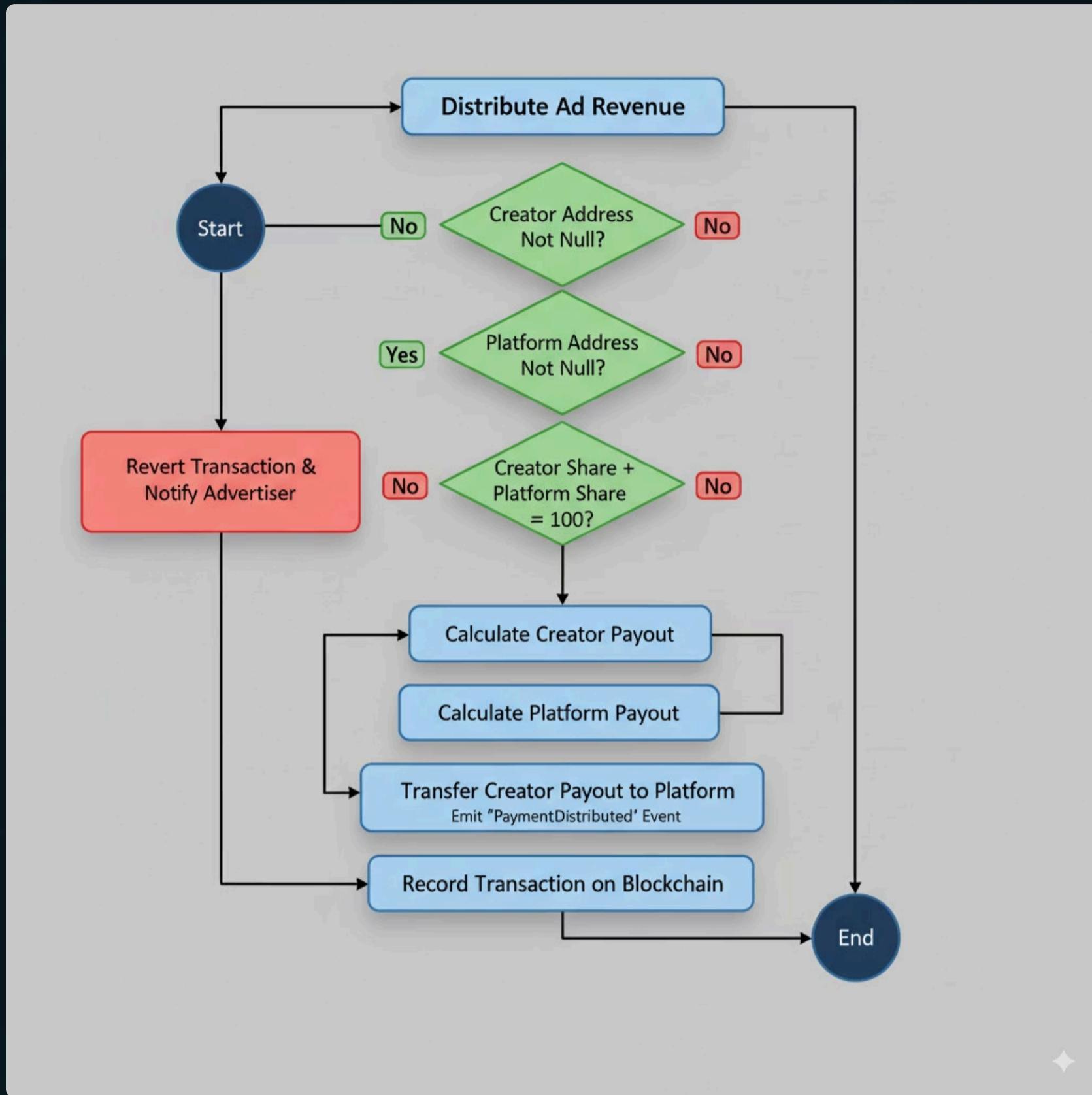
```
PROCEDURE DistributeAdRevenue(advertiser, creator, platform, amount, creatorShare, platformShare)
```

INPUT: Advertiser's payment transaction containing the specified parameters.

OUTPUT: Successful distribution or Rejection notice.

1. REQUIRE payment_amount > 0
 2. REQUIRE creator_address IS NOT NULL
 3. REQUIRE platform_address IS NOT NULL
 4. REQUIRE creator_share + platform_share == 100
 5. IF all requirements met THEN
 6. CALCULATE creator_payout = (amount * creator_share) / 100
 7. CALCULATE platform_payout = (amount * platform_share) / 100
 8. TRANSFER creator_payout TO creator_address
 9. TRANSFER platform_payout TO platform_address
 10. EMIT event 'PaymentDistributed' with transaction details
 11. RECORD transaction on blockchain
 12. ELSE
 13. REVERT transaction
 14. NOTIFY advertiser of failure
 15. END IF
- END PROCEDURE

Flowchart



Blockchain for Transparent Ad Revenue Sharing

Technology Stack: Tools of the Trade



Blockchain

Polygon (Amoy Testnet): Low gas fees, high scalability.

Smart Contract

Solidity, Remix IDE, Hardhat: For writing, debugging, testing.

Backend

Node.js, Express.js: API layer between frontend and blockchain.

Frontend

React.js, HTML/CSS: User-facing DApp.

Wallet & Provider

MetaMask, Alchemy: User auth, transaction signing, network connection.

Web3 Library

Web3.js: Communication between application and smart contract.