

Supply Chain Project Guide

Build Tamper-Proof Provenance Tracking

BSc Blockchain, Crypto Economy & NFTs

FS2026

By completing this project, you will:

- 1 Model multi-party business processes on-chain
- 2 Implement role-based permissions
- 3 Create immutable audit trails
- 4 Design verification interfaces

Related Lessons: L13, L15, L27, L28

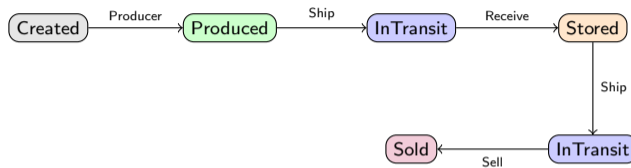
Problem: How do we create tamper-proof records of product journeys?

Role	Permissions	Responsibilities
Admin	Full access	Deploy, manage roles
Producer	Create products	Register new items
Transporter	Update location	Move between stages
Warehouse	Store/release	Custody management
Retailer	Final sale	Sell to consumers
Consumer	Read only	Verify authenticity

Permission Model:

- Role-based access control (RBAC)
- Each action restricted to specific roles
- Admin can grant/revoke roles

Clear roles prevent unauthorized state changes



State Machine Rules:

- Only valid transitions allowed
- Each transition logged with timestamp
- Ownership transfers with custody

State machines enforce valid business logic

Product Structure:

```
struct Product {
    uint256 id;
    string name;
    string origin;
    uint256 createdAt;
    address currentOwner;
    ProductState state;
    string metadata; // IPFS hash
}
```

History Entry:

```
struct HistoryEntry {
    uint256 timestamp;
    address actor;
    ProductState fromState;
    ProductState toState;
    string location;
    string notes;
}
```

Every state change creates an immutable history record

Using OpenZeppelin AccessControl:

```
bytes32 public constant PRODUCER_ROLE = keccak256("PRODUCER_ROLE");
bytes32 public constant TRANSPORTER_ROLE = keccak256("TRANSPORTER_ROLE");

function createProduct(...) external onlyRole(PRODUCER_ROLE) {
    // Only producers can create products
}

function addProducer(address account) external onlyRole(DEFAULT_ADMIN_ROLE) {
    _grantRole(PRODUCER_ROLE, account);
}
```

Benefits:

- Declarative permission checks
- Role hierarchy support
- Event logging for role changes

AccessControl provides enterprise-grade permission management

Events for Audit Trail:

```
event ProductCreated(uint256 indexed productId, string name, address producer);
event StateChanged(uint256 indexed productId, ProductState from, ProductState to);
event OwnershipTransferred(uint256 indexed productId, address from, address to);
```

Why Events Matter:

- Indexed for efficient querying
- Cannot be modified after emission
- Lower gas than storage
- Off-chain apps can listen in real-time

Query Pattern:

```
const events = await contract.getPastEvents('StateChanged', {
  filter: { productId: 123 },
  fromBlock: 0
});
```

Events create a searchable, immutable audit log

QR Code Integration:

- 1 Product has QR code linking to verification URL
- 2 Consumer scans with phone
- 3 Website queries blockchain for product history
- 4 Displays journey from origin to store

Verification Display:

- Authenticity badge (real/counterfeit)
- Complete journey timeline
- All handlers and locations
- Certifications attached

Trust Model:

- First-mile problem: Who enters initial data?
- IoT integration can automate some verification
- Reputation systems for actors

Consumer verification is the ultimate value proposition

Apply the 6 Questions:

Question	Supply Chain Answer
PROBLEM	Verify product authenticity and journey
INCENTIVES	Brands protect reputation, consumers get guarantees
BENEFITS/COSTS	Tamper-proof records; onboarding complexity
FAILURE MODE	Garbage in = garbage out (first-mile problem)
DESIGN CHOICES	On-chain vs off-chain, permissioned vs public
ALTERNATIVES	Traditional barcodes, centralized databases

Key Insight: Blockchain provides tamper-evidence, not truth. Data quality depends on input integrity.

The first-mile problem is the hardest challenge

Food Traceability:

- IBM Food Trust (Walmart, Carrefour)
- Origin tracking from farm to store
- Rapid recall capability (seconds vs days)

Luxury Goods:

- LVMH Aura (Louis Vuitton, Dior)
- Anti-counterfeiting
- Resale authentication

Pharmaceuticals:

- MediLedger Network
- Drug serialization compliance
- Counterfeit prevention

Supply chain blockchain has real enterprise adoption

Phase 1: Core Contract

- Deploy supply chain contract
- Configure roles for test accounts
- Create test products

Phase 2: State Transitions

- Test full product lifecycle
- Verify permission checks work
- Check history logging

Phase 3: Extensions

- Add batch tracking
- Implement certifications
- Add temperature logging

Phase 4: Frontend

- Build verification page
- Generate QR codes
- Create admin dashboard

Standards:

- GS1 Standards: gs1.org
- ISO 22739 (Blockchain terminology)

Platforms:

- Hyperledger Fabric: hyperledger.org/use/fabric
- IBM Food Trust: ibm.com/blockchain/solutions/food-trust

Project Materials:

- Notebook: [projects/notebooks/04_supply_chain.ipynb](https://github.com/ibm-blockchain-ecosystem/projects/notebooks/04_supply_chain.ipynb)
- Web: [.../projects/supply-chain/](https://github.com/ibm-blockchain-ecosystem/projects/supply-chain/)