A General Schema for Time Series Data Quality Guided by Real-World Use Cases and Based on International Standards

Ulrich Loup [1], Nicole Büttner [2], Romy Fösig [2], Marc Hanisch [4], Martin Ingenbleek [1], Ralf Kunkel [1], Christof Lorenz [2], Hylke van der Schaaf [5], David Schäfer [3], and Jürgen Sorg [1]

[1]Forschungszentrum Jülich
[2]Karlsruhe Institute of Technology
[3]Helmholtz Centre for Environmental Research GmbH – UFZ
[4]Helmholtz Centre Potsdam – GFZ German Research Centre for Geosciences
[5]Fraunhofer IOSB

April 2026

Author Contributions

Conceptualization: Ulrich Loup
Methodology: Ulrich Loup, Jürgen Sorg, Martin Ingenbleek, Nicole Büttner, Hylke van der Schaaf
Validation: David Schäfer, Marc Hanisch, Christof Lorenz
Writing – Original Draft: Ulrich Loup, David Schäfer, Hylke van der Schaaf
Writing – Review & Editing: All authors

Contributors

Data and Use Case Contributions:

Jannis Groh (Forschungszentrum Jülich, Leibniz-Zentrum für Agrarlandschaftsforschung (ZALF) e.V., Müncheberg)

Benjamin Louisot (Karlsruhe Institute of Technology)

Robert Wiesen (Helmholtz Centre for Environmental Research GmbH – UFZ,)

The World Wide Web Consortium (W3C) [1, Sec. 8.5] provides general best practices for including data quality information in data shared over the web. However, their implementation in practice often requires mapping or interpreting W3C concepts to the application domain. We report on a concrete approach to implementing the W3C Data Quality recommendations into time series data, thereby applying them to a wide range of scientific processes.

To that end, we propose a general schema for modeling data quality control information for time series data. The schema is guided by prominent use cases from the Earth and environmental sciences. It incorporates data quality flags, as well as processing information from automated quality control procedures and data inspections by domain experts. We provide a concrete implementation of the schema in the SensorThings API data model. Additionally, we demonstrate how file-based time series data can be annotated using the proposed schema in RO-Crates and the NetCDF format. By deeply integrating the W3C standard, we obtain a practice-oriented, semantically sound schema. We demonstrate the schema's implementation for its initial use cases and provide additional relevant examples.

The proposed schema realizes quality control in the SensorThings API data model and for file-based time series data. Our approach preserves the original domain-specific structures while ensuring compliance with the W3C recommendations. Thus, we offer a straightforward plan to improve the readability and machine actionability of existing data quality information and corresponding workflows across domains. We even enable their interoperability on an international level.

1 Introduction

Environmental research is increasingly driven by large volumes of time series data collected from complex and heterogeneous observation networks. These data underpin critical insights into climate change, air quality, hydrological processes, and ecosystem dynamics, and thus require a high degree of reliability, transparency, and interpretability. Consequently, the systematic description and communication of data quality information have become essential components of modern data management, particularly in the context of FAIR and interoperable research infrastructures. While general frameworks such as the W3C Data Quality Vocabulary provide conceptual guidance, their practical application to time series data and operational workflows remains a challenge.

The development of the proposed schema is guided by concrete requirements from two major environmental research infrastructures, namely TERENO (Terrestrial Environmental Observatories) and ACTRIS (Aerosol, Clouds and Trace Gases Research Infrastructure). Both initiatives operate large-scale, distributed observation networks that continuously generate heterogeneous time series data across domains such as hydrology, atmospheric composition, and ecosystem dynamics. These data form a critical basis for scientific analysis and policy-relevant assessments, which places strong emphasis on transparent and reliable data quality control (QC) procedures. As part of their data policies, both infrastructures mandate the application of automated QC methods, such as threshold-based filtering during data ingestion, complemented by subsequent expert-based validation prior to data publication.

While automated tools such as *SaQC* provide powerful mechanisms for scalable QC, they are typically embedded in specialized, often non-interoperable workflows. At the same time, manual QC remains indispensable for identifying context-dependent anomalies, yet is currently supported by fragmented and outdated tooling. Within the emerging ecosystem of the Helmholtz Earth and Environment DataHub, these limitations become increasingly critical, as infrastructures such as TERENO transition towards standardized, service-oriented data access based on the SensorThings API. The need to consistently represent both automated and manual QC processes across systems, while ensuring interoperability and alignment with FAIR principles, motivated the development of a general, standards-based schema. By grounding this work in the practical requirements of TERENO and ACTRIS, we ensure that the proposed approach addresses real-world challenges while remaining applicable to a broad range of time series data workflows.

A central objective of this work is to address practical design questions that arise when integrating quality information into time series data workflows, in particular where such information should be stored and to what level of completeness it should be represented. In general, two complementary approaches can be distinguished: quality information may either be embedded directly within individual data points of a time series, enabling immediate co-location with observations, or it may be maintained in dedicated, external entities that provide a richer and more expressive representation of quality assessment processes and results. While the former approach facilitates efficient access and straightforward integration into existing data structures, it is often limited in its ability to capture the full context of QC procedures. Conversely, the latter approach supports comprehensive, semantically explicit descriptions but may introduce additional complexity in data access and interpretation.

To reconcile these trade-offs, we pursue a hybrid approach that combines the strengths of both strategies. Specifically, we define a metadata-based information model, aligned with established standards such as the W3C Data Quality Vocabulary, to represent quality control processes, provenance, and detailed results in a structured and interoperable manner. In parallel, we introduce a lightweight and condensed representation of quality information directly at the level of individual data points. This enables efficient access and filtering in typical analysis workflows while maintaining a clear semantic linkage to the full, standards-compliant metadata model. In this way, the proposed schema supports both high-performance data handling and comprehensive, transparent documentation of quality information.

2 Background and Related Standards

2.1 W3C Data Quality Best Practices and Vocabulary.

The W3C Data on the Web Best Practices emphasize the importance of explicitly describing and publishing data quality information as an integral part of interoperable data ecosystems. In this context, the Data Quality Vocabulary (DQV) provides a lightweight, RDF-based framework for representing quality metadata in a standardized and machine-readable way. Central to its model is the concept of a `dqv:QualityMeasurement`, which captures the result of assessing a specific aspect of a dataset or resource. Each measurement is associated with a *dqv:Metric*, defining how the quality is evaluated, and further linked to a *dqv:Dimension*, representing a broader quality aspect such as accuracy, completeness, or timeliness. This layered structure enables flexible yet consistent modeling of quality information, supports provenance descriptions, and facilitates integration with existing standards such as PROV-O. As such, DQV provides a well-established conceptual foundation for expressing data quality in a way that can be aligned with time series models such as the SensorThings API.

2.2 SensorThings API

The OGC SensorThings API (STA) [2] provides a standardized, RESTful interface for managing and accessing sensor data in heterogeneous IoT environments. It is based on the Observations & Measurements (O&M) [3] model and adopts an observation-centric view on data. The core concept is that sensors associated with physical or virtual entities (`sta:Things`) produce observations (`sta:Observation`) of specific observed properties (`sta:ObservedProperties`). These observations are grouped into `sta:Datastream`, which ensure semantic consistency by linking a single `sta:ObservedProperty`, `sta:Sensor`, and `sta:Thing`. Each `sta:Thing` may be associated with spatial information via `sta:Locations` and each Observation may have a distinct spatial object that it is associated with via `sta:FeatureOfInterest`. This separation between metadata (e.g., `sta:Thing`, `sta:Sensor`, `sta:ObservedProperty`) and observational data enables flexible modeling of time series while preserving interoperability. As such, the SensorThings API provides a widely adopted conceptual foundation for structuring time series data, which is directly relevant when extending such models with standardized representations of data quality.

The forthcoming version 2.0 of STA introduces several extensions that further refine the conceptual model and improve its expressiveness for complex observation scenarios. In particular, the Observations & Measurements (O&M) extension adds the concept of an `sta:ObservingProcedure`, which generalizes the notion of a sensor by explicitly representing the procedure, method, or workflow used to derive an observation. This allows for a more flexible description of data generation processes, including simulations, aggregations, or quality assessment procedures, and thereby aligns well with provenance-oriented and data quality use cases. In addition, the Relations extension introduces `sta:RelatedDatastream` as a first-class construct to explicitly capture relationships between datastreams. This enables the formalization of dependencies such as derivation, aggregation, or transformation between time series, which is particularly relevant for representing quality-related processing chains. Together, these extensions strengthen the ability of the SensorThings API to model not only raw observations but also their processing context and interconnections, providing a useful foundation for integrating structured data quality information.

2.3 STAMPLATE Schema

The *STAMPLATE* (SensorThings API Metadata Profile for Linked dAta in environmenTal rEsearch) data model provides a structured approach for representing rich metadata in time series data infrastructures based on the OGC SensorThings API. It was developed within the context of Helmholtz Earth and Environment research activities to address limitations in the native SensorThings data model with respect to semantic expressiveness, interoperability, and extensibility. In particular, STAMPLATE introduces a consistent mechanism to attach domain-specific metadata to core SensorThings entities such as Datastreams, Observations, Sensors, and Things, while maintaining compatibility with existing service interfaces.

A key design principle of STAMPLATE is the integration of Linked Data concepts into operational data services. By leveraging JSON-LD [4], the model enables the annotation of SensorThings entities with semantic types and contextual information, thereby facilitating machine-actionable interoperability across systems. This includes the use of shared vocabularies, resolvable identifiers, and explicit typing of metadata constructs. As a result, STAMPLATE supports the alignment of observational data with external standards and ontologies, enabling more transparent and reusable data integration workflows.

Within the scope of quality control, STAMPLATE provides a flexible foundation for representing provenance information, processing steps, and quality-related metadata in a standardized way. It allows the extension of existing entities with structured properties that capture, for example, the methods applied during quality assessment, the context of execution, and links to derived data products. This makes it particularly suitable as a backbone for the schema proposed in this work, where both automated and manual QC processes need to be represented in a consistent, interoperable, and FAIR-compliant manner.

Further details on the STAMPLATE data model, its design principles, and implementation examples are available in the corresponding publication [5].

2.4 SaQC

The System for Automated Quality Control (SaQC) provides a flexible framework for the automated quality assessment of time series data, supporting both standalone operation and integration into heterogeneous data processing environments. It offers a comprehensive set of rule-based quality control procedures that can be readily applied to existing datasets. All core components of SaQC, including the procedure library and the representation of quality information, are designed to be extensible, enabling domain-specific adaptations while maintaining a consistent and reproducible execution framework.

A central design principle of SaQC is the strict separation of observations and quality metadata (flags). Rather than embedding quality indicators directly within data values, SaQC represents them as distinct but linked entities. This approach is consistent with the data modeling principles of the SensorThings API and the proposed representation of quality information in STA 2.0, and provides several important advantages:

- **Non-destructive processing:** Original observations remain unchanged, preserving data provenance and enabling reprocessing with updated QC procedures.
- **Multi-layered quality annotation:** Multiple independent QC procedures can contribute flags without overwriting or obscuring previous results.
- **Traceability and transparency:** Each flag can be explicitly associated with specific tests, thresholds, or standards, supporting reproducibility and auditing.

SaQC is designed to function both as an independent quality control application and as a modular quality control layer within existing data workflows, including data ingestion, post-processing, and dissemination. This integration-oriented design facilitates its use across diverse operational contexts. SaQC is currently integrated into several software systems, including:

- **Neptoon** (https://www.neptoon.org/en/latest/): SaQC provides core data cleaning functionality within a cosmic-ray neutron sensing processing pipeline.
- **time.IO** (https://doi.org/10.1016/j.softx.2025.102038): SaQC serves as the quality control and data processing layer within a fully automated sensor data ingestion pipeline, forming part of a comprehensive time series data infrastructure.

3 Use Cases

3.1 TERENO

The *TERrestrial ENvironmental Observatories* (TERENO) [6] is a long-term research infrastructure of the *Helmholtz Association* that investigates the impacts of global environmental change on terrestrial ecosystems in Germany. TERENO comprises several observatory regions, including the Eifel/Lower Rhine Valley, Harz/Central German Lowland, Bavarian Alps/Pre-Alps, and Northeastern German Lowland, each equipped with dense, multi-scale measurement networks. These observatories integrate in-situ sensors, lysimeters, eddy covariance systems, and remote sensing technologies to capture continuous time series data on hydrological, meteorological, and biogeochemical processes. The TERENO data management concept emphasizes standardized data acquisition, harmonization, and long-term accessibility, supported by centralized data services and adherence to FAIR principles. This enables interdisciplinary research and facilitates the reuse of environmental data across scientific domains.

QC in TERENO is an integral part of the data management lifecycle and is formalized through well-defined procedures described in the TERENO Data Management Plan [7]. As depicted in Fig. 3.1, QC processes are applied at multiple stages, beginning with automated checks during data ingestion, where raw sensor data are screened for technical errors, range violations, and temporal inconsistencies. These initial steps are complemented by standardized processing workflows that include calibration adjustments, plausibility checks, and the application of domain-specific QC algorithms. The use of reproducible, script-based QC methods—often implemented in dedicated frameworks—ensures consistency and scalability across the heterogeneous data streams collected within the observatories.

In addition to automated QC, TERENO places strong emphasis on expert-driven validation. Domain scientists review data products, assess flagged anomalies, and apply context-specific corrections where necessary. These manual interventions are essential for capturing complex environmental conditions that cannot be fully addressed by automated procedures alone. All QC steps, including applied methods, parameterizations, and resulting quality flags, are systematically documented and linked to the data via metadata. This ensures transparency, traceability, and reproducibility of quality assessments. By combining automated and manual QC within a structured data management framework, TERENO provides high-quality, reliable time series data that support both long-term environmental monitoring and advanced scientific analysis.

3.2 ACTRIS

The *Aerosol, Clouds and Trace Gases Research Infrastructure* (ACTRIS) is a pan-European distributed research infrastructure dedicated to the long-term observation and analysis of atmospheric composition. It integrates a network of highly instrumented observation stations, calibration facilities, and centralized data processing and archiving centres to provide harmonized and quality-controlled measurements of aerosols, clouds, and trace gases. ACTRIS is designed to support both fundamental atmospheric research and applications in climate science, air quality assessment, and environmental policy, with a strong emphasis on standardized measurement procedures and interoperable data services.

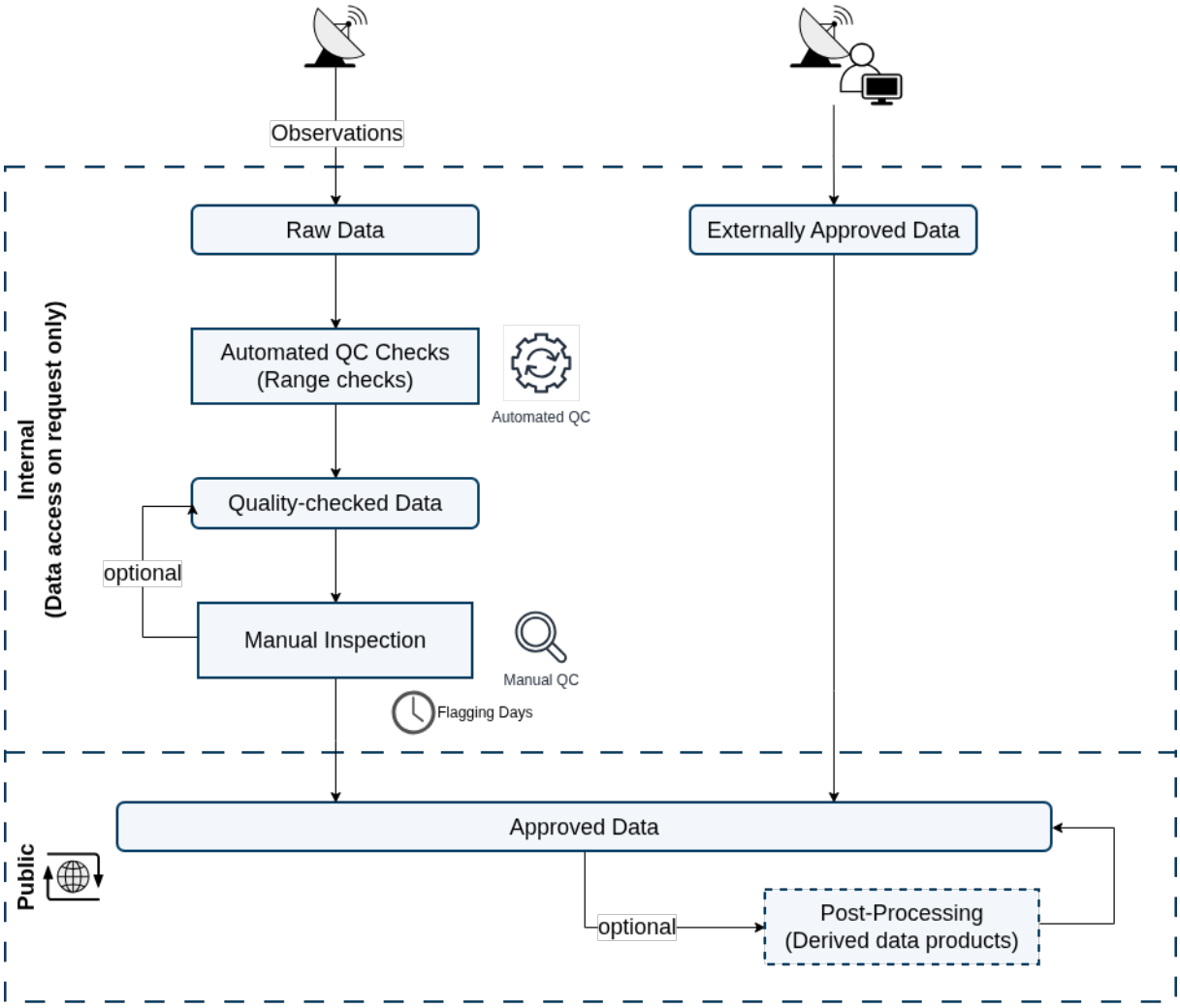

Figure 3.1: Quality control workflows in TERENO.

QC is a fundamental pillar of the ACTRIS data lifecycle and is explicitly formalized in the ACTRIS Data Management Plan [8]. The plan for the in-situ measurements defines a multi-stage QC strategy that spans from instrument-level calibration and automated pre-processing at observation stations to centralized quality assessment and validation at data processing centres. Automated QC routines are applied to detect instrument malfunctions, physically implausible values, and temporal inconsistencies using standardized algorithms and harmonized processing chains. These procedures ensure that data entering the ACTRIS data repository already meet a minimum level of consistency and reliability.

In addition to automated procedures, ACTRIS in-situ relies on expert-based manual quality assessment, particularly for complex atmospheric phenomena that require domain-specific interpretation. These manual QC steps are performed within dedicated data centre workflows and are tightly integrated with metadata and provenance tracking mechanisms. The ACTRIS Data Management Plan further emphasizes that all QC decisions, including processing steps, flags, and corrections, must be fully documented to ensure transparency and reproducibility. This structured approach ensures that quality information is not only applied consistently across the infrastructure but is also made interoperable and reusable in downstream scientific applications.

This is illustrated in Fig. 3.2.

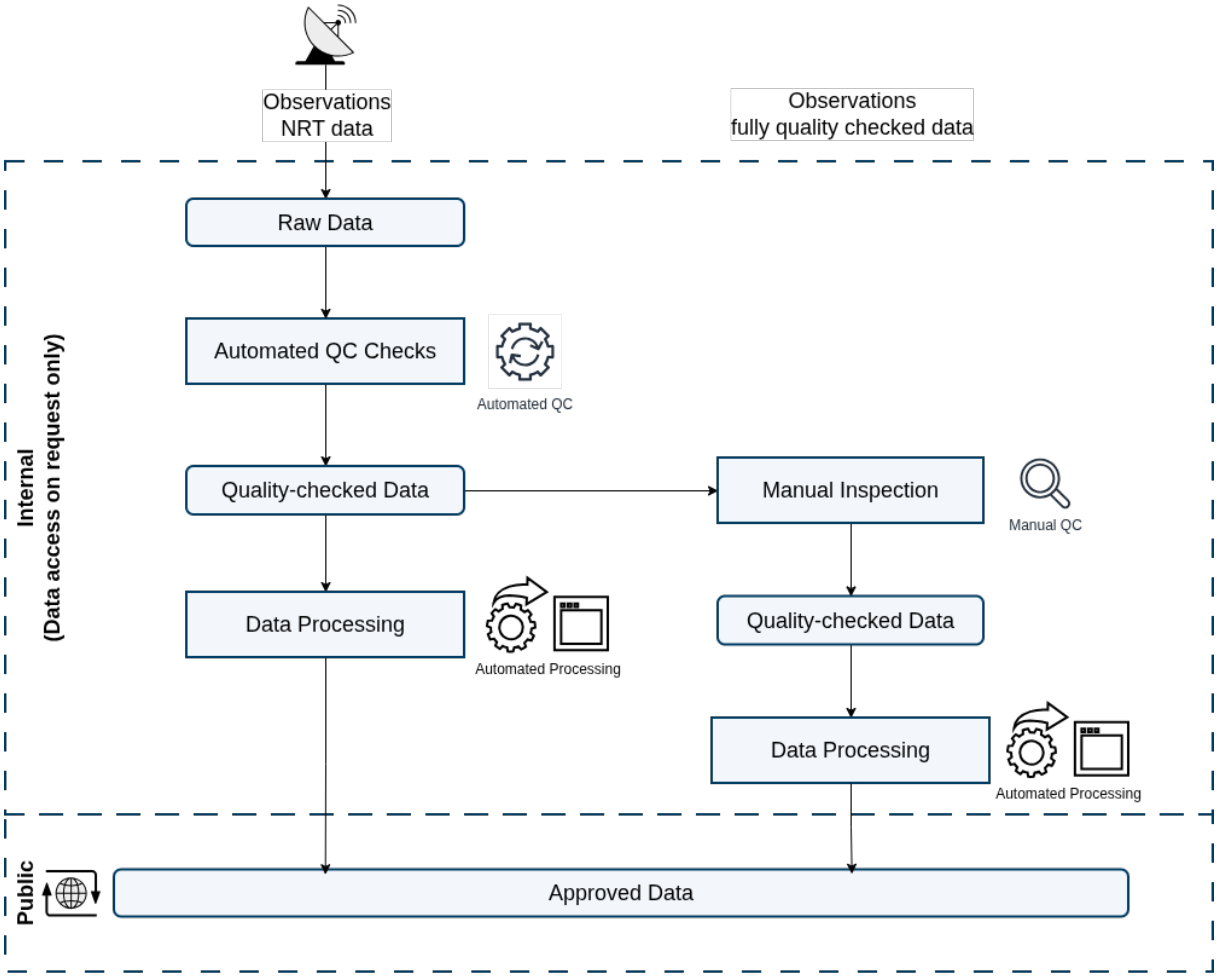

Figure 3.2: Quality control workflows in ACTRIS.

4 Proposed Schema

In the following sections, we describe the schema for QC of time series data as an intersection of both, the STA and the DQV terminology.

4.1 Initial Setting

We model time series using the STA data model notation. A `sta:Datastream` encapsulates a list of `sta:Observation`, each of which contains a timestamp and a value. The `sta:Observation` includes the timestamp fields `phenomenonTime` and `resultTime`. Observation time increases monotonically in STA. This is illustrated in Gig. 4.1.

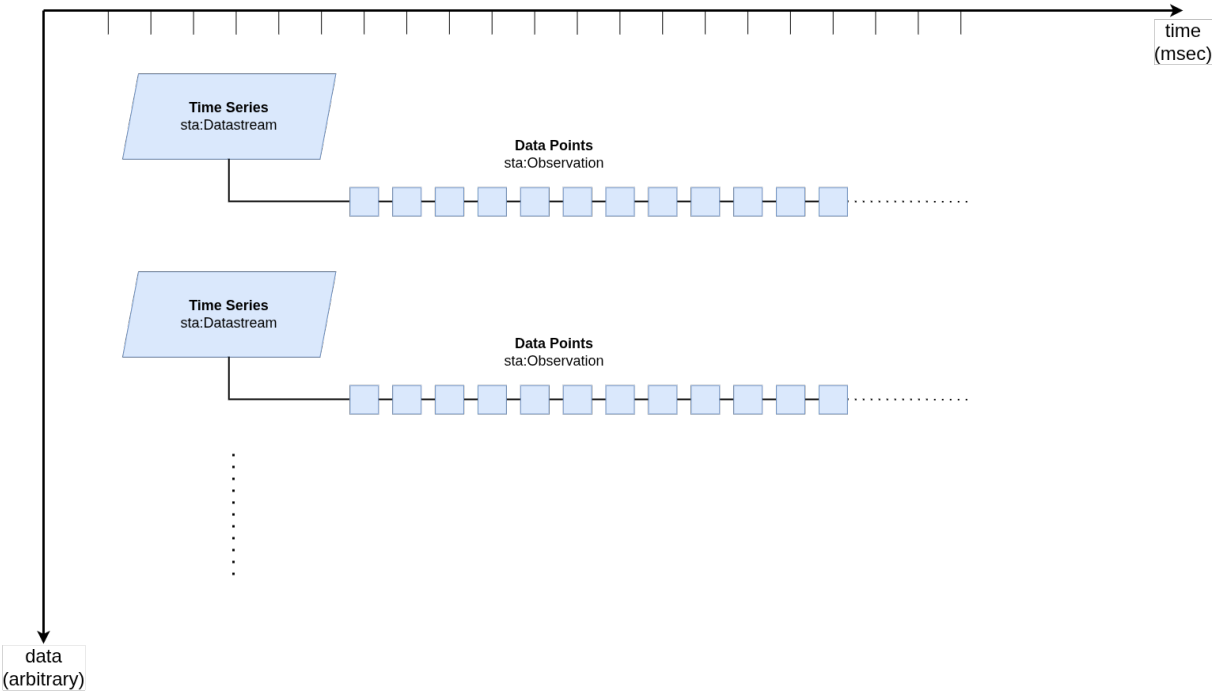

Figure 4.1: This diagram illustrates the time series data model employed in this paper, including the SensorThings API terminology indicated by the `sta:`-prefixed entity names.

4.2 General Approach

In our setting, a QC workflow takes a time series as input and measures the quality of each data point based on a certain *QC Procedure* representing the `dqv:Metric`, which creates *Quality Annotations* or *Flags* representing the `dqv:QualityMeasurements` with values in a certain *Quality Annotation Schema* representing the `dqv:Dimension`. The flags themselves are again modeled as `sta:Observation` and, thus, linked to a `sta:Datastream`– the *Quality Datastream*. This new `sta:Datastream` is related to the original `sta:Datastream`. In STA, `dqv:Dimension` is realized by `sta:ObservedProperty`, and `dqv:Metric` by the STA 2.0 `sta:ObservingProcedure`.

A visualization of this data model is given in Fig. 4.2.

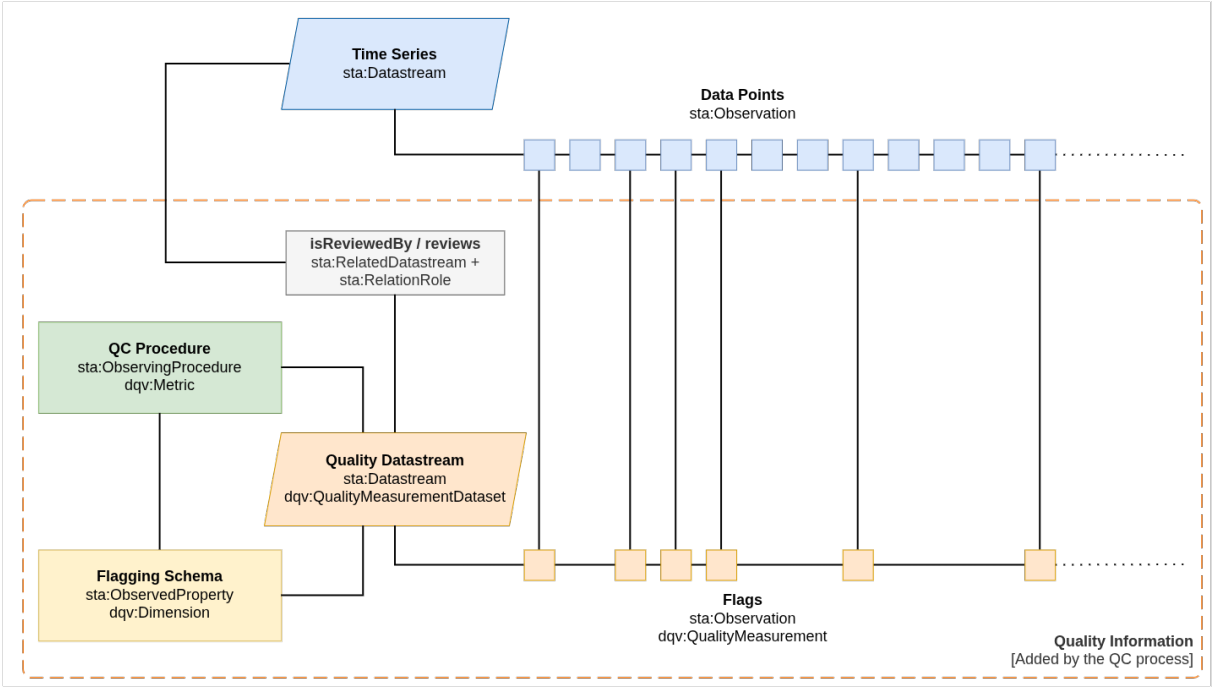

Figure 4.2: A visualization of the proposed data model for time series data quality. The diagram shows its entities starting from the *original* `sta:Datastream` and some exemplary `sta:Observations`. Each entity is tagged with its counterpart in STA by the `sta:` prefix and its semantic equivalent in the DQV by the `dqv:` prefix.

4.3 Schema Entities

4.3.1 Quality Datastream

The *Quality Datastream* is realized as an `sta:Datastream`. In the DQV semantics, it represents `dqv:QualityMeasurementDataset`. The STA field `observationType` is called `resultType` in version 2.0 and given by OGC's SWE Common[9] standard, which introduces a simple class diagram where all data types are extensions of the abstract class `AbstractDataComponent`. We simply assign the class `"Quality"` to `resultType`. The field `resultEncoding` is assigned the value `"JSONEncoding"`.

Links

In STA, `sta:Datastream` are linked to exactly one `sta:ObservedProperty` representing the measured quantity, exactly one `sta:Sensor` representing the measurement device producing data, and exactly one `sta:Thing` representing the platform the sensor is attached to. In the QC context, we use these natural links in the following way:

`sta:ObservingProcedures`: Links to the QC procedure (see Sec. 4.3.2)

`sta:ObservedProperty`: Links to the Flagging Schema (see Sec. 4.3.3)

`sta:SourceRelatedDatastream`: Link to the `sta:Datastream` representing the original time series through a `sta:RelatedDatastream` with the `sta:RelationRole` `isReviewedBy`

Table 4.1: Attribute specification of the *Quality Datastream* entity.

Attribute	Type	Description / Value Pattern
`id`	String	System-generated unique identifier
`name`	String	Unique representation of the quality datastream including the original datastream, the QC procedure and its parameters, the flagging schema and the time of the run
`description`	String	Human-readable description of the quality datastream
`resultType`	String	Constant value: `"Quality"`
`resultEncoding`	String	Constant value: `"JSONEncoding"`
`properties`	JSON object	• Add the STAMPLATE JSON-LD context • Add the type reference `dqv:QualityMeasurementDataset` • Add QC procedure parameters with values specific to this particular run of the QC procedure.

`sta:Observation`: Links to the *Quality Flags* (see Sec. 4.3.4)

`sta:Sensor`: Links to the `sta:Sensor` of the original time series

`sta:Thing`: Links to the `sta:Thing` of the original time series

4.3.2 Quality Control Procedure

We realize the *Quality Control Procedure* by `sta:ObservingProcedure`, which is an STA 2.0 concept. It realizes the `dqv:Metric` that performs the quality check on the `sta:Observation`.

Table 4.2: Attribute specification of the *Quality Control Procedure* entity.

Attribute	Type	Description / Value Pattern
`id`	String	System-generated unique identifier
`name`	String	Human-readable name identifying the QC procedure
`definition`	String	Link to the definition of this procedure, e.g., in a relevant vocabulary or a functions database
`description`	String	Human-readable description of the QC procedure
`resultType`	String	Constant value: `"Quality"`
`properties`	JSON object	• Add the STAMPLATE JSON-LD context • Add the type reference `dqv:Metric` • Add procedure parameters, optionally including default values

Links

In STA 2.0, at most one `sta:ObservingProcedure` may be linked to a `sta:Datastream`. An `sta:ObservingProcedure` has also one or more `sta:ObservedProperties`, which should link to the `sta:Datastream` of the `sta:ObservingProcedure`. In the QC context, we use these natural links in the following way:

`sta:ObservedProperties`: Links to the Flagging Schema (see Sec. 4.3.3), realizing the the relation `dqv:inDimension`

`sta:Datastream`: Links to the quality datastream (see Sec. 4.3.1)

4.3.3 Flagging Schema

The `sta:ObservedProperty` should define the schema which is used to provide the values in the fields of the "Quality Observation". In DQV terms, the `sta:ObservedProperty` defines the `dqv:Dimension` to be used by the `dqv:Metric` to express the `dqv:QualityMeasurement` result.

Table 4.3: Attribute specification of the *Flagging Schema* entity.

Attribute	Type	Description / Value Pattern
`id`	String	System-generated unique identifier
`name`	String	Human-readable name identifying the flagging schema
`definition`	String	Link to the definition of this schema, e.g., a relevant vocabulary or RDF schema
`properties`	JSON object	• Add the STAMPLATE JSON-LD context • Add the type reference `dqv:Dimension`

Links

`sta:ObservingProcedures`: Links to the QC procedure (see Sec. 4.3.2)

`sta:Datastream`: Links to the quality datastream (see Sec. 4.3.1)

4.3.4 Quality Flags

The `dqv:QualityMeasurements` are realized by `sta:Observation` with the same `phenomenonTime` as the original `sta:Observation`. The attribute `resultTime` contains the timestamp of when the QC procedure was performed. In DQV terms, this is when the `dqv:Metric` was used to produce the `dqv:QualityMeasurement`. The `result` field conforms to the `resultEncoding` based on SWE Common and contains the values defined by the flagging schema (see Sec. 4.3.3).

Table 4.4: Attribute specification of the *Quality Flags* entity.

Attribute	Type	Description / Value Pattern
`result`	String	Values defined by the flagging schema
`properties`	JSON object	• Add the STAMPLATE JSON-LD context • Add the type reference `dqv:QualityMeasurement`

Links

`sta:Datastream`: Links to the quality datastream (see Sec. 4.3.1)

4.4 Condensed Schema

In addition to the full, semantically explicit representation of quality information as separate entities, a condensed representation can be embedded directly within the original `sta:Observation`. This approach provides efficient, point-wise access to quality information while maintaining a clear semantic linkage to the complete quality model described in the previous sections. Conceptually, the embedded structure represents a projection of the full set of `dqv:QualityMeasurement` instances associated with an observation.

The condensed representation consists of a list of references to quality measurements, each corresponding to a quality flag generated by a specific quality control procedure. These entries may either fully inline selected attributes of the quality measurement or merely reference externally defined entities, for example those stored in a dedicated quality datastream. This flexibility allows implementations to balance between compactness and immediate accessibility of relevant information.

At the core of this representation are the key attributes defined by the DQV, which are implemented in the STAMPLATE JSON-LD context (see Tab. 4.5).

Table 4.5: Attribute specification of the Condensed Schema within the `properties` object of the `sta:Observation`.

Attribute	Type	Description / Value Pattern
`value`	String	value of the quality flag
`metric`	String	a reference to the QC procedure
`dimension`	String	a reference to the flagging schema

In addition, procedure-specific parameters may be included as part of the condensed structure. These parameters describe the configuration of a particular execution of the quality control procedure (e.g., threshold values or algorithm settings).

Even in cases where only references to external quality measurements are included, the condensed structure may optionally expose a subset of these attributes (e.g., value, metric, and dimension) for a selected or primary quality measurement. This enables efficient access to the most relevant quality information without requiring traversal of the full metadata graph. At the same time, all elements remain resolvable through their identifiers, ensuring consistency with the complete, standards-based representation of quality information across the system.

5 Implementation

5.1 SensorThings API / STAMPLATE

The proposed quality schema is implemented within the *STA* ecosystem through its integration into the *STAMPLATE* profile. In this context, the schema is not introduced as a standalone model, but rather embedded into the existing JSON-LD-based metadata extension mechanism provided by STAMPLATE. Concretely, the corresponding JSON-LD context definition is extended to include the semantic terms required to link the `sta:Observation` schema into the broader data model. This ensures that quality-related information is efficiently and consistently accessible and interpretable by machines within the SensorThings data structure.

In addition to the integration of the observation-level schema, core STA entities such as `sta:Datastream` and `sta:ObservedProperty` are extended with explicit links to quality-related semantics based on DQV. These links are incorporated into the STAMPLATE JSON-LD configuration and allow quality metadata to be associated in a standardized and interoperable manner across different levels of the data model. In particular, this enables a consistent representation of quality information that connects individual observations with higher-level semantic descriptions of the observed phenomena and their associated data streams.

By embedding the schema into STAMPLATE, the implementation leverages an existing, widely used extension mechanism for SensorThings while preserving full compatibility with standard STA services. At the same time, the explicit inclusion of DQV-based links ensures semantic alignment with international standards for data quality representation, thereby supporting interoperability, FAIR data principles, and cross-infrastructure integration.

5.1.1 Versioning and extension towards SensorThings API 2.0.

The current implementation of the STAMPLATE profile is primarily based on the SensorThings API version 1.1 specification. Within this version, the metadata extension mechanism and entity structure are well-established, but limited in their expressiveness with respect to explicit process representation and inter-entity relationships required for advanced quality control modelling. In contrast, the integration of the DQV links and the broader semantic modeling approach adopted in this work are aligned with the forthcoming SensorThings API 2.0 standard, which introduces enhanced support for process-oriented and relationship-centric data models.

To bridge this gap between the current operational standard and the upcoming specification, the STAMPLATE implementation of the proposed schema includes explicit serializations of SensorThings API 2.0 concepts, in particular the entities `sta:ObservingProcedure`, `sta:RelatedDatastream`, and `sta:RelationRole`. These constructs enable a more explicit representation of processing workflows, dependencies between datastreams, and the semantic roles of related entities within the quality control pipeline. By incorporating these elements into the existing STAMPLATE-based JSON-LD framework, the schema achieves backward compatibility with STA 1.1 while simultaneously preparing for seamless migration towards STA 2.0.

This dual compatibility approach ensures that the proposed quality schema can be deployed in current operational infrastructures without loss of functionality, while also aligning with the evolving SensorThings ecosystem. It thereby provides a future-proof foundation for representing complex quality control workflows in a semantically rich and standards-compliant manner.

5.2 NetCDF

In addition to the implementation within the SensorThings/STAMPLATE environment, a partial realization of the proposed quality schema has been explored for NetCDF-based data formats. This implementation follows the Helmholtz Metadata Collaboration's (HMC) recommendations for encoding quality information in NetCDF variables. In particular, it follows the HMC's guidelines on variable-specific quality flags [10]. These guidelines provide a lightweight and widely compatible mechanism to associate quality information directly with time series data.

In this approach, the concept of a quality datastream (see Sec. 4.3.1) is represented implicitly by introducing an additional variable for each observed variable, following a naming convention with the suffix `_qc`. This auxiliary variable stores quality flags corresponding to the primary data variable and thereby enables a direct, point-wise association between observations and their quality annotations. The flagging schema, including flag values and their semantic meaning, is encoded inline using NetCDF variable attributes as defined in the HMC recommendations. This allows efficient access and straightforward integration into existing NetCDF-based workflows while maintaining a certain level of semantic clarity.

However, this representation only captures a subset of the full quality schema proposed in this work. In particular, the modeling of quality control processes, such as the description of the applied QC procedures, parameterizations, and execution context, is not yet fully addressed within the NetCDF encoding. While the use of auxiliary variables provides a practical and interoperable solution for representing quality flags, the linkage to higher-level metadata describing the provenance and methodology of quality assessment remains an open challenge. Bridging this gap would require either extended metadata conventions within NetCDF or explicit connections to external metadata models, such as those provided by STAMPLATE and the DQV.

5.3 RO-Crates

To facilitate the exchange, publication, and reproducible reuse of quality-annotated time series data, the proposed schema can be embedded within the *Research Object Crate* (RO-Crate) framework [11]. RO-Crate provides a lightweight, JSON-LD-based packaging approach for research data and metadata, enabling the aggregation of datasets, contextual information, and semantic annotations in a single, portable structure. It is designed to align with FAIR principles by ensuring that both data and their descriptive metadata are machine-actionable and interoperable.

In the context of this work, an RO-Crate package serves as a container that bundles one or more time series datasets together with their associated quality information and metadata descriptions. At a high level, such a package consists of a root directory containing a mandatory `ro-crate-metadata.json` file and a set of referenced data resources (e.g., files representing datastreams). The metadata file encodes a graph of entities and relationships using JSON-LD, allowing the explicit representation of domain concepts such as datastreams, quality measurements, quality control procedures, and flagging schemas.

To construct an RO-Crate for the proposed schema, the original time series data and the corresponding quality annotations are included as separate but related data entities within the crate. These entities are

described as datasets or files and are semantically linked through the metadata graph to reflect their relationships as defined in the underlying model, for example by expressing that a quality dataset is derived from or associated with an original datastream. Additional entities represent the quality control procedures and the flagging schema, which are connected via typed relationships consistent with the semantics of the DQV and the SensorThings API. This approach enables a clear separation between data and metadata while preserving explicit, machine-readable links between all relevant components.

The RO-Crate metadata model further allows the inclusion of contextual information such as provenance, authorship, and usage conditions, thereby supporting transparent documentation of the quality control workflow. By leveraging established vocabularies and identifiers within the JSON-LD context, the resulting package can be integrated into broader data ecosystems and linked to external resources. As demonstrated in recent work on RO-Crate-based research data packaging, this approach provides a flexible and standards-aligned mechanism to represent complex data products and their associated quality information in a reproducible and interoperable manner.

6 Conclusion and Outlook

6.1 Conclusion

The authors demonstrate that the proposed schema offers practical solutions for adding information on data quality to both automated and manual workflows. The implementations they present show how seamlessly this general schema can be used in applications, particularly the ones considered here.

Direct implications are that both use cases can be implemented in a community solution that uses the proposed schema.

6.2 Outlook

The next step is merging the proposed schema into the STAMPLATE schema. Integration of the schema into the HMG NetCDF Guidelines [12] is planned. Additionally, an RO-Crate template that implements the proposed QC schema will be provided as an extension to this report.

Bibliography

[1] World Wide Web Consortium (W3C), "Data on the Web Best Practices," W3C, W3C Recommendation, 2017, Accessed: 2026-04-12. [Online]. Available: https://www.w3.org/TR/dwbp/

[2] Open Geospatial Consortium (OGC), "OGC SensorThings API Part 1: Sensing Version 1.1," OGC, OGC Implementation Standard OGC 18-088, 2021, Accessed: 2026-04-21. [Online]. Available: https://www.ogc.org/standards/swecommon/

[3] Open Geospatial Consortium (OGC), "Geographic information — Observations and measurements," OGC, OGC Abstract Specification OGC 20-082r4, 2013, Accessed: 2026-04-21. [Online]. Available: https://www.ogc.org/standards/om/

[4] World Wide Web Consortium (W3C), "JSON-LD 1.1: A JSON-based Serialization for Linked Data," W3C, W3C Recommendation, 2020, Accessed: 2026-04-28. [Online]. Available: https://www.w3.org/TR/json-ld11/

[5] N. Brinckmann, C. Faber, M. Hanisch, C. Lorenz, U. Loup, and D. Schäfer, *STAMPLATE: SensorThings API Metadata Profile for Linked dAta in environmenTal rEsearch*, Accessed: 2026-04-28, 2024. DOI: 10.5281/zenodo.17241283 [Online]. Available: https://doi.org/10.5281/zenodo.17241283

[6] S. Zacharias et al., "Fifteen years of integrated terrestrial environmental observatories (tereno) in germany: Functions, services, and lessons learned," *Earth's Future*, vol. 12, no. 6, e2024EF004510, 2024, e2024EF004510 2024EF004510. DOI: https://doi.org/10.1029/2024EF004510 eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2024EF004510. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2024EF004510

[7] TERENO Consortium, "TERENO Data Management Plan," Helmholtz Association, Tech. Rep., 2021, Version 1.0, accessed: 2026-04-28. [Online]. Available: https://www.tereno.net/joomla4/images/Resources/Downloads/Tereno-Data_management-plan-v10.pdf

[8] ACTRIS Data Centre Consortium, *ACTRIS Data Management Plan*, https://github.com/actris/data-management-plan/blob/master/DMP/ACTRIS-DMP.md, Accessed: 2026-04-28, 2024.

[9] Open Geospatial Consortium (OGC), "OGC SWE Common Data Model Encoding Standard," OGC, OGC Implementation Standard OGC 20-010, 2021, Accessed: 2026-04-12. [Online]. Available: https://www.ogc.org/standards/swecommon/

[10] Helmholtz Metadata Collaboration (HMC), *HMG NetCDF Guidelines: Variable-specific Quality Flags*, https://hmg-netcdf.readthedocs.io/en/latest/variables/variable-quality-flags.html, Accessed: 2026-04-28, 2024.

[11] RO-Crate Community, *RO-Crate: A Lightweight Approach to Research Object Packaging*, Accessed: 2026-04-28, 2023. DOI: 10.5281/zenodo.13751027 [Online]. Available: https://doi.org/10.5281/zenodo.13751027

[12] Helmholtz Metadata Collaboration (HMC), *HMG NetCDF Guidelines*, Accessed: 2026-04-28, 2024. DOI: 10.5281/zenodo.19691977 [Online]. Available: https://doi.org/10.5281/zenodo.19691977