Final report

Phase 2: Development of a single processor data feed

Project code: V.DIG.0020

Prepared by: Jodi Fleming, Andrew Cooke, Don Wilson, Keenan Tisch
Rezare Systems

Date published: 7 July 2021

PUBLISHED BY
Meat & Livestock Australia Limited
PO Box 1961
NORTH SYDNEY NSW 2059

Meat & Livestock Australia acknowledges the matching funds provided by the Australian Government to support the research and development detailed in this publication.

Abstract

Processors deliver carcase data to the Meat & Livestock Australia (MLA) Group in a variety of ways, resulting in redundant manual processes, additional reconciliations, and a substantial burden for MLA, processors, and processing software vendors that impacts the rate of adoption of new measures and technology in the industry.

This project developed a common data schema that could be used to represent the full range of carcase data currently delivered by processors to MLA, and which is extensible to support additional measures, technologies, and standards in the future.

Engagement with MLA programme teams, processors and software vendors was used to test the proposed schema, and a “walk-through” case study was used to conceptualise how the data schema might be used in a real-world implementation, supported by engagement with two software vendors.

The project delivered a data dictionary, schema model and technical representation, a process map of existing data flows and a conceptual model for future-state data flows. A key recommendation from the project is that MLA move to implement the schema and interfaces in a collaborative prototype project with at least two processors or software vendors.

Executive summary

Background

Currently processors deliver carcase data to the Meat & Livestock Australia (MLA) Group in a variety of ways, including automated delivery between systems, manual file upload through the web site, and even file transfers.

This variety of data transfer mechanisms for similar data sets causes challenges and redundancies:

- It increases costs for MLA programmes, processors, and software providers to support and maintain different delivery processes.
- It impacts data quality and security; particularly where manual uploads occur.
- Delivery of data to producers through LDL can be compromised when manual data uploads are delayed; and
- Processors and software providers perceive there is duplication of effort and are less willing to participate in further MLA programmes that would require more data transfers.

Work carried out by MLA and Rezare Systems during 2019 indicated support from processors and processing software vendors to moving to a single data schema that would address these challenges.

This project was designed to address the question of feasibility and to produce a single schema for carcase data, testing this with MLA programme teams, processors, and processing software vendors. Results from this project will be used to inform and support future pilot implementation programmes.

Project Objectives

The primary objective of this project was to develop a single data schema that could support delivery of carcase data from processors into MLA Group systems, with the aim of reducing effort, costs, and manual errors, and increasing data quality and security. Delivering this objective would involve:

- Documenting the current state data flows for the various types of carcase data provided to MLA programmes, including NLIS, LDL, MSA, DEXA and disease and defect data.
- Reviewing and aligning the data specifications for the MLA programmes that would be required in a single data schema.
- Proposing a conceptual data schema that would meet the data transfer needs.
- Documenting proposed future state data flows and testing these at a conceptual level with MLA.
- Preparing a "case study" to walk through how the new schema might work using input from processors and/or processing software vendors.
- Making recommendations to MLA, including identification of risks to data agreements from moving to the new data schema.

Project Methodology

This project was conducted using a waterfall methodology, with each milestone deliverable building on or from the previous milestone activities. Consultation was undertaken at each stage of the schema development to solicit information and feedback from MLA programme teams, processors, and software vendors.

The current process of processor data capture was documented. This information was then mapped to a proposed future state process, amalgamating all MLA products requirements into a single fit-for-purpose schema, with associated technical documentation.

A written future-state case study demonstrated the proposed carcase data schema could be used with existing processor software systems. Recommendations were provided to address potential risks raised by aggregating the data feeds into one schema for ingestion by the MLA Data Platform. Strategy guidance was also included for participants in the adoption of the single processor feed.

Results/key findings

The project demonstrated that it is feasible to create a new representation of carcase feedback data that can address the combined needs of NLIS, LDL, MSA, Disease and Defect, and Objective Measurement (Lean Meat Yield) programmes. The conceptual data model produced will allow the schema to be extended as new observations and measurement methods are defined and standardised.

Importantly, the extensible data model will allow processors to only supply data that they have collected and in line with the data sharing agreements they might have with MLA. For instance, processors handling only sheep data would not supply beef data observations, and vice versa. Processors that have not signed an MSA data agreement would not need to send MSA-specific data fields, nor would they receive MSA Index data in return. The same approach can be applied to other programmes, such as the Objective Measurement (Lean Meat Yield) programme.

Benefits to industry

The project has generated a single data schema that can represent carcase data (and related entities including processing lots and consignments) in a generalised and extensible way. Combined with effective Application Programming Interfaces (APIs) to connect systems, use of the schema will:

1. Remove current redundant data handling and software interfaces for processors and software vendors as they interact with several MLA programme solutions
2. Support the addition of future measurements, technologies, and standards within the framework
3. Reduce the burden carried by multiple processors, software vendors and MLA products in maintaining multiple data integrations
4. Support a greater degree of automated data reconciliation and more timely data correction, which will reduce manual processes and shorten the time to delivery of feedback
5. Allow MLA to move with greater agility to evolve its programme solutions over time while maintaining a consistent interface for processors

Future research and recommendations

Industry consultation and the further engagement with software vendors as the case study was developed demonstrate that software vendors and at least some processors have a level of enthusiasm for adopting both the schema and a consistent process for delivering carcase data to MLA.

Rezare Systems' primary recommendation is that MLA should proceed forward with a pilot development, creating the framework for an MLA carcase API using the schema collaboratively with at least two potential users (processors and processing software vendors).

The pilot should be carried out collaboratively and use a transparent and agile approach to facilitate learning and increase trust with processors and software vendors.

Table of contents

1. Project Background

Meat & Livestock Australia (MLA) and Integrity Systems Company (ISC) receive carcase feedback data feeds from processors to support industry programmes and producer tools. Programmes that utilise this data include the National Livestock Identification System (NLIS), Livestock Production Assurance (LPA), Meat Standards Australia (MSA), and Livestock Data Link (LDL). Processors deliver carcase data to MLA and ISC in a variety of ways, including automated delivery between systems, manual file upload through the web site, and even file transfers. This variety of data transfer mechanisms for conceptually similar data sets causes challenges:

- It increases costs for MLA programmes, processors, and software providers to support and maintain different delivery processes
- It impacts data quality and security, particularly where manual uploads occur
- Delivery of data to producers through MSA programmes can be compromised when manual data uploads are delayed, or additional reconciliation is required; and
- Implementing additional measurement technologies, standards, or MLA programmes is difficult and expensive for both MLA and all affected processors and processing software vendors, because of the need to create new interfaces.

This project sought to refine earlier data schema work to produce a conceptual data schema and supporting technical documentation for use by MLA technical specialists, processor IT teams and processing software vendors. The material produced in this project will allow these participants to evaluate suitability of the data schema to provide a consistent and modern interface for transferring carcase data to MLA, allowing them to plan implementations and collaborate on a future pilot programme.

Previous work

With support from industry, Rezare undertook a project (V.DIG.1902) to create a data schema for a single carcase data feed to enable processors and software vendors to upload carcase data into key MLA products in a single file. The project delivered the schema with the data requirements for NLIS, LDL, MSA and animal disease reporting and provided recommendations on what to consider in planning a transition to a single carcase schema and standardised methods for ingesting data into MLA and ISC systems.

The single schema included:

- A data dictionary, developed in both Microsoft Word and as GitHub Wiki documentation
- Messages in JSON Schema format, delivered into the Integrity Systems GitHub repository.

Scope of this project

The objective of this second phase project (V.DIG.0020), was to demonstrate how a fit-for-purpose single data schema that could be used to deliver and transform data from processors into the MLA Industry Data Platform.

Key outcomes of the project included mapping the current state for data delivery across multiple services and updating the proposed single carcase schema and data dictionary to meet the detailed needs identified.

The project scope also included development of a written “future state” case study to demonstrate how the process of delivering data could change to use the new schema, and still meet the requirements of both MLA programmes and the processors.

The project also included presentations to representatives of MLA programmes, processors, and processing software vendors. Project results were also presented to MLA, and risks and recommendations for a potential future industry pilot were discussed.

2. Project Objectives

The primary objective of this project was to develop a single data schema that could support delivery of carcase data from processors into MLA Group systems, with the aim of reducing effort, costs, and manual errors, and increasing data quality and security. MLA, processors, and their software vendors would need to be confident that the proposed single data schema would fit current processing workflows, support the needs of MLA and the processors, and be extensible for the future.

Delivering this objective involved:

- Documenting the current state data flows for the various types of carcase data provided to MLA programmes, including NLIS, LDL, MSA, DEXA and disease and defect data.
- Reviewing and aligning the data specifications for the MLA programmes that would be required in a single data schema.
- Proposing a conceptual data schema that would meet the data transfer needs.
- Documenting proposed future state data flows and testing these at a conceptual level with MLA.
- Preparing a “case study” to walk through how the new schema might work using input from processors and/or processing software vendors.
- Making recommendations to MLA, including identification of risks to data agreements from moving to the new data schema.

The project achieved this objective, and the project team received significant support and valuable feedback from both MLA technical specialists and the developers/vendors of processing software.

3. Methodology

Rezare Systems carried out several activities in this project to develop a future-state model of processor data capture and to demonstrate how this might be used. A waterfall methodology was used with five main milestones. The deliverables from each milestone informed subsequent milestones.

1. Undertake project initiation and planning activities, coordinating with MLA.
2. Meet with MLA programme teams to document current-state process flows for data delivery from processors to the various programmes. The deliverable from this work was a current-state process flow.
3. Using carcase data specifications from each of the MLA programmes under consideration (and previous data schema work), create a data dictionary and conceptual combined data schema. The

deliverables from this work were the data dictionary and conceptual schema in diagrammatic form. These were presented to MLA personnel to facilitate a go/no-go decision.

4. Undertake industry consultation with processors and processing software vendors to obtain feedback on the data schema, its structure, and the proposed mechanisms of use. The deliverable for this phase was a milestone report that included feedback and questions raised by industry participants. MLA were able to review the feedback from industry participants, and a go/no-go point allowed the project to be terminated if feedback indicated the single carcase data schema would not be acceptable.
5. Engage with at least two processors and/or processing software vendors to develop a case study that walks through the proposed new process flow to assess the feasibility of implementing the single carcase schema to provide data to MLA.
6. Final deliverables included the case study report, presentation of project results to ISC and MSA key representatives, discussion of key learnings from the project, and recommendations for a potential future industry pilot.

4. Results

The project demonstrated that it is feasible to create a new representation of carcase feedback data that can address the combined needs of NLIS, LDL, MSA, Disease and Defect, and Objective Measurement (Lean Meat Yield) programmes. The conceptual data model produced will allow the schema to be extended as new observations and measurement methods are defined and standardised.

The proposed schema is a substantial departure from the previous approaches. Other than recent iterations of MSA data transfer, most interactions between processors and MLA programmes have been based on flat-file communication, with specific columns for the various data elements. The proposed schema instead defines several conceptual entities that represent key data elements: consignments, processing lots, supply chain locations, and sets of extensible carcase observations. Depending on the needs of MLA and processors, these entities can be linked, combined, or embedded in different contexts or at different points of the data gathering and delivery process.

Importantly, the extensible data model allows processors to only supply data that they have collected and in line with the data sharing agreements they might have with MLA. For instance, processors handling only sheep data would not supply beef data observations, and vice versa. Processors that have not signed an MSA data agreement would not need to send MSA-specific data fields, nor would they receive MSA Index data in return. The same approach can be applied to other programmes, such as the Objective Measurement programme that is capturing DEXA data for Lean Meat Yield.

4.1 Project results and deliverables

Project deliverables included:

- Updated documentation of the **current-state process for processor data capture** noting how and when carcase data is collected (this diagram is shown in Figure 1, with further information available in the case study located in Appendix C).

- A **proposed future-state process for processor data capture** showing use of the new schema and APIs (Figure 2).
- Final versions of the **proposed single processor carcase schema**, **data dictionary**, and **JSON files** incorporating MSA, animal disease and Lean Meat Yield (DEXA) available at the project GitHub repository
- A **non-technical schema overview document** aimed at industry stakeholders (found in Appendix A).
- A **written short-form case study** (located in Appendix C) demonstrating how processor data could be transformed into a fit-for-purpose single data upload into the MLA Industry Data Platform, including:
    - **Prototype API specifications** to allow developers in MLA and processor or software vendor organisations to assess the potential feasibility of interacting with an API that implements the schema
    - **Recommendations on any risks** presented by current data sharing agreements created by aggregating the data feeds into one schema for ingestion by the MLA Data Platform.
    - **An implementation strategy** to guide participants in the adoption of the single processor feed.

Rezare Systems presented project results to ISC key representatives and discussed key learnings from the project, and recommendations for a potential future industry pilot (Appendix D).

4.1.1 Current-state process for processor data capture

During system analysis, Rezare Systems had developed a current state model that shows how information is exchanged with MLA programmes during the flow of animals and carcases from booking to boning and packing. This flow is illustrated in Figure 1.

- Processors currently obtain sets of property residue status and animal device extended residue programme status from NLIS overnight based on movements recorded by producers. These are used to query the status of animals on arrival.
- After completing full reconciliation of carcase data (after the end of a shift), and correcting any obvious data entry errors, slaughter records and carcase data is delivered to NLIS to meet the 48-hour requirement for return of such data.
- MSA and AUS-MEAT observations needed to support MSA are delivered to MSA so that MSA programme eligibility can be confirmed and the MSA Index returned.
- Data from NLIS, MSA, and other programmes is synchronised to the MLA data warehouse, and then made available in LDL.

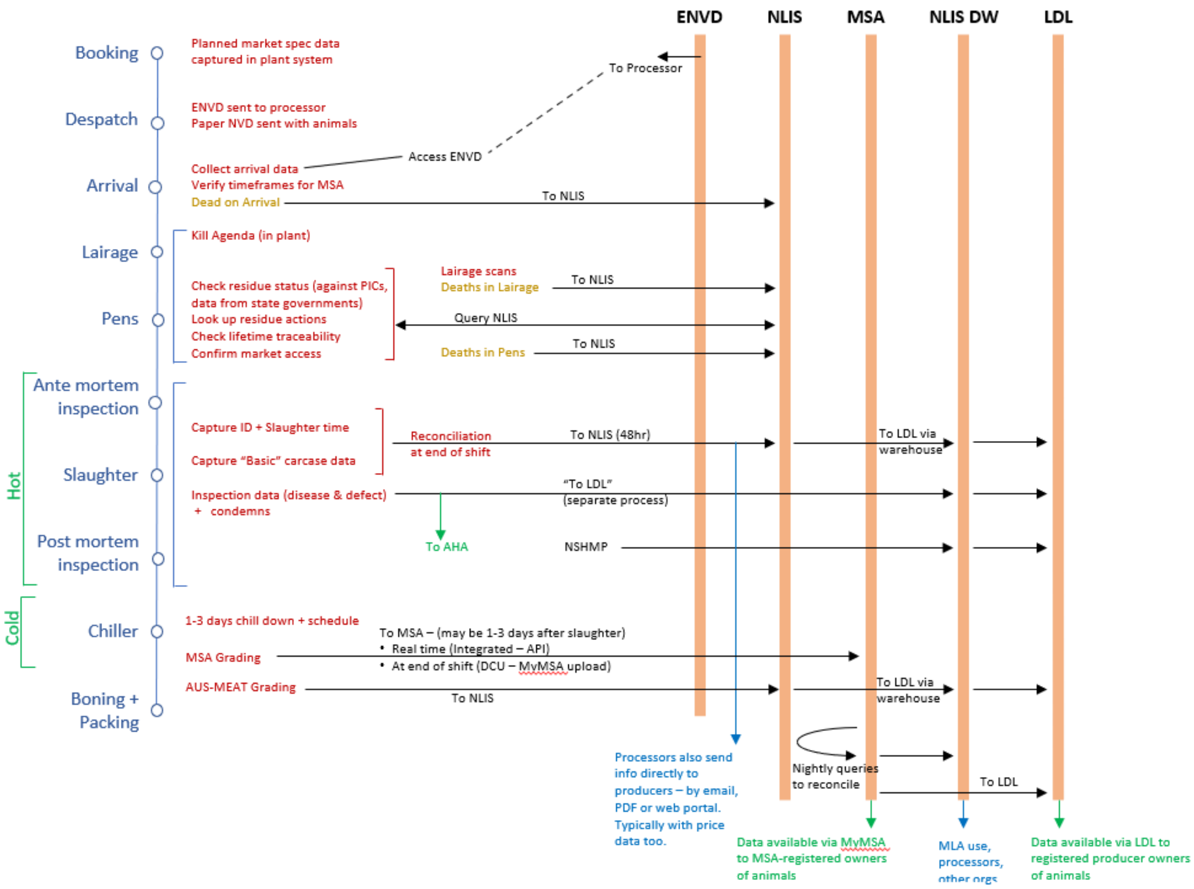


Figure 1: Carcase data process flow - current state

4.1.2 Conceptual future-state process for processor data capture

A future state process flow was developed and tested with MLA programme teams and processing software vendors. This conceptual process flow is shown in Figure 2. Numbers shown in the diagram are references to the data flows, elaborated further in **Appendix B: Details and examples of conceptual API.**

The primary feature of the conceptual design is a new API that mediates between processor software and the MLA programme systems. The new API:

- Releases processors and their software vendors from needing to understand the internal endpoints and detailed requirements of the various MLA programmes, allowing each programme to evolve at a pace that suits its needs without requiring changes by processors and vendors (apart from new metrics).
- Supports asynchronous and incremental delivery of data from processors to MLA. This allows processors to deliver data as it becomes available, and then to revise data if errors are found that need correction.
- Supports a common and flexible model for identifying participants that should have access to carcase data in a way that should work for MyMSA, LDL, and future programmes.
- Can support "smart measurement devices" and a variety of future architectures for processing software solutions, including distributed and microservice architectures.

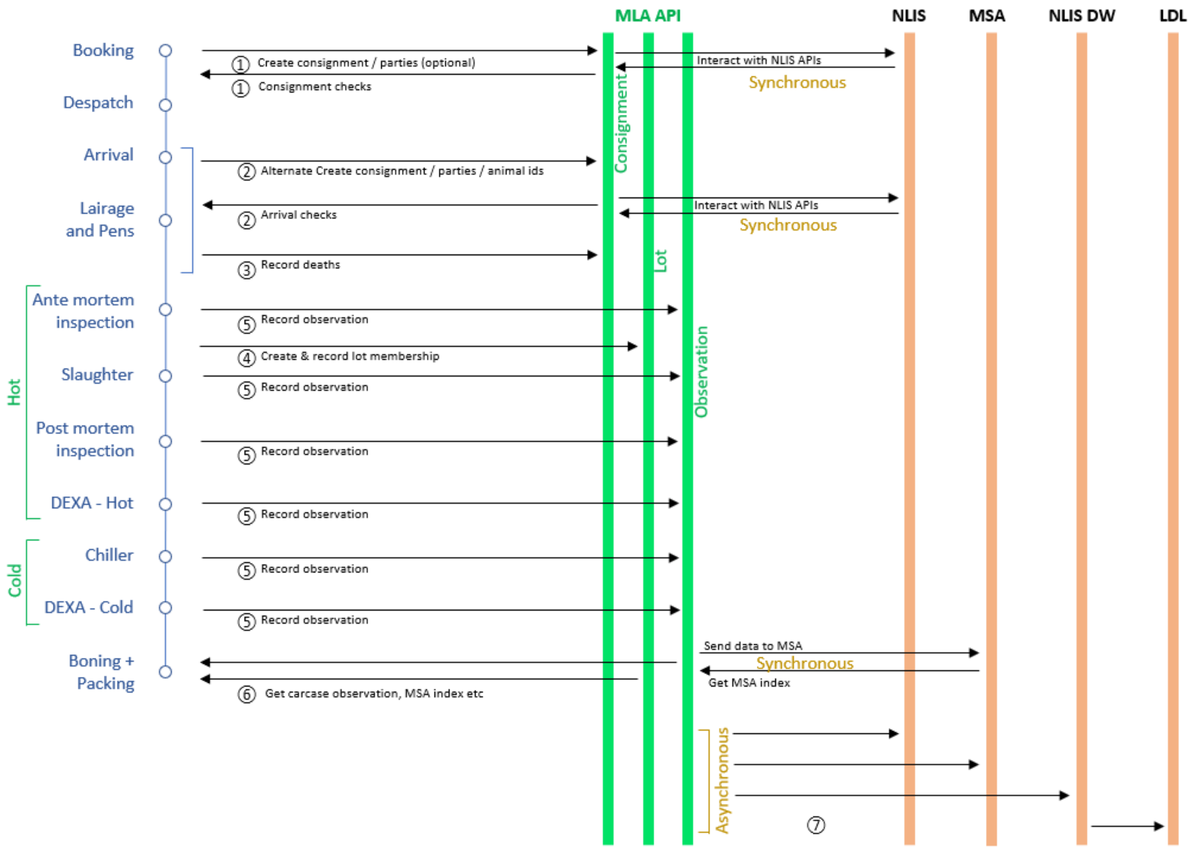


Figure 2: Carcase data process flow - future state

4.1.3 Conceptual Data Model

A conceptual data model shows the relationship between conceptual entities and provides examples of the data elements within those conceptual entities.

The conceptual data model[1] (Figure 3) designed by Rezare Systems for the single carcase schema was used as a prompt for discussions with MLA, ISC, software providers and processors throughout this project, and to clarify the relationship between data entities.

Rather than grouping specific carcase attributes into entities that may be collected in a single process, we have taken a more generalised tactic with each attribute recorded as an observation. Observations recorded by an operator at the same time are grouped as an observation set.

The final version of the conceptual data model is available at the project GitHub repository.

[1] The conceptual data model design is based on modelling undertaken by the W3C (World Wide Web Consortium) schema.org community group, and by the ICAR (International Committee for Animal Recording) animal data exchange working group and their body of work, though it is specific to carcase data and the information provided by the workshop participants.

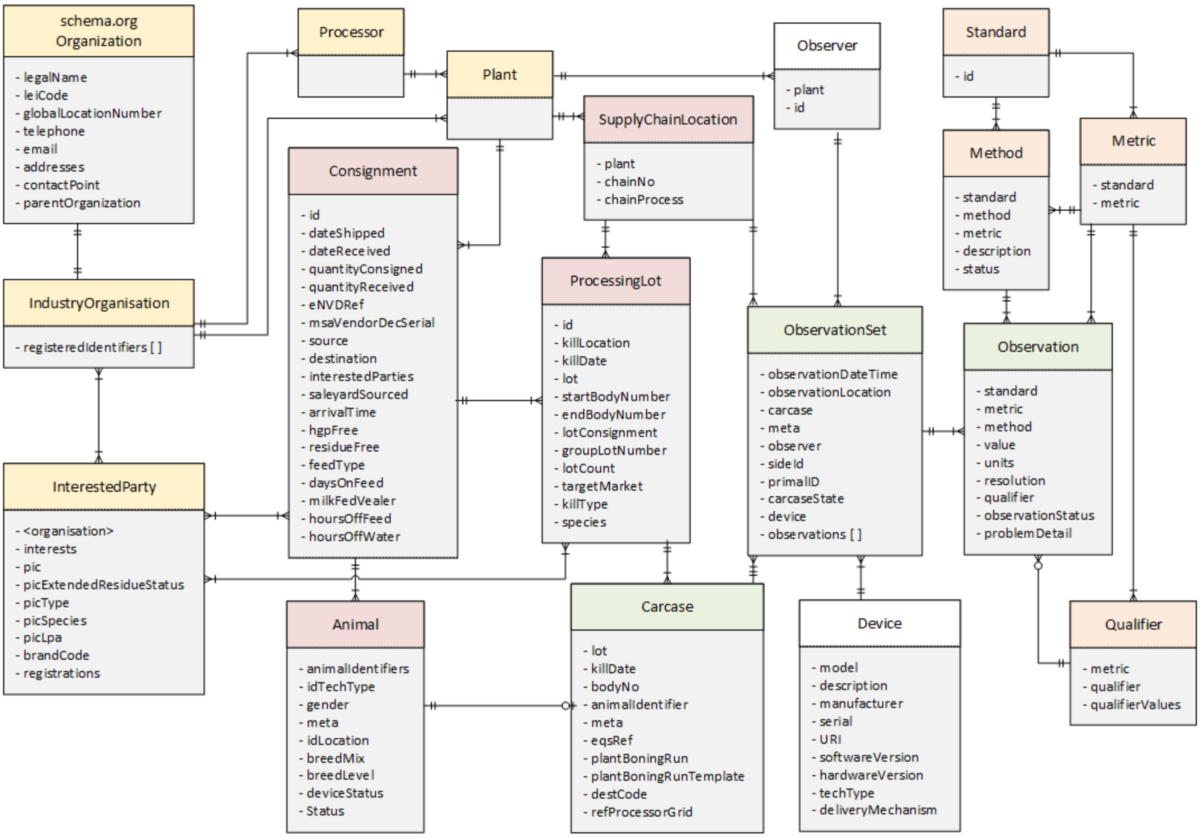


Figure 3: Conceptual data model

Note that the schema itself shows linkages – typically via identity (reference) or via containment. For instance, an ObservationSet contains Observations. Observations refer to the Standard, Metric, and Method by reference (name).

The data model is coloured to aid readability:

Yellow Entities represent industry participants such as processors, plants, source and producer PICs, and operators. In keeping with the approach pioneered by ICAR and used in the original draft schema, these are derived from the "Organization"[2] entity documented at schema.org. This entity provides many useful data elements, including the legal name and global location number of an organisation or site, contact information, and leiCode (legal entity identifier code -- the international representation of an ABN). Extensions to the "Organization" entity allow definition of industry organisations that have a set of identifiers (such as MSA and AUS-MEAT registrations for plants).

Red Entities provide information about a lot of animals and the context or supply chain location in which carcase information is recorded. These entities include a "consignment" of animals received by a plant with a declaration; a "lot" of animals from that consignment which

[2] Words in the conceptual data model, JSON and data dictionary use American spelling, to be consistent with ICAR. All other word spellings in this report adhere to Australian/New Zealand spelling dictionaries

are processed together in sequence; a "supply chain location" including plant, chain, and location identifier, and the identification details for each animal (such as NLIS identifiers).

Orange Entities are reference values that support identification of the specific standard (including AUS-MEAT and MSA standards), metric (for instance "Fat Depth", "Hot Standard Carcase Weight", or "Marbling") and method and qualifier used for an observation.

Green Entities will support delivery of carcase observations, identifying the carcase, the date, time, and point of observation, and a set of observation values -- each with its standard, metric, method, and optionally qualifier.

Rather than grouping specific carcase attributes into entities that may be collected in a single process, a more generalised approach is used, with each attribute recorded as an observation. Observations recorded by an operator at the same time are grouped as an observation set.

The benefits of this approach include much greater flexibility. As soon as an observation has been made it may be transmitted to the data warehouse without waiting for other observations, separating the dependency between data delivery and its use in calculations or analysis.

For instance, at this point the data may be incomplete from an MSA perspective, but it might be suitable for use in NLIS. When the MSA calculations are required, the system would need to check that all the data required for the calculation for an animal has been uploaded. The way the validation is done will need to change but it will have the same effect of ensuring all the required data is present.

Good API design practice would be to separate the data delivery to MLA from the request for results. For instance, a processor could POST/PUT new or updated metrics for some animals, then GET the results – if not all results were available, they might get results for animals where this was available, an "insufficient data" message where not all required data was available, and an error message if some metrics were invalid or out of range (including conditional mandatory fields).

Depending on how the APIs are defined to use this schema, data could be submitted for a shift (risky if poor internet) OR a processing lot (preferred at present, probably separate by hot or cold) OR as individual animals pass by a grading station or device (potential in the future).

There are advantages from working with smaller blocks of data. Currently data for an entire lot (or shift) can be held up if there are validation errors in one animal. If data is submitted by individual animal, a validation error need not delay calculations for other animals. Secondly, fewer transmission problems may be experienced by sites on a poorer internet connection by reducing the data payload blocks.

The generalised, flexible approach provides for operational differences between plants and copes better with any future changes or new technology. Plants that process just one species will only populate the fields that relate to their entities and animals. They will only send the metrics that match their species.

4.1.4 Data Dictionary

A data dictionary is a collection of names (including synonyms and past names), definitions, and attributes used in an information system. A Data Dictionary was compiled for the carcase schema based on the data dictionaries, file descriptions, and other documentation provided for each of the current and planned carcase integrations.

The **data dictionary** developed for the common carcase schema was added to the Wiki in the project GitHub repository, and is sorted by class and then attribute (https://github.com/Integrity-Systems-Company/common_carcase_schema/wiki/3.-Data-Dictionary).

For the common carcase schema, the data dictionary is simpler than might be anticipated by examining the source documentation or data dictionaries:

- Fields that were previously documented separately such as Establishment ABN and Operator ABN for Disease and MSA (two fields and two source documents) can be represented by a single definition -- the LEI Code or ABN -- because the conceptual object model allows multiple organisation entities, each with different "interests".
- Many observations that were previously represented separately in source document data dictionaries can now be represented by the "value" field for each observation entity, providing the observation entity also has a correct combination of standard, metric, method, and units. Lists of these standards and metrics, together with their methods, units, and any qualifiers will be required to implement the schema.

4.1.5 JSON Schema

A schema that would support representation of the data using JSON (JavaScript Object Notation[3]) has been developed, using the JSON-Schema specification[4] (draft 7).

The JSON Schema files are not included in this report but are available at the project GitHub repository https://github.com/Integrity-Systems-Company/common_carcase_schema.

JSON Schema files in the repository are organised into folders that include:

- "Types" – files that define data types which are included by reference in entities. An example of this is iscCarcaseMetric.json which defines metrics.
- "Enums" – files that define data types with a restricted set of values. An example of this is iscSpecieType.json which defines the set of livestock species values.
- "Resources" – files that define complex data types representing an entity. For instance, iscCarcaseObservation.json.

[3] JSON is the ECMA-404 data interchange standard defined at https://www.json.org/json-en.html
[4] The specification for JSON Schema can be found at https://json-schema.org/

4.1.6 Non-technical schema overview document

A non-technical schema overview document was created to provide industry participants with an overview of the schema, potential API processes and how to access and add to the information contained on the project GitHub repository.

The document covered the following topics:

- Overview of the schema structure
- Discussion of the main "components"
- How might we use this? API structures and processes
- Using GitHub issues for feedback
- Specific feedback items

This single carcase schema industry overview pack is attached to this report as **Appendix A**.

4.1.7 Written short-form case study and prototype API specifications

A future-state process flow was developed using the process diagram already referenced in Figure 2 and the conceptual data model shown in Figure 3. The full case study is available in Appendix C and is summarised here.

Key interactions through the conceptual API are shown in Figure 2 using numbered interactions. These interactions equate to:

1. *Consignment creation and checks at Booking*
    - *Send new consignment data to the MLA API*
    - *Retrieve the consignment from the MLA API to check validity and residue status*
2. *Consignment creation or update and checks at Arrival*
    - *Send new or updated consignment data to the MLA API*
    - *Retrieve the consignment from the MLA API to check validity and residue status*
    - *Send animal device identifiers for the consignment to the MLA API*
    - *Retrieve the list of animal device identifiers with extended residue programme status*
3. *Animal recording, checks, and death or dead-on-arrival (DOA) recording in Lairage*
    - *Send animal device identifiers and status to the MLA API*
4. *Assignment of carcases (and animal devices) to processing lots*
    - *Send processing lot identification (and link to consignment) to the MLA API*
    - *Send the carcases (and device identification) for a processing lot to the MLA API*
5. *Recording observations, including AUS-MEAT, MSA chiller assessments, DEXA, and disease and defects at any time or at shift end*
    - *Send one or more observations recorded for one or more carcases at a point in processing to the MLA API*
6. *Checking data and retrieving MSA eligibility confirmation and MSA Index.*
    - *Notify the MSA API that the processor considers all MSA required data for a processing lot has been delivered, receiving feedback of any problems*
    - *Retrieve any observations for carcases or a lot from the MSA API, including derived observations such as the MSA Index.*

Conceptual API Examples were provided to assist participants to understand how the APIs may interact with data collected by or required by processors. These are shown in more detail in **Appendix B**.

The lists of attributes included in these specifications are not exhaustive, rather they are a selection to demonstrate the approach. Similarly, the examples of observations show just a few of the many possible metrics.

Feedback from Processing Software Vendors

In addition to engaging with the MLA programme teams, Rezare Systems discussed the schema, conceptual API and process flow with two software vendors who produce processing software used in Australia. Both vendors provide solutions to small and large processing companies, including:

- A significant meat processor handling sheep, goats, and beef cattle in NSW, VIC, SA, and TAS
- A significant meat processor mainly handling beef in QLD, NSW, and SA.

The software vendors noted that there are three times primary that their software interacts with MLA systems (the software may interact with multiple systems at these times). These are:

- Receiving nightly data regarding PIC status and animal residue status based on NLIS movements
- Sending NLIS and other data to several MLA systems after a shift has concluded and all reconciliations and checks have been carried out
- Sending data to MSA in order to receive an MSA Index prior to animals for a lot being scheduled for boning and packing.

The vendors believed that they could move towards the following interactions:

- Interacting with the MLA API during the booking process when the consignment is first created, which would allow them to plan based on the residue status of the source PIC.
- Interacting with the MLA API during arrivals and in lairage to confirm the arrival of the animals and check both PIC residue status and animal extended residue status, and to check the association and identification of interested parties.
- One software vendor has been developing a standalone smart reader for use in lairage, and the MLA API would allow this device to interact in a distributed mode.
- Interacting with the MLA API in order to receive an MSA Index prior to animals for a lot being scheduled for boning and packing (as they do presently).
- Interacting with the MLA API to send or update all carcase observations and make any corrections to consignment details after a shift has concluded and all reconciliations and checks carried out. This is in line with what is presently done but could be earlier than at present if checks on interested parties and other identifiers can be carried out earlier.

Both vendors noted that communications environments and security at plants varies.

a. Plant control systems are usually not directly internet-connected so intermediary systems are used, which can influence timing of interactions.
b. Vendors appreciated the flexible way of structuring lot and carcase observation data so that data can be sent in small batches to overcome unreliable internet connections.

Further feedback from MLA Programme Teams

Rezare Systems also obtained feedback from representatives of some MLA programmes likely to use the new schema and API. MLA programme teams were broadly supportive of the move to a modern JSON schema, and at least one programme had already carried out work in this area. The MLA programme teams noted that:

- While the conceptual API provides a REST (representational state transfer) interface using JSON Schema, the Integrity Systems Company development team will make final decisions regarding implementation.
- Asynchronous interfaces from the conceptual MLA API to MLA programme systems allows data to initially be cached and delivered to MLA programmes much as it is presently, but also to evolve and become more real-time as those systems are updated.
- Synchronous interfaces between the MLA API and internal systems or cached data will be required to support queries for source property residue status and animal extended residue programme status.
- A synchronous interface from the MLA API to the MSA programme system will be required when a processor asserts completion of the MSA data for a lot or attempts to obtain the MSA Index for those values. This will allow additional checking of conditional mandatory fields.

5. Conclusion

Development of the common schema for carcase data focused on creating a conceptual schema that would meet the data needs of current MLA programmes (particularly NLIS, MSA, LDL, DEXA and disease and defects), yet be extensible for future programmes and new measurement types, methods, and standards.

In contrast to the “flat-file-oriented” approach used in many existing data flows, the proposed schema instead defines several conceptual entities that represent key data elements: consignments, processing lots, supply chain locations, and sets of extensible carcase observations. Depending on the needs of MLA and processors, these entities can be used in various combinations for unique APIs.

The new schema retains all the currently used identifiers but generalises how they are structured. Plants and the interested parties (properties, producers, and operators) are all based on a general “Organisation” which may be identified by an ABN (a LEI or legal entity identifier in international standards), a PIC (where relevant), or one of several registration identifiers (such as AUS-MEAT establishment, state or federal government registration, or MSA establishment or member registration).

Rather than a fixed, column-oriented approach for observation data, the proposed carcase data schema uses an extensible model that supports any observation, provided the standard (e.g., AUS-MEAT), metric (HSCW), method (e.g., chain scale), and units (KGM) are specified. The schema removes potential confusion by making these components explicit rather than implied.

The project generated several reports and reusable intellectual property. This includes:

- A data dictionary in GitHub Wiki format

- Schema documentation in both diagrammatic form and as JSON Schema format (in the GitHub repository)
- Example API definitions in OpenAPI format (in the GitHub repository).

5.1 Key findings

Participating processors and vendors of processing software showed enthusiasm for the changes in approach and a willingness to embrace both the new schema and the resulting changes to connectivity (APIs) that would be needed to adopt the schema. These changes include:

- Delivering data to all MLA programmes through a common schema and interface.
- Delivering observations and carcase information that is needed for the programmes in which the processor is enrolled.
- Adding and updating data through new, common API methods, for instance using "PUT" to send data for consignments, processing lots, and sets of carcase observations.
- Sending data in smaller "chunks" (including at end of lots, or even from intelligent devices in the chain or in lairage), which has the added benefit of reducing transmission failures in remote sites. Where a processor wishes to send all data at the end of shift, the data can still be transmitted in logical chunks.
- Identifying the interested parties to each consignment of animals (a similar process to that followed for MSA data at present, and potentially valuable for LDL permissions too).
- Using standard, metric, and method to define each of the observations – an extensible scheme that will support current MSA and AUS-MEAT standards as well as DEXA, Disease and Defect, and future standards.

In discussions sessions with processors, they expressed enthusiasm to see how the single carcase schema to deliver data to the MLA Industry Data Platform would work in the form of a processor and MLA pilot.

Importantly, the extensible data model will allow processors to only supply data that they have collected and in line with the data sharing agreements they might have with MLA. For instance, processors handling only sheep data would not supply beef data observations, and vice versa. Processors that have not signed an MSA data agreement would not need to send MSA-specific data fields, nor would they receive MSA Index data in return. The same approach can be applied to other programmes, such as the DEXA lean meat yield programme.

5.2 Benefits to industry

The project has generated a single data schema that can represent carcase data (and related entities including processing lots and consignments) in a generalised and extensible way. Combined with effective Application Programming Interfaces (APIs) to connect systems, use of the schema will:

1. Remove current redundant data handling and software interfaces for processors and software vendors as they interact with several MLA programme solutions

2. Support the addition of future measurements, technologies and standards within the framework
3. Reduce the burden carried by multiple processors, software vendors and MLA teams in maintaining multiple data integrations
4. Support a greater degree of automated data reconciliation and more timely data correction, which will reduce manual processes and shorten the time to delivery of feedback
5. Allow MLA to move with greater agility to evolve its programme solutions over time while maintaining a consistent interface for processors

Processing software vendors also identified opportunities for improved efficiency and data handling, primarily by being able to use the Consignment object to check PIC status and parties after a booking is received. This will allow more effective planning of cuts and markets and improve data quality (and potentially reduce reconciliation time).

6. Future research and recommendations

Industry consultation about the carcase data schema, and the engagement with software vendors as the case study was developed demonstrate that software vendors and at least some processors have a level of enthusiasm for adopting both the schema and a consistent process for delivering carcase data to MLA.

We make the following recommendations to MLA for future investigation and implementation.

1. Use of a standard, extensible JSON schema and compatible APIs is generally welcomed by industry. This should be developed in a collaborative approach with industry. The recommended next step is a pilot co-development programme during which MLA would define the API alongside early-adopter processors or software vendors who would develop prototype data-delivery functionality in their systems.
2. Deliver the proposed API as an integration layer to reduce the resource and timing impacts of simultaneously changing the interfaces for several MLA programme solutions. An integration layer could cache transmitted data and translate requests into formats that the existing MLA programme solutions understand. Once stabilised and proven, the existing solutions could be modified to make direct use of the integration layer as required. This will improve agility in the development of individual MLA programme systems and improve testability.
3. Emphasise communication and fit-for-purpose documentation during the collaborative pilot initiative and beyond. This should include publishing data dictionaries, standards, metrics and methods into GitHub repositories, and publishing API specifications as OpenAPI specifications so they can be readily adopted by software developers. We note that a detailed list of metrics, methods, and qualifiers for each of the standards (AUS-MEAT, MSA, and disease and defect and DEXA programmes) was not fully undertaken for this project and could be considered for future activities.

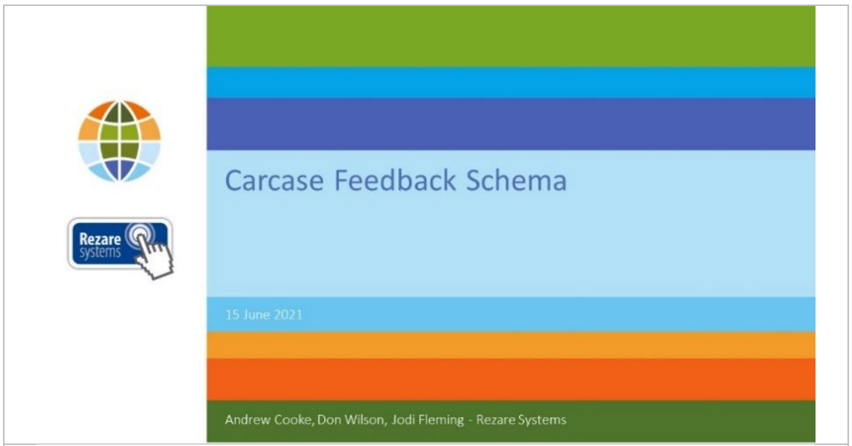
7. References

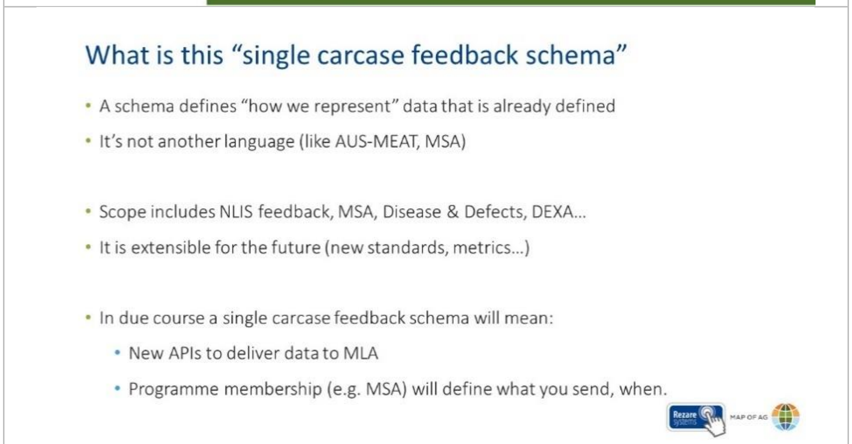
7.1 Background IP

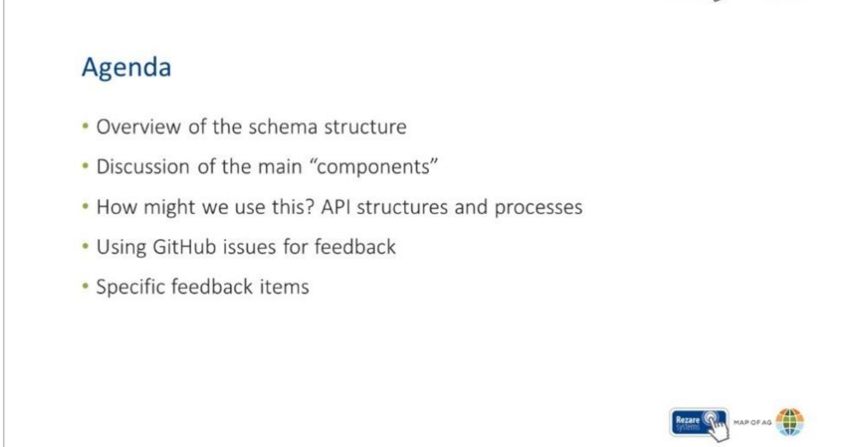
This project utilises the JSON schema specifications for livestock data based on the ICAR ADE (https://github.com/adewg/icar) and licensed under the Apache Licence 2.0. and the JSON specifications for DataLinker (https://github.com/Datalinker-Org), also licensed under the Apache Licence 2.0
JSON is the ECMA-404 data interchange standard defined at https://www.json.org/json-en.html. The specification for JSON Schema can be found at https://json-schema.org/

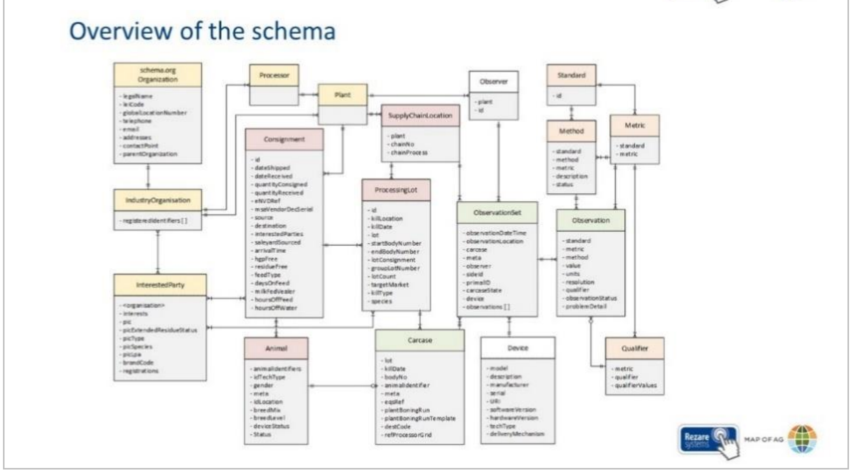
The conceptual data model design is based on modelling undertaken by the W3C (World Wide Web Consortium) schema.org community group, and by the ICAR (International Committee for Animal Recording) animal data exchange working group and their body of work, though it is specific to carcase data and the information provided by MLA and the workshop participants.

Appendix A: Single carcase schema industry overview pack





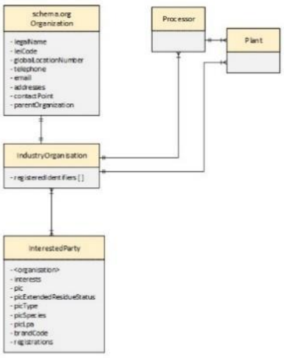




The "main components"

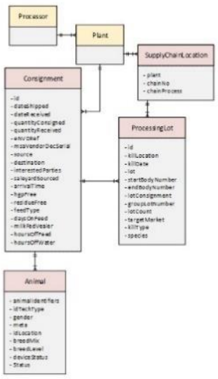
Discuss the entities and how they link

Organisations and interested parties



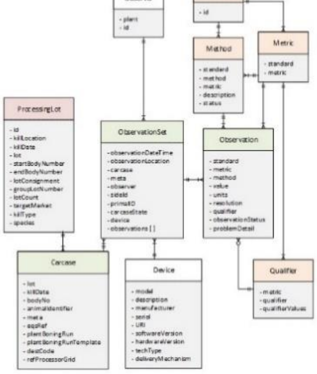
- Key organisation attributes:
  - Legal Name
  - Address, email, etc (but privacy)
  - LEICode (Legal Entity ID = ABN)
- Add Registered Identifiers
  - (AUS-MEAT and MLA establishment IDs)
- Add relationships for feedback
  - Identify producer, owner, brand owner/retailer, etc.

Consignments and lots



- Consignment
  - Arrival of animals at plant
  - Vendor declarations
  - MSA declarations…
- Processing Lot
  - Set of animals of the same consignment, processed together

Carcase observations



Sets of observations for carcases

- Typically recorded "together"
  - e.g., hot, cold, grading station
- Each carcase may have many observations
- Each observation is a "metric"
  - Belonging to a "standard", and measured with a "method"
  - Optionally with a qualifier and device

Metric Examples

Standard	Metric	Method	Qualifier
AUS-MEAT	HSCW	Scales	
AUS-MEAT	Marbling	Visual	
MSA	Marbling	Visual	
AUS-MEAT	EMA	Grid	12th Rib
Disease	Liver Fluke	Visual	
MSA	Loin Temperature	Thermometer	
MSA	Index	MSA Index	

How might we use this?

Discuss APIs and processes

How might we use this?

- GET, POST, PUT semantics
  Often you will PUT data, because you want to insert OR update it.
- You could POST/PUT a:
  - Consignment – to register details of a consignment
  - Processing Lot – with or without animals?
  - Carcases and Observation Sets (hot, cold, or by station)
  - Prefer "per Processing Lot" to "end of shift"
- You might GET MSA values / index for a set of animals

How to provide feedback

Making use of GitHub and an "open source" process

Find this project on GitHub

- https://github.com/Integrity-Systems-Company/common_carcase_schema
- See the Wiki for documentation
- JSON code is in "Code" – along with auto-generated markdown
- Use "Issues" for feedback

Appendix B: Details and examples of conceptual API

The following sections and tables describe the flows depicted in **Error! Reference source not found.** and refers to the numbered references on the diagram. Section 0 provides examples of some of the schema data that may be handled by the API.

① Consignment Checks

A consignment may optionally be created at booking to document a specific set of animals to be delivered from a source property to a processing plant. Details include the source and destination of the consignment, the reference of the NVD or eNVD, and identifies the parties with an interest in the consignment.

The Consignment endpoint creates or updates the consignment including identifying the interested parties. The response returns a unique MLA id for the consignment.

There is a synchronous interaction with the NLIS APIs to return to the processor checks on the consignment such as PIC extended residue programme (ERP) status values.

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment		iscConsignmentResource	iscConsignmentResource with Id filled in.
GET	host/consignment/{Id}	Id		iscConsignmentResource

② Arrival Checks

If the consignment was not created at booking, it will be created on arrival.

The consignment will be updated with details including when the consignment was shipped and received, and the number of animals consigned and received. The entity may also include any animal data recorded for the group or consignment such as feed type, residue status, hours off feed and water.

A Consignment Animals endpoint allows specification of the animals in the consignment.

There is a synchronous interaction with the NLIS APIs to return to the processor checks on the consignment such as animal Device ERP status values.

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment		iscConsignmentResource	iscConsignmentResource with Id filled in.
GET	host/consignment/{Id}	Id		iscConsignmentResource
PUT	host/consignment-animals/{Id}	Id	Array of iscAnimalResource	ProblemDetails
GET	host/consignment-animals/{Id}	Id		Array of iscAnimalResource

- Id: The Id of the consignment
- ProblemDetails: an error based JSON object: RFC7807.

③ Record Deaths

The Consignment endpoint also records any deaths on arrival, in lairage and in pens.

The Consignment Animals endpoint can be used here too, setting the status of any dead animals.

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment-animals/{Id}	Id	Array of iscAnimalResource	ProblemDetails
GET	host/consignment-animals/{Id}	Id		Array of iscAnimalResource

④ Lot Membership

A processing lot describes a group or batch of animals from a single consignment that are processed together. It identifies the chain where the animals were killed, when the lot was slaughtered, the species, the body numbers, and parties interested in the lot and the target market. The Lot Membership may be recorded at slaughter.

The Processing Lot endpoint creates or updates the processing lot (and can return details of the lot for reconciliation), while the Processing Lot Carcases endpoint allows specification of the carcases in the lot (linking carcase body numbers to animal identifications where these are available).

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/processinglot		iscProcessingLotResource	iscProcessingLotResource with lotId filled in.
GET	host/processinglot/{lotId}	lotId		iscProcessingLotResource
PUT	host/processinglot-carcases/{lotId}	lotId	Array of iscCarcaseResource	ProblemDetails
GET	host/processinglot-carcases/{lotId}	lotId		Array of iscCarcaseResource

- lotId: The Id of the Processing Lot
- ProblemDetails: an error based JSON object: RFC7807.

⑤ Record Observation

Observations recorded by an operator at the same time are grouped as an observation set. As soon as an observation has been made it may be transmitted to the data warehouse without waiting for other observations. However, it does not have to be done this way. Data may also be transmitted for a lot or for an entire shift or whatever aggregation that suits the processor.

Observations and sets of observations are handled by the Observation endpoint. A processor could POST/PUT new or updated metrics for some animals, then GET the results – if not all results were available, they might get results for animals where this was available, an "insufficient data" message where not all required data was available, and an error message if some metrics were invalid or out of range (including conditional mandatory fields).

Depending on how the APIs are defined to use this schema, data could be submitted for a shift (risky if poor internet) OR a processing lot (preferred at present, probably separate by hot or cold) OR as individual animals pass by a grading station or device (potential in the future). The generalised, flexible approach provides for operational differences between plants and copes better with any future changes or new technology. Plants that process just one species will only populate the fields that relate to their entities and animals. They will only send the metrics that match their species.

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	/processinglot-observationset/{lotId}	lotId	Array of iscObservationSetResource	ProblemDetails or success
GET	/processinglot-observationset/{lotId}	lotId, *Optional filters:* carcaseId,		Array of iscObservationSetResource

		observationDateTime, standard, metric		

- lotId: The Id of the Processing Lot
- ProblemDetails: an error based JSON object: RFC7807.

⑥ Getting Results

Requests can be made to the Lot and the Consignment endpoints for carcase observations or for results such as MSA Indexes. These are synchronous interactions.

Action	Resource (path)	Parameters	Request JSON	Response JSON
POST	host/processinglot-msa-complete/{lotId}	lotId, *Optional filter:* carcaseId		Array of iscObservationSetResource Or ProblemDetails
GET	host/processinglot-observationset/{lotId}	lotId, *Optional filters:* carcaseId, observationDateTime, standard, metric		Array of iscObservationSetResource

- lotId: The Id of the Processing Lot
- LotObservationSet: Is a JSON object that contains a reference to a processingLot and an array of ObservationSets.

⑦ Interactions with MLA Systems

The MLA systems (NLIS, MSA, NLIS Data Warehouse and LDL) interact with the API asynchronously.

Examples of conceptual API calls with sample data

The following provides examples of some of the attributes that may be handled by the API calls. The purpose of this is to assist understanding of how the APIs may interact with data collected by or required by processors.

The lists of attributes below are not exhaustive. Rather they are a selection to demonstrate the approach. Similarly, the examples of observations show just a few of the many possible metrics.

① Consignment Checks

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment		iscConsignmentResource	iscConsignmentResource with Id filled in.
GET	host/consignment/{Id}	Id		iscConsignmentResource

Consignment Resource data:

Attribute	Value
id	123e4567-e89b-12d3-a456-426614174000
eNVDref	CO125
Source	ND250250
Destination	NS121093
interestedParties	see below

Interested Parties:

Attribute	Party 1	Party 2	Party 3
Interests	Producer	Retailer	Saleyard
pic	ND250250		NA477352
leiCode	74 412 829 978	44 201 372 937	89 645 123 456
legalName	Berra Farm	Bob's Farm Butchery	XYZ Saleyard No 1 Pty Ltd
registrations	NLIS, 123456 LPA, 875432		
picExtendedResidueStatus	<returned>		

② Arrival Checks

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment		iscConsignmentResource	iscConsignmentResource with Id filled in.
GET	host/consignment/{Id}	Id		iscConsignmentResource
PUT	host/consignment-animals/{Id}	Id	Array of iscAnimalResource	ProblemDetails
GET	host/consignment-animals/{Id}	Id		Array of iscAnimalResource

Consignment Resource data:

Attribute	Value
id	123e4567-e89b-12d3-a456-426614174000
dateShipped	15/03/2021
dateReceived	16/03/2021
quantityConsigned	83
quantityReceived	83
arrivalTime	16/03/2021 07:43am
hgpfree	Y
feedType	Grass
hoursOffWater	10

Consignment Animals data

Attribute	Animal 1	Animal 2
id (Consignment)	123e4567-e89b-12d3-a456-426614174000	123e4567-e89b-12d3-a456-426614174000
animalIdentifiers	EID, 982 000145027095 NLIS, 3HGET065XBCD0071	EID, 982 000145017814
gender	Male	Male
idLocation	LE	LE
breedMix	AA	AA
breedLevelQualifier	A0	S1
deviceStatus	<returned>	<returned>

③ Record Deaths

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/consignment-animals/{Id}	Id	Array of iscAnimalResource	ProblemDetails
GET	host/consignment-animals/{Id}	Id		Array of iscAnimalResource

Consignment Animals data

Attribute	Animal 1	Animal 2
id (Consignment)	123e4567-e89b-12d3-a456-426614174000	123e4567-e89b-12d3-a456-426614174000
animalIdentifiers	EID, 982 000145017814 NLIS, 3HGET065XBCD0071	EID, 982 000145043588
status	Dead on Arrival	Death at Lairage

④ Lot Membership

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	host/processinglot		iscProcessingLotResource	iscProcessingLotResource with lotId filled in.
GET	host/processinglot/{lotId}	lotId		iscProcessingLotResource
PUT	host/processinglot-carcases/{lotId}	lotId	Array of iscCarcaseResource	ProblemDetails
GET	host/processinglot-carcases/{lotId}	lotId		Array of iscCarcaseResource

Processing Lot data:

Attribute	Value
id	123e4567-e89b-12d3-a456-426614174534
killDate	16/03/2021 11:10am
killLocation	3
lot	6
startBodyNumber	84
endBodyNumber	190
lotConsignment	123e4567-e89b-12d3-a456-426614174000
lotCount	107
species	Cattle

Processing Lot Carcases data

Attribute	Carcase 1	Carcase 2
id (Processinglot)	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534
animalIdentifier	EID, 982 000445566778	EID, 982 000776655448
bodyNo	84	85
plantBoningRun	9999-2	9999-2
plantBoningRunTemplate	YZQ123	YZQ123
destCode	ABC12	ABC12
refProcessorGrid	011-21	011-21

⑤ Record Observation

Action	Resource (path)	Parameters	Request JSON	Response JSON
PUT	/processinglot-observationset/{lotId}	lotId	Array of iscObservationSetResource	ProblemDetails or success
GET	/processinglot-observationset/{lotId}	lotId, *Optional filters:* carcaseId, observationDateTime, standard, metric		Array of iscObservationSetResource

Hot Observations:

Attribute	Observation 1	Observation 2	Observation 3
lotid	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534
killDate	16/03/2021 11:10am	16/03/2021 11:10am	16/03/2021 11:10am
observationDateTime	16/03/2021 2:14pm	16/03/2021 2:14pm	16/03/2021 2:14pm
observationLocation	3	3	3
carcase	84	84	85
observer	ABCD01	ABCD01	ABCD01
sideID	R	R	R
carcaseState	H	H	H
standard	AUS-MEAT	AUS-MEAT	AUS-MEAT
metric	hotCarcaseWt	marbling	hotCarcaseWt
method	Scales	Visual	Weigh
value	241.2	3.1	288.0
units	kg		kg
observationStatus	Measured	Calculated	Measured

Disease & Defect Observations:

Attribute	Observation 1
lotid	123e4567-e89b-12d3-a456-426614174534
killDate	16/03/2021 11:10am
observationDateTime	16/03/2021 3:18pm
observationLocation	3
carcase	93
observer	BAX02
sideID	R
carcaseState	H
standard	Disease
metric	Liver Fluke
method	Visual
value	Y
units	
observationStatus	

Cold Observations:

Attribute	Observation 1	Observation 2	Observation 3
lotid	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534
killDate	16/03/2021 11:10am	16/03/2021 11:10am	16/03/2021 11:10am
observationDateTime	17/03/2021 1:23pm	17/03/2021 1:23pm	17/03/2021 1:25pm
observationLocation	8	8	8
carcase	84	84	85
observer	XYCD01	XYCD01	XYCD01
sideID	R	R	R
carcaseState	C	C	C
standard	MSA	MSA Chiller v3.1	MSA Chiller v3.1
metric	Eye Muscle Area	marbling	loinTemp
method	Grid	Visual	Thermometer
value	77	320	7.1
units	sq cm		ºC
observationStatus	Measured	Measured	Measured

⑥ Getting Results

Action	Resource (path)	Parameters	Request JSON	Response JSON
POST	host/processinglot-msa-complete/{lotId}	lotId, *Optional filter:* carcaseId		Array of iscObservationSetResource Or ProblemDetails
GET	host/processinglot-observationset/{lotId}	lotId, *Optional filters:* carcaseId, observationDateTime, standard, metric		Array of iscObservationSetResource

Result Observations:

Attribute	Observation 1	Observation 3
lotid	123e4567-e89b-12d3-a456-426614174534	123e4567-e89b-12d3-a456-426614174534
observationDateTime	17/03/2021 1:23pm	17/03/2021 1:25pm
carcase	84	87
sideID	R	R
standard	MSA	MSA
metric	index	index
method	MSAIndex	MSAIndex
value	55.42	N/A
units		
observationStatus	Calculated	Calculated

Appendix C: Carcase Schema potential use case study

This “walk-through” case study conceptualises how the data schema might be used in a real-world implementation, supported by engagement with two software vendors, and MLA programme teams.

V.DIG.0020 - Rezare - Carcase Schema pot

Appendix D: Single Carcase Data Schema Project Summary and Recommendations

Rezare Systems presented project results to ISC key representatives and discussed key learnings from the project, and recommendations for a potential future industry pilot.

MLA Carcase Schema
- Final Report Summa