26th International Congress on Modelling and Simulation, Adelaide, SA, Australia, 30 Nov–4 Dec 2025
mssanz.org.au/modsim2025

The Minerva data platform – towards a data platform for research applications

ST Bell [a], D Liu [a], A Philip, C Cerecke [b], J Tang [b], M Nichols [c], Z Gilmore [a] and HT Lin [d]

[a] *New Zealand Institute for Bioeconomy Science Limited, Lincoln, Christchurch 8140, New Zealand*
[b] *New Zealand Institute for Bioeconomy Science Limited, Auckland 1142, New Zealand*
[c] *New Zealand Institute for Bioeconomy Science Limited, Port Nelson, Nelson 7010, New Zealand*
[d] *New Zealand Institute for Bioeconomy Science Limited, Palmerston North 4442, New Zealand*
Email: stephen.bell@plantandfood.co.nz

Abstract: The Minerva data platform is designed to streamline data integration, management and analysis across diverse research domains. It features a unified architecture that supports decentralised object ID assignment, modular metadata structures and good interoperability with both existing and custom tools. Drawing on proven techniques from high-integrity global industries such as medical imaging, Minerva has the potential to deliver a scalable, interoperable data platform that facilitates accelerated scientific discovery by providing simple standardised tools for data modelling and management that integrate well with existing and future science applications and workflows. Maintaining science data integrity, traceability and governance is a central theme in its design principles.

Keywords: *Integrated modelling, digital twin, metadata, object identification, data reuse, traceability*

1. INTRODUCTION

Scientific research generates increasingly large and diverse datasets, ranging from sensor measurements and high-resolution images to simulation datasets and experimental records and a wide range of quantitative and molecular genetics datasets. Hardware systems may range from small field deployed embedded systems, laboratory systems and workstations through to larger scale high-performance computing systems. Managing, integrating and analysing data across these heterogeneous systems presents significant challenges, including inconsistent formats, fragmented storage and limited interoperability across research tools. The landscape of available software tools and frameworks is complex and bespoke solutions in specialised domains are common (Romera et al, 2024).

The Digital Horticultural Systems (DHS) initiative within the New Zealand Institute for Bioeconomy Science is developing a digital twin for perennial horticulture and comprises multiple research programmes (Mawson and Stanley et al. 2023). The Minerva system is being developed to meet the data platform requirements across all DHS research programmes. Long horizon digital enablement is an aspirational goal for the DHS initiative, and this along with a desire to address some of the challenges around interoperability and sharing of science data, have guided our approach.

This paper presents the design principles and core capabilities of the Minerva platform, including its data structure, modelling support, tool ecosystem and processing workflows. We further illustrate its application through real-world examples, such as sensor platform data capture, orchard imaging and spatial visualisation, demonstrating how Minerva can potentially accelerate scientific discovery while maintaining robust data management practices.

2. BACKGROUND

Scientific research increasingly involves heterogeneous data types, evolving models, and interactions across multiple tools and subsystems. Interoperability and re-use of data between systems is a significant challenge. Maintaining traceability of data back to its origin and linking a wide variety of contextual metadata is also of high importance, along with data governance and stewardship.

High value science datasets have a long lifecycle and can easily outlive the hardware and software systems used in their creation. In that regard, data is a first-class citizen. Science datasets such as measurements made during an experimental trial are typically treated as immutable; any data cleaning and reprocessing actions are documented and recorded along with the data. A good science data platform needs to be able to capture this detail, in a flexible and efficient way.

We identified several large global industries that have a high degree of data interoperability between a multiplicity of stakeholders, systems and vendors, to use as case studies for better understanding how to facilitate interoperability-at-scale in a science data platform. The medical imaging industry which ubiquitously uses the DICOM and HL7 standards is used as a case study for this paper.

Key aspects of the DICOM standard are a) the use of IEC/ISO object identifiers, b) standardised container data formats and c) a modular system of tags (for metadata) stored with each data object. The object IDs have a graph structure which allows delegation of nodes to a third party; we describe these further in Section 3.2. The metadata requirements of the medical industry are diverse and mission critical, with the structure and use of these being well described in the main standards.

Container data formats are commonly used by applications that need to share and exchange complex data. For example, audio and visual media formats (avi), PDF documents, and image formats such as tiff and exif. Other relevant data format examples include gpx for navigation and geolocation data, and geojson which is used to describe geospatial features. While some of these storage formats such as the audio-visual media container formats include detailed metadata, other storage formats may only have limited provision or omit this entirely. Symbiotic piggyback metadata schemes have emerged in some data storage formats, arising through the use of tokens and symbols inside sections set aside for comment text.

A defining characteristic of industry scale interoperability is the use of unifying standards, rather than reliance on single vendor solutions and software frameworks. Lower-level protocols and formats defined in RFCs[1] form the basis of many common data tools and techniques.

In contrast to these examples, CORBA was an early standard designed in the 1990s to enable communication between software components across platforms and languages. It initially gained attention but ultimately declined in use due to complexity, inconsistent application programming interfaces (APIs) and low compatibility with emerging web technologies (Henning, 2008); CORBA provides a good example of how short the lifecycle of even quite popular software tools can be.

There are several widely used tools and technologies in high-performance science computing. Common systems include Beowulf type clusters (with rackmount server nodes) using batch processing systems based on schedulers such as Slurm, and high availability cluster systems such as Kubernetes, commonly used by cloud service providers and enterprise data centres. Parallel computing techniques are common in some application domains. OpenMP is a useful parallelisation tool for improving performance on a single node and is applied at the time applications are created. When running large scale simulations, it is sometimes desirable to split parts of the simulation across multiple compute nodes, and the message-passing-interface protocol (MPI) is often used in this situation but increases the technical complexity of the solution (Jin *et al.*, 2011). Simulations running across multiple compute nodes typically utilise bespoke solutions with systems programming often needed to effect the process-to-process communication, leading to increased time and complexity costs.

Several aspects of science computing applications can lead to difficulties in portability. Those relating to data interoperability include coding references to storage services using infrastructure naming schemes (hostnames and port numbers) and building for deployment on only one execution platform. The general solution to this problem is relatively straightforward. That is to place all the infrastructure details into a configuration space.

Some of the initial design considerations for Minerva arise from the following question:

> Supposing we have two computing systems, A and B, within different organisations. What are the best ways to exchange or transfer an application or model and its associated data from one system to another with no change needed to either the application codebase or its associated data?

It is the answers to this question that lead to the recipes for interoperability …

3. MINERVA DATA STRUCTURE

Fulfilling the data platform requirements of the DHS digital twin programme is a key focus for our work. On the physical twin side there is data capture from orchard sensor networks, monitoring and control data from automated systems, orchard management data, and data interactions with robotic systems.

---

[1] https://en.wikipedia.org/wiki/Request_for_Comments An RFC (request for comment) is a proposal document describing an internet technical standard.

The modelling and simulation components of the digital twin include the ability to easily manage and orchestrate models and use selected datasets. Different combinations of parameters may be needed for simulations. For example, when running simulations to explore differences between two genetic variants of a crop or investigating the effects between notably dry or wet years of production. Tracking which datasets and parameters are associated with each simulation is highly important. Ideally, any given simulation run should be able to be run independently and to be verified by a third party in a relatively simple way.

Data visualisation is important in both a traditional science data visualisation sense but also in the context of the 3D orchard structures and 2D geospatial domains. We will discuss this further in relation to the Sight Green 3D visualisation tool example in Section 4.5.

3.1. Basic data object structure

The basic structure of a Minerva data object comprises three components: 1) an object identifier; 2) a structured metadata section; and 3) a data payload section. Minerva data objects may be either a composite object where all three elements are embedded in a single data structure, database or file; or split into two parts, the first combining the object identifier and metadata structures, and the second the data payload. This second type is used to easily integrate the many existing data file objects and formats from other frameworks and domains (Figure 1).

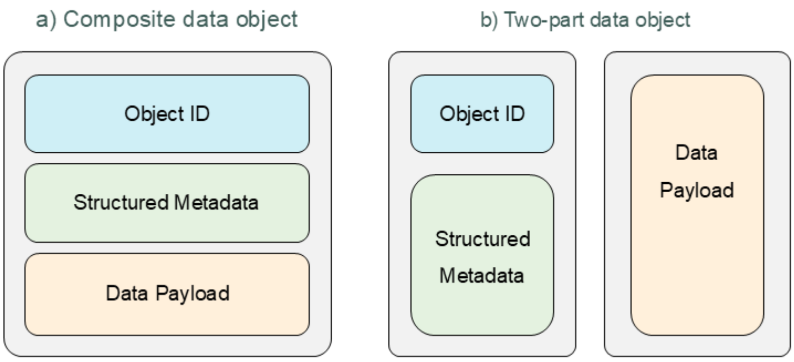

Figure 1 Minerva data object structure: a) composite data objects contain all the information in a single structure; b) two-part objects separate the header and data payload, allowing use of different backend storage technologies for each part.

A third type of data referencing is needed for linking with data stored in existing databases. For this we have linked a Minerva object identifier and metadata with a *graph-resource-name.* A graph resource name uses a graph or tree structure to describe the node or location where data are stored. Simple examples include a *file-system path*, or web URL. However, we can go further than these simple examples, for instance, the location of data-tables in a SQL database can be described by the following scheme: *database/table-name/fieldname.*

3.2. Object identifiers

Minerva uses ISO/IEC 6523 based object identifiers (OIDs)[2]. OIDs allow independent organisations to generate identifiers without conflict with other organisations. These OIDs have a tree-hierarchy structure, which provides an intrinsic ability to trace the origin of each OID. The Internet Assigned Numbers Authority (IANA) provides an OID prefix and a simplified process for organisations to obtain an assigned private enterprise number[3] which, is combined with the IANA prefix to create an OID prefix for the requesting organisation.

This structure is described as having three main elements: a) a prefix describing the origin; b) a node with delegated authority; and c) subsequent nodes reflecting a structure and function within the delegation. In the following example from Wikipedia[2], an OID prefix is constructed and described as follows:

- 1.3.6.1.4.1 is the IANA prefix for private enterprise numbers
- 343 is the assigned private enterprise number for the company, Intel
- 1.3.6.1.4.1.343 becomes the OID prefix for the company, Intel.

[2] https://en.wikipedia.org/wiki/Object_identifier
[3] https://www.iana.org/assignments/enterprise-numbers/

A key aspect is the decentralised allocation of OIDs containing structural information about their origin and function. This system also allows OID generation from embedded field devices that have limited network connectivity, and no prospect of connecting reliably to a centralised issuer.

3.3. Structured metadata

We were inspired to explore the use of structured metadata by its successful use in the medical imaging industry. Key aspects we considered were this industry's use of stakeholder groups and committees to formulate and guide specific metadata sections, and the modular structure of the metadata tag system. Obviously, we needed something more general than what the medical imaging examples provide, and the form our metadata structures should take was not immediately clear.

Tree-hierarchy data formats have sufficient flexibility to handle most of the anticipated uses. However, we were concerned about proliferating complexity if the schemas were not constrained. Initially, we opted for a tree-hierarchy and core components using three levels of nesting: *domain, category, fieldname.*

Key categories used for grouping and data modelling are *collection*, *series* and *station*. Each of these categories contains the fields *id* and *name*.

The *collection* group is defined by either the data-originators or user community and typically are also used by Minerva to define a boundary or scope for applying access and security measures.

A *series* is a related sequence or set of items, such as a sequence of images or the set of input data and parameter-set objects required to run a simulation.

The *station* identifier provides for recording the location and equipment pertinent to the recording of the data stored in the data object.

Other metadata categories cater to recording details such as units and field-name descriptions, attribution and licencing information, guardianship, and project and organisation details. We have considered techniques for delegating extended metadata definitions beyond the core Minerva components, but these go beyond the scope of this paper.

3.4. Data payload

The data payload component is the main information content of the data object. The two-part data objects described in Section 3.1 allow the data payload component to be stored in any file storage format by using filesystem storage, or by using a Minio[4] backend, which provides S3 compatible block storage. However, providing such file-object based storage, although useful, is primarily a catchall fallback solution to ensure Minerva is always able to provide a data storage and retrieval service at some level.

To support and provide a versatile range of data transformation and analysis tools as the system evolves, particularly for the modelling and simulation components, we have utilised abstract data models representing data structures such as key-value data, table data, images, time-series measurements and tree-hierarchies. Our approach for the key-value pair data model was developed in consideration with the requirements of naming and storing model parameters. Here we represent the key for a model parameter as a tuple with the following items: *domain, category* and *fieldname*. This allows the values for a set of related parameters to be stored in a tree hierarchy (e.g. json), which we define as a *parameter-set.* The *domain, category* and *fieldname* key tuples provide a simple ontology-like namespace and are utilised throughout our modelling and simulation components to facilitate the connection and exchange of data.

With an internal abstraction for table data, it is relatively straightforward to load, transform and store data for formats such as csv, SQL-tables, data-frames in R and python using existing software frameworks. As such, we use the term *metaclass* to refer to an abstract data model and its structure. Metaclasses are used by Minerva to implement support for features based on each abstract data model. Applications can, in turn, define classes providing the actual data schema, based on these *metaclasses*.

The image and time-series measurement data models are briefly described in later sections in relation to their application examples (see Sections 4.3 and 4.4).

[4] https://www.min.io/ Minio: A software framework for exascale data object storage.

4. TOOLS AND APPLICATION EXAMPLES

4.1. Minerva tools and Python API

Minerva provides a set of command-line tools for storing (*m-store*), searching (*m-find*) and retrieving (*m-get*) data in a Minerva service. Additionally, the *m-move* command provides for transferring data between two Minerva services. The *m-find* tool facilitates searches using indexed metadata keys, followed by an optional filtering stage which may use any metadata key to refine the search to retrieve a precise set of objects.

Data transformations may be made during data import and export. This functionality is supported by an extensible plugin system which allows for both built-in and local customised transformations to be extended over time.

Minerva also provides a Python API for programmatic access to data using the same underlying functionality and principles as the *m-find, m-get, m-store* and *m-move* tools. Additionally, we have incorporated *rest-api* access following a similar structure.

4.2. Modelling and simulation

Modelling and simulation applications are a key motivation for this work. The two-part data objects from section 3.1 in conjunction with the block storage capabilities of Minio allow us to support arbitrarily complex science data formats. This effectively enables Minerva to support data storage requirements of all model file formats.

The key-value pair and table data models are commonly used in modelling and simulation datasets. Our key-value pair data objects are typically used for grouping and organising model parameters, and the table data format is useful for storing the tables of time-dependent model inputs and outputs used in dynamic systems models. The metadata grouping identifiers, *series* and *collection* are useful for data management in more complex simulations. The identifier and metadata components of our data objects are useful for meeting tracking and provenance requirements of simulation datasets.

4.3. Sensor platform

We have created a platform service for describing deployed in-orchard sensors and capturing the data these produce. The data model for a deployed sensor is a two-part Minerva data object, comprising the *measurement-channel* descriptor (id and metadata) and a linked data payload component stored in an influx database. *Station* and *site* metadata categories are included for each sensor. The *series* identifier is used to group and link multiple sensors from a given installation (*sensor-group)*. An example of a *sensor-group* is the *temperature, rainfall, relative-humidity, and wind-speed* sensors at one site (see Figure 4).

The sensor platform supports multiple communications protocols commonly used by data logging systems, including FTP and MQTT, and facilitates connecting the data streams originating from the sensors to the storage services. Status monitoring and logging are provided, and we are currently developing the standards and techniques for recovering from faults and handling out of range measurements when these are detected.

4.4. Mobile orchard imaging

Minerva provides capabilities for transferring data between systems, supporting automated workflows. In mobile orchard imaging applications, a robot imaging rig moves through the orchard to capture images of plant structure and surrounding environments.

Figure 2 shows a simplified workflow for automated data capture on a robotic imaging system. The imaging system includes multiple camera channels and a geolocation data stream. The captured image data and metadata are received by an embedded on-device Minerva that queues and transfers the images to a central storage system.

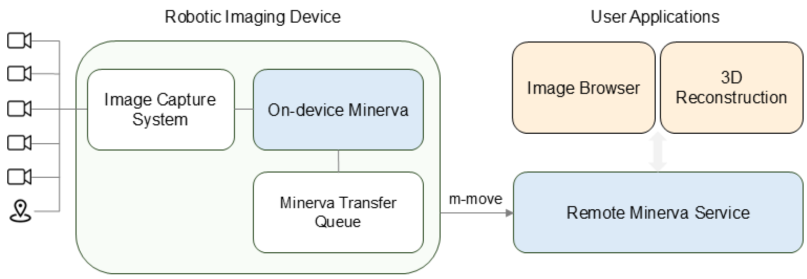

Figure 2 Minerva service deployment on a robotic imaging system. Captured images are stored by the embedded Minerva service and forwarded to a remote storage location (or gateway).

The queue system provides fault tolerant automated transfers to help maintain data integrity and consistency throughout the process. Events in Minerva, such as the start or end of data storage and retrieval operations, are inspected by a process that can trigger downstream processing tasks when selected criteria are met – some examples of processing tasks include compression/decompression, data format conversions, and user-application defined processing tasks. This facilitates timely, efficient, and continuous research data-processing workflows.

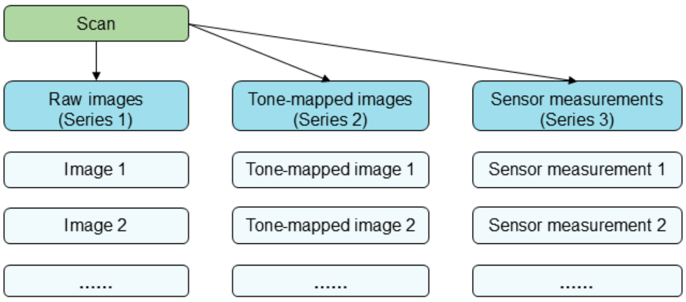

Figure 3 Data groups in the mobile imaging application. A scan comprises a series of raw images. Additional series may contain derived images (e.g. tone-mapped images) or heterogeneous related data (e.g. sensor measurements).

Figure 3 illustrates how the mobile imaging system utilises the Minerva data model. The system captures images of orchard elements along with optional sensor measurements. We refer to the images gathered in a single trip as a *scan* dataset. These datasets are organised using a hierarchical data model, where a *collection* represents all data associated with a given site or scan, and all the images captured in a single *scan* are grouped into an initial *series* (raw images). Each image within a series can be single- or multi-channel from a single exposure and includes associated metadata.

Downstream applications like 3D reconstruction use these datasets, with derived results stored back as a new series (e.g. tone-mapped images) for traceable management and further analysis. Derivative elements like thumbnails or processed outputs are linked back to their original data, ensuring efficient organisation and provenance.

4.5. Sight Green – spatial visualisation

Sight Green is a 3D spatial visualisation application built using Minerva as a data modelling and storage management service. In Sight Green, each site visualisation is organised as a base-configuration comprising layers of static features such as orchards, blocks and roads. This base-configuration can be extended with variations that are unique to a specific experiment or project context by including additional user data layers. Environmental time-series sensor measurements from selected *sensor groups* are linked to the base-configuration. A *base-configuration* identifier included in each variation provides the linkages used to align data from the base-configuration across the derived variations for the site or within the orchard.

Together, this data model (Figure 4) supports the construction of a digital twin of the physical orchard, allowing Sight Green to manage data efficiently without requiring custom application database programming.

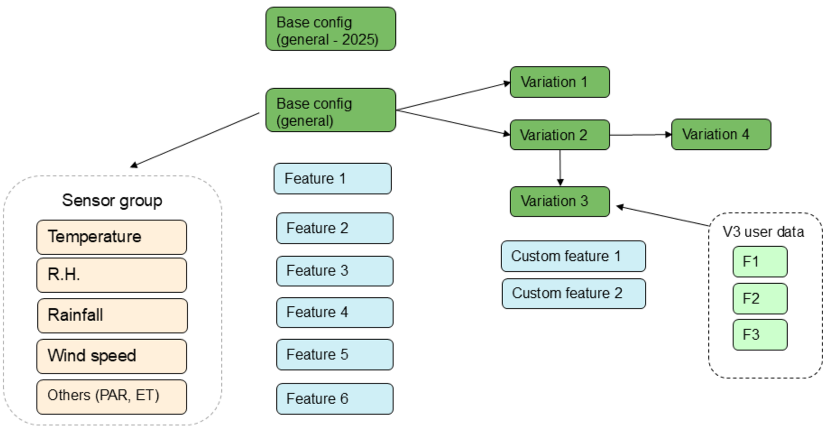

Figure 4 The Sight Green application data structures and linking. Dark green: visualisation base-configurations and variations. Blue: features for visualisation layers associated with each visualisation. Salmon: linked group of environmental measurements from sensors. Light green: visualisation preferences for a user specific to one variation.

5. DISCUSSION AND CONCLUSION

Achieving a high degree of interoperability for models and data across a wide range of domains is a challenging process. A key approach has been to develop the Minerva composite and two-part data object concepts, and the techniques and tool ecosystem associated with these.

The decentralised OID allocation has worked well in the mobile imaging rig application. Although we made good initial progress with the metadata structures for the modelling and simulation related components, it took somewhat longer to establish consistent practices across the multiple applications we have been working with. Our focus on long term use has started to pay off and we are now seeing benefits in terms of flexibility in the Sight Green data models and ease of linking with the sensor measurement platform. The sensor platform has gained interest within our organisation and several groups in beyond the core DHS research programmes are starting to use this service. Of interest to many is the ability to view and access the sensor measurement data within the Sight Green tool. Feedback from early users of these systems will be invaluable as we move towards visualisations of the (yet to be) deployed physical twin systems.

One of the challenges in this work has been the tension between needing to make near term delivery of early components with useful functionality, versus achieving progress with features targeted at the longer-term aspirations. Our approach to balancing these needs has been to deliver a series of functional prototypes for evaluation and feedback, while progressively developing the modular metadata structures, abstract data models and other foundational components towards the longer horizon goals.

The best uptake of this work was in collaborations with projects working through their own development programmes as parallel activities. These situations have mutual benefits with opportunities to brainstorm on new initiatives, feedback about successes and failures, and engage on innovations with the new capabilities. A good example of this has been our collaboration with the Modelling High-performance Apple Systems team in our institute (Stanley and Zhu et al. 2024), who have adapted their functional structural fruit crop model to use our composite and two-part data objects for all their input data and parameters. This has advantages when running the model in complex simulation environments.

Minerva can support automation of data capture and storage workflows, as seen in our sensor platform and mobile imaging rig work. We can extend this using our current work on event driven processing, and the flexibility of the structured metadata for linking and annotating datasets, to provide a wide range of data driven applications such as machine-learning detection-based tools. Minerva is well suited for deployment at different scales such as within an embedded system, workstations and servers, and data-centre platforms.

The outcome in the Sight Green application, where we were able to largely eliminate custom database programming in the core-application, was both spectacular and somewhat unexpected. The role of this use case goes well beyond demonstrating some of the advantages Minerva can provide. It showcases an approach to building applications around data resources through curating the digital content.

ACKNOWLEDGMENTS

We acknowledge Bryn Gibson, and Ou-An Chuang for their contributions to the Minerva codebase.

This research was funded by the New Zealand Ministry of Business, Innovation, and Employment (MBIE) through the Strategic Science Investment Fund (contract number C11X1702) as part of the Digital Horticultural Systems Ngā Pou Rangahau, a Growing Futures™ programme at The New Zealand Institute for Plant and Food Research Limited.

REFERENCES

Henning M (2008) The rise and fall of CORBA. Communications of the ACM 51(8), 53–57. doi:10.1145/1378704.1378718

Jin H, Jespersen D, Mehrotra P, Biswas R, Huang L and Chapman B (2011) High performance computing using MPI and OpenMP on multi-core parallel systems. Parallel Computing 37(9), 562–575.

Mawson AJ, Stanley CJ, Zhu J, Pattemore DE, Chooi KM, Oliver RJ, Lin HT and Harker FR (2023) Developing a digital twin of apple production and supply chain ecosystems. Acta Horticulturae, 1360, 129–136. doi: 10.17660/ActaHortic.2023.1360.17

Romera AJ, Sharifi M, Charters S (2024) Digitalization in agriculture. Towards an integrative approach. Computers and Electronics in Agriculture, 219, 108817.

Stanley J, Zhu J, Rojo F, Kaneko T, Breen K, Friend A, Teixeira E, Niemann N and Mawson J (2024) Advancing high-performance apple production systems through digital twin modelling. Acta Horticulturae, 1395, 125–132. doi: 10.17660/ActaHortic.2024.1395.17

