

DELIVERABLE 3.1
INTERIM REPORT REVIEWING
BATTERY DOMAIN
ONTOLOGIES AND
GUIDELINES FOR THEIR USE
BATTERY 2030+ CSA3

BATTERY
2030+

At the heart of a connected green society

Document Classification

Deliverable Number & Title	D3.1 Interim report reviewing battery domain ontologies and guidelines for their use
Work Package	WP3 Common Best Practices in Battery Research
Author(s)	SINTEF – Simon Clark
Delivery due date	2024-10-31
Dissemination Level	Public

Document History

Internal review	2024-10-25	All partners
Contributions	2024-10-30	Pn KIT – Christian Punckt
Validation	2024-10-30	Project Coordinator P1 UPPSALA – Kristina Edström kristina.edstrom@kemi.uu.se

Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure and Environment Executive Agency (CINEA). Neither the European Union nor the granting authority can be held responsible for them.



This project has received funding from the European Union's Horizon Europe research and innovation programme under grant number No. 101104022.

TABLE OF CONTENTS

1	Introduction.....	5
2	A Linked Data Primer.....	6
2.1	Background and motivation	6
2.2	Key concepts in linked data.....	6
2.3	Linked data resources.....	7
2.3.1	RDF Schema (RDFS)	7
2.3.2	Dublin Core terms.....	7
2.3.3	Schema.org.....	7
2.3.4	CSV on the Web (CSVW).....	7
2.3.5	Data Catalogue Vocabulary (DCAT)	8
2.4	Linked data formats.....	8
2.5	Linked data examples.....	9
2.5.1	Example 1: A person.....	9
2.5.2	Example 2: A place.....	10
2.5.3	Example 3: A person who shares a name with a place	11
2.5.4	Example 4: A tabular dataset	11
3	Battery Domain Ontologies.....	13
3.1	EMMO-based ontologies relevant for batteries	13
3.1.1	EMMO Electrochemistry domain	13
3.1.2	EMMO Battery domain.....	13
3.1.3	BattINFO	14
3.2	Best practice guidelines for publishing battery linked data.....	14
3.2.1	Best Practice 1: Publish your battery data online	15
3.2.2	Best Practice 2: Apply an open and clear license	15
3.2.3	Best Practice 3: Use structured data formats	15
3.2.4	Best Practice 4: Use non-proprietary, open formats	16
3.2.5	Best Practice 5: Use RDF vocabularies to describe and identify your data.....	16
3.2.6	Best Practice 6: Provide minimum metadata for reuse	16
3.2.7	Best Practice 7 : Ensure comprehensive metadata for clarity and reusability.....	17
3.2.8	Best Practice 8: Document your data with human-readable descriptions	17
3.2.9	Best Practice 9: Link your data to other datasets.....	18
3.2.10	Best Practice 10: Stay up-to-date with ontology developments.....	18
3.3	Battery linked data examples.....	20
3.3.1	Example 5: A simple battery cell	20
3.3.2	Example 6: An active material.....	20

3.3.3	Example 7: An electrolyte.....	21
3.3.4	Example 8: A detailed battery cell.....	22
3.3.5	Example 9: A Battery test data schema.....	23
4	Conclusion	28

1 INTRODUCTION

Battery data from research projects is often underutilized due to inconsistent formats and lack of standardization, making it difficult to share, analyse, or reuse effectively. This limits insights into areas like material optimization and performance improvements. Better data organization through ontology-based linked data approaches can help overcome these challenges and accelerate innovation.

Ontologies provide a structured vocabulary for describing materials, components, processes, and performance data in a consistent way. They are organized in layers, with top-level ontologies covering general concepts and domain-level ontologies, like those for batteries, focusing on specific terms relevant to the field. By aligning with a top-level framework, battery ontologies ensure greater interoperability and data reuse across domains like materials science and manufacturing, while preserving the necessary details.

Battery research spans multiple disciplines, each with its own terminology, making a shared ontology essential for consistent communication. As new battery technologies emerge, ontologies play a critical role in maintaining data consistency and meeting regulatory requirements.

This report reviews the key battery domain ontology resources developed within the top-level Elementary Multiperspective Materials Ontology (EMMO) framework. By using battery ontologies, researchers and industry stakeholders can create more efficient workflows, enhance collaboration, and unlock new insights from battery data that were previously siloed or difficult to integrate.

The remainder of this report is structured as follows:

- A linked data primer introduces the basic concepts that underpin the approach and presents some examples from other domains in Section 2.
- Battery domain ontologies are then presented and reviewed with a focus on resources developed within the EMMO framework in Section 3.1.
- Best Practices for published linked battery data are provided in Section 3.2.
- Usage examples with some key guidelines are then presented in Section 3.3.

2 A LINKED DATA PRIMER

Linked data is a powerful framework for connecting and sharing structured data across diverse domains in a way that allows it to be understood and processed by both humans and machines. It provides a methodology for representing data as a network of interrelated entities, each described using a shared vocabulary and unique identifiers. By linking pieces of information, linked data facilitates interoperability between systems and enables users to extract meaningful insights across different datasets.

This section is intended for readers who are new to the concept of linked data. Readers with a working knowledge of linked data, RDF, and JSON-LD can skip to Section 3.

2.1 Background and motivation

Linked data is a key component of the Semantic Web, an idea introduced by Sir Tim Berners-Lee in the early 2000s. The goal of the Semantic Web was to make the web smarter by allowing machines to not only display information but also understand and process it. This would improve the connection between data, making information more useful and accessible.

At the core of linked data is the Resource Description Framework (RDF), which organizes information as triples—subject, predicate, and object. These triples are like node-edge-node structures, such as linking "Marie Curie" (subject) to "birthDate" (predicate) to "1867-11-07" (object). When combined, these triples form a graph of interconnected data, enabling the definition of relationships that make it easier to explore related information.

In 2006, Tim Berners-Lee introduced the Linked Data Principles, explaining how to use URIs (Uniform Resource Identifiers) to uniquely identify resources, make data accessible via the web, and link datasets.

Today, linked data is foundational to modern search engines and knowledge graphs. For instance, Google uses linked data in its Knowledge Graph, which powers rich search results by connecting facts about people, places, and things from various sources. Wikidata, a structured data hub for Wikipedia, relies heavily on linked data principles to link information about millions of topics. These technologies enable search engines to deliver more relevant and precise results by understanding relationships between data, making the web more interconnected and useful for both people and machines.

2.2 Key concepts in linked data

The **Resource Description Framework (RDF)** is the foundational model for representing information in linked data. It structures data as a set of triples, each consisting of a subject, predicate, and object. These triples are used to create graphs where each node (subject or object) is connected by edges (predicates).

In linked data, every resource (such as a battery, material, or test result) is identified by a **Uniform Resource Identifier (URI)**. URIs ensure that each entity is globally unique and can be referenced unambiguously across datasets. For example, a battery cell can have a unique URI like ``http://example.org/batteryCell/123``.

An **ontology** is a formal vocabulary that defines the relationships and concepts within a domain, providing a shared language for describing data. Ontologies in linked data use RDF to standardize the meanings of terms, ensuring that the same concept (e.g., "capacity") is understood consistently across different datasets. In the battery domain, ontologies like the Domain Battery Ontology (DBO) and the Elementary Multiperspective Materials Ontology (EMMO) are used to describe batteries, materials, and test processes.

SPARQL is the query language for extracting and manipulating data stored in RDF format. It allows users to query across datasets using relationships between entities. For example, a SPARQL query could be used to find all batteries with a capacity greater than 3000 mAh that use lithium-ion chemistry.

2.3 Linked data resources

Linked data resources are the vocabularies that make linked data possible. These vocabularies define the structure and meaning of data, allowing different datasets to connect and communicate in a standardized way. There are many vocabularies available, each suited to specific needs or domains, but all of them work together to support interoperability and meaningful data exchange. Below are some of the key linked data resources that are widely used to organize and describe information across various fields.

2.3.1 RDF Schema (RDFS)

RDF Schema (RDFS) is a framework that extends RDF (Resource Description Framework) by providing a basic vocabulary to describe relationships between resources. RDFS introduces concepts like classes and properties to allow more expressive descriptions of data. It also supports hierarchical structures through subclasses and sub properties, which allow users to define more specific relationships under broader categories. For example, you can define that a "Student" is a subclass of "Person," or that "birthPlace" is a sub property of "location."

RDFS helps establish the foundational structure of linked data, allowing different resources to be described in a standardized and meaningful way, making it a key building block for ontologies.

2.3.2 Dublin Core terms

Dublin Core is a set of standardized metadata terms used for describing digital resources, such as titles, creators, and subjects. These terms provide a flexible, widely adopted vocabulary for describing web content in a structured way.

2.3.3 Schema.org

Schema.org is a collaborative effort supported by major search engines like Google, Bing, and Yahoo to provide a shared vocabulary for structuring data on the web. Schema.org defines types and properties to describe common entities, such as people, places, and events, allowing websites and data providers to structure their information in a way that search engines and applications can easily interpret.

For example, the Person type in Schema.org can have properties like name, birthDate, and jobTitle, while the Event type includes properties like startDate and location. Schema.org's simplicity and broad adoption make it an ideal starting point for building interoperable linked data on the web.

2.3.4 CSV on the Web (CSVW)

CSV on the Web (CSVW) is a standard that defines how to represent metadata about CSV files to make them more accessible as linked data. CSVW helps transform tabular data (like CSV files) into structured, machine-readable formats by describing the columns, datatypes, and relationships in a CSV file.

With CSVW, you can:

- Define the structure of a CSV file (e.g., what each column represents).
- Validate the data (e.g., ensure columns contain the correct datatypes).
- Link the CSV data to external resources (e.g., linking values to related entities via URIs).

By using CSVW, even simple CSV files can be turned into linked data that is interoperable across the web, bridging the gap between traditional tabular data and more advanced linked data formats.

2.3.5 Data Catalogue Vocabulary (DCAT)

DCAT (Data Catalogue Vocabulary) is a W3C standard used for describing datasets and data catalogues, making it easier to share, discover, and integrate data across different platforms. DCAT provides a vocabulary for publishing metadata about datasets, including:

- **Dataset:** Describes a collection of data, such as a research dataset or a government dataset.
- **Catalogue:** Describes a collection of datasets, such as a data repository or a catalogue of open data.
- **Distribution:** Describes how the dataset is made available (e.g., as a downloadable file or API).

DCAT plays a crucial role in the open data movement, helping governments, organizations, and researchers publish their datasets in a standardized way, making them easier to find and reuse. By linking datasets to their descriptions, DCAT enables data cataloguing systems to interoperate, allowing users to search for and retrieve datasets more efficiently.

2.4 Linked data formats

Linked data can be expressed in several formats, but one of the most commonly used is JSON-LD (JavaScript Object Notation for Linked Data). JSON-LD is particularly appealing because it's both human-readable and machine-readable, making it a great choice for web applications and APIs.

What sets JSON-LD apart is its familiar structure using key-value pairs. Since it's based on JSON, a format widely used in web development, developers don't have to learn a new syntax to implement linked data. This ease of use makes it accessible and practical for integrating linked data into existing systems.

```
{
  "@context": "https://schema.org/",
  "@type": "Person",
  "@id": "https://www.wikidata.org/wiki/Q7186",
  "name": "Marie Curie",
  "alternateName": "Maria Skłodowska-Curie",
  "birthDate": "1867-11-07",
  "deathDate": "1934-07-04"
}
```

FIGURE 1. AN EXAMPLE OF JSON-LD MARKUP.

Let's walk through this example line-by-line to understand the core components of JSON-LD:

- **@context** This field tells us which vocabulary to use to interpret the data. In this case, "@context": "https://schema.org/" specifies that the data is structured according to the Schema.org vocabulary. Schema.org is a common set of terms used to describe things like people, places, and events. The @context ensures that terms like "name" or "birthDate" have the same meaning wherever they are used.
- **@type** This field defines what kind of thing the data is describing. Here, "@type": "Person" tells us that the entity being described is a Person. This helps systems understand that the data refers to a person and not, for example, an organization or a place.
- **@id** This field provides a unique identifier for the entity. In this case, "@id": "https://www.wikidata.org/wiki/Q7186" is a link to Marie Curie's profile on Wikidata. This makes it possible for systems to reference this exact person across datasets, ensuring that all data about this person is connected and consistent.

- **name** This field simply provides the person's full name, "Marie Curie". It's a straightforward way to label the entity, and because it follows the Schema.org vocabulary, it's recognized and used consistently across the web. If you look into the context file you will see that the string "name" maps to the URI <https://schema.org/name>.
- **alternateName** This field is used to specify another name by which the person is known. Here, "Maria Skłodowska-Curie" is Marie Curie's birth name, which is helpful for contexts where alternate names or aliases are relevant. Again, the string "alternateName" maps to the URI <https://schema.org/alternateName>.
- **birthDate** This field provides the person's birth date in YYYY-MM-DD format: "1867-11-07". This follows the Schema.org structure, allowing systems to interpret it correctly as a date, rather than a plain string. The context file maps the string "birthDate" to the URI <https://schema.org/birthDate>.
- **deathDate** This field follows the same format as "birthDate", marking the date of Marie Curie's death: "1934-07-04". The context file maps the string "deathDate" to <https://schema.org/deathDate>.

JSON-LD's ability to link data from different sources also makes it highly interoperable. By using URIs to reference objects, it ensures that data remains connected and meaningful, even when gathered from various domains. This linking capability supports the creation of rich, interconnected datasets, which are essential for many modern applications.

The key benefits of JSON-LD lie in its simplicity and linking capabilities. It's easy to use, thanks to its JSON foundation, and flexible enough to be embedded into web documents or exchanged between systems. Its ability to link data through unique identifiers (URIs) enables the creation of rich, interconnected datasets, making it a valuable tool for implementing linked data on the web.

2.5 Linked data examples

To understand linked data, it's helpful to look at practical examples that demonstrate how entities and relationships are represented using structured data formats like JSON-LD. The following examples show how people, places, and tabular datasets can be described and linked using schema.org and CSVW, highlighting core principles of linked data.

2.5.1 Example 1: A person

In this example, the fictional character Sherlock Holmes is described using schema.org terms. The structured data includes basic information such as his name, job title, birth date, and address. Additionally, relationships to other entities (e.g., his employer and colleague) are defined, and topics Sherlock is knowledgeable about are listed.

```
{
  "@context": "https://schema.org/",
  "@type": "Person",
  "name": "Sherlock Holmes",
  "jobTitle": "Detective",
  "birthDate": "1854-01-06",
  "worksFor": {
    "@type": "Organization",
    "name": "Self-employed"
  },
  "knowsAbout": ["Criminal Investigation", "Chemistry", "Forensic Science", "Violin"],
  "sameAs": "https://en.wikipedia.org/wiki/Sherlock_Holmes",
}
```

```

    "address": {
      "@type": "PostalAddress",
      "streetAddress": "221B Baker Street",
      "addressLocality": "London",
      "postalCode": "NW1"
    },
    "colleague": {
      "@type": "Person",
      "name": "Dr. John Watson"
    }
  }
}

```

FIGURE 2. JSON-LD INSTANCE DESCRIBING THE FICTIONAL PERSON SHERLOCK HOLMES

In this example:

- **@context** defines the vocabulary to be used (in this case, schema.org).
- **@type** specifies that this is a description of a Person.
- **worksFor** links to an Organization that describes his employment status as "Self-employed."
- **knowsAbout** lists areas of expertise (criminal investigation, chemistry, etc.), demonstrating relationships between entities.
- **sameAs** links to an external resource, connecting this description to a more detailed page on Wikipedia.
- **address** specifies where Sherlock lives, using a structured postal address.
- **colleague** describes his relationship with another person, Dr. John Watson.

2.5.2 Example 2: A place

Places can also be described using linked data. For example, the Taj Mahal can be described as a place with attributes like name, address, and description.

```

{
  "@context": "https://schema.org/",
  "@type": "Place",
  "name": "Taj Mahal",
  "description": "A famous monument located in Agra, India.",
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "Dharmapuri, Forest Colony",
    "addressLocality": "Agra",
    "addressRegion": "Uttar Pradesh",
    "postalCode": "282001",
    "addressCountry": "India"
  },
  "geo": {
    "@type": "GeoCoordinates",
    "latitude": 27.1751,
    "longitude": 78.0421
  },
  "sameAs": "https://en.wikipedia.org/wiki/Taj_Mahal"
}

```

FIGURE 3. JSON-LD INSTANCE DESCRIBING THE PLACE TAJ MAHAL.

In this example,

- **@type** defines the entity as a Place.
- The **address** includes structured details like **streetAddress**, **addressRegion**, and **postalCode**.
- **geo** provides the geographic coordinates (**latitude** and **longitude**).
- **sameAs** again links to an external resource (in this case, the Wikipedia page for the Taj Mahal).

2.5.3 Example 3: A person who shares a name with a place

To show how linked data can distinguish between entities that share the same name, we can describe Taj Mahal (the musician) alongside Taj Mahal (the monument). Even though these two things share the same name, they are clearly distinguished by the metadata linking them to different concepts in the schema.org vocabulary.

```
{
  "@context": "https://schema.org/",
  "@type": "Person",
  "name": "Taj Mahal",
  "jobTitle": "Musician",
  "birthDate": "1942-05-17",
  "sameAs": "https://en.wikipedia.org/wiki/Taj_Mahal_(musician)"
}
```

FIGURE 4. JSON-LD INSTANCE DESCRIBING A PERSON WITH THE NAME TAJ MAHAL.

2.5.4 Example 4: A tabular dataset

Tabular data can also be represented as linked data using CSVW (CSV on the Web). In this example, a dataset that tracks the height of a ball over time is described in a structured way, using metadata to explain the contents and provide machine-readable context.

TABLE 1. AN EXAMPLE TABULAR DATASET OF THE HEIGHT VS. TIME FOR A THROWN BALL.

Time_s	Height_m
0	0
1	4
2	7
3	9
4	10
5	9
6	7
7	4
8	0

```
{
  "@context": "http://www.w3.org/ns/csvw",
  "url": "thrown_ball.csv",
}
```

```

{
  "tableSchema": {
    "columns": [
      {
        "name": "Time_s",
        "titles": "Time (s)",
        "datatype": "number",
        "description": "Time in seconds since the ball was thrown."
      },
      {
        "name": "Height_m",
        "titles": "Height (m)",
        "datatype": "number",
        "description": "The height of the ball in meters at the given
time."
      }
    ]
  }
}

```

FIGURE 5. JSON-LD METADATA DESCRIBING A TABULAR DATASET USING CSVW TERMS.

In this example,

- url links to the CSV file containing the data.
- tableSchema describes the structure of the table, with columns for Time_s and Height_m.

3 BATTERY DOMAIN ONTOLOGIES

3.1 EMMO-based ontologies relevant for batteries

The Elementary Multiperspective Materials Ontology (EMMO) offers a comprehensive framework for structuring domain-specific ontologies in scientific fields, with a strong emphasis on formal knowledge representation. In the context of battery research, several EMMO-based ontologies provide a structured, interoperable way to describe data related to electrochemical processes, battery components, and testing methodologies. This approach enables consistent data sharing and integration across different research platforms and disciplines. This section provides a detailed technical overview of the key EMMO-based ontologies used in battery research:

- Domain Electrochemistry Ontology
- Domain Battery Ontology
- BattINFO

These ontologies play a crucial role in standardizing how data is described, facilitating interoperability and enhancing data reuse across different platforms and disciplines.

3.1.1 EMMO Electrochemistry domain

The Electrochemistry Domain Ontology provides a foundational framework for describing concepts and processes common to all electrochemical systems, including key elements such as electrodes, electrolytes, active materials, and electrochemical quantities. These concepts are central not only to battery systems but also to a wide range of other electrochemical applications, including fuel cells, supercapacitors, and electrolyzers. By formalizing these core concepts, the ontology enables consistent and standardized descriptions of the various processes, materials, and parameters involved in electrochemical systems.

It is designed as an independent domain ontology, allowing it to be reused across multiple domains, not limited to batteries. This flexibility is critical for ensuring that researchers in different fields, such as fuel cell research, can use the same terminology and structure to describe their systems, promoting interoperability and data reuse across disciplines.

The Electrochemistry Domain Ontology adheres to best practices for ontology development, following guidelines that ensure it is robust, scalable, and easy to maintain. These practices include clear documentation, the use of established vocabularies, and regular updates to accommodate new developments in the field. The ontology currently has a FOOPS score of 91%, reflecting its high level of formalization and adherence to best practices. The FOOPS score measures how well an ontology supports FAIR data principles, ensuring that it meets the needs of modern research environments.

This high score indicates that the ontology is not only comprehensive but also well-positioned for broad adoption across various electrochemical research fields. It enables interdisciplinary collaboration by providing a common language for describing electrochemical processes, materials, and test data, making it an essential tool for researchers in battery science and beyond.

3.1.2 EMMO Battery domain

The Battery Domain Ontology is an extension of the Electrochemistry Domain Ontology, specifically designed to cover the components and processes unique to battery systems. While the Electrochemistry Ontology provides foundational concepts such as electrodes, electrolytes, and electrochemical processes common to all electrochemical systems, the Battery Ontology builds on these to provide a more detailed representation of the physical structure, components, and performance metrics specific to batteries. This extension ensures that battery-specific details can be fully captured, enhancing the representation of complex systems within battery research.

The Battery Ontology expands the scope of the Electrochemistry Domain by introducing detailed descriptions of battery cells, including their formats, chemistries, and interfaces. It allows for the formalized representation of materials and components used in batteries. Additionally, it enables detailed descriptions of key processes like charge and discharge cycles, degradation mechanisms, and battery lifecycle performance.

A notable feature of the Domain Battery Ontology is that it incorporates terminology from established standards, including those defined by the International Electrotechnical Commission (IEC). By aligning with these globally recognized standards, the ontology provides a comprehensive vocabulary that supports the accurate description of battery resources, ensuring consistency with industry practices. This adherence to established terminology allows for greater interoperability between academic research, industrial applications, and regulatory compliance.

The Battery Domain Ontology currently has a FOOPS score of 75%, indicating a need for further refinement, particularly in areas such as documentation, extensibility, and broader community input. However, this score still reflects a robust and well-structured ontology, capable of accurately representing the physical components of battery systems, their KPIs, and common research workflows.

3.1.3 BattINFO

BattINFO is a higher-level ontology that serves as a wrapper for both the Domain Battery Ontology and the Domain Electrochemistry Ontology, streamlining the integration and use of battery-related data from multiple sources. By encapsulating these two core ontologies, BattINFO simplifies the representation of battery systems, offering a user-friendly framework for describing battery components, electrochemical processes, and performance metrics. Its primary purpose is to facilitate easier data management and interoperability, allowing researchers to describe battery systems without needing to engage deeply with the complex technical structures of the underlying ontologies.

BattINFO currently acts as an interface between the more detailed domain ontologies, offering predefined templates for describing common battery-related resources and processes. These templates ensure that data can be structured in a consistent, interoperable manner, while still adhering to the formal vocabularies provided by the Domain Battery and Electrochemistry ontologies. By streamlining the process, BattINFO makes it easier for researchers to organize and share their data in a standardized format.

In the future, BattINFO will be merged with the Battery Domain Ontology, consolidating the efforts of both ontologies into a single, more cohesive framework. This integration will eliminate redundancy while retaining the high level of detail and standardization needed to describe the complexities of battery systems. The merged ontology will provide a unified platform for representing everything from material properties and electrochemical reactions to battery performance and lifecycle data, ensuring that the combined resources are accessible, reusable, and easily integrated across different research domains.

3.2 Best practice guidelines for publishing battery linked data

Publishing battery-related data in a way that ensures it is findable, accessible, interoperable, and reusable is critical for advancing research and fostering collaboration. These best practices build on the W3C Best Practices for Publishing Linked Data and the 5-star principles of linked data, emphasizing the importance of structured and open data formats. Key principles include using open formats, providing complete metadata, applying clear licenses, and ensuring compatibility with evolving ontologies. By adhering to these guidelines, researchers can improve data quality, facilitate integration with other datasets, and promote transparency and reproducibility, ensuring that battery data is

valuable and useful within the wider scientific community. The alignment with these foundational best practices guarantees that battery data is not only well-structured but also maximizes its potential for reuse and linking across diverse research platforms and domains.

These best practices are additive, with each step building on the previous to progressively increase the value and usability of battery data. Starting with basic accessibility, each practice enhances findability, interoperability, and reusability, ensuring data is not only available but fully optimized for integration and collaboration. Following all ten practices maximizes data's impact within the wider research ecosystem.

3.2.1 Best Practice 1: Publish your battery data online

Description: The first step to making your battery data reusable and accessible is to publish it online. By placing your data on the web, you ensure that other researchers can access and work with it. This could involve publishing raw data, processed results, or other associated metadata.

Why: Making data available online is the foundation of open science. It encourages collaboration, supports reproducibility, and enables wider dissemination of research results.

How: Use reliable and open-access repositories (e.g., institutional repositories, Zenodo, etc.) to host your data. Ensure that the dataset has a unique and accessible URL.

Example: Publishing test data as a downloadable CSV file from an open repository ensures that others can access the dataset: <https://github.com/emmo-repo/domain-battery/blob/master/docs/assets/data/example-cyclic-voltammetry.csv>

3.2.2 Best Practice 2: Apply an open and clear license

Description: When you publish your battery data, it is crucial to apply a clear and open license that defines how others can use, modify, and share your data. This removes ambiguity and promotes open collaboration by clearly stating the terms of reuse.

Why: Without an explicit license, other researchers may be unsure how they are allowed to use your data, limiting its potential for reuse. Open licenses like Creative Commons enable the free use and distribution of data with clear conditions.

How: Choose an open license that fits your needs. For example, a Creative Commons Attribution (CC BY) license allows others to use your data as long as they provide appropriate credit. Ensure that the license information is visible in your dataset's metadata or documentation.

Example: A dataset with a creative commons license.

```
{
  "name": "My Dataset",
  "license": "https://creativecommons.org/licenses/by/4.0/"
}
```

FIGURE 6. JSON METADATA CONTAINING LICENSE INFORMATION

3.2.3 Best Practice 3: Use structured data formats

Description: To make your battery data more accessible and machine-readable, structure it using well-defined formats like XLSX, CSV, or PARQUET. Structured data formats provide consistency and make it easier for other researchers to understand and analyse the data.

Why: Structured data improves the usability of datasets, enabling easier processing, analysis, and integration with other tools and systems. It also makes your data compatible with automated systems that rely on structured formats.

How: Ensure that each dataset follows a consistent structure. For tabular data, use a CSV file with clearly labeled columns. For more complex data, use hierarchical formats like JSON or XML.

Example: An excel file with structured battery test data might include columns for time, voltage, and current.

3.2.4 Best Practice 4: Use non-proprietary, open formats

Description: To maximize accessibility and interoperability, publish your battery data in non-proprietary formats that are open and widely supported, such as CSV, JSON, HDF5, or PARQUET. Avoid proprietary formats like Excel that may restrict data usability across platforms.

Why: Non-proprietary formats ensure that your data can be accessed and processed by any system, promoting long-term reuse and avoiding issues with software compatibility.

How: Convert your data into open formats like CSV for tabular data or JSON-LD for linked data. Ensure that the format is machine-readable and that the structure is standardized.

Example: A battery test dataset published in an open CSV format:

3.2.5 Best Practice 5: Use RDF vocabularies to describe and identify your data

Description: Use RDF (Resource Description Framework) vocabularies to describe and identify key components, materials, and processes in your battery data. RDF provides a structured, linked data approach that allows your data to be shared and understood across different systems.

Why: RDF vocabularies ensure that data entities are described consistently using globally recognized standards. This allows for better integration with other datasets and improves data interoperability.

How: Use RDF vocabularies such as Schema.org, Dublin Core, or domain-specific ontologies (e.g., Domain Battery Ontology) to define key terms. Assign URIs to materials, components, and processes to uniquely identify them.

Example: A JSON-LD representation of battery data using RDF vocabularies. In this case, the context file maps the string “dcat:Distribution” to the URI <https://www.w3.org/ns/dcat#Distribution>

```
{
  "@context": "https://w3id.org/emmo/domain/battery/context",
  "@type": "dcat:Distribution",
  "schema:name": "My Dataset",
  "dcat:license": "https://creativecommons.org/licenses/by/4.0"
}
```

FIGURE 7. AN EXAMPLE OF SIMPLE JSON-LD MARKUP FOR A DISTRIBUTION

3.2.6 Best Practice 6: Provide minimum metadata for reuse

Description: At a minimum, include metadata that clearly states what the data is and where the data is located. This ensures that other researchers can easily discover, interpret, and access the data.

Why: Without basic metadata, datasets are hard to find and understand, reducing their potential for reuse. Metadata that describes the content and location of the dataset ensures it is discoverable and accessible.

How: Provide basic metadata fields such as the dataset’s name, a brief description, and a URL to access it. Include this metadata in both human-readable and machine-readable formats.

Example:

```
{
  "@context": "https://w3id.org/emmo/domain/battery/context",
```

```

    "@type": "dcat:Distribution",
    "schema:name": "My Dataset",
    "dcat:license": "https://creativecommons.org/licenses/by/4.0",
    "dcat:downloadUrl": "https://raw.githubusercontent.com/emmo-repo/domain-battery/refs/heads/master/docs/assets/data/example-cyclic-voltammetry.csv"
  }

```

3.2.7 Best Practice 7: Ensure comprehensive metadata for clarity and reusability

Description: To ensure that others can effectively interpret and use your dataset, include complete and detailed metadata that describes all relevant aspects of the data, such as its structure, variables, units, data collection methods, and any assumptions or limitations. Complete metadata ensures that the dataset can be accurately understood and reused in different contexts.

Why: Without complete metadata, datasets are often difficult to interpret or use, limiting their potential for reuse. Metadata provides the necessary context for understanding the data, including what the data represents, how it was collected, and how it should be interpreted.

How: Provide detailed descriptions for each variable (e.g., time, voltage, current), the units of measurement, and any specific conditions under which the data was collected (e.g., temperature, test environment). Include metadata on data collection protocols, instrumentation, and any pre-processing steps.

Example: A metadata record explaining the variables in a battery test dataset:

```

{
  "@context": "https://w3id.org/emmo/domain/battery/context",
  "@type": "dcat:Distribution",
  "schema:name": "My Dataset",
  "dcat:license": "https://creativecommons.org/licenses/by/4.0",
  "dcat:downloadUrl": "https://raw.githubusercontent.com/emmo-repo/domain-battery/refs/heads/master/docs/assets/data/example-cyclic-voltammetry.csv",
  "csvw:dialect": {
    "@type": "csvw:Dialect",
    "csvw:skipRows": 2,
    "csvw:delimiter": "\t"
  },
  "csvw:tableSchema": "https://raw.githubusercontent.com/emmo-repo/domain-electrochemistry/master/docs/assets/jsonld/csvw_table_schema_maccor_2.json"
}

```

3.2.8 Best Practice 8: Document your data with human-readable descriptions

Description: Provide clear and comprehensive documentation that explains the structure, contents, and context of your data. This ensures that others can easily understand and reuse your dataset.

Why: Documentation is essential for interpreting datasets, especially for external researchers who were not involved in the original project. Without detailed descriptions, data can be misinterpreted or underutilized.

How: Include a README file or comprehensive metadata that explains the variables, units, methods, and conditions under which the data was collected. Ensure that this documentation is available alongside the dataset.

Example: A README file explaining the variables in a battery cycle life dataset.

```
{
  "@context": "https://w3id.org/emmo/domain/battery/context",
  "@type": "dcat:Distribution",
  "schema:name": "My Dataset",
  "dcat:license": "https://creativecommons.org/licenses/by/4.0",
  "dcat:downloadUrl": "https://raw.githubusercontent.com/emmo-repo/domain-battery/refs/heads/master/docs/assets/data/example-cyclic-voltammetry.csv",
  "csvw:dialect": {
    "@type": "csvw:Dialect",
    "csvw:skipRows": 2,
    "csvw:delimiter": "\t"
  },
  "csvw:tableSchema": "https://raw.githubusercontent.com/emmo-repo/domain-electrochemistry/master/docs/assets/jsonld/csvw_table_schema_maccor_2.json",
  "dc:title": "BG_Maccor_Testdata - 079",
  "dc:description": "A TXT file containing constant current constant voltage (CCCV) test data from an Maccor cycler"
}
```

3.2.9 Best Practice 9: Link your data to other datasets

Description: Link your battery data to related datasets to provide richer context and facilitate cross-domain analysis. This could include linking to material properties databases, external test results, or environmental impact datasets.

Why: Linking data increases its utility by enabling connections with other relevant information. This enhances the value of your dataset by allowing researchers to integrate it with other sources of knowledge.

How: Use URIs to link your dataset to other open datasets or databases. For example, link to external material databases or published research that provides additional context.

Example: Linking a dataset to an external material database.

3.2.10 Best Practice 10: Stay up-to-date with ontology developments

Description: Ensure that your use of ontologies remains current by monitoring updates to the ontologies you rely on. As battery research and technologies evolve, so do the ontologies that represent them.

Why: Ontologies are constantly updated to reflect new materials, technologies, and processes. Staying updated ensures that your data remains relevant and interoperable with new datasets and standards.

How: To simplify managing ontology updates, many user-friendly Python libraries offer built-in support for versioning, usage, and seamless integration of ontology updates. Tools such as rdflib, OWLready2, EMMOntoPy, and other ontology management libraries provide functions to easily query, update, and synchronize datasets with ontology repositories, often with minimal coding.

For practical management, regularly reviewing ontology repositories and leveraging libraries that automate these updates allow you to revise datasets efficiently. This approach ensures that your data remains aligned with the latest standards and structures, facilitating smoother integration with emerging research and maintaining long-term interoperability.

Example: Tracking the version of the ontology used in your dataset

3.3 Battery linked data examples

This section provides practical examples of how battery-related data can be represented using linked data principles. By utilizing battery ontologies these examples demonstrate how different components and properties of a battery system can be described in a structured, machine-readable format using JSON-LD. Each example illustrates a key aspect of battery data, from simple descriptions of battery cells to more detailed representations of materials, electrolytes, and testing data.

It is important to note that, in practice, researchers are not expected to manually enter JSON-LD or directly manage the intricacies of linked data. Instead, these linked data representations are best created through user-friendly data management interfaces. Ideally, these interfaces are integrated directly within the devices, control software, or simulation tools that generate the data, allowing researchers to capture metadata and create linked data seamlessly, with minimal effort. This approach supports researchers by managing data linking "in the backend" without requiring manual input. By embedding linked data tools into familiar systems, researchers can focus on their work while benefiting from the advantages of linked data for future sharing, analysis, and reuse.

3.3.1 Example 5: A simple battery cell

In this example, a CR2032 coin cell is described using a combination of schema.org and the Domain Battery Ontology. This representation includes the name, manufacturer, and key properties such as the nominal capacity. By linking to external resources like Wikidata, it becomes easy to identify and reference relevant entities such as the manufacturer.

```
{
  "@context": "https://raw.githubusercontent.com/emmo-repo/domain-battery/master/context.json",
  "@type": "CR2032",
  "schema:name": "My CR2032 Coin Cell",
  "schema:manufacturer": {
    "@id": "https://www.wikidata.org/wiki/Q3041255",
    "schema:name": "SINTEF"
  },
  "hasProperty": {
    "@type": ["NominalCapacity", "ConventionalProperty"],
    "hasNumericalPart": {
      "@type": "Real",
      "hasNumericalValue": 230
    },
    "hasMeasurementUnit": "emmo:MilliAmpereHour"
  }
}
```

3.3.2 Example 6: An active material

This example describes Lithium Iron Phosphate (LFP), a common active material used in battery electrodes. It shows how different properties, such as particle size distribution, density, and specific surface area, can be described using RDF vocabularies. The @reverse keyword is used to create a custom property definition, demonstrating flexibility in linked data representation.

```
{
  "@context": [
    "https://w3id.org/emmo/domain/electrochemistry/context",
```

```

    {
      "isOutputOf": { "@reverse": "hasOutput" }
    }
  ],
  "@type": ["ActiveMaterial", "LithiumIronPhosphate", "Powder"],
  "rdfs:comment": [
    "This example shows how to describe an active material using
    information typically found in specifications or safety data sheets (SDS).",
  ],
  "molecularFormula": "LiFePO4",
  "pubChemReference": "https://pubchem.ncbi.nlm.nih.gov/compound/15320824",
  "hasProperty": [
    {
      "@type": "D10ParticleSize",
      "hasNumericalPart": { "@type": "Real", "hasNumericalValue": 1.0 },
      "hasMeasurementUnit": "emmo:MicroMetre"
    },
    {
      "@type": "D50ParticleSize",
      "hasNumericalPart": { "@type": "Real", "hasNumericalValue": 3.5 },
      "hasMeasurementUnit": "emmo:MicroMetre"
    },
    {
      "@type": "TappedDensity",
      "hasNumericalPart": { "@type": "Real", "hasNumericalValue": 1.45
    },
    "hasMeasurementUnit": "emmo:GramPerCubicCentiMetre"
  ]
}

```

3.3.3 Example 7: An electrolyte

In this example, a custom electrolyte composition, LP57, is described. The solvent and solute components, along with their respective mass fractions, are detailed. This shows how to handle complex compositions with constituents and the use of mass fractions relative to the total solution.

```

{
  "@context": "https://w3id.org/emmo/domain/electrochemistry/context",
  "@type": "Electrolyte",
  "rdfs:label": "LP57",
  "rdfs:comment": ["This example demonstrates custom electrolyte
  composition descriptions using properties 'hasSolvent' and 'hasSolute'.",
  "Quantities like 'MassFraction' that are defined with respect to some total
  value refer back to the nearest holistic whole that they are part of. For
  example, the mass fraction of ethylene carbonate is defined with respect to
  the total mass of the solvent, while the mass fraction of the lithium salt
  is defined with respect to the whole electrolyte.", "The notation LP57 is
  derived from a manufacturer-specific product line and doesn't have general

```

```

applicability. It has been adopted as a shorthand in the domain, but it is
not a regulated descriptive code."],
  "hasSolvent": {
    "@type": "Solvent",
    "hasConstituent": [
      {
        "@type": "EthyleneCarbonate",
        "hasProperty": {
          "@type": "MassFraction",
          "hasNumericalPart": {
            "@type": "Real",
            "hasNumericalValue": 0.3
          },
          "hasMeasurementUnit": "emmo:UnitOne"
        }
      },
      {
        "@type": "EthylMethylCarbonate",
        "hasProperty": {
          "@type": "MassFraction",
          "hasNumericalPart": {
            "@type": "Real",
            "hasNumericalValue": 0.7
          },
          "hasMeasurementUnit": "emmo:UnitOne"
        }
      }
    ]
  },
  "hasSolute": {
    "@type": "LithiumHexafluorophosphate",
    "hasProperty": {
      "@type": "MassFraction",
      "hasNumericalPart": {
        "@type": "Real",
        "hasNumericalValue": 0.127
      },
      "hasMeasurementUnit": "emmo:UnitOne"
    }
  }
}

```

3.3.4 Example 8: A detailed battery cell

Here, a more complex battery cell description is provided, including the positive and negative electrodes, case type, and nominal voltage. This example showcases how multiple components and properties can be combined to give a complete picture of the battery.

```
{
```

```

"@context": "https://w3id.org/emmo/domain/battery/context",
"@type": "BatteryCell",
"schema:name": "My LFP-Graphite R2032 Coin Cell",
"schema:manufacturer": {
  "@id": "https://www.wikidata.org/wiki/Q3041255",
  "schema:name": "SINTEF"
},
"hasPositiveElectrode": {
  "@type": "Electrode",
  "hasActiveMaterial": {
    "@type": "LithiumIronPhosphate"
  }
},
"hasNegativeElectrode": {
  "@type": "Electrode",
  "hasActiveMaterial": {
    "@type": "Graphite"
  }
},
"hasCase": {
  "@type": "R2032"
},
"hasProperty": {
  "@type": ["NominalVoltage", "ConventionalProperty"],
  "hasNumericalPart": {
    "@type": "Real",
    "hasNumericalValue": 3.2
  },
  "hasMeasurementUnit": "emmo:Volt"
}
}

```

3.3.5 Example 9: A Battery test data schema

This final example describes the schema for test data using CSVW (CSV on the Web). It defines the structure of the data, including columns for time, cycle number, voltage, current, and capacity. This schema ensures that test data can be consistently represented and easily interpreted.

```

{
  "@context": "https://w3id.org/emmo/domain/battery/context",
  "@type": "csvw:TableSchema",
  "csvw:columns": [
    {
      "@type": "csvw:Column",
      "csvw:name": "time",
      "csvw:titles": "Test Time",
      "dc:description": "time of the current data point relative to the  
start of the test",
      "csvw:propertyUrl": {

```

```

    "@type": "TestTime"
  },
  "hasMeasurementUnit": [
    {
      "@id": "emmo:Hour"
    },
    {
      "@id": "unit:HR"
    }
  ],
  "schema:unitCode": [
    {
      "@id": "emmo:Hour"
    },
    {
      "@id": "unit:HR"
    }
  ],
  "csvw:datatype": "xsd:decimal",
  "csvw:required": "true"
},
{
  "@type": "csvw:Column",
  "csvw:name": "cycle",
  "csvw:titles": "Cycle Index",
  "dc:description": "the current cycle number in the test",
  "csvw:propertyUrl": {
    "@type": "CycleIndex"
  },
  "hasMeasurementUnit": [
    {
      "@id": "emmo:UnitOne"
    },
    {
      "@id": "unit:UNITLESS"
    }
  ],
  "schema:unitCode": [
    {
      "@id": "emmo:UnitOne"
    },
    {
      "@id": "unit:UNITLESS"
    }
  ],
  "csvw:datatype": "xsd:integer",
  "csvw:required": "false"
},
{

```

```

    "@type": "csvw:Column",
    "csvw:name": "volt",
    "csvw:titles": "Voltage",
    "csvw:propertyUrl": {
      "@type": "CellVoltage"
    },
    "hasMeasurementUnit": [
      {
        "@id": "emmo:Volt"
      },
      {
        "@id": "unit:V"
      }
    ],
    "schema:unitCode": [
      {
        "@id": "emmo:Volt"
      },
      {
        "@id": "unit:V"
      }
    ],
    "csvw:datatype": "xsd:decimal",
    "csvw:required": "true"
  },
  {
    "@type": "csvw:Column",
    "csvw:name": "curr",
    "csvw:titles": "Current",
    "csvw:propertyUrl": {
      "@type": "CellCurrent"
    },
    "hasMeasurementUnit": [
      {
        "@id": "emmo:Ampere"
      },
      {
        "@id": "unit:A"
      }
    ],
    "schema:unitCode": [
      {
        "@id": "emmo:Ampere"
      },
      {
        "@id": "unit:A"
      }
    ],
    "csvw:datatype": "xsd:decimal",

```

```

    "csvw:required": "true"
  },
  {
    "@type": "csvw:Column",
    "csvw:name": "ah",
    "csvw:titles": "Capacity",
    "csvw:propertyUrl": {
      "@type": "Capacity"
    },
    "hasMeasurementUnit": [
      {
        "@id": "emmo:AmpereHour"
      },
      {
        "@id": "unit:A-HR"
      }
    ],
    "schema:unitCode": [
      {
        "@id": "emmo:AmpereHour"
      },
      {
        "@id": "unit:A-HR"
      }
    ],
    "csvw:datatype": "xsd:decimal",
    "csvw:required": "false"
  },
  {
    "@type": "csvw:Column",
    "csvw:name": "ah-step",
    "csvw:titles": "Step Capacity",
    "csvw:propertyUrl": {
      "@type": "StepCapacity"
    },
    "hasMeasurementUnit": [
      {
        "@id": "emmo:AmpereHour"
      },
      {
        "@id": "unit:A-HR"
      }
    ],
    "schema:unitCode": [
      {
        "@id": "emmo:AmpereHour"
      },
      {
        "@id": "unit:A-HR"
      }
    ]
  }

```

```
    }  
  ],  
  "csvw:datatype": "xsd:decimal",  
  "csvw:required": "false"  
}  
],  
"csvw:primaryKey": "time"  
}
```

4 CONCLUSION

In this report, we have reviewed the role that ontology-based linked data approaches can play in advancing battery research. As the complexity and volume of battery data increase, standardizing the way this data is structured is essential for driving innovation and collaboration. By leveraging well-developed domain ontologies such as the Electrochemistry Domain Ontology, Battery Domain Ontology, and BattINFO, researchers can more effectively capture the nuances of battery materials, components, processes, and performance metrics. These ontologies not only ensure consistency and interoperability across research platforms but also promote data reuse, making it easier for researchers to build on each other's work.

The application of linked data principles and structured data formats, such as RDF and JSON-LD, enables researchers to integrate experimental data with computational models, simulations, and external datasets, offering a holistic view of the battery. This capability is especially important in multidisciplinary fields like battery research, where data must be exchanged and understood by a wide range of experts.

To maximize the impact of battery data, researchers must follow best practices for publishing and sharing their data, including using open and non-proprietary formats, applying clear licenses, and ensuring comprehensive metadata. These practices facilitate collaboration, enhance the value of published data, and support the evolving needs of the research community.

As battery research continues to evolve, the ontologies and frameworks discussed in this report will need to be continually updated to reflect new technologies, materials, and regulatory requirements. Staying up-to-date with these developments will be crucial for maintaining the relevance and usefulness of battery linked data in the future.

In conclusion, the use of battery ontologies in linked data offers immense potential to enhance data sharing, collaboration, and discovery within the field. By adopting these approaches, the battery research community can overcome the challenges of data silos and realize the full potential of the data they generate.