COGNITE

BUILDING A CONTEXTUALIZED POWER SYSTEM NETWORK MODEL

CONTENT

INTRODUCTION

Statnett is both the system operator of the Norwegian power system network as well as the owner and maintainer of the high-voltage transmission network. For Statnett as the system operator of the Norwegian power system, the natural way to represent data about the power system is not in a hierarchical tree structure, but rather in a graph that supports advanced graph-related queries naturally. Statnett models its functional power system network in adherence to the Common Information Model (CIM) standard and is actively involved in its ongoing development. For Statnett's maintenance organization, however, data is organized in an Enterprise Resource Planning (ERP) system with a hierarchical structure. While those components form the backbone of the high-voltage transmission network in the real world, there is to date no data model that fuses the physical equipment view with the functional power system model.

The Norwegian grid structure is complex and conducting analysis on grid asset health often involves combining several data sources. Retrieving and querying data is today to a large degree static and time consuming and this process is difficult to automate due to the current data platform's lack of flexibility hence, data is retrieved manually. Nonstructured data makes it difficult to see relevant data in context, not just combining ERP and CIM information, but also information from other data sources such as redispatch and metering data for settlement.

Data is stored and used in silos and its structure is in many ways a reflection of the organizational setup where different parts of the organization have developed a culture of working in isolation. Combining data from different sources to conduct more accurate analysis for power system operations or long-term planning of grid asset maintenance opens up a huge value potential and can lead to smarter investment decisions and better grid utilization.

Statnett and Cognite, a global industrial AI Software-as-a-Service (SaaS) company, have from 2018 to 2020 been engaged in an R&D partnership. As part of this R&D engagement, it was explored how to construct a unified data model (starting with a functional power system model) that can be populated with information from a multitude of data sources and queried efficiently by domain experts using a domain language that is familiar to them. Such a data model would considerably extend what is available in the CIM with information from a variety of other data sources such as special regulations redispatching events and net settlements, which in turn would enable Statnett's analysts to build applications that improve the company's operations, maintenance and analysis activities.

This paper describes the work conducted and is organized as follows. Section 1 introduces the relevant data modelling terminology within Cognite Data Fusion (CDF) and explains the main underlying principles. Section 2 and 3 introduce aspects of the functional and physical representation of the power system, respectively. Together, they lay out the principles on which the unified data model in CDF is built. Section 4 presents our approach of contextualization, that is the process of (semi)automatically joining information from multiple data sets together and extracting the relevant associations between information entities. Section 5 then discusses our approach to expose the richness and complexity of the data model to end users in a domain specific language. Section 6 offers a conclusion.

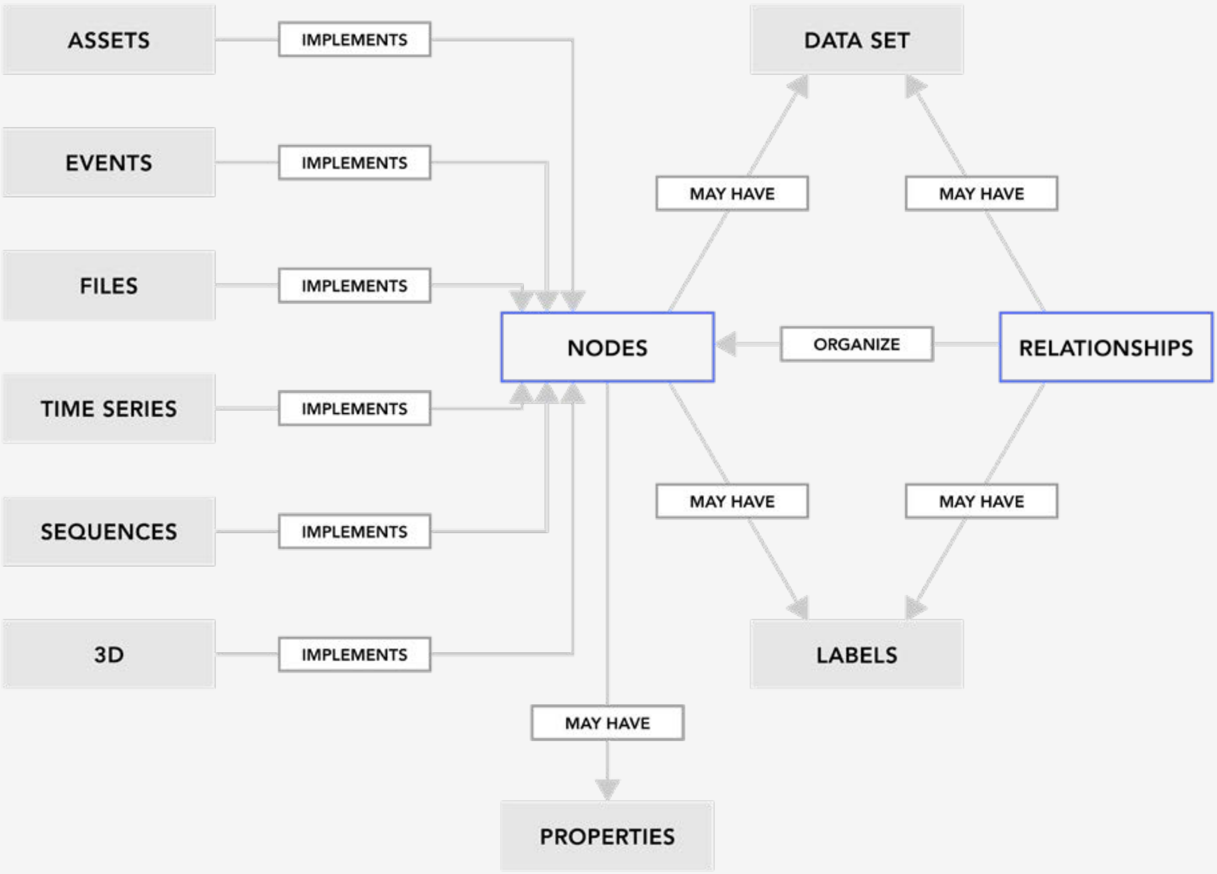
1. CDF DATA MODEL TERMINOLOGY

Cognite Data Fusion (CDF) provides a data model with fixed resource types so that users and application developers can make assumptions about the structure of the data and rapidly build applications on top of the data model.

Figure 1: CDF data model

A resource type is an individual data entity with a unique identifier. As seen in Figure 1, resource types include, among others, *Assets, Time Series, and Events* which form the nodes in a labeled property graph. A data model in CDF is contextualized through establishing associations between related resources. These *Relationships* are a resource type that is used to model associations between any two other resource types.

It is through the notion of *Relationships* that graph-based data models, such as power system models, can be represented in CDF. Another key feature of CDF is that Relationships are typed and timestamped. Types allow for semantic distinction and are user-defined. The various types used in the power industry context will be addressed in more detail below. Timestamping refers to the fact that each relationship has a *valid_from* and *valid_to* timestamp that allows for efficient versioning of any association over time. This allows, for instance, a power system network model to be versioned natively and enables queries for how the power system looked at any given point in time and when it changed.



connectsTo

This relationship type is used to express electrical connectivity in the power system network context. It is understood that this type is bidirectional, but directionality follows the CIM implementation at Statnett, e.g. Terminal connectsTo ConnectivityNode.

implements

This relationship type is used to express which physical equipment implemented which functional role, e.g. PhysicalTransformer implements FunctionalTransformer. This allows us to track moving (physical) components in a (functional) power system network model.

belongsTo

This relationship type is more generic and can be used in several ways:

Associating data from events, time series, sequences with functional or physical assets, e.g. TimeSeriesA belongsTo AnalogB

Associating assets to higher levels of aggregation, e.g. PowerTransformerA belongsTo SubstationB, or SubstationA belongsTo PriceAreaB

2. FUNCTIONAL POWER SYSTEM NETWORK MODEL

A starting point for the data model in CDF is the functional power system network model which Statnett stores in a dedicated graph database. As many other grid operators, Statnett adheres to the CIM standard. The CIM is an open data model standard developed with the intent to facilitate the exchange of power system network information between TSOs, DSOs, and regulators and also with ENTSO-E, the European Network of Transmission System Operators for Electricity. For the purposes of this R&D project, we adhered to version 15 of the CIM standard.

It is important to note that Statnett's power system network model is a functional description of the Norwegian power system (including interconnections to and parts of neighboring countries) to the extent it needs to provide a model for *system operations*. In practice, this means that while it models the entire Norwegian power system, it only does so to the degree necessary for operating the Norwegian power system. For example, some parts of the regional grid may be modeled as loads for simplification whereas other parts may be extensively modeled, even for the same voltage level.

As Statnett adheres to the CIM standard when representing its power system network model, it was important for any solution built in CDF to be compliant with the CIM data modeling framework. As a consequence, the data model in CDF is at the core a mirrored representation of the CIM model, particularly its equipment profile, but extended with information from additional data sources and enhanced querying capabilities that are provided through grid-specific extensions to the Cognite Python SDK which is discussed in further detail below.

In the existing data model, the CIM model serves as the master data source for all functional assets and their attributes and metadata. This includes mainly substations (and all components contained in a substation such as transformers, breakers, disconnectors, etc.), and transmission lines. In addition to asset information, we also extract electrical connectivity information where connectivity between two instances of conducting equipment is established through a Connectivity Node to which Terminals connect (see Figure 2 below).

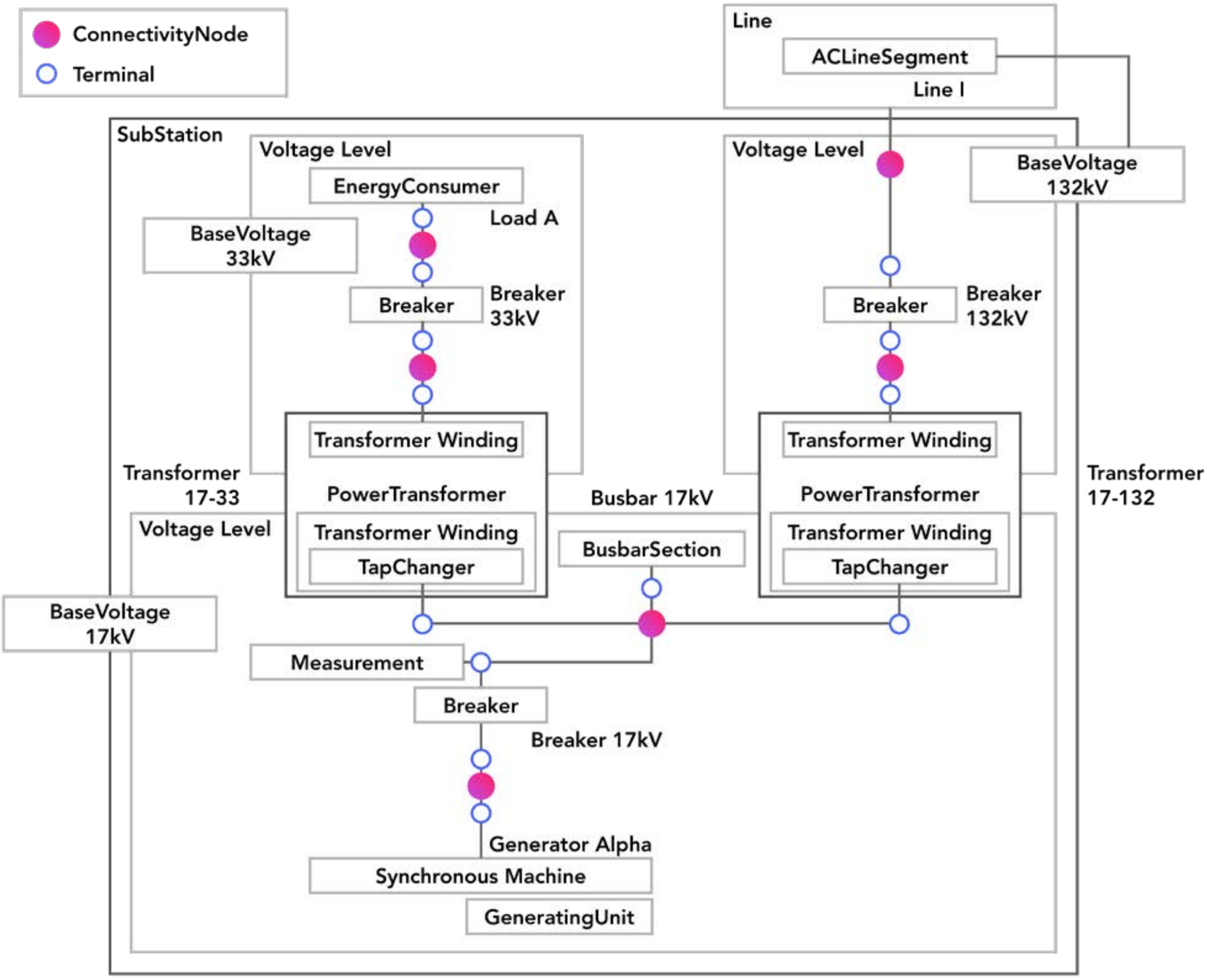


Figure 2: Example circuit with full CIM mappings (see Figure 3.13 in McMorran, A.W. (2007). An Introduction to IEC 61970-301 & 61968-11: The Common Information Model.)

This node-breaker representation is mirrored in CDF, where the following convention on how connectsTo relationships are used is shown in Figure 3.

While we attempted to stay as true to the CIM model as possible, there is an important difference in the way CDF handles temporal changes to the power system network model. In practice, Statnett stores snapshots of its power system network model in time which makes queries for changes in the network model impossible. In CDF, we leverage the fact that relationships are a resource type that can be time stamped, which means any association between nodes in the data model graph can be given a time range in which said association is valid. This allows for both the persistence of assets, time series, and other information that is no longer present in the current network model, but also allows for advanced queries against the entire history of the power system.

Operationally relevant time series data such as active and reactive power, current, and voltage but also oil pressure/temperature in transformers and cable systems is provided by a centralized SCADA/EMS system at Statnett. The CIM model provides asset classes for Analogs and Discretes that are used to link the relevant time series to. Each Analog or Discrete is, like any other object in the CIM model, identified through a unique master resource identifier (mRID). Time series from the SCADA system contain the same mRID in their meta information so that matching of time series to analog/discrete assets in CDF is straightforward. An example schema is given in Figure 4 below.

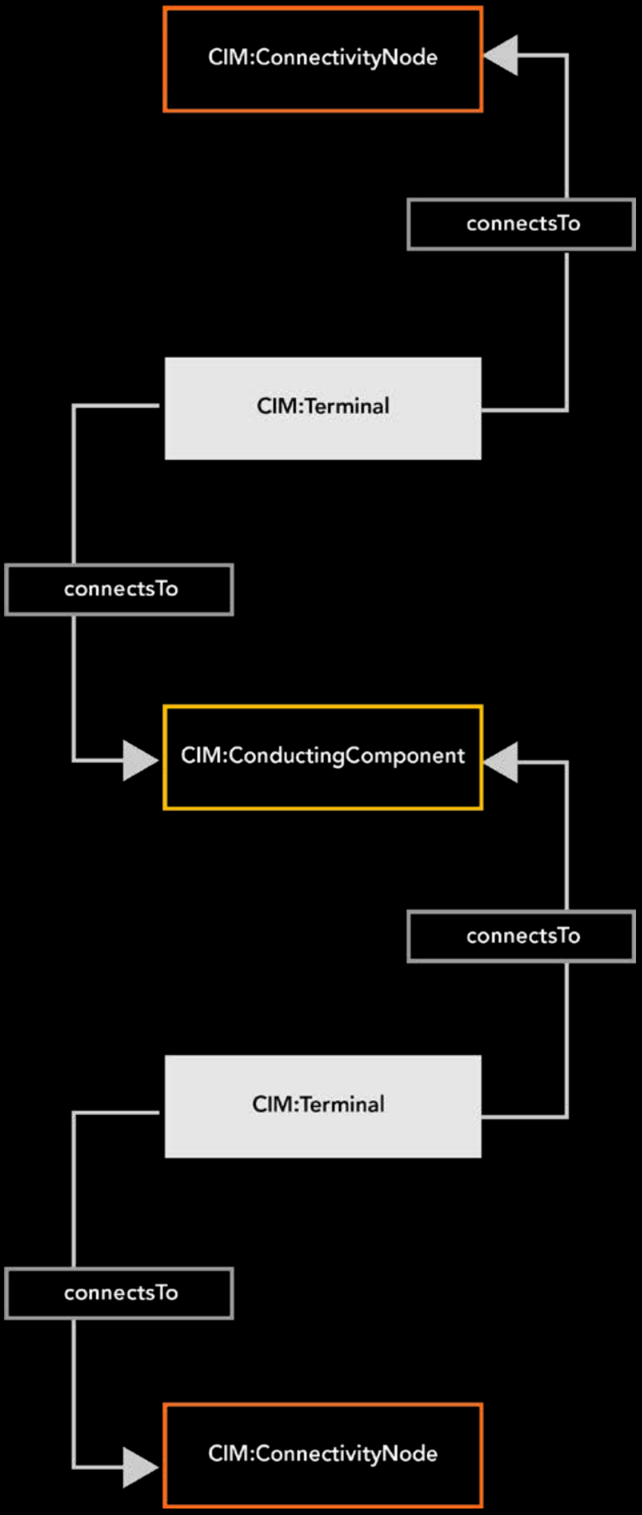


Figure 3: CDF node-breaker model

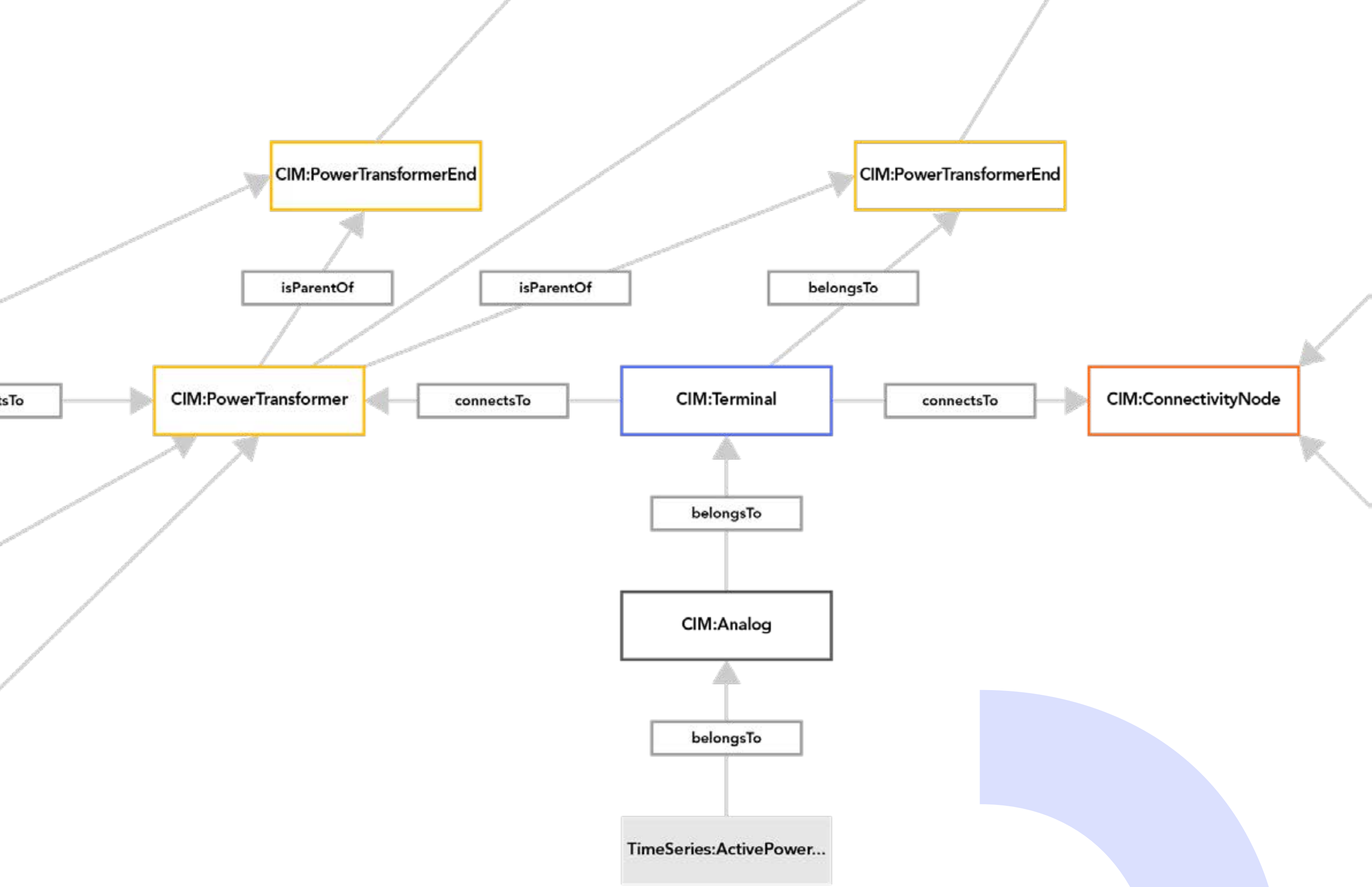


Figure 4: Example schema of how time series from SCADA are linked to CIM analogs/discretes

Another important concept within a power system model is the notion of a Power Transfer Corridor (PTC). A PTC is a set of conducting equipment (transmission lines or transformers) defined over a critical transmission corridor. Statnett maintains a set of PTCs to ensure integrity of the grid/power system/transmission in interfaces between two regions of the power system, or sets of conductive equipment that are exposed to a significant portion of the transmission exchange between two parts of the system (click here). Monitoring the power flow over a PTC enables the transmission service operator (TSO) to ensure that the consequences of contingencies (e.g. outages) in the power system are limited. Safe operating conditions are ensured by keeping the power system within its operating limits.

At present, the PTCs modeled in CDF are defined in a separate system from where Statnett stores and maintains its power system network model. In the source, the PTCs are served as a nested data structure which contain its constituent equipment, as well as pertinent metadata. In the metadata of the PTC we find information on the limiting factors, the type of monitoring the power flow is under and whether the PTC is market limiting. The groupings of components from the source are persisted in CDF as shown in Figure 5.

In Figure 5, a ConductingEquipment is either a transmission line, or power transformer. The spsComponents container can also contain generating units. The ConductingEquipment components are modeled as the PTC view of the pertinent component. For instance, if a transmission line contributes 40% of its power flow to a PTC this information is persisted in the ConductingEquipment node. The ConductingEquipment node is a view of the equipment in CIM, because relationships in CDF are not weighted.

We also note that each of the container nodes has only one child for illustrative purposes, and that each container holds a set of components.

In addition to the architecture, we also persist the power flow through a PTC in CDF in *time series*.

Figure 5: Example schema for Power Transfer Corridors

3. PHYSICAL EQUIPMENT

A complementary part of the power system network model is information about the physical equipment that implements a given functional role, its technical specification, and maintenance related data. In principle, a functional equipment should never contain information about physical attributes that would change once said physical equipment is altered, updated, or exchanged. In practice, however, a clear distinction between functional and physical attributes has not been upheld in the source systems.

Most of the asset-specific information about equipment that Statnett owns and maintains itself is provided by the centralized ERP system. As such it upholds a functional asset hierarchy which is structured through the perspective of the maintenance organization. At the lowest level of detail, a given functional asset has a child asset representing a physical piece of equipment, identified typically through its *serial number*. It is these physical assets that are ingested into CDF and linked to functional counterparts with relationships of type *implements*. As before, timestamps on those relationships allow users to track equipment as it moves across functional locations and enables advanced queries for (functional) time series based on an equipment-centric perspective. For example, condition monitoring models may require the entire history of load on a power transformer, irrespective of where in the grid it has been placed. Such queries involve several graph traversal steps in order to retrieve the relevant time series data from potentially multiple functional transformers.

To these physical assets other information is linked through relationships that pertain to the physical equipment, such as work orders, oil sample lab results, or field measurements on bushings. Example schemas for different asset classes are presented below.

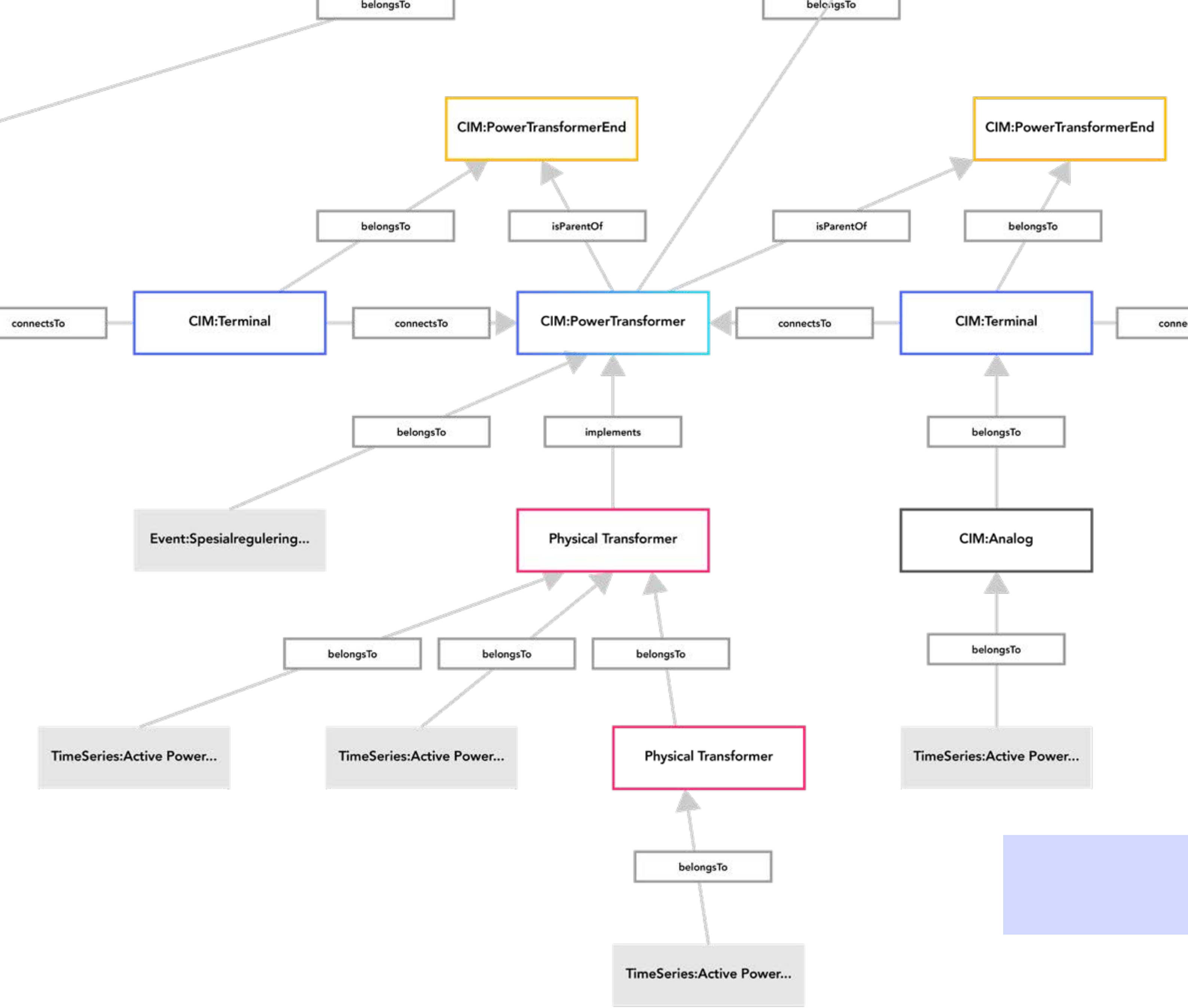


Figure 6: Example schema for Power Transformers

Power Transformers

Power Transformer are critical components in the power system. Consequently, they are well instrumented and relevant data about their operation and maintenance history are available from multiple source systems.

For the functional perspective we follow the conventions laid out in the CIM standard and model both objects (PowerTransformer, PowertransformerEnd, Terminal, Analog, etc.) and their associations with Assets and Relationships in CDF. Functional time series that are available from the centralized SCADA system are then linked to the respective Analog assets obtained from the CIM representation of the power system model.

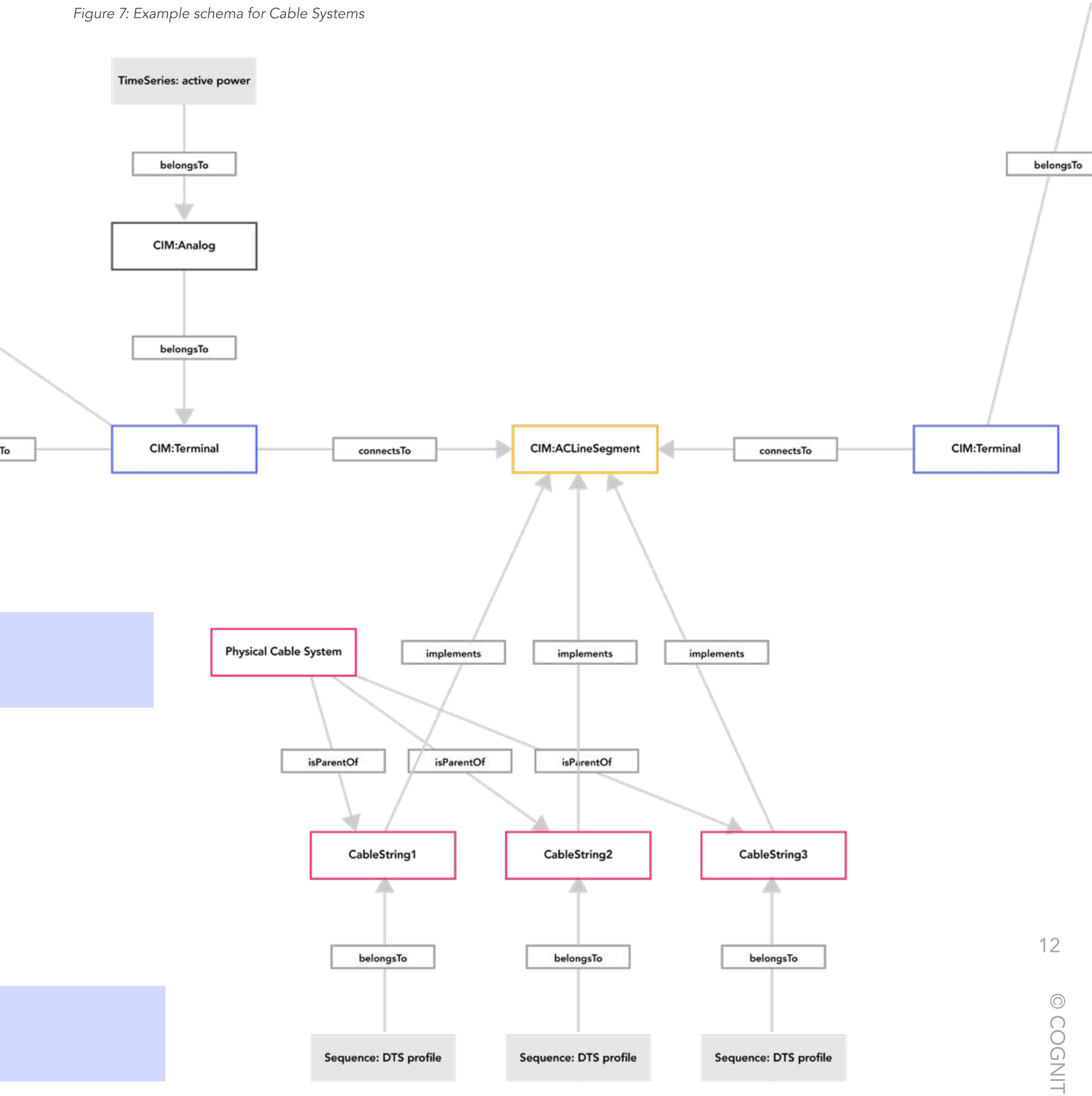
For the physical equipment perspective, we retrieve information from Statnett's ERP system. To date, such transformer assets are not broken down further into subcomponents in the source system and constitute a monolithic structure. As there is, however, a need to contextualize further information (such as, e.g., field measurements on bushings) an additional structure of subcomponents is introduced. It is to these subcomponents that relevant data is linked, where appropriate. An example schema is given in Figure 6.

Cables

From a functional perspective, cable systems are a special instance of AC Line Segments, and modeled as such in Statnett's CIM representation. Physically, however, there is one cable string per phase, such that the resulting implements relationships are *many-to-one*.

Whereas most power system components are attributed with (univariate) time series data from sensors, more and more cable systems are equipped with Distributed Temperature Sensing (DTS) technology that generate data that is defined over a spatial domain. These spatial profiles contain several thousand observations for a single timestamp and are stored in CDF as Sequences. An example schema is given in Figure 7.

Figure 7: Example schema for Cable Systems



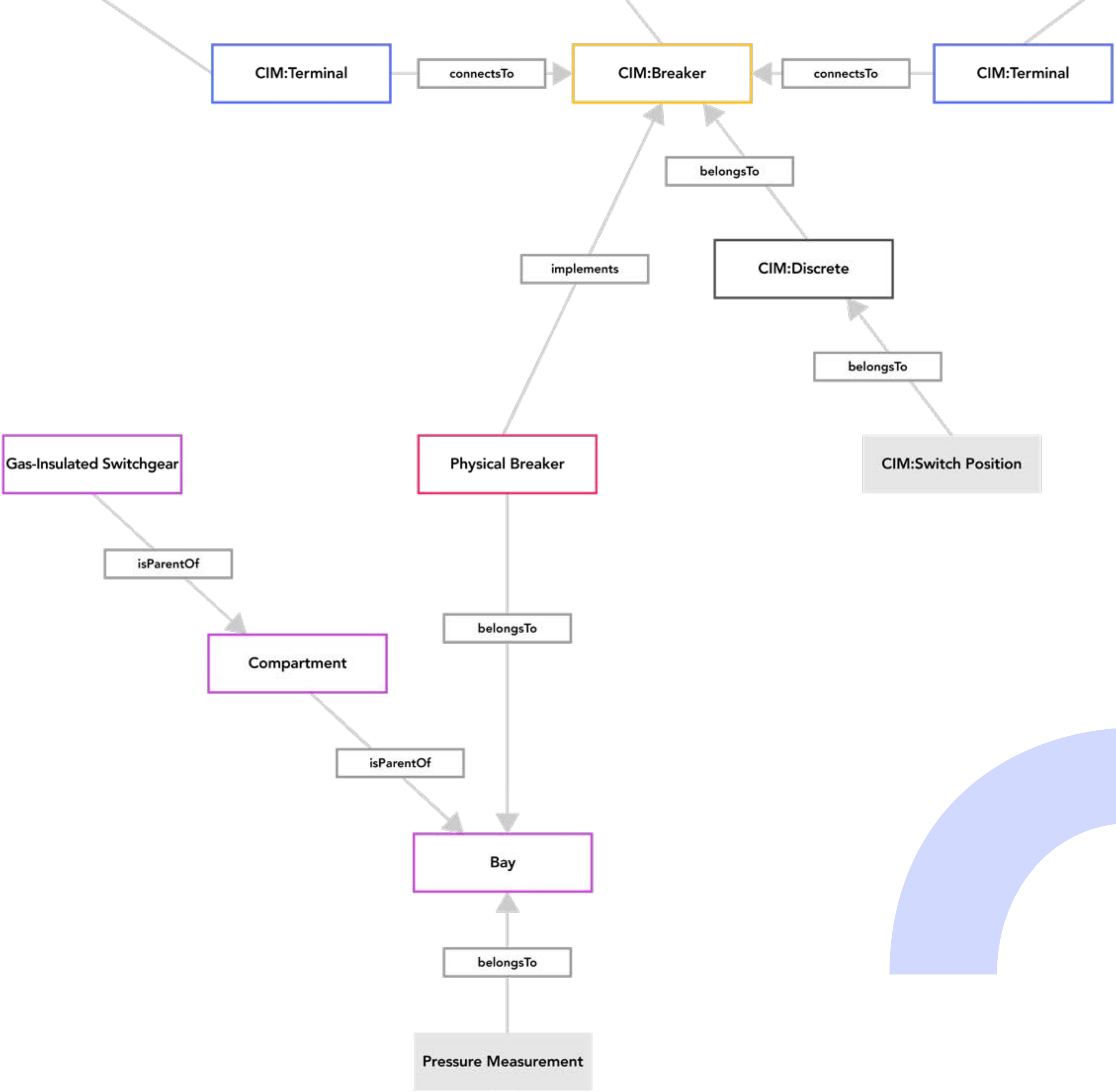


Figure 8: Example Schema for Gas Insulated System

SF6 Gas-insulated Systems

Gas-insulated switchgears are becoming increasingly more common in situations where small substation footprints are important, for example in urban areas. These consist of compartments that are filled with SF6 gas, an excellent insulator but also a very potent greenhouse gas. Monitoring of SF6 leakage is therefore becoming paramount and requires access to relevant operational data.

While each breaker contained in these compartments is represented (physically) in the ERP system and (functionally) in the power system model, an additional structure is required that allows a given breaker to be placed correctly in the respective switchgear compartments and bays. Information about these bays and compartments are retrieved from technical documentation provided by the manufacturer and used to create the relevant assets and their associations in CDF. An example schema is given in Figure 8.

4. CONTEXTUALIZATION

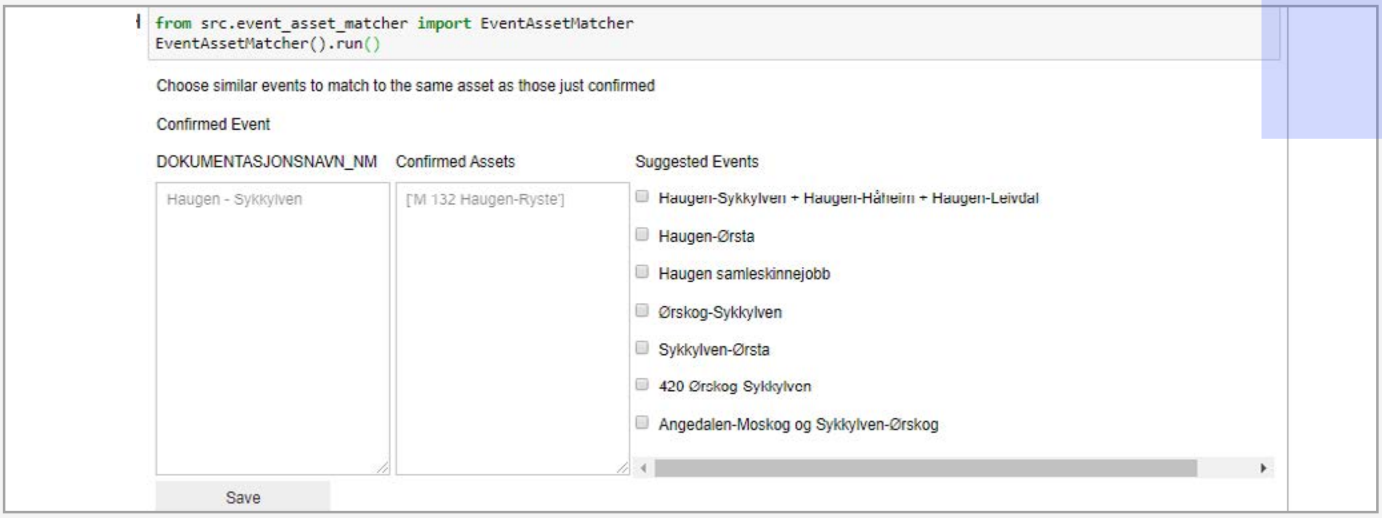
Statnett's analysts need to be able to assess and analyze data from several data sources on top of what is provided through CIM and the ERP system. However, as many of these source systems historically have not been linked in a meaningful way, the analysts have often found themselves manually mapping data between these source systems.

Data originating from redispatch events provides a good example. Redispatching is a measure used by TSOs to relieve physical congestions in the transmission grid or otherwise ensure system security. It can be applied both within a bidding zone or between different bidding zones. It involves altering generation or load patterns in order to alter the power flow. TSOs must remunerate activated internal or cross-zonal redispatching. If the expected accumulated cost of relieving a bottleneck by redispatching exceeds the cost of expanding the capacity of the grid, the TSO should make the appropriate investment. The required investment analyses are rather complex. This is because many factors interact to determine the state of the grid at any time, such as planned line outages due to maintenance, unforeseen incidents causing outages or inaccurate weather forecasts causing generation from renewables to deviate from plan.

In order to perform such analyses Statnett must be able to map the generating unit or load affected by a redispatch event to the correct power transfer corridor, power line or transformer. In Statnetts systems, when a “redispatch event” occurs, a set of inputs are added to the source system, among others a free-text field that includes the affected component/asset. This results in a labor intensive and manual work process to match the free-text field to the affected components. Historically this has been a time consuming task performed in Excel on a needs basis. This approach has resulted in matched data living in silos and not available in context to other data sources.

To enable the analysts to fully leverage the unified data model and solve use cases such as the cost analysis for redispatching events, a "matching tool" for contextualizing data from different data sources was developed to complement the data model. It provides a jupyter notebook-based GUI as shown in Figure 9.

Figure 9: Jupyter notebook-based GUI for "matching tool".



```
from src.event_asset_matcher import EventAssetMatcher
EventAssetMatcher().run()
```

Choose similar events to match to the same asset as those just confirmed

Confirmed Event

DOKUMENTASJONSNAVN_NM: Haugen - Sykkylven

Confirmed Assets: ['M 132 Haugen-Ryste']

Suggested Events

☐ Haugen-Sykkylven + Haugen-Håheim + Haugen-Leivdal
☐ Haugen-Ørsta
☐ Haugen samleskinnejobb
☐ Ørskog-Sykkylven
☐ Sykkylven-Ørsta
☐ 420 Ørskog Sykkylven
☐ Angedalen-Moskog og Sykkylven-Ørskog

The "matching tool" takes items from two datasets as input, e.g. events and assets, and suggests a mapping rule connecting the events (or group of events based on identical metadata fields) and assets together based on similarity on the objects data and metadata (text strings). The developed algorithm determining the similarity between text strings uses tokenization, where each string is divided into tokens that consist of consecutive letters, or consecutive digits.

Example: "99 test-example T3" will result in the following representation [99, test, example, T, 3]. The similarity between two strings is first determined by the number of common tokens. If the same token is repeated 3 times in an asset and 2 times in an event, it counts as a minimum of (2, 3) = 2 common tokens. With the same number of identical tokens between an event group and two different assets, one counts the number of letter tokens that are within edit distance 1, then 2 and finally 3.

The actual matches are furthermore stored in CDF as events, so that Statnett can keep an audit trail of who matched what, what rule was applied, how certain one is that the match is correct etc. The algorithm tends to suggest assets with long names, since they are not penalized for having many extra tokens. However, the correct assets still appear among the top suggestions that are presented to the end user, which is what the user needs.

By using the problem of contextualizing and matching "redispatching" events to drive the development of this tool, while at the same time also using other source systems and data (s.a. meter data for settlements etc.) to validate our solution, we ensured that our "matching tool" was generalizable and scalable to solve similar matching problems for other source data. Key to this development was also to start with a simple solution solving the problem end-to-end which the end users could interact with through a simple UI. While the first iteration of the logical functionality represented the most basic functionality and smartness, we gradually added more and more smartness to the logic, allowing us to iterate continuously with the end user and optimize the workflow end to end.

This matching tool now allows Statnett to complement the CIM data model with additional contextualized data sources and for analysts to solve additional use cases, s.a. the redispatching cost analysis, on top of the CDF data model. All redispatching events and meter data for settlements have already been matched and verified by the end users, with significant time savings.

5. POWER SDK

In the previous sections we described our approach to constructing and populating a contextualized power system model that is of relevance to a large number of end users and can cater for different perspectives into the power system and provide richness through the availability of multiple data sources in context.

Inevitably, however, such a data model increases the complexity which is detrimental to its efficient use and, as a consequence, an important part of the R&D work has been to explore ways to efficiently query such a data model from a domain expert view.

In CDF, physical items, such as Substations and line segments, are stored as Assets, and the connections between the assets that make up the data model are stored as Relationships. While it is natural to think of assets and relationships as nodes and edges in a graph, that does not represent how a power analyst thinks about their data. For a power analyst, it is more natural to think of the substations as nodes, and the line segments between them as edges. For this reason, building graph query capabilities into the CDF back end would not provide an analyst with a good way of exploring their data in a way that feels natural to them, regardless of the technical implementation (Gremlin, Cypher, etc).

The solution we came up with, was to construct a conceptual map from the way the analyst thinks about the data, to the way it is represented in CDF. This map is implemented in the Power SDK.

The choice to implement this in Python was made on the basis that the ideal power analyst is proficient in using Jupyter notebooks. The Power SDK adds functionality on top of the regular Cognite Python SDK, which allows the users to query the data in ways that are natural to them, rather than ways that reflect how the data is stored in CDF.

Some of the most basic, but most useful features are the addition of better typing on objects that are of interest to Statnett users. In the regular SDK, users work with objects that are typed as Assets, which are identified as substations, ac line segments or transformers only by examining metadata on the asset. In the Power SDK, these variants are raised as proper types, so that rather than working on an Asset object which represents a substation, the user can work on a Substation object. This subtle difference has turned out to make a huge difference in how efficiently users are able to query their data and showed how valuable a close cooperation with domain experts is in building the right solution that provides immediate value. Examples of common queries for power system components in the Power SDK are given in Figure 10. To illustrate the ease of use, the equivalent queries are presented using the standard Cognite Python SDK.

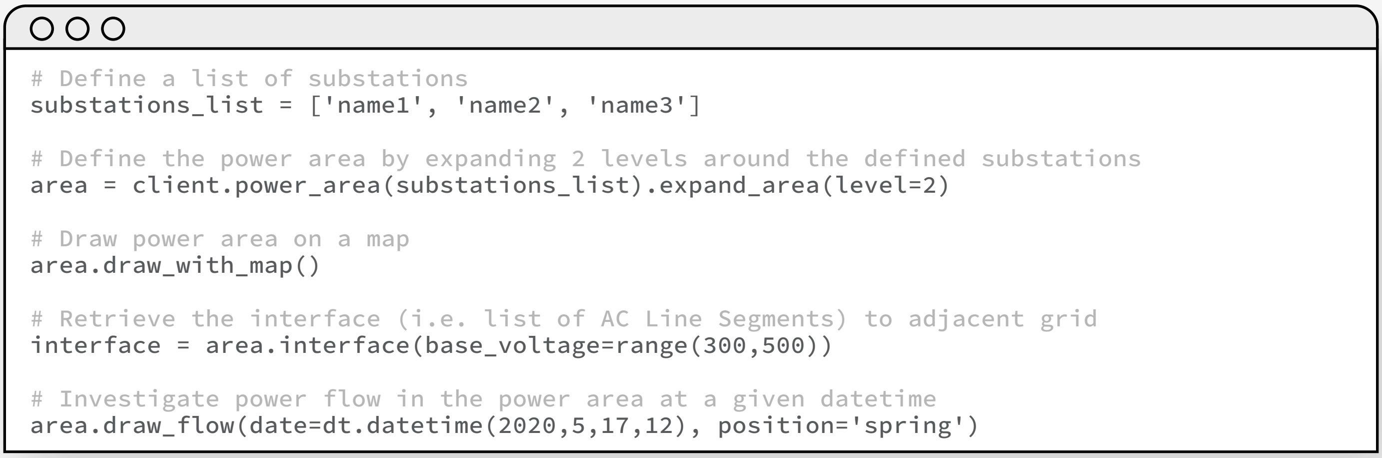
```
# get all substations (standard sdk)
client.assets.list(metadata = {'type':'Substation'})

# get all substations (power-sdk)
client.substations.list()

# list all transmission lines at voltage levels between 132 kV and 300 kV (standard sdk)
AssetList([acl
           for acl in client.assets.list(metadata = {'type':'ACLineSegment', 'Equipment.
gridType': 'GridTypeKind.main'})
           if int(float(acl.metadata['BaseVoltage_nominalVoltage'])) in range(132, 301)
          ])

# list all transmission lines at voltage levels between 132 kV and 300 kV (power-sdk)
client.ac_line_segments.list(base_voltage=range(132, 301), grid_type='main')

# list all hydro power plants in bidding area Elspot NO5 (standard sdk)
bidding_area_no5_extid = client.assets.list(name='Elspot NO5')[0].external_id
assets_in_bidding_area = client.relationships.list(target_resource_id=bidding_area_no5_
extid).to_pandas()['source'].map(lambda x: x['resourceId'])
client.assets.retrieve_multiple(external_ids=list(assets_in_bidding_area))

# list all hydro power plants in bidding area Elspot NO5 (power-sdk)
client.hydro_generating_units.list(bidding_area='Elspot NO5')
```

Figure 10: Example component queries in the standard Cognite SDK as well as the Cognite Power SDK

As part of the Power SDK work we have also implemented more sophisticated features that are tailored for the power industry. One user story that came up was the wish to define geographical areas in order to study the power consumption within them. We were able to solve this by iteratively working with Statnett domain experts, and the result is the PowerArea class, which has methods for easily collecting historical data for power flow across the interface of a geographical region, and allows analysts to gain quicker insight into when there is surplus/demand of power within the area, how much of the available production is used within the area, and how it is dispatched – uncovering, for example, situations where more redispatching could be achieved because of a power surplus.

Further functionality includes methods to plot the substations and line segments in the area in order to get a better understanding of the data. Example queries to define a power area and investigate its interface with adjacent parts of the grid as well as historic power flow are given in Figure 11, whereas an example plot of a user defined power area is shown on a map in Figure 12.

Figure 11: Example queries for power area in the Cognite Power SDK



```
# Define a list of substations
substations_list = ['name1', 'name2', 'name3']

# Define the power area by expanding 2 levels around the defined substations
area = client.power_area(substations_list).expand_area(level=2)

# Draw power area on a map
area.draw_with_map()

# Retrieve the interface (i.e. list of AC Line Segments) to adjacent grid
interface = area.interface(base_voltage=range(300,500))

# Investigate power flow in the power area at a given datetime
area.draw_flow(date=dt.datetime(2020,5,17,12), position='spring')
```

The Power Area is implemented by using the Python library Networkx to construct a graph client side where substations are nodes and AC line segments are edges. We initially conceptualized the power area generation through a CDF API endpoint, but it quickly became clear that the requirements of a power area were too domain-specific to be useful as a general endpoint for multiple industry verticals. For this reason, we made the decision to implement the power area generation client side through the Power SDK. This choice also allowed the user to regenerate and alter power areas much more rapidly. This turned out to be useful, as one of the requested features was to expand power areas incrementally by following AC line segments.

As the Power SDK serves as a map from the user's mental model of the power grid to the Statnett data model in CDF, the Power SDK will need to be updated whenever the data model is updated with breaking changes. Updating the Power SDK can therefore be thought of as a natural part of iterating on the data model.

Figure 12: Displaying power area on map with the Cognite power-sdk

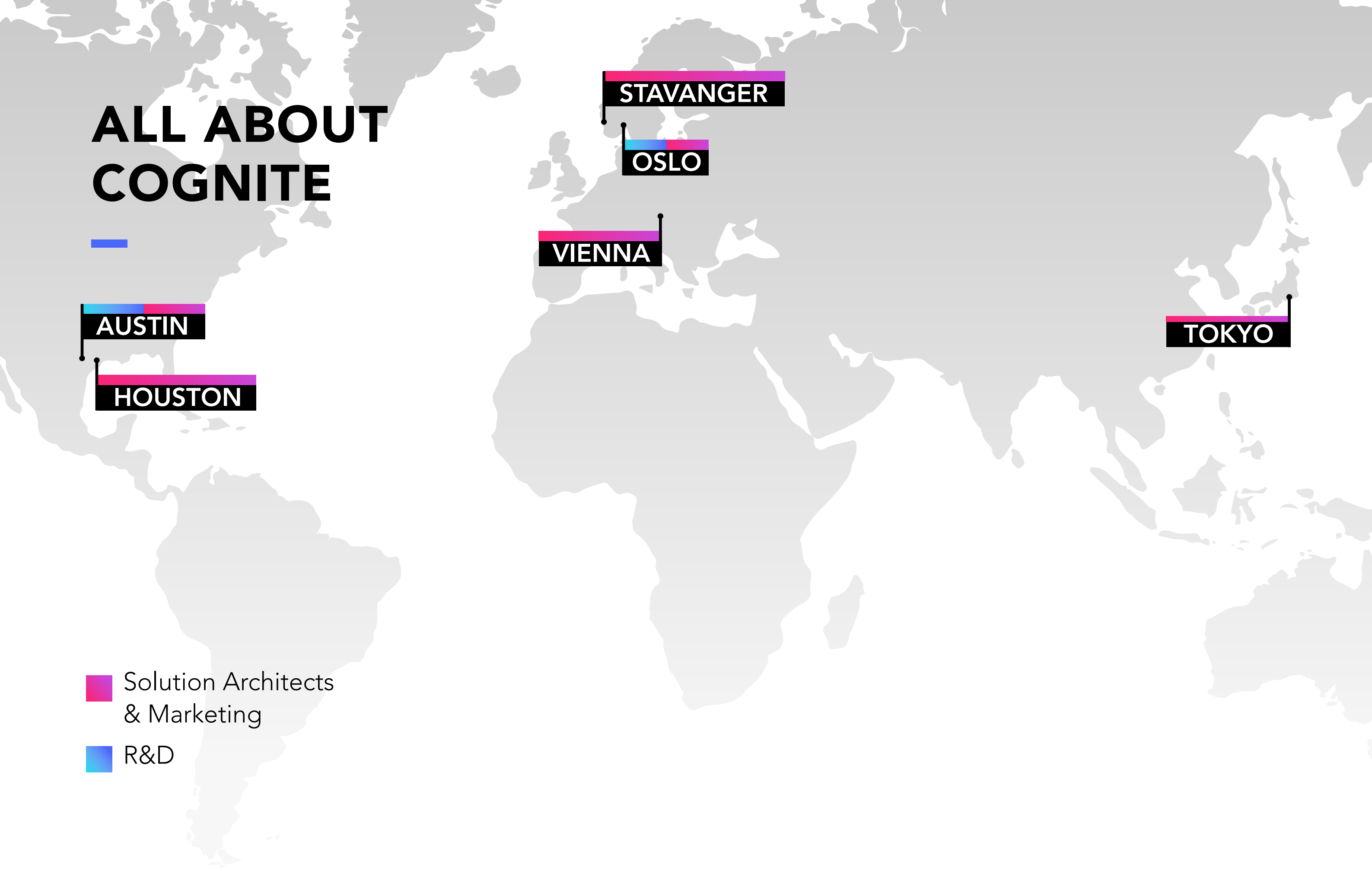
SUMMARY

In the previous sections we have described and discussed our approach on how to build, populate, and expose a rich, contextualized data model of the Norwegian power system. It was a comprehensive and iterative process that laid the foundation of design principles that further work on contextualization and Power SDK functionality bear fruit. We liberated data from multiple data sources and allowed analysts at Statnett to find and retrieve relevant information for their analysis tasks more efficiently, leading to better data-driven decision-making.

Some immediate examples of where these benefits can already be seen and could be further developed is in Statnett's yearly transformer capacity study or, more generally, the analysis of grid connections where the Power SDK can help documenting and updating the analysis more easily.

Furthermore, there is a vast value potential that can be explored in further development activities, such as the integration with power flow simulators for more advanced analysis tasks and the more active use of the node breaker representation in conjunction with breaker positions.

ALL ABOUT COGNITE



Our Company

Cognite is one of the fastest-growing software providers in the field of industrial digitalization. With 350 employees from more than 40 different countries, our interdisciplinary team combines world-class software competence and deep domain expertise. Cognite's offices include Oslo and Stavanger, Norway; Austin and Houston, Texas; Palo Alto, California; Tokyo, Japan; Vienna, Austria; and Singapore.

Cognite was founded to enable heavy-asset industries to generate value from their digital transformations by overcoming the obstacles of data trapped in silos, data type disconnectedness, data quality variance, the chasm between proof-of-concept and production (business value), the rigidness and slowness of legacy software development approaches, and the limited ability to leverage pre-existing data logic and flows.

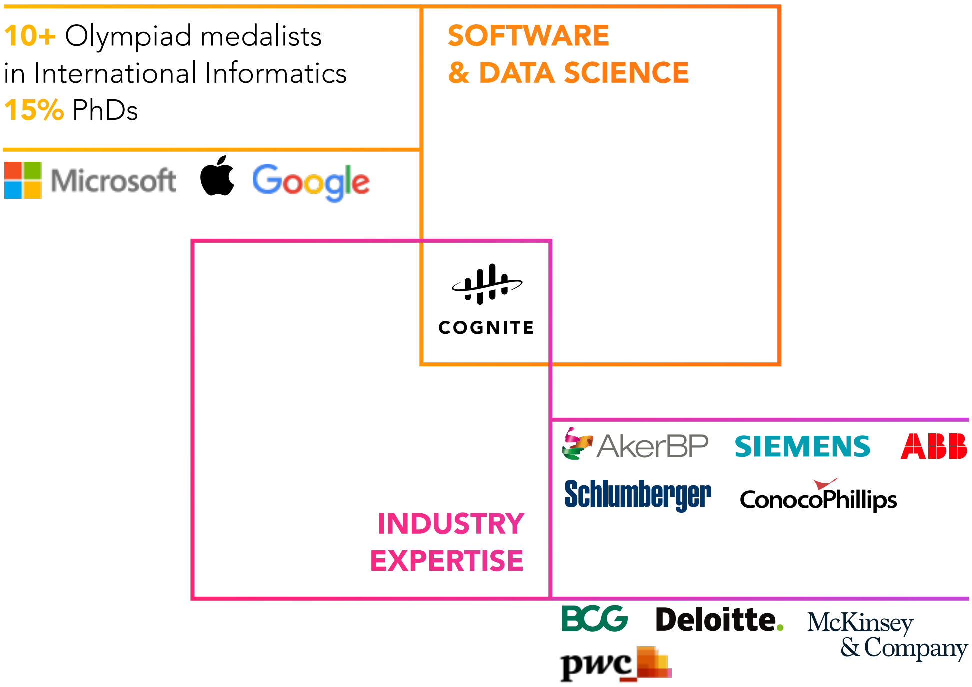
Our Mission

At Cognite, we make industrial data more valuable than ever. Empowering users with contextualized data as a service, delivering industrial AI at scale to unlock the power and value of your data.

The key to industrial digitalization lies in data liberation. Heavy-asset industries already have data. Now they need software to collect, clean, and put it to use. A resource to transform the data into information and stimulate a thriving ecosystem of industrial applications.

Embrace change and take control of your industrial transformation opportunity today. Empower your industrial data consumers to build, operationalize, and scale both models and applications with Cognite Data Fusion, the leading industrial DataOps software.

A Unique Global Team that Combines Software with Deep Industry Expertise



10+ Olympiad medalists in International Informatics

15% PhDs

SOFTWARE & DATA SCIENCE

INDUSTRY EXPERTISE

**To learn more,
get in touch with us.**

ALEXANDER GLEIM
Data Science Lead
Power & Utilities
alexander.gleim@cognite.com

THOMAS TRÖTSCHER
Chief Data Officer
Statnett
thomas.trotscher@statnett.no

COGNITE

COGNITE.COM
CONTACT@COGNITE.COM