Volume: 13 Issue: XI Month of publication: November 2025
DOI: https://doi.org/10.22214/ijraset.2025.75652

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

NFT: What's in a Name? Everything

Jasmine Cathrine Mathew
Department of Computer Science, St. Joseph's Academy of Higher Education and Research, Moolamattom

Abstract: NFT (Non-Fungible Token) has emerged as a trending topic in the digital world. This article focuses on the working principle of NFTs and their practical applications in real-world scenarios. Ethereum blockchain serves as the foundational technology that powers NFTs. This document provides a comprehensive technical overview of Ethereum blockchain technology applied in the textile industry for maintaining product ownership verification and authenticity.
Keywords: NFT, Ethereum, Blockchain, Cryptocurrency, Smart Contracts, Textile Industry

I. INTRODUCTION

The economic term NFT stands for "Non-fungible Token" **-** an Ethereum-based digital asset that can represent anything uniquely. A digital asset exists in digital form with distinct usage patterns and is stored on digital devices such as tablets, data storage devices, and laptops. NFTs represent physical or creative digital works including music, games, GIFs, iconic video clips, virtual avatars, video game skins, real estate, and various forms of creative works. The term "non-fungible" means non-replaceable **-** NFTs are not mutually interchangeable. While one Bitcoin can be exchanged with another Bitcoin, no two NFTs are the same.
NFTs provide exclusive ownership to items, addressing drawbacks in online markets such as counterfeits and authenticity issues through digital signatures **-** built-in authentication serving as proof of ownership in each token.
Cryptocurrency is decentralized digital money built on blockchain technology, verified by encryption techniques and distributed among authorized users only. Its supply is not controlled by central banks and exists only in networks.

II. BLOCKCHAIN TECHNOLOGY

A. Overview

The technology behind NFTs is Ethereum blockchain **-** a distributed public ledger that records all transactions. The transparent and "immutable" property of blockchain makes it popular among online markets. Anything recorded by blockchain technology cannot be tampered with or modified, even by administrators. Blockchain is a decentralized network consisting of a sequence of blocks where all parties can share and record transactions. All transactions must be validated and verified by users in the blockchain network, ensuring no one can modify the ownership record of each NFT.

B. Block Structure

In blockchain, each block consists of: **-** Block Header: Contains metadata including: **-** Cryptographic hash of the previous block **-** Block size **-** Timestamp **-** Nonce value **-** Block Data: Contains validated transactions and ledger events

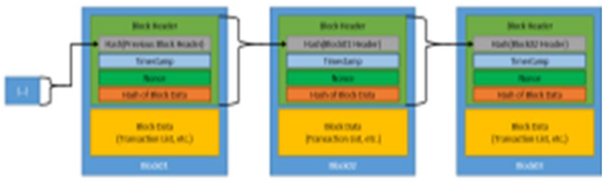
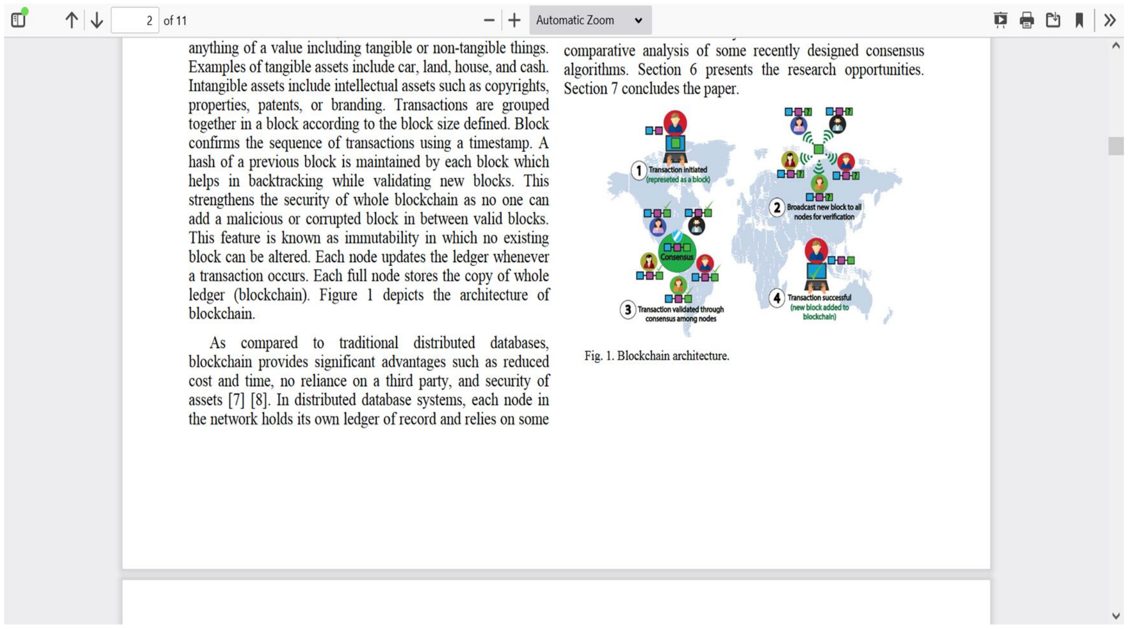


Figure 1: Detailed blockchain block structure showing header components and data sections

Blocks have storage capacities, and once filled, new blocks are added to the chain. All other nodes check the validity and authenticity of transactions in newly added blocks.

C. Consensus Algorithm

A key aspect of blockchain technology is determining which user publishes the next block. This is solved through consensus models where users joining a blockchain network are recorded in a pre-configured genesis block. All nodes must agree on a consensus model, and each block must be independently validated.



anything of a value including tangible or non-tangible things. Examples of tangible assets include car, land, house, and cash. Intangible assets include intellectual assets such as copyrights, properties, patents, or branding. Transactions are grouped together in a block according to the block size defined. Block confirms the sequence of transactions using a timestamp. A hash of a previous block is maintained by each block which helps in backtracking while validating new blocks. This strengthens the security of whole blockchain as no one can add a malicious or corrupted block in between valid blocks. This feature is known as immutability in which no existing block can be altered. Each node updates the ledger whenever a transaction occurs. Each full node stores the copy of whole ledger (blockchain). Figure 1 depicts the architecture of blockchain.

As compared to traditional distributed databases, blockchain provides significant advantages such as reduced cost and time, no reliance on a third party, and security of assets [7] [8]. In distributed database systems, each node in the network holds its own ledger of record and relies on some

comparative analysis of some recently designed consensus algorithms. Section 6 presents the research opportunities. Section 7 concludes the paper.

1 Transaction initiated (represeted as a block)
2 Broadcast new block to all nodes for verification
Consensus
3 Transaction validated through consensus among nodes
4 Transaction successful (new block added to blockchain)

Fig. 1. Blockchain architecture.

Figure 2: Blockchain architecture showing transaction flow from initiation to consensus

III. PROOF-OF-STAKE (POS)

Most NFTs use Ethereum-based blockchain technology. Ethereum relies on a consensus mechanism called “Proof of Stake” for maintaining a time-ordered ledger and security.

Proof-of-Stake Process: - Validators (cryptocurrency owners) lock up their coins - They are randomly selected by the protocol to create new blocks - Users staking larger amounts have higher chances of being selected - No expensive hardware or massive electricity consumption required - Regular PCs with sufficient coins can participate

IV. SMART CONTRACTS

A smart contract is a computerized transaction protocol where buyer-seller agreements are directly encoded into lines of code. Smart contracts run on top of Ethereum blockchain to maintain network safety and organization.

NFTs are minted through smart contracts that provide ownership and control transferability. Authenticity is achieved through unique IDs and metadata that no other token can reproduce. The creator maintains royalty rights without handing them to online platforms.

Ethereum Smart Contract Languages: - Solidity (most widely used) - Vyper

V. APPLICATION IN SILK INDUSTRY

A. *Industry Challenges*

India is the 2nd largest silk manufacturer and exporter globally, contributing 12% to export earnings in FY20. However, the industry faces significant challenges:

1) Common Problems: - Price fluctuation - Absence of proper markets - Lack of transport and storage facilities - Poor market trend information - Finance limitations - High production costs with low productivity - Technology penetration gaps - Rural industry nature
2) Counterfeit Issues: - Mechanization led to fake handlooms and handicrafts flooding markets - Well-engineered replicas marketed as original handmade pieces - Indian artisans losing market share and income - Middlemen dominating profits over artisans - Only 10% of weavers currently work on handlooms; 90% abandoned for power looms

B. *NFT Solution for Silk Industry*

NFTs can provide ownership verification and authenticity for silk products by: - Creating unique digital certificates for each authentic piece - Enabling traceability from artisan to consumer - Reducing counterfeit products - Ensuring fair compensation to artisans - Creating permanent records of authenticity

VI. MODERN IMPLEMENTATION GUIDE

A. Prerequisites (Updated for 2025)

Required Tools: - Node.js (version 18 or higher) - npm (latest version) - MetaMask wallet - Alchemy account (or similar blockchain API provider)

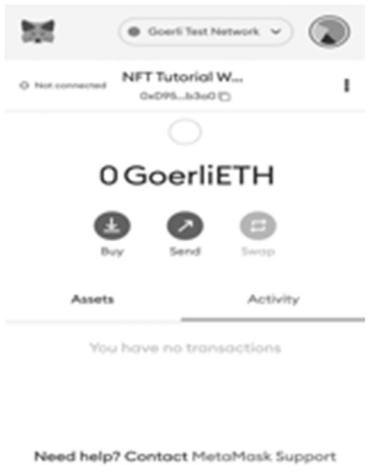


Figure 5: MetaMask wallet interface showing test network setup

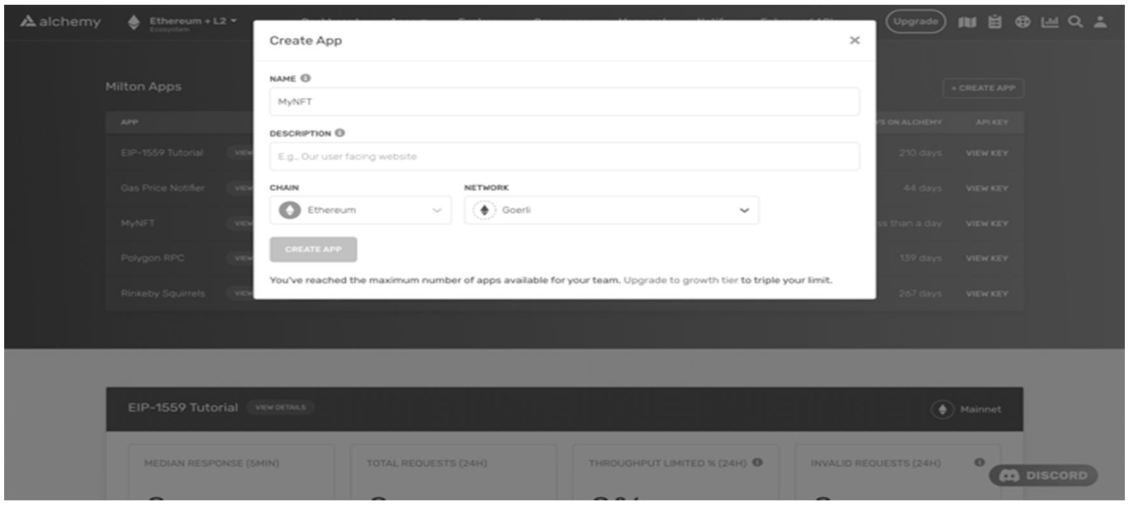
Setup Commands:

```
# Check Node.js version
node -v  # Should be v18.0.0 or higher

# Check npm version
npm -v
```

B. Create Alchemy App (Updated)

1. Sign up at Alchemy.com
2. Navigate to Apps → Create App
3. Configure:
   – Name: SilkNFT-Demo
   – Description: NFT application for silk industry authentication
   – Chain: Ethereum
   – Network: Sepolia (updated testnet for 2025)
4. Save the API KEY



NAME: MyNFT
DESCRIPTION: E.g., Our user facing website
CHAIN: Ethereum
NETWORK: Goerli

You've reached the maximum number of apps available for your team. Upgrade to growth tier to triple your limit.

APP		...S ON ALCHEMY	API KEY
EIP-1559 Tutorial	VIEW	210 days	VIEW KEY
Gas Price Notifier	VIEW	44 days	VIEW KEY
MyNFT	VIEW	...s than a day	VIEW KEY
Polygon RPC	VIEW	139 days	VIEW KEY
Rinkeby Squirrels	VIEW	267 days	VIEW KEY

EIP-1559 Tutorial VIEW DETAILS Mainnet

MEDIAN RESPONSE (5MIN) | TOTAL REQUESTS (24H) | THROUGHPUT LIMITED % (24H) | INVALID REQUESTS (24H)

Figure 3: Alchemy platform interface showing app creation process

C. Project Setup (Modernized)

```
# Create project directory
mkdir silk-nft-project && cd silk-nft-project

# Initialize Node.js project
npm init -y

# Install dependencies
npm install --save-dev hardhat @openzeppelin/contracts ethers dotenv

# Initialize Hardhat
npx hardhat
```

D. Updated Hardhat Configuration (2025 Standards)

hardhat.config.js:

```
require('@nomicfoundation/hardhat-toolbox');
require('@nomicfoundation/hardhat-chai-matchers');
require('dotenv').config();

const { API_URL, PRIVATE_KEY } = process.env;

/** @type import('hardhat/config').HardhatUserConfig */
module.exports = {
  solidity: {
    version: "0.8.24",
    settings: {
      optimizer: {
        enabled: true,
        runs: 200,
      },
    },
  },
  defaultNetwork: "sepolia",
  networks: {
    sepolia: {
      url: API_URL,
      accounts: [`0x${PRIVATE_KEY}`],
    },
    localhost: {
      url: "http://127.0.0.1:8545",
    },
  },
  etherscan: {
    apiKey: process.env.ETHERSCAN_API_KEY,
  },
};
```

E. Modern Smart Contract Implementation

contracts/SilkNFT.sol:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.24;

import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";
import "@openzeppelin/contracts/utils/Counters.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/security/ReentrancyGuard.sol";
import "@openzeppelin/contracts/token/common/ERC2981.sol";

contract SilkNFT is ERC721URIStorage, Ownable, ReentrancyGuard, ERC2981 {
    using Counters for Counters.Counter;

    Counters.Counter private _tokenIds;

    // Royalty percentage (2.5% = 250 basis points)
    uint96 public constant ROYALTY_PERCENTAGE = 250;
    uint96 public constant MAX_ROYALTY_PERCENTAGE = 1000; // 10% max

    // Product authenticity levels
    enum AuthenticityLevel {
        HANDWOVEN,
        POWERLOOM,
        MACHINE_MADE,
        MIXED
    }

    // Silk product metadata
    struct SilkProduct {
        string artisanName;
        string region;
        string silkType;
        AuthenticityLevel authenticity;
        uint256 timestamp;
        string qualityGrade;
    }

    mapping(uint256 => SilkProduct) public silkProducts;

    constructor(string memory name_, string memory symbol_)
        ERC721(name_, symbol_) {
        // Set default royalty
        _setDefaultRoyalty(msg.sender, ROYALTY_PERCENTAGE);
    }

    /**
     * @dev Mint a new NFT for silk product
     * @param recipient Address that will receive the NFT
```

```
 * @param tokenURI URI pointing to metadata JSON
 * @param artisanName Name of the artisan who created the product
 * @param region Geographic region of production
 * @param silkType Type of silk (Mulberry, Tussar, Eri, etc.)
 * @param authenticity Level of authenticity
 * @param qualityGrade Quality rating (A, B, C, etc.)
 * @return tokenId The ID of the newly minted token
 */
function mintSilkNFT(
    address recipient,
    string memory tokenURI,
    string memory artisanName,
    string memory region,
    string memory silkType,
    AuthenticityLevel authenticity,
    string memory qualityGrade
) public onlyOwner nonReentrant returns (uint256) {
    _tokenIds.increment();
    uint256 newItemId = _tokenIds.current();

    _mint(recipient, newItemId);
    _setTokenURI(newItemId, tokenURI);

    // Store silk product information
    silkProducts[newItemId] = SilkProduct({
        artisanName: artisanName,
        region: region,
        silkType: silkType,
        authenticity: authenticity,
        timestamp: block.timestamp,
        qualityGrade: qualityGrade
    });

    // Transfer event for transparency
    emit Transfer(address(0), recipient, newItemId);

    return newItemId;
}

/**
 * @dev Transfer NFT with automatic royalty distribution
 */
function safeTransferFrom(
    address from,
    address to,
    uint256 tokenId
) public virtual override(ERC721, ERC721URIStorage) {
    super.safeTransferFrom(from, to, tokenId);
}
```

```
/**
 * @dev Update royalty for all future sales
 */
function setDefaultRoyalty(address recipient, uint96 feeNumerator)
    public onlyOwner {
    require(feeNumerator <= MAX_ROYALTY_PERCENTAGE, "Royalty too high");
    _setDefaultRoyalty(recipient, feeNumerator);
}

/**
 * @dev Get silk product information
 */
function getSilkProduct(uint256 tokenId)
    external view returns (SilkProduct memory) {
    require(_exists(tokenId), "Token does not exist");
    return silkProducts[tokenId];
}

/**
 * @dev Batch mint NFTs for efficiency
 */
function batchMintSilkNFTs(
    address[] memory recipients,
    string[] memory tokenURIs,
    string[] memory artisanNames,
    string[] memory regions,
    string[] memory silkTypes,
    AuthenticityLevel[] memory authenticities,
    string[] memory qualityGrades
) public onlyOwner nonReentrant returns (uint256[] memory) {
    require(
        recipients.length == tokenURIs.length &&
        tokenURIs.length == artisanNames.length &&
        artisanNames.length == regions.length &&
        regions.length == silkTypes.length &&
        silkTypes.length == authenticities.length &&
        authenticities.length == qualityGrades.length,
        "Array length mismatch"
    );

    uint256[] memory tokenIds = new uint256[](recipients.length);

    for (uint256 i = 0; i < recipients.length; i++) {
        _tokenIds.increment();
        uint256 newItemId = _tokenIds.current();

        _mint(recipients[i], newItemId);
        _setTokenURI(newItemId, tokenURIs[i]);

        // Store silk product information
```

```
            silkProducts[newItemId] = SilkProduct({
                artisanName: artisanNames[i],
                region: regions[i],
                silkType: silkTypes[i],
                authenticity: authenticities[i],
                timestamp: block.timestamp,
                qualityGrade: qualityGrades[i]
            });

            tokenIds[i] = newItemId;
        }

        return tokenIds;
    }

    // The following functions are overrides required by Solidity.
    function _burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) {
        super._burn(tokenId);
    }

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
    }

    // ERC2981 implementation
    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721, ERC721URIStorage, ERC2981)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}
```

F. Environment Configuration

.env:

```
# Alchemy API URL for Sepolia testnet
API_URL="https://eth-sepolia.g.alchemy.com/v2/YOUR_API_KEY"

# Private key from MetaMask
PRIVATE_KEY="your-metamask-private-key-without-0x-prefix"

# Optional: Etherscan API key for contract verification
ETHERSCAN_API_KEY="your-etherscan-api-key"
```

G. Deployment Script

```
scripts/deploy.js:
const { ethers } = require("hardhat");

async function main() {
  console.log("□ Starting SilkNFT deployment...");

  // Grab the contract factory
  const SilkNFT = await ethers.getContractFactory("SilkNFT");

  // Deploy the contract
  console.log("□ Deploying contract...");
  const silkNFT = await SilkNFT.deploy("SilkAuthentication", "SILK");

  // Wait for deployment to be mined
  await silkNFT.deployed();

  console.log("✅SilkNFT deployed to:", silkNFT.address);
  console.log("□ Contract Explorer:", `https://sepolia.etherscan.io/address/${silkNFT.address}`);

  // Optional: Verify contract on Etherscan
  if (process.env.ETHERSCAN_API_KEY) {
    console.log("□ Verifying contract on Etherscan...");
    await hre.run("verify:verify", {
      address: silkNFT.address,
      constructorArguments: ["SilkAuthentication", "SILK"],
    });
    console.log("✅Contract verified on Etherscan");
  }

  return silkNFT.address;
}

// Execute deployment
main()
  .then((address) => process.exit(0))
  .catch((error) => {
    console.error("❌Deployment failed:", error);
    process.exit(1);
  });
```

H. Deployment Commands

```
# Compile contracts
npx hardhat compile

# Deploy to Sepolia testnet
npx hardhat run scripts/deploy.js --network sepolia

# Expected output:
# □ Starting SilkNFT deployment...
# □ Deploying contract...
# ✅SilkNFT deployed to: 0xYourContractAddress
# □ Contract Explorer: https://sepolia.etherscan.io/address/0xYourContractAddress
```

VII. NFT MARKETPLACE INTEGRATION

A. *Modern NFT Platforms (2025)*

Primary Ethereum NFT Marketplaces: 1. OpenSea - Largest NFT marketplace 2. Rarible - Community-owned marketplace 3. SuperRare - Curated fine art platform 4. Foundation - Artist-focused platform 5. Nifty Gateway - Fiat on-ramp supported 6. Axie Infinity Marketplace - Gaming-focused 7. Async Art - Programmable art platform

B. *Listing Process*

Steps to List Silk NFTs: 1. Connect wallet to marketplace 2. Select “Create” or “Mint” option 3. Upload product images and metadata 4. Set pricing (ETH or fiat) 5. Configure royalty percentages 6. Add authenticity certificates 7. Set auction duration or fixed price

VIII. BENEFITS FOR SILK INDUSTRY

1) For Artisans
- Authenticity Proof: Digital certificates verifying handwoven origin
- Fair Compensation: Direct sales without middleman exploitation
- Global Reach: Access to international buyers
- Royalty Income: Ongoing revenue from secondary sales
- Brand Building: Personal reputation system

2) For Buyers
- Guaranteed Authenticity: Blockchain-verified original products
- Complete Traceability: From artisan to final owner
- Investment Value: Potential appreciation of authentic pieces
- Quality Assurance: Detailed product information and grading

3) For Industry
- Market Transparency: Real-time pricing and demand data
- Supply Chain Optimization: Reduced counterfeit circulation
- Economic Growth: Support for rural artisans
- Technology Adoption: Modernizing traditional industry

IX. TECHNICAL IMPLEMENTATION ROADMAP

1) Phase 1: Foundation (Weeks 1-2)
- Set up development environment
- Deploy basic smart contract
- Create MetaMask wallet integration
- Implement basic minting functionality

2) Phase 2: Enhanced Features (Weeks 3-4)
- Add authentication levels and product tracking
- Implement batch minting capabilities
- Create metadata management system
- Add royalty distribution mechanism

3) Phase 3: Marketplace Integration (Weeks 5-6)
- Connect to major NFT platforms
- Implement cross-platform compatibility
- Add bulk listing capabilities
- Create automated pricing mechanisms

4) *Phase 4: Industry Integration (Weeks 7-8)*
- Partner with silk cooperatives
- Implement artisan onboarding process
- Create quality verification system
- Develop mobile application

X. FUTURE CONSIDERATIONS

1) Scalability Solutions
- Layer 2 Integration: Consider Polygon or Arbitrum for reduced fees
- Batch Processing: Optimize for high-volume transactions
- Off-chain Storage: IPFS integration for large metadata files

2) Enhanced Features
- IoT Integration: RFID tags for physical-digital connection
- AI Authentication: Computer vision for authenticity verification
- Supply Chain Tracking: Complete journey from cocoon to consumer

3) Regulatory Compliance
- KYC Integration: Know Your Customer verification for high-value transactions
- Tax Reporting: Automated royalty and capital gains tracking
- Cross-border Regulations: International trade compliance

XI. CONCLUSION

The integration of NFT technology with the silk industry presents transformative opportunities for authenticity verification, fair compensation, and market transparency. While blockchain development research remains in its early stages, the practical applications in business demonstrate significant potential for automating transactions and reducing manual processes.

The combination of digital verification, smart contracts, and NFT marketplaces can address critical challenges faced by the silk industry, particularly in combating counterfeit products and ensuring fair compensation for artisans. This technology enables transparent, honest transactions through networked computation, replacing traditional manual supervision.

As blockchain technology matures and becomes more accessible, we can expect to see increased adoption in traditional industries, leading to more efficient, transparent, and equitable business models. The future of blockchain applications in business research appears promising, with potential for revolutionary changes in how we conduct and verify transactions.

REFERENCES

[1] Investopedia - Blockchain Technology
[2] Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). “Blockchain Technology Overview.” arXiv: Cryptography and Security. doi: 10.6028/nist.ir.8202
[3] Ante, L. (2021). “Non-fungible token (NFT) Markets on the Ethereum Blockchain: Temporal Development, Cointegration and Interrelations.” SSRN Electronic Journal. doi: 10.2139/ssrn.3904683
[4] MIT Technology Review - Ethereum Proof of Stake
[5] Investopedia - Proof of Stake
[6] The Better India - Handlooms and Handicrafts
[7] Chaudhry, N., & Yousaf, M. (2018). “Consensus Algorithms in Blockchain: Comparative Analysis, Challenges and Opportunities.” 12th International Conference on Open Source Systems and Technologies (ICOSST), 54-63. doi: 10.1109/ICOSST.2018.8632190
[8] Zhao, J.L., Fan, S., & Yan, J. (2016). “Overview of business innovations and research opportunities in blockchain and introduction to the special issue.” Financial Innovation, 2, 28.