SMART FOOD SUPPLY CHAIN

Web Platform & Blockchain Module

Complete Technical Specification Document

1. Executive Summary

The Smart Food Supply Chain platform is a comprehensive food traceability system designed for school cafeterias in Italy. The system creates an auditable, blockchain-verified chain of custody from food vendors to the meals served to students, ensuring transparency, food safety compliance, and parent trust.

The platform leverages a hybrid blockchain architecture combining fast private database operations with immutable public chain anchoring, providing both operational efficiency and cryptographic proof of document integrity.

1.1 Key Objectives

- Enable complete ingredient traceability from vendor to student meal
- Provide tamper-proof document verification using blockchain technology
- Allow parents and guardians to verify meal sources via simple QR code scanning
- Ensure compliance with Italian food safety regulations (DDT documentation)
- Create a transparent and trustless verification system

1.2 System Overview

The platform consists of three primary portals (Vendor, School Administration, Consumer) connected through a central blockchain verification service. Documents are hashed using SHA-256, stored in a private MySQL database for fast operations, and anchored daily to a public blockchain for immutability guarantees.

2. Technology Stack

2.1 Core Infrastructure

Component	Technology
Backend Framework	PHP 8.2+ with RESTful API architecture
Database	MySQL 8.0+ (primary data store & private chain)
Content Management	WordPress or Drupal (configurable)
Blockchain	Hybrid: Private DB + Public chain hash anchoring
Document Storage	XML (structured data) + PDF (archive copies)
Hash Algorithm	SHA-256 (document + photo composite)
QR Generation	Server-side QR code library with embedded verification URL

2.2 Blockchain Architecture

The system implements a hybrid blockchain model optimized for cost-efficiency and performance:

Private Chain Layer (MySQL)

- Immediate hash storage upon document upload
- Sub-second verification response times
- Zero transaction costs for internal operations
- Full ACID compliance for data integrity

Public Chain Layer

- Immutable proof of existence timestamps
- Cost-effective gas optimization through batching
- Compatible with Ethereum or Hyperledger public networks

3. User Roles & Permissions

Role	Access Level	Key Permissions
Super Admin	Full System Control	User management, blockchain config, system settings, audit logs
Administration	Operational Management	Menu management, vendor oversight, school admin, reports generation
Vendor	Document Upload	DDT/Invoice upload, photo capture, delivery document management
School	Receipt & Menu Mgmt	Delivery receipt, QR scan verification, menu-DDT linking, menu publication
Family/Student	Public Read-Only	Menu viewing, QR verification, full traceability access (NO LOGIN REQUIRED)

3.1 Role Relationships

- Vendor → School: One-to-Many (single vendor serves multiple schools)
- School → Menu: One-to-Many (school creates multiple menus over time)
- Menu → DDT: Many-to-Many (menu contains multiple DDTs, DDT can appear in multiple menus)

4. System Modules

4.1 Vendor Portal

The Vendor Portal enables food suppliers to manage their delivery documentation and maintain relationships with schools.

Features

- DDT/Invoice upload form with field validation
- Integrated camera for document photo capture
- School selection interface (supports 1-to-many vendor-school relationships)
- Transaction history with status tracking
- Delivery confirmation receipts

Document Upload Requirements

Field	Type	Validation
DDT Number	String	Required, unique
Document Date	Date	Required, not future
Vendor ID	Integer	Required, valid vendor
School ID	Integer	Required, valid school
Product List	JSON Array	Required, min 1 item
Document Photo	Image (JPEG/PNG)	Required, max 10MB
PDF Upload	PDF	Optional, max 25MB

4.2 School Administration Panel

The School Admin Panel is the central hub for managing deliveries, creating menus, and publishing verified meal information.

Features

- Delivery receipt confirmation workflow
- QR scanning interface for blockchain verification
- Menu composition tool with DDT linking
- QR code generation for published menus
- Historical menu and delivery archive

Critical Requirement: Menu-DDT Coverage

> **IMPORTANT: Every menu item MUST be linked to at least one verified DDT before publication. The system will reject any menu that has incomplete DDT coverage.**

4.3 Blockchain Service

The Blockchain Service handles all cryptographic operations and provides verification APIs.

Core Functions

- SHA-256 hash generation from document content + photo binary
- Private chain: MySQL storage for immediate operations
- Public chain: Daily batch anchoring for immutability
- Verification API for hash validation requests

4.4 Consumer Portal

The Consumer Portal provides public access to menu information and full traceability data.

Features

- QR code scanner with camera integration
- Full menu display with ingredient breakdown
- DDT/Invoice visibility for each menu item
- Vendor information display
- Real-time blockchain verification status

Access Model

> NO LOGIN REQUIRED: The Consumer Portal is fully public. Parents and guardians can verify meal sources without creating an account or providing any personal information.

5. Core Data Flows

5.1 Vendor → School Flow (DDT Upload)

This flow describes how delivery documents are captured, hashed, and prepared for blockchain anchoring.

1. Vendor physically delivers products to the school
2. School staff logs into School Administration Panel
3. Staff uploads DDT/Invoice document (PDF or form entry)
4. Staff captures photo of physical document using integrated camera
5. System generates SHA-256 hash combining: document data + photo binary
6. Hash is immediately stored in MySQL (private chain)
7. Hash is queued for daily batch anchoring to public blockchain
8. School receives confirmation with verification status

Hash Generation Algorithm

```
hash = SHA256(document_content || photo_binary || timestamp || school_id)
```

5.2 Menu Publication Flow

This flow describes how school administrators create menus, link ingredients to verified DDTs, and generate the single QR code for consumer verification.

1. School admin creates new menu (daily/weekly)
2. Admin adds menu items (dishes/meals)
3. For each menu item, admin links ALL ingredients to their source DDTs
4. System validates: every ingredient has at least one verified DDT reference
5. If validation fails, system blocks publication and shows missing DDTs
6. Once validated, menu is saved as structured data with full DDT references
7. System generates combined hash: menu content + all linked DDT hashes
8. Single QR code is generated containing verification URL
9. Menu is published to public Consumer Portal

Combined Hash Algorithm

```
menu_hash = SHA256(menu_content || DDT_hash_1 || DDT_hash_2 || ... ||
DDT_hash_n)
```

QR Code Content Structure

```
https://verify.smartfood.example.com/menu/{menu_id}?h={menu_hash_prefix}
```

5.3 Consumer Verification Flow

This flow describes how parents and guardians verify meal sources using the QR code.

1. Consumer scans QR code (printed menu, school display, or digital)
2. Browser/app opens Consumer Portal verification page
3. Portal loads menu data and initiates blockchain verification
4. System checks hash against private chain (MySQL)
5. System verifies anchoring status on public blockchain

6. Portal displays: complete menu with all ingredients
7. For each ingredient, consumer can view: DDT details, vendor info, delivery date
8. Verification status badge shows: Verified ✓ / Pending / Unverified

6. Database Schema

6.1 Entity Relationship Summary

Relationship	Cardinality	Description
School → Document	One-to-Many	Each school receives multiple DDT/Invoices
School → Menu	One-to-Many	Each school creates multiple menus over time
Menu → Document	Many-to-Many	Menu contains multiple DDTs; DDT can be in multiple menus (junction table)
Document → Blockchain	One-to-One	Each document has exactly one blockchain record
Vendor → School	Many-to-Many	Vendors can serve multiple schools; schools have multiple vendors

6.2 Core Tables

schools

```
id (PK) | name | address | region | contact_email | created_at | updated_at
```

vendors

```
id (PK) | company_name | vat_number | address | contact_email | verified |
created_at
```

documents (DDT/Invoices)

```
id (PK) | ddt_number | school_id (FK) | vendor_id (FK) | document_date |
products (JSON) | pdf_path | photo_path | created_at
```

menus

```
id (PK) | school_id (FK) | menu_date | menu_type (daily/weekly) | items
(JSON) | status | published_at | qr_code_path
```

menu_documents (Junction Table)

```
id (PK) | menu_id (FK) | document_id (FK) | menu_item_name | created_at
```

blockchain_records

```
id (PK) | document_id (FK) | hash | hash_type (document/menu) |
private_chain_stored_at | public_chain_tx_hash | public_chain_anchored_at |
verification_count
```

7. API Specifications

7.1 Document Upload API

POST `/api/v1/documents`

Uploads a new DDT/Invoice document with photo capture.

Request Body (multipart/form-data)

1. ddt_number: string (required)
2. school_id: integer (required)
3. vendor_id: integer (required)
4. document_date: date (required, format: YYYY-MM-DD)
5. products: JSON array (required)
6. document_photo: file (required, JPEG/PNG)
7. document_pdf: file (optional, PDF)

Response (201 Created)

```
{ "id": 123, "hash": "a1b2c3...", "status": "pending_anchor", "created_at": "2025-01-15T10:30:00Z" }
```

7.2 Menu Creation API

POST `/api/v1/menus`

Creates a new menu with DDT linkages.

Request Body (application/json)

1. school_id: integer (required)
2. menu_date: date (required)
3. menu_type: enum (daily, weekly)
4. items: array of menu item objects (required)
5. items[].name: string (dish name)
6. items[].ingredients: array of ingredient objects
7. items[].ingredients[].name: string
8. items[].ingredients[].document_ids: array of DDT IDs

7.3 Menu Publication API

POST `/api/v1/menus/{menu_id}/publish`

Validates DDT coverage, generates combined hash, creates QR code, and publishes menu.

Response (200 OK)

```
{ "menu_id": 456, "menu_hash": "x9y8z7...", "qr_code_url": "/qr/456.png", "verification_url": "https://verify.example.com/menu/456", "published_at": "2025-01-15T11:00:00Z" }
```

Error Response (400 Bad Request - Incomplete DDT Coverage)

```
{ "error": "incomplete_ddt_coverage", "missing": [{ "item": "Pasta al Pomodoro", "ingredient": "Tomatoes", "message": "No DDT linked" }] }
```

7.4 Verification API

GET `/api/v1/verify/menu/{menu_id}`

Public API for consumer verification. No authentication required.

Response (200 OK)

1. menu: complete menu object with all items
2. documents: array of all linked DDTs with vendor info
3. verification: blockchain verification status object
4. verification.private_chain: boolean (stored in MySQL)
5. verification.public_chain: boolean (anchored to public blockchain)
6. verification.last_verified: timestamp

7.5 Hash Verification API

GET `/api/v1/verify/hash/{hash}`

Verifies a specific hash against both private and public chains.

Response (200 OK)

1. hash: the queried hash string
2. exists: boolean
3. type: enum (document, menu)
4. private_chain_timestamp: datetime or null
5. public_chain_tx: transaction hash or null
6. public_chain_block: block number or null

8. Security Architecture

8.1 Authentication & Authorization

- JWT-based authentication for admin portals (Vendor, School, Admin)
- Role-based access control (RBAC) with granular permissions
- Session timeout: 30 minutes of inactivity
- Consumer Portal: No authentication (public access by design)

8.2 Data Integrity

- SHA-256 hashing for all documents and menus
- Composite hash includes: document content + photo + timestamp + identifiers
- Hash immutability: once stored, hashes cannot be modified
- Public chain anchoring provides external proof of existence

8.3 Document Security

- Uploaded documents stored in secure, non-public directory
- Access controlled through signed URLs with expiration
- Photo captures include EXIF data stripping for privacy
- PDF documents scanned for malware before storage

8.4 Tamper Detection

The system provides multi-layer tamper detection:

1. Document Modification: Any change to document content invalidates the hash
2. Photo Replacement: Composite hash includes photo binary, preventing substitution
3. Menu Tampering: Menu hash includes all DDT hashes, any DDT change breaks menu verification
4. Database Compromise: Public chain anchoring provides external verification independent of internal database

8.5 Audit Logging

- All document uploads logged with user, timestamp, IP address
- All verification requests logged for analytics
- Menu publication events create immutable audit records
- Failed verification attempts flagged for review

9. QR Code System

9.1 QR Code Generation Strategy

The system generates a SINGLE QR code per menu, not per DDT or per ingredient. This provides a clean user experience while maintaining full traceability.

Approach	Status	Rationale
QR per DDT	**NOT USED**	Too many codes, impractical for cafeteria display
QR per Ingredient	**NOT USED**	Overwhelming for parents, complex printing
QR per Menu	**IMPLEMENTED**	One scan = complete traceability for entire meal

9.2 QR Code Contents

Each QR code encodes a verification URL that includes:

- Base verification domain (configurable per deployment)
- Menu unique identifier
- Hash prefix for quick validation (first 8 characters)
- Optional: school identifier for multi-tenant deployments

9.3 QR Code Lifecycle

- Generated: Upon successful menu publication
- Active: From publication until menu expiration date
- Archived: Menu data remains accessible but marked as historical
- Verification: Always available, even for archived menus

10. Deployment Architecture

10.1 Infrastructure Requirements

Component	Specification
Web Server	Apache 2.4+ or Nginx 1.18+ with PHP-FPM
PHP Runtime	PHP 8.2+ with extensions: mysqli, gd, json, curl, openssl
Database	MySQL 8.0+ with InnoDB engine, utf8mb4 charset
Storage	Minimum 100GB for documents, recommend S3-compatible object storage
SSL/TLS	Required for all endpoints, recommend Let's Encrypt
Blockchain Node	Connection to Ethereum node (Infura/Alchemy) or Hyperledger network

10.2 Batch Anchoring Schedule

Public blockchain anchoring occurs daily to optimize gas costs:

- Anchor Time: 02:00 UTC (configurable)
- Batch Size: All pending hashes since last anchor
- Merkle Root: Single transaction contains root of all daily hashes
- Retry Logic: 3 attempts with exponential backoff on failure

10.3 Monitoring & Alerts

- Health check endpoint: /api/v1/health
- Metrics: document upload rate, verification requests, anchor status
- Alerts: failed anchoring, high verification failure rate, storage threshold

11. Appendix

11.1 Glossary

Term	Definition
DDT	Documento di Trasporto - Italian transport/delivery document required for goods transfer
Hash	Cryptographic fingerprint of data; any change to input produces completely different output
SHA-256	Secure Hash Algorithm producing 256-bit (64 character hex) output
Private Chain	Internal MySQL database storing hashes for fast operations
Public Chain	External blockchain (Ethereum/Hyperledger) providing immutable proof
Anchoring	Process of writing hash data to public blockchain
Merkle Tree	Data structure allowing efficient verification of large datasets with single root hash
Traceability	Ability to track product journey from vendor through to consumer meal

11.2 Compliance Notes

- System designed for compliance with Italian food safety regulations
- DDT documentation requirements per Italian law (DPR 472/96)
- GDPR considerations: minimal personal data collection, no consumer tracking
- Data retention: configurable per regional requirements (default: 5 years)

11.3 Future Enhancements

- Mobile application for iOS/Android with native QR scanning
- Integration with school ERP systems
- Allergen tracking and automated warnings
- Nutritional information integration
- Multi-language support for international deployments