Open Government Data as Multi-dimensional 5 Star Data: cube.link

Michael Luggen[1,2,4], Benedikt Hitz[2], Julien Audiffren[4], Djellel Difallah[3], Jean-Luc Cochard, and Philippe Cudré-Mauroux[4]

[1] Swiss Federal Chancellery
`firstname.lastname@bk.admin.ch`
[2] Berne University of Applied Sciences
`firstname.lastname@bfh.ch`
[3] NYU Abu Dhabi
`firstname@nyu.edu`
[4] University of Fribourg
`firstname.lastname@unifr.ch`

Abstract. Many governments made commitments to publish data collected and created by taxpayers as Open Government Data (OGD). Yet, a common challenge is that data producers are not always end-users, leading to inefficiencies, gaps, and inconsistencies in how data is used and interpreted. OGD can originate from various sources and can sometimes be noisy. For instance, smart cities can produce fine-grained and precise data via sensor networks such as power grids or transportation networks, whereas data automatically extracted from government documents can be highly noisy. A common pattern, however, is that this data often describes spatiotemporal phenomena with multidimensional facets pertaining to real-world entities. Having a common ontology for such observations can help standardize their downstream usage. This paper describes the effort made by the Swiss Government to create an ecosystem that allows to publish open government data in a pragmatic and efficient manner while maintaining the goal of publishing high-quality data that are integrated and interoperable. The paper introduces the domain-independent ontology *Cube Schema* (https://cube.link), which is supported by open-source tools to publish, integrate, and validate diverse data sources, as well as multiple end-user tools for providing and visualizing data. With underlying statistics, we show the usage of these tools by both data providers and data consumers with several deployment use-cases.

Keywords: Open Government Data · Linked Open Data · Open Source · Data Cube · Publishing · Visualization.

1 Introduction

While the commitment to publish open data has been made by many governments [1], publishing the data in a cost-effective and efficient manner remains a common challenge [2]. Furthermore, one often needs to strike a balance between fulfilling the basic task of publishing the data (as 1–3 star data[5]) and enhancing the data to be self-descriptive and well-integrated across various governmental domains.

Ideally, each branch of the government could spend resources to identify, create, and then adapt a domain-specific ontology to their own needs [3]. At the same time, a substantial share of available governmental data is structured in a form that resembles or is close to a structure known as tidy data [4], i.e., data that are easy to manipulate, model, and visualize, and have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table[6].

In this paper, we introduce the *Cube Schema*[7] (`cube.link`), a generic ontology that allows publishing tidy data in a structure similar to OLAP cubes [5]. The ontology not only facilitates structuring multidimensional data but also allows describing the data through cube metadata, providing context (metadata on units, scale naming, and aggregations) for the various dimensions (columns in tidy data), and supplying context related to the concepts used in the observations (rows in tidy data). A crucial feature of `cube.link` is its ability to reuse and share concepts across multiple cubes. Additionally, `cube.link` provides metadata designated as hints for downstream software, enhancing the User Experience (UX) when directly consuming the data.

Building on this foundation, the newly proposed `cube.link` ontology has been rigorously tested for performance and usability. The core ontology supports a wide range of use-cases, from straightforward data management to complex visualization tasks. Its adaptability to various scenarios is achieved through extensions to the core ontology. Furthermore, we describe multiple open-source tools developed to support the ecosystem around this ontology, tools actively in use and development at the Swiss Confederation and adopted by other organizations, as described below. Additionally, we provide data transformation pipelines (code-based), a GUI named *Cube-Creator* for transforming inputs from tidy (CSV) data, a technical viewing and verification tool (*Cube-Viewer*), as well as an end-user visualization tool (*visualize.admin.ch*) that enables non-technical users, guided by metadata, to create complex visualizations.

In the following sections, we start by describing some related work in Section 2, highlighting how `cube.link` differs from previous initiatives. Section 3 introduces the conceptual foundations, explaining the philosophy guiding our ontology development. Section 4 details the structure of `cube.link` and discusses some of its extensions. We outline applications and tools supporting the cube ecosystem along with usage statistics in Section 5. Finally, Section 6 describes the future roadmap for our ontology.

[5] https://www.w3.org/2011/gld/wiki/5_Star_Linked_Data
[6] https://data.europa.eu/apps/data-visualisation-guide/intro-to-tidy-data
[7] https://cube.link

2 Related Work

The RDF Data Cube Vocabulary (QB) [6, 7] is one of the earliest vocabularies developed in the RDF domain for representing data cubes. It was heavily influenced by the Statistical Data and Metadata eXchange standard (SDMX)[8]. While QB has the advantage of being very close to SDMX, it was also mainly designed to publish statistical data [8, 9]. This is not (anymore) aligned with the wide breadth of data domains that are today processed and published in governments. Before our work on `cube.link` started, we actively worked with QB and identified a number of issues that were at the root of our decision to develop a new ontology.

One primary challenge arises from the vocabulary's alignment with SDMX, which, while beneficial for statistical use-cases, imposes design constraints that complicate broader applicability. The model exhibits a non-uniform linking strategy, combining both forward and backward references within its structure. This inconsistency disrupts the "follow-your-nose" principle commonly embraced in linked data paradigms, thus reducing the utility of RDF-native traversal and discovery mechanisms. Another fundamental issue lies in the open-ended nature of the specification. While designed to be flexible, this openness has resulted in divergent interpretations among implementers. There is frequently more than one way to model a given dataset leveraging the vocabulary, leading to a lack of interoperability across RDF Data Cube applications [10]. As a consequence, software libraries intended to consume generic RDF Data Cubes must account for a wide range of modeling styles, which is both resource-intensive and error-prone. Additionally, the reuse of dimension definitions, which is a key aspect for enabling data comparability and harmonization across sources, is uncommon within the RDF Data Cube vocabulary [11]. The absence of standardized dimension reuse practices poses interoperability challenges, making cross-provider data alignment more difficult.

The increased availability of multidimensional data on the Semantic Web highlighted QB's limited support for advanced analytical capabilities, specifically Online Analytical Processing (OLAP). To address these limitations, QB4OLAP was proposed as a semantic extension of QB, explicitly supporting structured dimension hierarchies, levels, and aggregate functions for measures [5]. QB4OLAP enables traditional OLAP operations (roll-up, slice, dice) directly via standard SPARQL queries.

Other efforts, such as *schema.org*, introduced statistical observation support (`schema:Observation`) for representing data cubes. In the context of Open Government Data, this construct suffers from similar downsides as QB, since it was primarily designed to represent statistical values and because the schema.org ontology is notoriously open in terms of how ranges and domains can be applied to specific use-cases.

[8] https://sdmx.org/

3 Conceptual Design Choices

Based on the observations made above in the Related Work section (2), we took a series of conceptual design choices to develop `cube.link`. In the following, we highlight the most important ones, starting with the possibility of adding metadata and descriptions within the data of the cube, continuing with compatibility with external linked datasets (e.g., for controlled vocabularies), and concluding with the assurance that `cube.link` can be further extended while ensuring it remains queryable in real-life scenarios.

Enabling Data-Providers to Provide Context on their Data A fair amount of data in organizations is kept in tables with multiple columns, typically in the form of Excel or CSV files. This data often includes a time dimension, along with additional dimensions for classification and columns representing observed values or counts. Although these tables hold data in a structured format, they often lack the necessary information to interpret the columns correctly. Metadata required to create meaningful charts and visualizations is missing. In many cases, this context is only available through non-machine-readable documents, such as PDFs or descriptions on separate websites. With `cube.link`, data providers and domain specialists can augment and annotate data directly within the published dataset. This makes the data more transparent and usable. Fully annotated cubes can also be used immediately to generate interactive visualizations, improving accessibility and understandability.

Reuse of Controlled Vocabularies and Ontologies The `cube.link` ontology was designed to enable heavy reuse of available Controlled Vocabularies inside the cubes. For example, if a column holds the municipalities that a series of data points belongs to, it is possible and advised to use the unique resources of such municipalities inside the knowledge graph of the data provider. In concrete use-cases of the Swiss Government, the authority defining the identifiers of such communities is the *Federal Statistics Office*. The geographical boundaries are provided by the *Federal Office of Topography*, called *Swisstopo*, while the official translations are provided by the *Swiss Federal Chancellery*. Each organization inside the Swiss Government provide parts of the data which together define a full-featured representation of a municipality. (See Figure 1 for an example on cantonal level, extended with data from the *Federal Office for Energy*.)

This allows to profit from additional, well-defined and maintained information, e.g., from higher-level administrative units such as cantons or the federal state, allowing one to filter and sort the data by these units without any further effort by the data provider. Another dimension of integrated information is use-case specific geographical boundaries provided by Swisstopo, which immediately allow to create spatial maps such as choropleth maps. Finally and most importantly, the use of shared concepts allows to compare datasets from different domains implicitly and without coordination from the data providers.

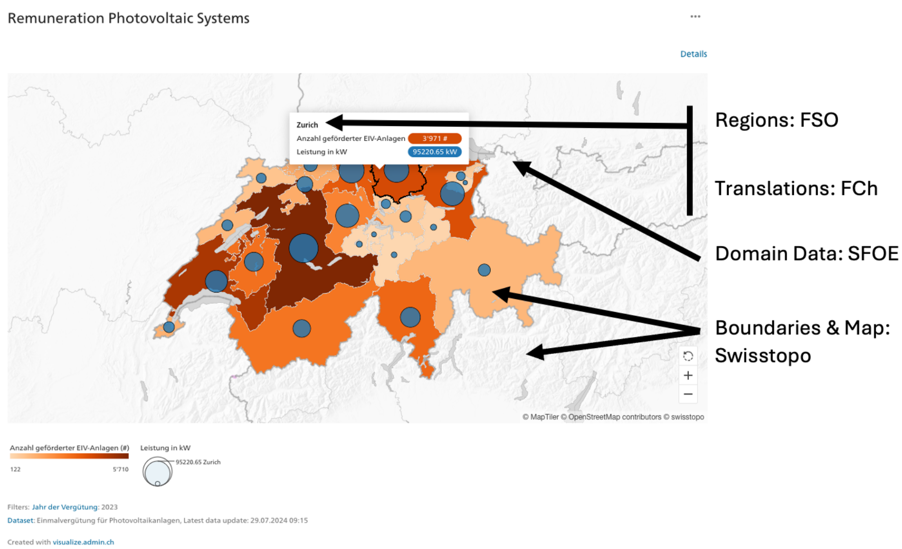

Fig. 1: Example visualization based on a domain-specific cube about remunerations of photovoltaic systems in Switzerland. Various data sources are integrated from several data providers including: FSO (the *Federal Statistics Office*), FCh (the *Federal Chancellery*), SFOE (the *Federal Office of Energy*), and Swisstopo (the *Federal Office of Topography*). The depicted chart is accessible live and is based on the cube that can be dereferenced as follows: https://energy.ld.admin.ch/sfoe/bfe_ogd84_einmalverguetung_fuer_photovoltaikanlagen/10.

Supporting Once-Only Data Publication While not mandatory, we support a strict once-only principle of data publication as demanded by the Tallinn Declaration[9] on eGovernment. Different use-cases often need different mixes and unions of data points, which can be provided in a once-only fashion. This allows to have a single point of correction in case of errors, as well as a single point for general annotations. Therefore, `cube.link` is nudging data providers to publish multiple smaller cubes that are providing unique data points. Through the connections to shared resources, it is naturally possible to query multiple cubes in union for a multitude of different use-cases.

Covering Statistical Data and Scientific Observations The ontologies covered in the Related Work section are targeting statistical data, which often boil down to counts and percentages that can be aggregated along various dimensions. Aggregation is also important for scientific observations, such as water quality data or noise measurement, which also raises the need for additional metadata like units or scales of measurement for proper visualization. Our goal

[9] https://ec.europa.eu/newsroom/dae/redirection/document/47559

was to provide one ontology for both use-cases and even allow mixed settings, where statistical data can be combined with scientific observations.

Extensibility Based on a concise and powerful core ontology, our `cube.link` was built for extensibility. For instance, we already provide an extension to describe the extent of a dimension based on SHACL. Furthermore, a User eXperience Extension focuses on the description of metadata to automatically consume and visualize the cubes. Finally, we also provide extensions for advanced topics such as versioning of cubes, the relation between dimensions (e.g., standard deviation annotations), as well as concepts for declaring hierarchies within a dimension as a basis for expressing possible aggregations. Further planned extensions are mentioned in the Future Development section below.

Optimized for Fast Query Access Finally, the best data structure defeats its purpose if it cannot be accessed and queried with acceptable performance. `cube.link` imposes strict limitations on handling missing values and defining constraints on dimensions, which ensures fast querying of large datasets—up to multiple gigabytes.

4 The `cube.link` Ontology

4.1 Observations at the Core

The core of the `cube.link` ontology[10] consists of a set of observations, each of which links all the dimensions (or variables) associated with a particular observation. In other words, a `cube:Observation` represents a row in a tidy data format. The collection of observations is connected to the `cube:Cube` through the `cube:ObservationSet` container. This setup allows to navigate from the cube forward to each individual observation. (See also Figure 2.)

For each dimension, one can point (i.e., integrate) to already published concepts (see also Section 4.3) and add numeric data points. The predicates used to connect an observation to a value can reuse pre-existing ontologies or can be created ad hoc. With this simple core, there is a deterministic way to transform all data that is provided in the form of tidy data into `cube.link`, while remaining flexible on the reuse of already published concepts. The use of `cube:observedBy` is mandatory to connect each individual observation to an observer. This ensures the possibility to combine, query and compare identically structured cubes from different publishers.

4.2 The Closed World Cube

Linked Data and, more generally, RDF is based on the Open World assumption, which provides flexibility for extending datasets and their models. But tion, which provides flexibility for extending datasets and their models. But

[10] https://cube.link/#core-schema

data without any specific constraints is notoriously difficult to be used for visualizations and front-end applications [12]. With `cube.link`, our approach was to keep the attributes of the core ontology as described above open, and even allow the reuse of already published data when structured sufficiently close to an `cube:Observation`.

To enable end-user applications to understand what can be expected from a cube in regard to its structure, the different dimensions and metadata of the cubes can leverage the `cube:Constraint` concept[11]. Constraints are added in a separate structure to the cube and are described as SHACL[12] shapes.

With this approach, it is possible to express and constrain the data that is expected in a cube. At the same time, the same description of the data as described in the SHACL shape can be used to validate the concrete data inside the cube (see section 5.4). For each dimension (or column in tidy data), the necessary attribute to be provided for a valid constraint is the predicate (with `shacl:path`) connecting an observation to its value. This allows to use a subset of the information attached to an observation.

Finally, it is optionally possible to attach with `shacl:in` the list of all distinct values to be expected in a dimension. While this can always be introspected in the data, this addition allows end-user tools to quickly query the possible values to populate user interface components.

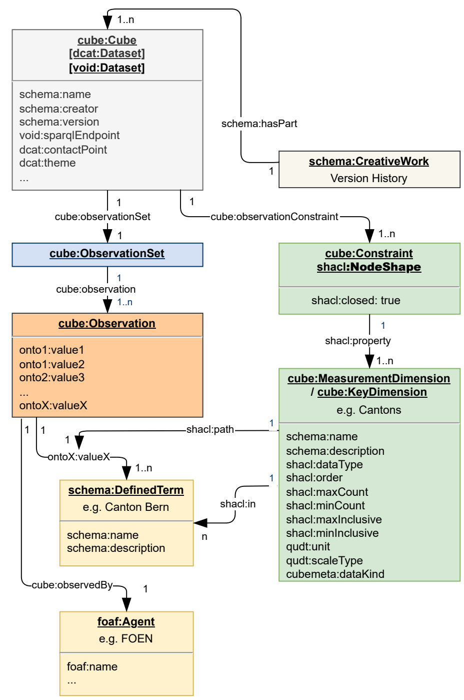

Fig. 2: An overview of the core classes of `cube.link`. Orange represents the observations (a row) itself. Yellow are the connected concepts (CV) to a row. Green represents the metadata on dimensions (a column).

4.3 Shared Dimensions a.k.a. Controlled Vocabularies

We identified two main reasons to introduce *Shared Dimensions* (a term adapted from controlled vocabularies to be more intuitive for data providers).

[11] https://cube.link/#constraints
[12] https://www.w3.org/TR/shacl/

First, Shared Dimensions promote the reuse of predefined, consistent dimension values across datasets. When multiple data cubes use the same Shared Dimensions, end-user tools can recognize the similarities and enable combined visualizations across datasets. (See for an example the screenshot in Figure 3.)

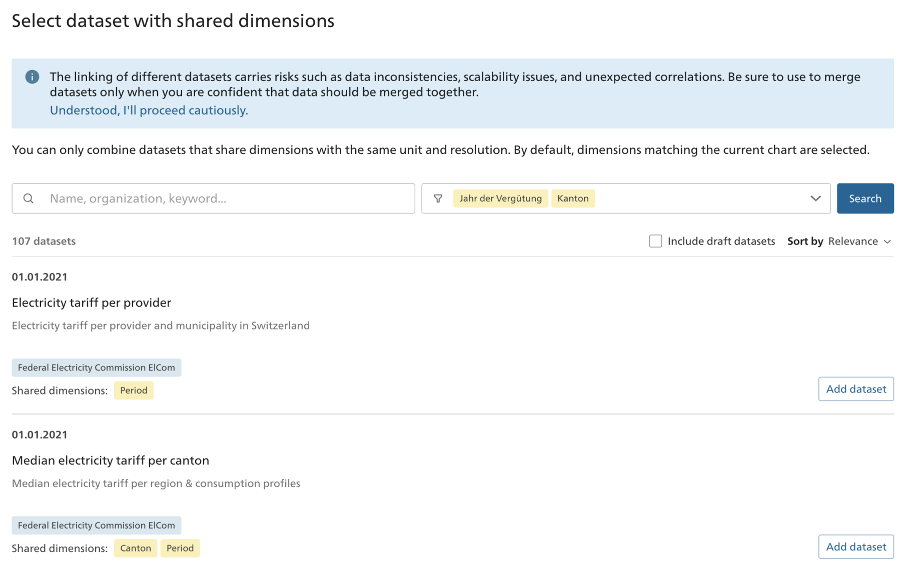

Fig. 3: In visualize.admin.ch, it is possible to combine other datasets on the same chart, based on the common usage of Shared Dimensions (yellow highlight).

Second, in the multilingual context of Switzerland, the possibility to properly translate concepts in multiple languages for later use in a visualization provides ample motivation to use concepts instead of strings in the data itself. Instead of having a string "Summer", the cube might hence have concepts that have the labels "Summer", "Été", "Estate" and "Sommer" attached. In addition, a description, other attributes or even links to further concepts can be attached to such a concept. Finally, concepts can represent spatial regions, which in turn allows to create arbitrary choropleth maps (as shown in Fig. 1) for diverse use-cases.

4.4 User Experience Extension

Often, the information necessary to interpret a published dataset is provided in a secondary source like a website or a PDF document. Additional information on the content, or detailed descriptions are often necessary to consume the data and create sensible charts and visualizations. In that case, information that improves the user experience (UX) of the data consumer provided as machine-readable semantically defined data can be added through the *User Experience*

Extension[13]. First and foremost, this extension helps distinguishing between two types of dimension (columns).

The `cube:KeyDimensions` are part of the combined key uniquely identifying one observation (these dimensions are also often used to be part of the URI template identifying one observation – for example Year and Canton in Figure 4).

The `cube:MeasurementDimensions` are the columns providing the observed values that are attached to one observation. This distinction allows to automatically create a graphical user interface to choose the dimensions shown and filtered for a given end-user application like *visualize.admin.ch* (see Section 5.3). Finally, the User Experience Extension allows to provide multilingual naming and descriptions of the dimensions, the unit (count, percentage, KG, CHF/month and other) and scale (nominal, ordinal, interval, rational) of a dimension based on QUDT[14], the *data kind* (i.e. type of data), and the order between the dimensions. The *data kind* is a concept we introduced in `cube.link` that provides higher level hints on the data types for spatial dimensions (e.g. if there are coordinates or shapes provided) and temporal dimensions (e.g. the time resolution to be shown).

Further extensions of `cube.link` define advanced concepts and features, such as handling multiple versions of cubes, publishing the state of a cube, defining relations between measurement dimensions, and the expression of hierarchies inside a dimension[15].

5 In-Use: Producers, Tools and Consumers in the `cube.link` ecosystem

This section provides a detailed overview of the tools that form the ecosystem surrounding `cube.link`, along with usage statistics on published cubes and their views.

5.1 Adoption from Data publishers and Publication Tools

Many organizations, mostly offices of the Swiss Confederation, independently publish cubes covering many different domains. Table 1 shows the number of cubes per organization as of May 2025.

5.2 Publication Tools

While developing `cube.link`, we could identify two major target groups of data providers: 1) Data providers who manage data using files – for example spreadsheets close to a tidy data format. For this group, the specialized GUI application *Cube-Creator* was built; 2) Data providers who manage their data in a database.

[13] https://cube.link/#ux
[14] https://qudt.org/
[15] see https://cube.link/#advanced

Publishing Organisation	**Draft**	**Published**
(Not attributed)	49	95
Federal Electricity Commission ElCom	0	3
Federal Food Safety and Veterinary Office FSVO	39	143
Federal Office for Agriculture FOAG	0	42
Federal Office for the Environment FOEN	19	188
Federal Office of Civil Aviation FOCA	1	4
Federal Office of Communications OFCOM	0	18
Federal Office of Public Health FOPH	3	0
Federal Statistical Office FSO	1	2
Financial Statistic FFA	1	1
Swiss Federal Archives SFA	0	41
Swiss Federal Office of Energy SFOE	34	44
Statistics Office of the City of Zurich	0	440

Table 1: The count of published cubes by creator and status (Draft vs Published) as of May 2025.

Two further sub-groups can be identified in this context: a) Organizations that have the necessary know-how to directly transform their data into a `cube.link` format and b) Organizations that outsource this task to specialized companies.

While the first target group might probably reuse an existing pipeline for its needs, the second one often has the tendency to create its own technology stack.

The Graphical User Interface Application *Cube-Creator* The goal of the *Cube-Creator*[16] is to support the transformation of tabular CSV data into RDF data that fulfill the requirements of the `cube.link` ontology. Experience shows that people tasked with such a transformation seldom have the necessary technical background in data science. Therefore, the *Cube-Creator* assists this transformation in a step-by-step manner.

An important aspect of the *Cube-Creator* is to help re-using existing vocabulary from well-known sources (e.g., schema.org, SKOS, etc.) and enriching the data with shared dimensions and hierarchies as well as allowing to localize the data in different languages.

A shared dimension in `cube.link` is a column in the tabular input data that also exists in other datasets and can be reused with all existing metadata. A shared dimension could for example be all the different spatial divisions of Switzerland - like municipalities, districts and cantons. Another example would be arbitrary classification schemes like ICD-10 codes from medicine or NACE/NOGA codes from economics. The *Cube-Creator* helps with mapping input data onto existing shared dimensions, or with the creation of new shared dimensions.

[16] https://github.com/zazuko/cube-creator

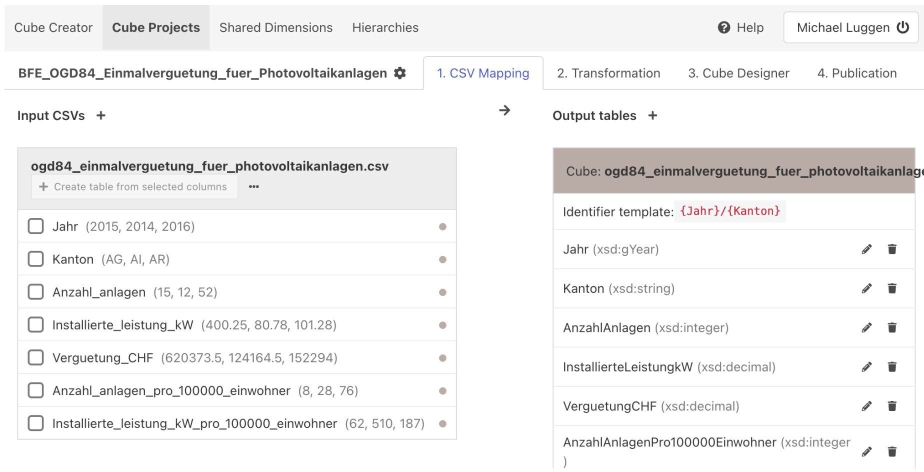

Fig. 4: The first step after creating a Cube Project encompasses the mapping from the input CSV to a properly defined cube. The *Cube-Creator* also proposes an identifier template based on the input. The data type of the different dimensions (columns) is specified.

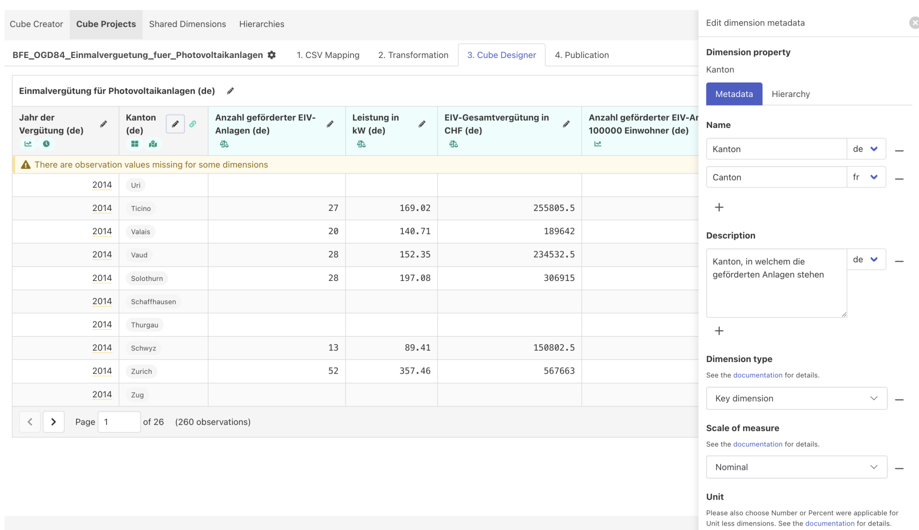

Fig. 5: After an initial transformation (step 2 – not shown), the imported data is presented in the Cube Designer. In step 3, the data providers annotate the cube with the necessary metadata as described above. The visible side-panel in this example depicts the Dimension Properties of the Region of the Cantons. The final step (step 4 – not shown) publishes the data online.

Publishing Pipelines and Frameworks To accommodate the data products of the second target group, several frameworks have been designed to help publish data held in databases.

barnard59 A pipeline component[17] that is part of *barnard59* allows one to add metadata to a dataset and publish it in a `cube.link` compliant manner. This framework written in JavaScript, usually run on Node.js, is optimized to process the data as streams. Therefore, it is also possible to transform large datasets, up to multiple gigabytes, with a low memory footprint. This component is also used in the backend of *Cube-Creator*.

pylindas Based on the Python Pandas Dataframe, the pylindas[18] framework built by the *Federal Office for the Environment* facilitate the process of merging the data prepared in a `DataFrame` with the necessary metadata to create a `cube.link` compliant cube.

5.3 End-User Applications and Dashboards

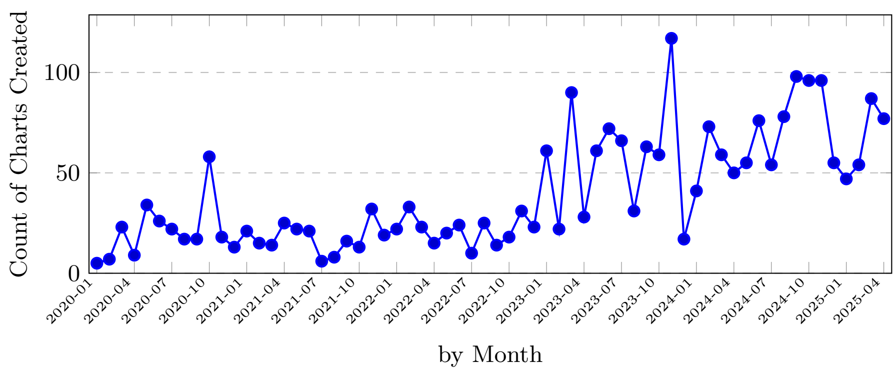

Fig. 6: Monthly number of charts created with visualize.admin.ch from 2020 until April 2025.

visualize.admin.ch One main advantage of standardizing data is the ability to build data-consuming tools that can rely on having the data in a specific form. One such tool is the visualization tool (https://visualize.admin.ch) that allows to visualize `cube.link`-compliant data through an easy-to-use graphical user interface.

As `cube.link` compliant data is enriched with specific metadata, visualization tools can be restricted to allow only sensible options when visualizing the data. For example, targeting only data that is enriched with the necessary spatial information (mostly by connecting to the corresponding shared dimensions), map-based visualizations can be offered as an option. Once a chart

[17] https://github.com/zazuko/barnard59/tree/master/packages/cube
[18] https://github.com/Kronmar-Bafu/lindas-pylindas

(or a table) is configured in *visualize.admin.ch*, the chart can be embedded in social media, documents or websites. In Figure 6, we report the number of charts created by end-users over time. Since August 2024 also, the views on charts are tracked as reported in Figure 7. Up-to-date statistics are available at https://visualize.admin.ch/statistics.

Visualize is actively developed by the Swiss Government and is available as an open source package[19].

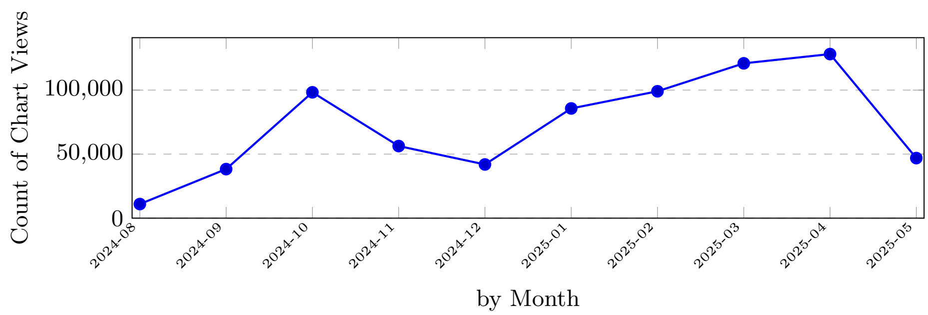

Fig. 7: Monthly Charts Views on *visualize.admin.ch*

Domain-Specific Dashboards Leveraging the core ontology, domain-specific dashboards have been built for different offices of the Swiss federal administration:

- https://www.strompreis.elcom.admin.ch A dashboard providing the current and historical electricity prices from the *Federal Electricity Commission* for all Swiss municipalities. This dashboard is built independently of visualize.admin.ch to cater to the specific use-case of a fine-grained overview down to the municipalities. Figure 8 shows the monthly traffic on the dashboard.
- https://www.dashboard.blv.admin.ch/overview A dashboard showing key up-to-date indicators of the *Federal Food Safety and Veterinary Office*. This dashboard is based on multiple components created with visualize.admin.ch
- https://www.agrarmarktdaten.ch/daten A data download portal containing agricultural data collected by the *Federal Office for Agriculture*. This interface allows to select and download specific parts of the data based on `cube.link`

5.4 Lessons Learned

Co-creation and Agile Development As one of the major success factors of the overall effort described in this paper, we identify the iterative and interac-

[19] https://github.com/visualize-admin/visualization-tool

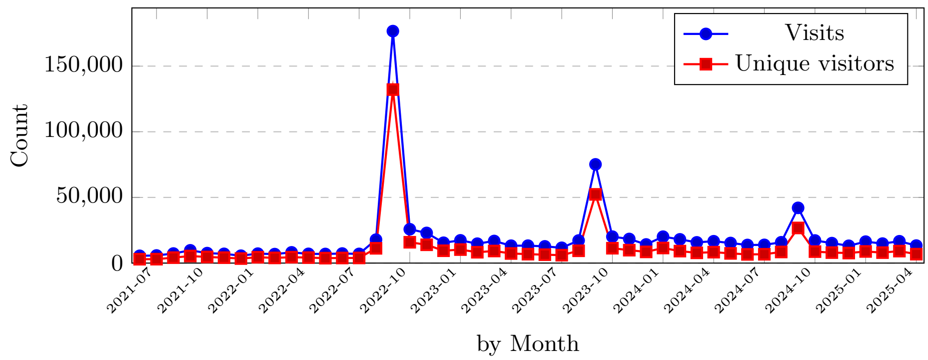

Fig. 8: Website Traffic on https://www.strompreis.elcom.admin.ch over time: unique visitors and visits as measured by a Matomo instance. The peaks occurring in September show increased interest in the data when the prices for the following year are announced. (In 2022 there was a bigger than usual price increase announced.)

tive process between the definition of the `cube.link` ontology and the design of end-to-end tools that occurred over the same period of time. After a solid basis on how to write and consume the main structure of a cube was established, the necessary metadata and extensions were built iteratively across several teams. On one hand, end-user tools like visualize.admin.ch brought requirements to the table to optimize the automatic generation of graphs. These often triggered additions to the `cube.link` ontology, which were discussed and finally tested with *Cube-Creator*. Once stable and working end-to-end, the final additions were often retroactively documented in the `cube.link` ontology. This allowed to put a spotlight on specific challenges from all involved experts, which was crucial, especially with technical intricacies such as the description of hierarchies or related dimensions.

The Attraction Field Surrounding GUI Tools Even though the *Cube-Creator* was originally designed for transforming and annotating data in CSV and other tabular formats, the tool is not well-suited for direct use with relational databases. While some data providers choose to export their databases to CSV and process them with Cube-Creator, this approach can be inefficient and is generally not recommended for handling complex relational data structures. Although this is of course technically possible, the *Cube-Creator* was optimized for manual use-cases and was not intended to be integrated into automated data pipelines. Therefore, it became necessary to clarify in such cases that using *Cube-Creator* as an intermediate step is unnecessary – and that data publishers are better off exporting their data directly as Linked Data using the frameworks mentioned in Section 5.2.

Validation and Introspection As organizations started to develop their own data pipelines, questions on how to create cubes were directed to the development teams around *Cube-Creator* and *visualize.admin.ch*. While supporting more developers was a goal of the overall project, repeated questions and requests took resources from extensions and further developments. Therefore, projects on validation and technical tools to introspect data structures were launched. We describe below the *Cube-Viewer*, which was designed to introspect data on a technical level. We also describe some further tools that were provided through a GUI application called Cube-Validator[20].

Cube-Viewer The Cube-Viewer[21] is a Web application that displays `cube.link`-compliant data as tabular data drawing on the possibilities of localization, e.g., to show the column headers in different languages. Furthermore, the metadata of the cube itself as well as all its different dimensions can also be visualized. The Cube-Viewer allows for some basic data analysis by means of filtering the data. For example, the dataset from Figure 1 can be introspected in the Cube-Viewer directly.

Cube Validator The Cube-Validator is another Web application that allows to select a cube and a validation profile, and provides a detailed report targeted to developers creating cubes through their own software. The aforementioned cube can be preselected and validated within the Cube-Validator.

6 Future Developments and Summary

The overall project and funding framework for the LINDAS project[22], which enabled the work around the `cube.link` ecosystem, was renewed in early 2025 with a timeframe of an additional ten years. This ensures that the gained experience and the built ecosystem around `cube.link` will be further developed.

As many datasets at the Swiss Confederation are handled in a C# .NET programming context, a dedicated library providing support to publish valid `cube.link` data is in the works also, and will be published in the future as open source. The *Cube-Creator* provides, apart from the open issues on GitHub, a roadmap for long-term development. In addition, *visualize.admin.ch* is in active development on GitHub.

To summarize, we introduced in this paper `cube.link`, a generic ontology designed to efficiently publish government data in tidy, multidimensional formats as cubes. Our approach supports rich metadata descriptions and concept reuse across datasets, thereby enhancing usability and enabling advanced visualizations.

The ecosystem built around `cube.link` demonstrates its ability to address key limitations of existing approaches and confirms its suitability for practical applications.

[20] https://cube.link/#validating-cubes
[21] https://github.com/zazuko/cube-viewer
[22] https://lindas.admin.ch/

References

[1] United Nations Department of Economic and Social Affairs. *Strategy Note on Open Government Data for Effective, Accountable and Inclusive Institutions and Public Service Delivery*. Strategy Note. United Nations Department of Economic and Social Affairs, Dec. 2024. URL: https://publicadministration.desa.un.org/sites/default/files/cepa-sessions/Strategy%20note%20open%20government%20data%20Dec%202024.pdf.

[2] Bernd W. Wirtz, Jan C. Weyerer, Marcel Becker, and Wilhelm M. Müller. "Open Government Data: A systematic literature review of empirical research". In: *Electron. Mark.* 32.4 (2022), pp. 2381–2404. DOI: 10.1007/S12525-022-00582-8.

[3] Bruno Elias Penteado, José Carlos Maldonado, and Seiji Isotani. "Methodologies for publishing linked open government data on the Web: A systematic mapping and a unified process model". In: *Semantic Web* 14.3 (2023), pp. 585–610. DOI: 10.3233/SW-222896.

[4] Hadley Wickham. "Tidy Data". In: *Journal of Statistical Software* 59.10 (2014), pp. 1–23. DOI: 10.18637/jss.v059.i10.

[5] Lorena Etcheverry and Alejandro A. Vaisman. "QB4OLAP: A Vocabulary for OLAP Cubes on the Semantic Web". In: *COLD*. Vol. 905. CEUR Workshop Proceedings. CEUR-WS.org, 2012.

[6] W3C Working Group. *RDF Data Cube Vocabulary*. W3C Recommendation. Accessed: 2025-05-13. World Wide Web Consortium (W3C), Jan. 2014. URL: https://www.w3.org/TR/vocab-data-cube/.

[7] W3C Spatial Data on the Web Working Group. *RDF Data Cube Extensions for Spatio-Temporal Components (QB4ST)*. W3C Working Group Note. Accessed: 2025-05-13. World Wide Web Consortium (W3C), Oct. 2017. URL: https://www.w3.org/TR/qb4st/.

[8] Richard Cyganiak, Simon Field, Arofan Gregory, Wolfgang Halb, and Jeni Tennison. "Semantic Statistics: Bringing Together SDMX and SCOVO". In: *LDOW*. Vol. 628. CEUR Workshop Proceedings. CEUR-WS.org, 2010.

[9] Sarven Capadisli, Sören Auer, and Axel-Cyrille Ngonga Ngomo. "Linked SDMX Data: Path to high fidelity Statistical Linked Data". In: *Semantic Web* 6.2 (2015), pp. 105–112. DOI: 10.3233/SW-130123.

[10] Evangelos Kalampokis, Bill Roberts, Areti Karamanou, Efthimios Tambouris, and Konstantinos A. Tarabanis. "Challenges on Developing Tools for Exploiting Linked Open Data Cubes". In: *SemStats@ISWC*. Vol. 1551. CEUR Workshop Proceedings. CEUR-WS.org, 2015.

[11] Areti Karamanou, Evangelos Kalampokis, Efthimios Tambouris, and Konstantinos A. Tarabanis. "Linked Data Cubes: Research results so far". In: *SemStats@ISWC*. Vol. 1654. CEUR Workshop Proceedings. CEUR-WS.org, 2016.

[12] Michael Luggen, Adrian Gschwend, Bernhard Anrig, and Philippe Cudré-Mauroux. "Uduvudu: a Graph-Aware and Adaptive UI Engine for Linked Data". In: *LDOW@WWW*. Vol. 1409. CEUR Workshop Proceedings. CEUR-WS.org, 2015.