

FoodBlock: A Universal Data Primitive for the Food Industry

Whitepaper, Version 1.2 February 2026

ABSTRACT

The global food industry generates over three trillion pounds in annual economic activity across fourteen sectors, yet operates without a shared data standard. Existing standards address narrow domains: barcodes identify products, regulations mandate reporting formats, labelling rules govern packaging. None provides a universal primitive. This paper introduces FoodBlock: a minimal, content-addressable data structure consisting of three fields (type, state, refs) and six base types capable of representing any food industry operation. The protocol's core axiom, that a block's identity is derived deterministically from its content, provides tamper evidence, offline validity, and deduplication without centralised infrastructure. The key technical insight is that reasoning AI systems now make universal protocol adoption viable for the first time. A protocol can be both rigorous enough for a multinational supply chain (typed, hashed, cryptographically signed) and accessible to a market stall trader who describes a transaction in plain English. The protocol draws on three converging developments: decentralised data models that eliminated the need for central authorities, AI systems capable of reasoning about structured data and acting on behalf of their operators, and social network architectures that demonstrated how trust and reputation emerge from graphs of interactions between participants. Because the protocol is three fields of valid JSON, reasoning AI systems can read, write, and converse through it natively. An agent does not need a custom integration for every system it encounters. It needs only the protocol. This means agents can act on behalf of food businesses, negotiating with other agents, querying businesses directly, verifying certifications with regulatory authorities, and managing inventory, with human approval when necessary. Crucially, these agents do not merely automate existing tasks. They create interactions that would never have occurred: a bakery agent negotiating prices with a supplier agent, a regulatory agent contacting a business to verify an expiring certificate, supplier comparisons that would take days to conduct manually. As humans and AI agents participate, the network becomes denser and more verified, producing network effects that increase value for every participant. Over time, agents develop their own relationships: a bakery agent learns which supplier agents deliver reliably, a supplier agent learns which buyers pay promptly, and these accumulated interactions form an emergent economy of trust and preference that operates continuously on behalf of its human participants. Developers extend this ecosystem by building connected tools: smart fridges that monitor stock levels, POS systems that record every sale, cold-chain sensors that track temperature from warehouse to shelf, all producing the same universal data structure. Food data requires authenticity, traceability, and interoperability, not consensus or

scarcity enforcement. All three are achievable with JSON, SHA-256, and a database. This paper presents the protocol's design rationale, the agent economy it enables, the developer ecosystem it supports, and its implications for traceability, trust, and economic growth across the entire food system.

1. THE PROBLEM

Every major industry operates on a shared data standard. Banking relies on SWIFT, a single message format enabling any institution to transact with any other. The web relies on HTTP, enabling any client to communicate with any server. Logistics relies on GS1 barcodes. Healthcare relies on HL7. In each case, a common data format allows systems built by different organisations, in different countries, using different technologies, to interoperate.

The food industry is the oldest and largest on earth: three trillion pounds of annual economic activity, fourteen sectors spanning farming to retail, regulation to waste recovery, hundreds of millions of businesses, and billions of daily transactions. It has no equivalent standard. A farmer in Wales records a harvest in a spreadsheet. A mill in Somerset logs production in a separate system. A bakery in London tracks inventory in another. A restaurant orders supplies via messaging applications. A food safety inspector files reports as PDFs. None of these systems interoperate. Food data is scattered across millions of incompatible formats, locked inside platforms that do not connect.

This fragmentation has direct consequences. When a food safety alert is issued, regulators cannot instantly identify every retailer selling the affected product; the process relies on phone calls and emails while contaminated food remains on shelves. Consumers have no means of independently verifying provenance claims such as "organic" or "free-range." Small producers face certification processes designed for large corporations, creating disproportionate barriers to market access. Consumer reviews sit on third-party platforms, disconnected from the businesses and products they describe. A restaurant's reputation is scattered across Google, TripAdvisor, and Instagram with no way to link a review to a verified meal or a verified inspection. Community food initiatives (food banks, surplus redistribution, cooperative buying groups) operate in isolation, unable to coordinate because they share no common data layer. The social dimension of food is as fragmented as the commercial one.

Previous attempts at food data standardisation have addressed narrow domains: GS1 barcodes identify products, FDA regulations mandate specific reporting formats, EU labelling rules dictate packaging requirements. Each solves one problem for one jurisdiction. None provides a universal primitive: a

single, minimal data structure capable of representing any food operation, commercial or social, supply chain or consumer-facing, in any sector, in any jurisdiction.

2. THE PRIMITIVE

A FoodBlock is a data structure consisting of three fields: **type** (what the block represents), **state** (the block's properties), and **refs** (references to other blocks). These three fields are sufficient to represent every food industry operation.

Six base types classify every food operation:

Actor: any participant in the food system: a farmer, bakery, restaurant, food safety authority, or consumer.

Place: any location: a farm, warehouse, kitchen, market stall, or delivery vehicle.

Substance: any food item or material: an ingredient, finished product, meal, surplus, or commodity.

Transform: any process that changes food: harvesting, milling, baking, fermenting, cooking, or composting.

Transfer: any movement of food or value: a sale, shipment, donation, or booking.

Observe: any record concerning food: a review, inspection, certification, sensor reading, or publication.

These six types, three representing entities (actor, place, substance) and three representing events (transform, transfer, observe), provide complete coverage of the food industry. Whether the context is a fish market, dairy farm, multinational distributor, or fine dining restaurant, all participants create actors, operate at places, handle substances, perform transforms, execute transfers, and record observations. The types are universal; domain-specific details are held in state.

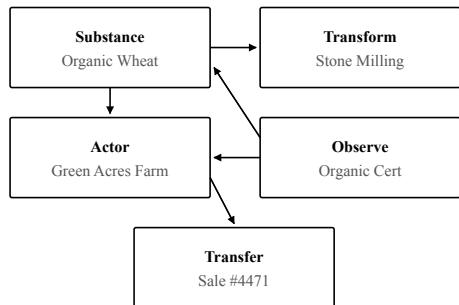


Figure 1. Five blocks linked by refs.

Provenance

The third field, **refs**, is what transforms isolated blocks into a connected graph. Because every block can reference other blocks, a provenance chain forms naturally as food moves through the supply chain. The wheat block references the farm. The flour block references the wheat. The bread block references the flour. The sale block references the bread. Each step is recorded; each step points to its predecessor.

Each participant records what they know; the chain assembles itself as each participant contributes their segment. In a food safety incident, this graph is traversable: beginning at a contaminated block, following every forward reference identifies every affected product, business, and consumer with precision and immediacy.

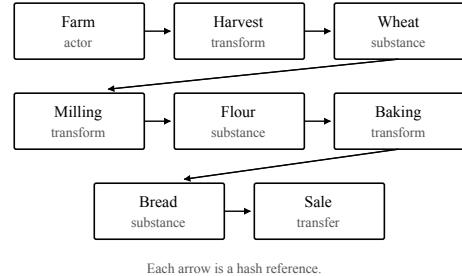


Figure 2. Provenance chain from farm to sale.

Content-Addressed Identity

The provenance chain is only valuable if its data cannot be altered after the fact. FoodBlock achieves this through content-addressed identity: when a block is created, its contents are processed through a deterministic hash function that produces a unique identifier. This identifier becomes the block's permanent identity. Identical content always produces an identical identifier, regardless of when, where, or by whom the block was created.

Any modification (altering a price, changing a certification date, substituting a supplier name) produces a completely different identifier. Every block in the chain is therefore tamper-evident by design. If any link has been altered, the mismatch is immediately detectable by any party. Because identifiers require no network connectivity to compute, blocks are valid at the moment of creation. A producer at a market stall without internet access can record a transaction that is complete and verifiable before it reaches any database.

3. TRUST

In the current food system, trust is primarily acquired through purchase: certification badges, platform listings, and audits. These are point-in-time assessments by single organisations, typically renewed on annual cycles regardless of intervening events.

FoodBlock implements a fundamentally different model. Trust is not a credential conferred by an authority; it is computed from the graph of interactions that accumulates as businesses, customers, inspectors, and certifiers create blocks over time. This model draws on the same principle that underpins social networks: reputation emerges from the pattern of interactions between participants, not from a central score assigned by a single entity.

Three layers of evidence contribute to trust computation:

Authenticity. Every block is cryptographically signed by the creating party. Any recipient can verify that a block genuinely originated from the claimed author.

Verification depth. A claim made by a single party constitutes a self-declaration. The same claim confirmed by an independent customer constitutes peer verification. The same claim certified by a recognised authority (the Soil Association, the Food Standards Agency) constitutes authority verification. Each level carries greater weight due to the increasing difficulty of fabrication.

Consistency over time. A farm that has recorded organic harvests, received positive inspections, and accumulated verified orders over three years possesses a deeper evidence trail than one registered the previous day. Historical depth across multiple independent parties is resistant to manipulation.

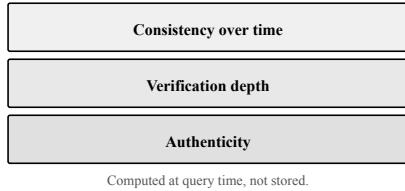


Figure 3. Trust computation layers.

No single entity controls trust computation. Different contexts may weight evidence differently: an organic marketplace may prioritise certifications; a wholesale buyer may prioritise verified transaction history; a consumer may prioritise peer reviews. The evidence is consistent; the interpretation is context-dependent. Trust is always current, computed at the moment of query, not frozen in an annual certificate.

4. THE NETWORK

FoodBlock is not an application. It is a layer that applications, devices, and agents build on top of.

The primitive is deliberately minimal (three fields, six types, JSON) so that any system capable of producing structured data can participate. A POS terminal writes blocks as a side effect of processing sales. A temperature sensor in a refrigerator writes blocks at regular intervals. An AI agent writes blocks in response to a conversation. The protocol does not prescribe how data enters. It prescribes what data looks like once it arrives. This means adoption is incremental: a single bakery can start by recording its products and sales, and derive immediate value. As suppliers, customers, and inspectors join and contribute their own blocks, the bakery's graph grows richer without any additional effort from the bakery itself. The protocol sits beneath existing workflows rather than displacing them.

Each new participant adds value to every existing participant. This is the dynamic that transforms a data format into a network, and it is what creates the surface area for an ecosystem of AI agents, connected devices, and developer tools to emerge.

5. THE AGENT ECONOMY

When AI agents can read and write a shared data protocol, they can act on behalf of every participant in the food system. Not only businesses, but consumers, regulators, community organisations, and the network itself.

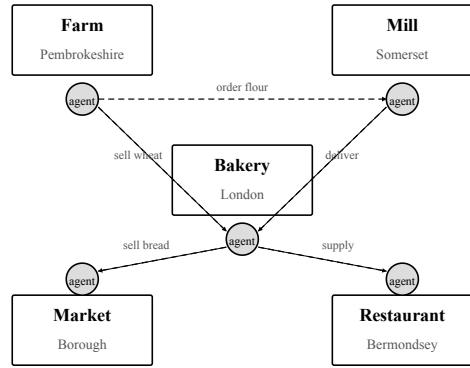


Figure 4. The network. Businesses and agents transacting.

A baker spends the morning making bread, not managing supply chains. A bakery agent that detects low flour inventory queries the network for suppliers, evaluates offers against the operator's stored preferences, negotiates with supplier agents, and presents a draft order for the operator to approve. The operator taps approve and returns to baking. Every step in the exchange is a FoodBlock, signed by its author and referencing the blocks that preceded it. The complete negotiation history is permanent and auditable. Over time the agent learns which suppliers deliver on time, which offer the best prices, and which the operator prefers. Routine procurement proceeds in the background while the owner does what they are actually good at.

But the agent economy extends far beyond procurement. Not every agent negotiates with another agent. A delivery tracking agent monitors shipments in real time, verifying that temperature stayed within range and flagging delays before they affect production schedules. A stock management agent watches inventory levels across multiple locations and triggers reorders before shortages occur. A consumer agent discovers local producers, reads verified reviews, and traces the provenance of a product from farm to shelf before recommending a purchase. A regulatory agent monitors the network for expired certifications, broken cold chains, and provenance gaps, escalating anomalies to human inspectors rather than waiting for scheduled audits. A food rescue agent detects surplus posted by restaurants and bakeries, matches it to nearby food banks and community kitchens, and coordinates redistribution before waste occurs.

The cumulative effect of these agents is not merely automation. It is amplification. A bakery owner would never spend an afternoon contacting the Food Standards Agency to verify that a new supplier's hygiene certificate is current, or reaching out to five farms to compare organic flour prices, or checking whether a delivery driver's cold-chain records show a temperature breach. But an agent does all of this in the background, continuously, without being asked. Each verification, each comparison, each proactive check creates a new block in the graph. The network becomes denser, more trustworthy, and more economically active than any human-only system could sustain. Agents do not replace human judgment. They increase the number of informed interactions in the system, so that human judgment, when it is applied, operates on better evidence.

In each case the pattern is the same: an agent reads the graph, applies its operator's preferences, takes action within defined boundaries, and defers to a human for decisions that exceed its authority. The protocol provides the shared language that makes this possible. Without a common data structure, agents would need custom integrations with every system they interact with. With one, they can transact with any participant in the network from the moment they connect.

6. THE DEVELOPER ECOSYSTEM

A protocol is only as valuable as what people build on top of it. The minimal structure of FoodBlock is designed to enable creativity: not to prescribe what the food industry should look like, but to give developers, businesses, and entrepreneurs the building blocks to shape it themselves.

The barrier to integration is deliberately low. The protocol specifies only the block format, not the hardware, the communication method, or the programming language. A temperature sensor in a walk-in refrigerator writes blocks recording temperature, humidity, and timestamp. A GPS tracker in a delivery vehicle writes blocks continuously, building an unbroken environmental record from warehouse to destination. A POS terminal writes a block for every sale, turning end-of-day reconciliation into a graph query rather than a manual count.

For existing companies, adoption means expanding tools they already sell. A distributor's warehouse management system writes blocks when goods leave the facility. A retailer's receiving system writes blocks when goods arrive. The blocks reference each other across organisational boundaries without requiring shared infrastructure. A POS vendor adds a FoodBlock export to its existing product, and overnight every merchant using that system becomes a participant in the network. The integration cost is low because the protocol asks so little: three fields, valid JSON, a hash.

For new companies, the protocol creates a foundation to build on from day one. A hardware startup designs a smart fridge that tracks stock levels, monitors expiry dates, and triggers reorders through the agent economy described in the previous section. A cold-chain startup builds temperature monitors that produce verifiable, tamper-evident records from warehouse to shelf. A mobile app startup builds a consumer product that scans a QR code on any item and renders its full provenance chain, with every certification, inspection, and review attached as verifiable evidence. These companies do not need permission from FoodBlock, do not pay licensing fees, and do not depend on a proprietary API that might change or disappear. They build on an open protocol.

The developer ecosystem is not limited to professional software engineers, and this is arguably the most significant consequence of the protocol's minimal design. Previous food data standards failed at adoption because the people who produce food data could not write structured data. Reasoning AI systems eliminate this barrier. The protocol's simplicity, three fields of valid JSON, means that an AI assistant can generate working integrations

from a natural language description. A restaurant owner who has never written code can describe what they need and receive a working tool that reads and writes FoodBlocks. The barrier to participation drops from technical literacy to imagination: anyone who can describe a problem can build a solution.

The result is an ecosystem where hardware manufacturers, software vendors, application developers, and non-technical business owners all produce and consume the same data structure. A sensor built in Shenzhen, a POS system built in London, a mobile app built in Lagos, and a bakery owner's custom inventory tool built through conversation all interoperate without ever coordinating with each other. The protocol is the integration layer.

7. OPEN AND FEDERATED

Everything described in the previous sections depends on one condition: no single entity can own the protocol. A food data standard controlled by one company is not a standard. It is a product, subject to changing terms, increasing prices, and discontinuation. FoodBlock is released under the MIT licence with no patents, no royalties, and no registration requirements. The specification is public. The reference implementations are open source. Any party can implement it: university research groups, government agencies, startups, individual producers, and multinational retailers. A standard must function as a public good.

The deployment model reinforces this principle. The protocol is federated, analogous to email. No single entity controls the network. Any organisation can operate its own server. A farm in Pembrokeshire, a mill in Somerset, and a bakery in Bermondsey each retain ownership of their data, on their own infrastructure, under their own control, while maintaining the ability to discover peers, reference each other's blocks, and construct provenance chains that span organisational boundaries. Because a block's identity is derived from its content, the same block stored on multiple servers is provably identical. If one server goes offline, a cached copy on a peer server is equally valid.

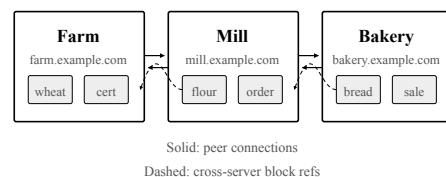


Figure 5. Federated deployment.

This matters because the alternative is platform dependency. When a business builds operations on a proprietary platform, it does not own its data. If the platform modifies its terms or ceases operation, the business loses access to its own operational history. Federation eliminates single points of failure, single points of control, and platform lock-in. Every participant owns their data, chooses their infrastructure, and interoperates with every other participant through the protocol alone.

The protocol itself evolves through its own mechanism. Improvements are proposed as blocks, the community adopts those that prove effective, and consensus emerges from use rather than from authority. Contributing to the protocol requires the same capabilities as using it. The standard grows with the industry it serves.

8. TECHNICAL PROPERTIES

The previous sections describe what the protocol enables. This section describes the technical properties that make it work. These are not implementation details buried in source code. They are architectural decisions that directly affect the reliability, security, and practicality of every operation described in this paper.

Immutability and append-only history. A FoodBlock cannot be edited once created. Its identity is its content, so modifying any field produces a different identity entirely. Updates are expressed as new blocks that reference their predecessors, preserving the full history of every change. A product's price history, a supplier's evolving certifications, an agent's growing preference model are all preserved as chains of blocks. Nothing is overwritten. Nothing is lost. This property is what makes the provenance chains described in Section 2 trustworthy: if any block in a chain had been silently altered, the hash mismatch would be immediately detectable by any party.

Offline validity. Hashing requires no server, no network connection, and no authority. A block's identity can be computed anywhere, by anyone, using only the SHA-256 algorithm and the block's three fields. This means a farmer at a rural market without mobile signal, a fisherman at sea, or a delivery driver in a tunnel can all create valid, signed blocks that are complete and verifiable before they ever reach a database. When connectivity returns, blocks sync to the network and deduplicate automatically: if the same block was created independently by two systems, the identical hash ensures only one copy exists.

The same data always produces the same identity. For content-addressed identity to work across different systems, languages, and platforms, the same block must always produce the same hash. FoodBlock achieves this through a canonical JSON specification: keys are sorted lexicographically at every level of nesting, numbers use no trailing zeros, strings are Unicode-normalised, and null values are omitted. Reference implementations exist in JavaScript, Python, Go, and Swift, and a suite of 124 cross-language test vectors verifies that all four produce identical hashes for identical inputs. If they ever disagree, the protocol is broken.

Every claim has a verifiable author. Every block can be signed by its author using Ed25519 digital signatures. The authentication wrapper records the block, the author's identity, and the signature. Any recipient can verify that a block genuinely originated from the claimed author without contacting any central authority. Signing is what transforms self-declared

data into verifiable claims: a certification signed by the Soil Association carries weight because the signature is cryptographically bound to the Soil Association's public key.

Businesses control who sees what. Not all data should be public. A supplier's cost price, a business's margin, an agent's learned preferences may need to remain confidential. FoodBlock supports field-level encryption using X25519 key exchange and AES-256 symmetric encryption. Individual fields within a block can be encrypted to specific recipients, so a supplier can share product information publicly while keeping wholesale pricing visible only to verified buyers. The encrypted data is part of the block's hash, so the set of recipients is committed at creation and cannot be changed without creating a new block.

Prove one fact without revealing everything. A business may need to prove a specific claim about a block without revealing its entire contents. Merkle proofs enable this: the block's state is structured as a Merkle tree, and a proof for any individual field can be generated and verified without exposing the other fields. A producer can prove that a product is certified organic without revealing its price, its supplier relationships, or its sales volume. The proof is cryptographically verifiable by any party.

Data can be erased without breaking the chain. The append-only model preserves auditability, but legal requirements such as GDPR and CCPA may require content erasure. Tombstone blocks resolve this tension. When erasure is required, the target block's content is replaced with a tombstone marker, but its hash, type, and references are preserved so that the graph structure remains intact. Downstream blocks that reference the erased block can still traverse the chain. The tombstone itself records the reason, the requester, and the timestamp of erasure, maintaining an audit trail of the deletion itself.

Structure is optional but verifiable. The protocol is schemaless by default, accepting any valid JSON in state. This ensures the primitive remains universal. However, interoperability benefits from shared expectations about what fields a given type should contain. Schema conventions, themselves stored as FoodBlocks, define expected fields, required references, and validation rules for specific block types. Validation is always opt-in. A block without a schema declaration is valid. A block with one can be checked against its declared schema by any consuming system.

Conflict resolution. When two parties independently update the same block, the protocol detects the conflict through its author-scope update model. Only the original author of an update chain may create successor blocks. An update from a different author is treated as a fork, a new chain, rather than a replacement. When legitimate multi-author updates are needed, an explicit approval block, signed by the original author, grants update rights to the new party. The approval is itself part of the graph, making every transfer of control visible and auditable.

No consensus mechanism required. Blockchain solved a specific problem: preventing the same digital token from being spent twice without a trusted intermediary. Food data does not

have this problem. Two restaurants can independently claim to serve the best carbonara, and both blocks are valid. No scarcity needs enforcing. No consensus needs reaching. FoodBlock adopts the hash-linked, append-only, signed architecture of distributed ledgers and discards the consensus mechanism entirely. What food data requires is authenticity (signatures), traceability (provenance chains), and interoperability (a universal primitive). All three are achieved with JSON, SHA-256, and a database. The simplification is itself a technical insight: recognising what an industry does not need is as important as providing what it does.

9. IMPLICATIONS

None of the capabilities described in this paper exist in isolation. Provenance is only useful if the data is tamper-evident. Trust is only computable if the graph is open and federated. Agents are only effective if they share a common language. The developer ecosystem is only viable if the protocol is minimal enough for anyone to implement. Each section of this paper depends on every other. The implication is that the food industry does not need better tools in any single category. It needs the layer beneath the tools.

When that layer exists, the economic consequences are significant. Every hour a business owner currently spends chasing invoices, comparing suppliers, or manually reconciling inventory is an hour not spent on the work that actually generates value. A shared data protocol, combined with AI agents that handle routine operations, moves humans higher in the decision-making process. The baker decides what to bake, not which supplier to call. The farmer decides what to grow, not how to format a compliance report. The inspector decides where risk lies, not how to assemble documentation. Human attention shifts from operational mechanics to judgment, creativity, and strategy. The economic activity of the food industry does not shrink when routine tasks are automated. It grows, because the people who understand food best are freed to do more of what they are good at.

The food industry's data fragmentation is not a technology deficit. Databases, platforms, and applications are abundant. It is a primitives problem: the fundamental data structure that allows incompatible systems to produce compatible data does not yet exist. FoodBlock provides that primitive. Three fields, six types, one axiom. The structure is minimal enough for a market stall trader to use through conversation with an AI assistant, and rigorous enough for a multinational supply chain to build automated operations upon. The protocol is open, federated, and designed to be owned by no one and used by everyone.

APPENDIX: SECTOR COVERAGE

The food industry comprises fourteen interconnected sectors. The following reference demonstrates how six base types provide complete coverage without requiring any sector to modify its operations.

Primary Production: farms, fisheries, dairies. Actors produce substances through transforms at places. A dairy farm (actor) at a location (place) produces milk (substance) through milking (transform), recorded with organic certification (observe).

Processing and Manufacturing: mills, factories, abattoirs. Raw substances are transformed into processed substances. A mill (actor) grinds wheat (substance) into flour (substance) through milling (transform), tracked by batch inspections (observe).

Distribution and Logistics: warehouses, shipping, cold chains. Substances are transferred between places. A shipment (transfer) moves flour from a mill (place) to a bakery (place), monitored by temperature readings (observe).

Retail: shops, markets, online stores. Actors sell substances to consumers through transfers.

Hospitality: restaurants, cafes, hotels. Actors transform substances into meals and serve them at places, reviewed by customers (observe).

Food Service: catering, institutional kitchens, meal planning. Transforms of substances at scale, recorded as observations.

Waste and Sustainability: food rescue, composting, redistribution. Surplus substances are transferred to new actors through donations (transfer).

Regulation and Food Safety: inspectors, certifying bodies, standards agencies. Authority actors observe other actors through certifications and inspections.

Education and Media: food writers, educators, content creators. Actors create observations about substances and places.

Community and Social Food: food cooperatives, community kitchens, social enterprises. Groups of actors share substances through transfers and record community events as observations.

Health and Nutrition: dietitians, nutritional analysis, health claims. Professional actors observe substances with assessments and dietary plans.

Food Finance: commodity trading, agricultural investment, insurance. Value is transferred between actors, observed with market data.

Cultural Food: heritage foods, geographical indications, traditional methods. Substances and transforms are observed with cultural certifications tied to places and regions.

Food Technology: novel ingredients, novel processes, food science research. Innovator actors experiment with transforms of substances, observed through publications and experimental data.

No sector requires a type outside the six base types. The three-field structure adapts to every context through subtypes and domain-specific state properties.

FoodBlock is an open protocol released under the MIT licence. This document, like the protocol it describes, is available for unrestricted public use.