

Traceability in the Food Supply Chain

A Hyperledger Fabric Solution

# Points of Contact

|  |  |
| --- | --- |
| Mohammad Jamshed Qureshi | mohammadjamshed.qureshi@georgebrown.ca |
| Shilpa Kaushik | shilpa.kaushik@georgebrown.ca |
| Pratik Patil | pratik.patil@georgebrown.ca |
| Andrew Starling | andrew.starling@georgebrown.ca |

# Project Business Case

Food supply chain is a blockchain based solution which solves the tracking and tracing of food products in the supply chain so that any product can be traced back to its roots. There are many use cases around provenance for food supply chain as it can help troubleshoot supply chain bottlenecks, forecast the shelf life of the fresh produce as its moving through the supply chain, and help pin-point the origin of the product in case of recalls.

* Describe the strategic need for the project.
* Consideration of options: what options were considered before this course of action (undertaking the project) was adopted.
* Expected benefits and timescales: what benefits will the organization receive once this project is complete and how long will these benefits take to achieve.
* Costs: what is the project estimated to cost and how will this be funded.
* Return on Investment: based on the expected benefits and costs, what is the expected return on investment.
* Risks: what are the risks that could affect the business case.

# Solution Design

## System Overview

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

This section provides a brief description of the Systems Design Document’s purpose and scope.

The system design must satisfy the following requirements:

* Tracking and tracing of products through the supply chain. In this particular case we are tracking apples from a farmer/supplier apple orchard. The apples are packaged in boxes onto a palette. Each palette would have a unique identification code. (The tracking granularity is at the palette level for this project).
* The shipment of apples is received at a manufacturer where the apples are processed and made into apple sauce. The apple sauce is put in jars which are then put in boxes. The boxes are placed on a palette. This palette would have a unique identification code. The palette is shipped to a warehouse.
* The palette of apple sauce is unloaded at the warehouse. The ID is scanned to record the shipment received. This triggers payment for the shipment to the manufacturer.
* A palette of products is shipped to the retailer. This palette would most likely consist of different products rather than apple sauce only. The palette would have a unique identification number.
* Enable various stakeholders to have access to the products. Note that the entities upstream in the supply chain may have access to wider data than the entities downstream. So, the ACL scenarios have to be thought out to implement this.
* The data standards – they are optional for this project but GS1 is preferred if possible.
* Business transactions such as PO (purchase order), shipment notification, receipt confirmation.

## High Level System Diagram

A picture containing text, map

Description automatically generated

## Design Constraints

This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design.

## Future Contingencies

This section describes any contingencies that might arise in the design of the system that may change the development direction. Possibilities include lack of interface agreements with outside agencies or unstable architectures at the time this document is produced. Address any possible workarounds or alternative plans.

# SYSTEM ARCHITECTURE

In this section, describe the system architecture for the project.

## System Hardware Architecture

In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components.

## System Software Architecture

In this section, describe the overall system software and organization. Include a list of software modules (this could include functions, subroutines, or classes), computer languages, and programming computer-aided software engineering tools (with a brief description of the function of each item).

# FILE AND DATABASE DESIGN

The section should reveal the final design of all database management system (DBMS) files and the non-DBMS files associated with the system. Additional information may add as required for the particular project. Provide a comprehensive data dictionary showing data element name, type, length, source, validation rules, maintenance (create, read, update, delete (CRUD) capability), data stores, outputs, aliases, and description.

|  |  |  |
| --- | --- | --- |
| GROWER/SUPPLIER | | |
| Field | Description | Format |
| GTIN | Global Trade Item Number | 9504000059378 |
| LOT | Batch/Lot Number |  |
| DATETIME | Date and Time Stamp |  |
| GLN\_FARM | Global Location Number (physical location of farm) | 9504000059361 |
| * Global Trade Item Numbers + Lot Numbers are used to identify raw produce. * Global Serial Shipping Container Codes are used for products distributed from the farms with batch/lot information. * Each physical location is identified with a Global Location Number. | | |

|  |  |  |
| --- | --- | --- |
| MANUFACTURER | | |
| INPUT | | |
| Field | Description | Format |
| GTIN | Global Trade Item Number of Raw Materials | 9504000059378 |
| DATETIME\_RECEIVED | Date and Time Shipment Received |  |
| GLN\_FACTORY | Global Location Number of Factory |  |
| OUTPUT | | |
| SSCC | Serial Shipping Container Code |  |
| DATETIME\_PRODUCED | Date and Time Product Produced |  |
| DATETIME\_PACKED | Date Product Packed |  |
| * Internal traceability is maintained in the key processes: receiving, transformation of raw materials to semi-finished or finished goods, storage and distribution. * GTINs and production data (batch/lot & expiry date) are printed on products to be despatched. * Links between goods received and despatch of final product are always maintained through the GTINs and batch/lot numbers. | | |
| WAREHOUSE | | |
| INPUT | | |
| Field | Description | Format |
| SSCC | Serial Shipping Container Code | 095060001111000017 |
| GLN | Global Location Number of Warehouse | 9506000111247 |
| DATETIME\_RECEIVED | Date and Time Received |  |
| OUTPUT | | |
| SSCC | Serial Shipping Container Code | 095060001112000016 |
| GLN | Global Location Number of Warehouse | 9506000111247 |
| DATETIME\_SHIPPED | Date and Time Shipped |  |
| * Internal traceability is maintained in the processes of receiving, storage and distribution. * Global Trade Item Numbers + batch/lot or expiry date is used to assure accuracy of products when aggregated and disaggregated. These cases are linked and tracked with a pallet using the Serial Shipping Container Code. * Despatch Advice containing Global Trade Item Numbers, Serial Shipping Container Code and Global Location Numbers are sent to customers. | | |

|  |  |  |
| --- | --- | --- |
| RETAILER | | |
| INPUT | | |
| Field | Description | Format |
| GTIN | Global Trade Item Number of Carton/Case |  |
| GLN | Global Location Number of Retail Store |  |
| DATETIME\_RECEIVED | Date and Time Received |  |
| OUTPUT | | |
| GTIN | Global Trade Item Number of Carton/Case |  |
| DATETIME\_UNPACKED | Date and Time Unpacked and put on Shelves |  |
| DATETIME\_SOLD | Date + Time Sold to Customer |  |
| * Retailers provide product information to their customers at Point of Sale. * Master data and event data (for traceability purposes) can be provided to consumers on product labels or through scanning of barcodes. | | |

## Database Management System Files

This section reveals the final design of the DBMS files and includes the following information, as appropriate (refer to the data dictionary):

* Refined logical model; provide normalized table layouts, entity relationship diagrams, and other logical design information
* A physical description of the DBMS schemas, sub-schemas, records, sets, tables, storage page sizes, etc.
* Access methods (such as indexed, via set, sequential, random access, sorted pointer array, etc.)
* Estimate of the DBMS file size or volume of data within the file, and data pages, including overhead resulting from access methods and free space
* Definition of the update frequency of the database tables, views, files, areas, records, sets, and data pages; estimate the number of transactions if the database is an online transaction-based system

## Non-Database Management System Files

In this section, provide the detailed description of all non-DBMS files and include a narrative description of the usage of each file—including if the file is used for input, output, or both; if this file is a temporary file; an indication of which modules read and write the file, etc.; and file structures (refer to the data dictionary). As appropriate, the file structure information should:

* Identify record structures, record keys or indexes, and reference data elements within the records
* Define record length (fixed or maximum variable length) and blocking factors
* Define file access method—for example, index sequential, virtual sequential, random access, etc.
* Estimate the file size or volume of data within the file, including overhead resulting from file access methods
* Define the update frequency of the file; if the file is part of an online transaction-based system, provide the estimated number of transactions per unit time, and the statistical mean, mode, and distribution of those transactions

# INTERFACE

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs.

## Inputs

This section is a description of the input media used by the operator for providing information to the system; show a mapping to the high-level data flows described in Section 1 .2.1, System Overview. For example, data entry screens, optical character readers, bar scanners, etc. If appropriate, the input record types, file structures, and database structures provided in Section 3, File and Database Design, may be referenced. Include data element definitions, or refer to the data dictionary.

Provide the layout of all input data screens or graphical user interfaces (GUTs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary.

This section should contain edit criteria for the data elements, including specific values, range of values, mandatory/optional, alphanumeric values, and length. Also address data entry controls to prevent edit bypassing.

Discuss the miscellaneous messages associated with operator inputs, including the following:

* Copies of form(s) if the input data are keyed or scanned for data entry from printed forms
* Description of any access restrictions or security considerations
* Each transaction name, code, and definition, if the system is a transaction-based processing system

## Outputs

This section describes the system output design relative to the user/operator; show a mapping to the high-level data flows described in Section 1.2.1. System outputs include reports, data display screens and GUIs, query results, etc. The output files are described in Section 3 and may be referenced in this section. The following should be provided, if appropriate:

* Identification of codes and names for reports and data display screens
* Description of report and screen contents (provide a graphic representation of each layout and define all data elements associated with the layout or reference the data dictionary)
* Description of the purpose of the output, including identification of the primary users
* Report distribution requirements, if any (include frequency for periodic reports)
* Description of any access restrictions or security considerations

# DETAILED DESIGN

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

## Hardware Detailed Design

A hardware component is the lowest level of design granularity in the system. Depending on the design requirements, there may be one or more components per system. This section should provide enough detailed information about individual component requirements to

## Software Detailed Design

A software module is the lowest level of design granularity in the system. Depending on the software development approach, there may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs).

If there are many modules or if the module documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Industry-standard module specification practices should be followed. Include the following information in the detailed module designs:

* A narrative description of each module, its function(s), the conditions under which it is used (called or scheduled for execution), its overall processing, logic, interfaces to other modules, interfaces to external systems, security requirements, etc.; explain any algorithms used by the module in detail
* Data elements, record structures, and file structures associated with module input and output
* Graphical representation of the module processing, logic, flow of control, and algorithms, using an accepted diagramming approach (for example, structure charts, action diagrams, flowcharts, etc.)
* Data entry and data output graphics; define or reference associated data elements; if the project is large and complex or if the detailed module designs will be incorporated into a separate document, then it may be appropriate to repeat the screen information in this section
* Report layout

# SYSTEM INTEGRITY

Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled.

Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:

* Internal security to restrict access of critical data items to only those access types required by users
* Audit procedures to meet control, reporting, and retention period requirements for operational and management reports
* Application audit trails to dynamically audit retrieval access to designated critical data
* Standard Tables to be used or requested for validating data fields
* Verification processes for additions, deletions, or updates of critical data

Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.