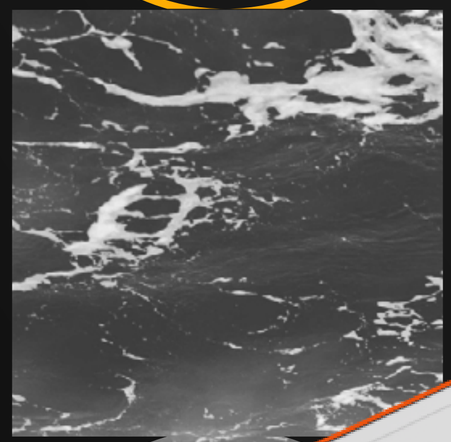
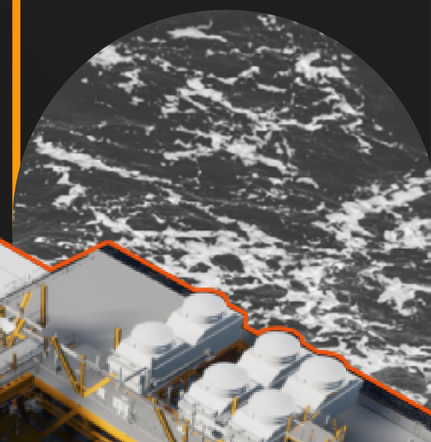
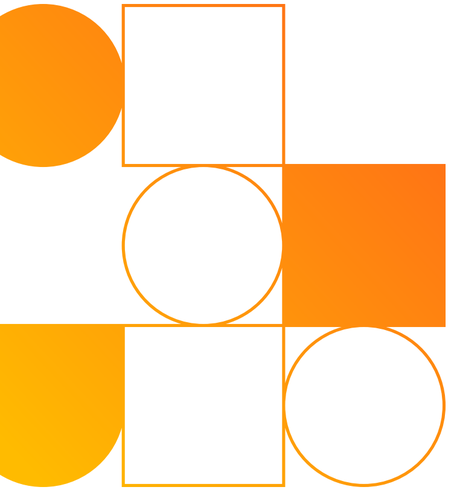
COGNITE

Advancing Digital Twins with Data Modeling

Use data product practices to provide simple access and governance of industrial data across source data models, domain data models, and solution data models







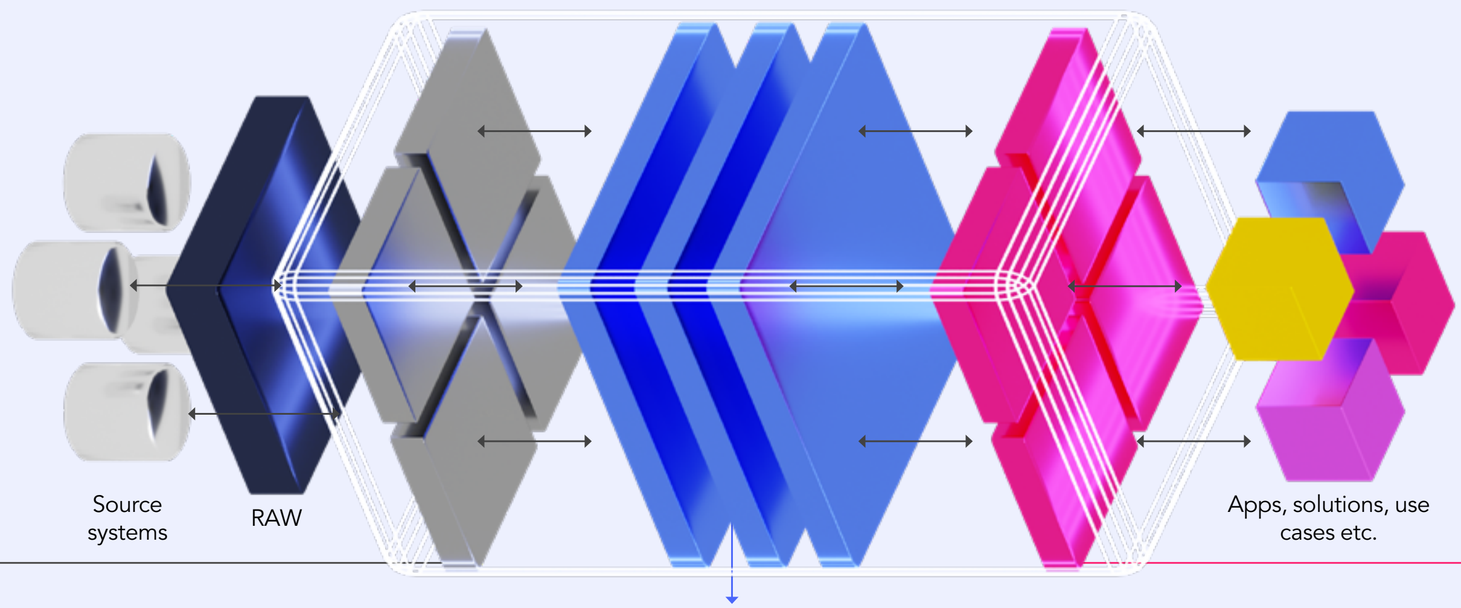
↘ Executive Summary

To increase the impact industrial digital twins have across operations, industry focus must shift to the data integration layer that powers digital twins. Today, much of our industrial data is still siloed, difficult to understand, and used by very few stakeholders in business. For this to change, digital twins must easily integrate all industrial data, provide simple access to this data, and accelerate the deployment and scaling of solutions.

Data modelling is a core component of turning siloed data into scalable solutions. Below is the data modelling framework Cognite Data Fusion® uses to power our Open Industrial Digital Twin:

- **Source data models** - Data is liberated from source systems and made available in its original state.
- **Domain data models** - Siloed data is unified through contextualization and structured into industry standards.
- **Solution data models** - Data from the source and domain models can be reused to deploy and scale solutions.

With Cognite Data Fusion®, data from all industrial sources can be integrated, contextualized, and scaled to solve challenges across asset performance, production optimization, product quality, and more.



Source data model

Asset hierarchy from SAP ■ OPC from Kepware ■ BOP/BOM from MES

Liberate data in source systems by having them modeled and queryable in Cognite Data Fusion® through the same API interfaces.

Domain data model

Asset or process centric ■ Follow standards (e.g. OPC, ISA95) ■ Custom semantic types (CNC)

Domain data models are optimized for reuse between application developers, data scientists and solution developers. We offer a set of industry standard OOTB data models that can be extended.

Solution data model

Data model targeting data consumption for use cases

Optimized data models for applications, data scientists and other solutions. We provide a GraphQL interface with auto generated SDKs for minimal code. This provides Backend as a Service (BaaS) to reduce cost of development and maintenance of solutions.

The Industrial Data Problem

The physical world of industrial data is a messy place. Equipment wear, fluctuating operational targets, and work orders are all important factors in determining the cause of operational issues. A process value cannot be observed and interpreted in isolation. Is our value too high? What is the maximum pressure recommended by the vendor? Are there any deviations from the recommended value? When was this pump last inspected and what were the observations?

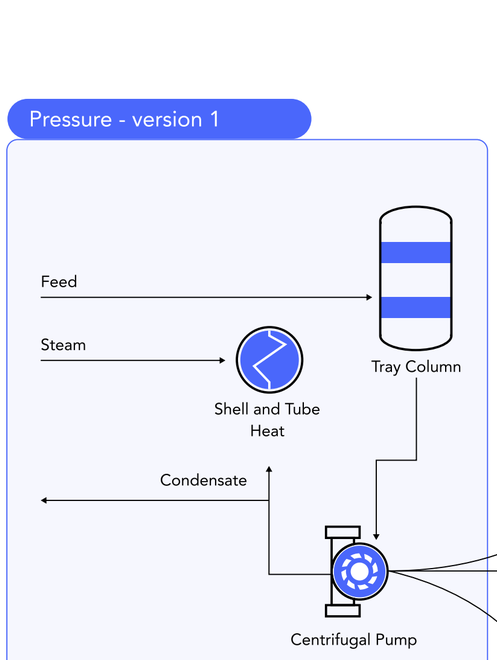
These questions are the concerns of the subject matter experts, the engineers responsible for keeping equipment running and continuously operating at a level that maximizes efficient production both short-term and long-term. Increasingly, documentation and electronic trails of maintenance and operational history can be accessed digitally. However, diagrams and vendor documentation are still found in flat PDF files, maintenance records are scanned paper documents, and 3D models are not up-to-date with the actual physical world as modifications and maintenance happen. How do we enable subject matter experts across operations, maintenance, and quality to solve their problems with digital tools? Data science models combined with physical simulators promise huge optimisations. Yet, you will also need to move beyond the constraints of the vendors' models and tie together sensor data from your end to end operations with models that can predict fault situations or optimize your production.

Herein lies a fundamental question; You can store all this industrial data in databases, but how do you access information and understand the relationship between two physical components that are found in a PDF diagram? It could be as simple as a maximum pressure recommendation, or the last maintenance report. Finding the right information and enough context to understand what you are looking at is not as simple as using Google search. You need a much richer, contextual picture to understand what is going on, what might be the problem, and how it can be solved.

From Siloed Industrial Data to a Digital Twin

At the core of the problem is what we call “contextualization”. Time series sensor values like [1.1, 1.2, 1.3, …] have little meaning without knowing from which equipment the sensor values are collected from and the wider context of the process this equipment is a part of. Also, other data like the unit of the sensor values, fault state of the equipment, nearby sensor values, connected equipment, maintenance history, and other data can be essential to interpreting the sensor values and finding the root cause of an operational issue. Contextualization is the process of establishing meaningful relationships between data sources and types to traverse and find data through a digital representation of the relationships that exists in the physical world. Through contextualization, you build what we refer to as an industrial knowledge graph. This knowledge graph is continuously evolving and spans many dimensions and different data types, from the time series values to diagrams showing the process flows to a node in, e.g., a facility 3D model and recent images from an inspection.

The diagram below illustrates a simplified version of an industrial knowledge graph of a centrifugal pump.. Depending on where you are starting to explore a problem with the centrifugal pump, you may start from the maintenance report (which shows a deviation) or maybe from the work order or the engineering diagram (e.g. Piping and Instrumentation Diagram, P&ID). If you are looking into a problem



Work Order

Start: Nov 20, 2022
Stop: Nov 23, 2022
Action: Check

Nov 23, 2022 - Maintenance Report

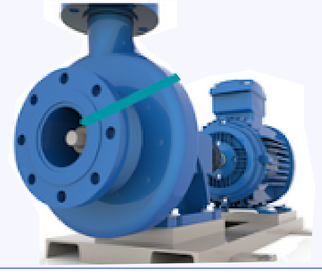
Project Name		Package No.	
Area		Date	
Unit No.		Subcon. Name	

S.NO	ITEM TO CHECK	Y	N	N/A
1	Check that the pipework and equipment are drained as per procedure	V		
2	Check that the pipework and equipment (re)installation conforms to the specifications, drawings and to the P&ID's	V		
3	Check that new gasket have been installed after pipework and equipment reinstallation	V		
4	Check that the pipework, valves, check valves are undamaged. Also check the ancillaries and any other components.		V	
5	After the (re)installation of a valve check that it operates freely.	V		
6	Check that the flow direction of the valves and ckeck-valve is in the correct sense/direction.	V		
7	Check that the valves have been correctly lubricated and that they have the correct spindle packing	V		
8	Check that all the bolts and fasteners are suitably lubricated.	V		
9	Check that after pre-commissioning is finished that the painting of piping systems and of the equipment is "touched up" to conform to the specifications	V		

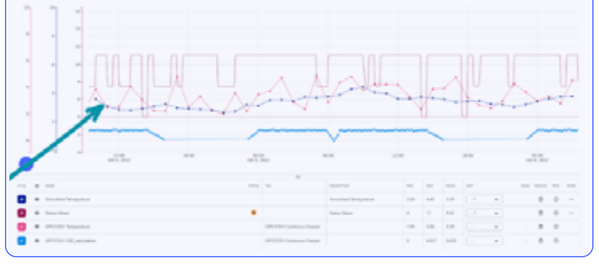
Contrifugal Pump Specification

Model	Power kW	Power HP	Inlet (Inch)	Outlet (Inch)	Max Flow (L/min)	Max Head (m)	Dim (mm)	G.W. (kg)
MGA - 1A	0.75	1	1 1/2”	1 1/2”	270	20	365x210x275	14
MHF - 5B	0.75	1	2”	2”	600	13.5	370x215x285	15
MHF - 5A	1.1	1.5	2”	2”	600	15	415x220x300	16
MHF - 5AM	1.5	2	2”	2”	600	22.2	415x210x290	24
MHF - 5AMF	1.5	2	2”	2”	600	22.5	470x270x340	26
MHF - 6C	1.1	1.5	3”	3”	1100	11.9	460x265x340	25
MHF - 6CF	1.1	1.5	3”	3”	1100	11.9	480x290x390	25
MHF - 6B	1.5	2	3”	3”	1200	14.7	430x270x375	28.5
MHF - 6BF	1.5	2	3”	3”	1200	14.7	480x270x390	28.5
MHF - 6AR	2.2	3	4”	4”	1200	18.5	505x270x370	37.5
MHF - 6ARF	2.2	3	4”	4”	1200	18.5	530x270x390	37.5
MHF - 7BR	3	4	4”	4”	1300	21	505x270x370	44
MHF - 8BRF	4	5	4”	4”	1600	22	560x270x390	48

3D Model of Pump and Winder Context



Pump Sensor Data



with a pump operations and starting out from a view where you are inspecting the time series sensor values coming from the pump, the latest work order on the pump and the maintenance report is critical to your problem solving. Since the maintenance report, the work order, and the time series sensor values are often found in separate systems, it is not a trivial task to gather all the relevant information necessary. This simple example above illustrates the importance of data contextualization across different systems.

Cognite Data Fusion® allows industrial companies to represent these real-world components, how they relate to each other and their exact location, as well as all the possible connections that are created, changed, and augmented throughout the lifespan of the components and the asset. The type of data that needs to be connected is somewhat stable, but the data itself is continuously generated by disparate systems and the data needs to be continuously contextualized to make all data accessible for problem solving.

However, there is another complexity that also comes into play. The diagram above shows some obvious connections between different data relevant to the operations, but what if the pump is replaced with a newer version that offers more sensor values? To use this new sensor, you need to add a new sensor value to the data model of this pump. Can you do this without breaking the dashboard you created to monitor operations of all similar equipment?

A continuously evolving industrial knowledge graph is the foundation of creating industrial digital twins that solve real-world problems. Industrial digital twins are powerful representations of the physical world that can help you better understand how your assets are impacting your operations. A digital twin is only as useful as what you can do with it, and there is never only one all-encompassing digital twin. Your maintenance view of a physical installation will need to be different from the operational view, which is different from the engineering view for planning and construction.

Having all your data in an industrial knowledge graph that can present different industrial digital twins adapted to your needs is important, but it is unfortunately not the only thing you need to solve operational problems with your vast amounts of industrial data. Many ambitious projects fail to try to structure data into a common model that can be used by all data consumers.. The data is there and you can find it, but can you trust the data? Do you have robust data pipelines and change management in place to ensure reliability? Are you able to evolve your digital twin representation with the life cycle of the assets or when new data is needed for more valuable insights? Have you used an industry-standard to represent your data, and can you share the data with your vendors and partners? And how do you handle the information you need that has no representation in industry standards such as CFIHOS or ISA:95?

To tackle these problems, you need to actively control, monitor, and manage the data through a governance and management process. This is a continuous process to ensure that data does not become stale and unusable, and that data is properly governed and secured.

When you treat data as a product in your organization - and proactively make it available and valuable for everyone in your organization - you are practicing DataOps, or rather Industrial DataOps. Industrial DataOps is about breaking down silos and optimizing the broad availability and usability of industrial data, specifically in asset-heavy environments.

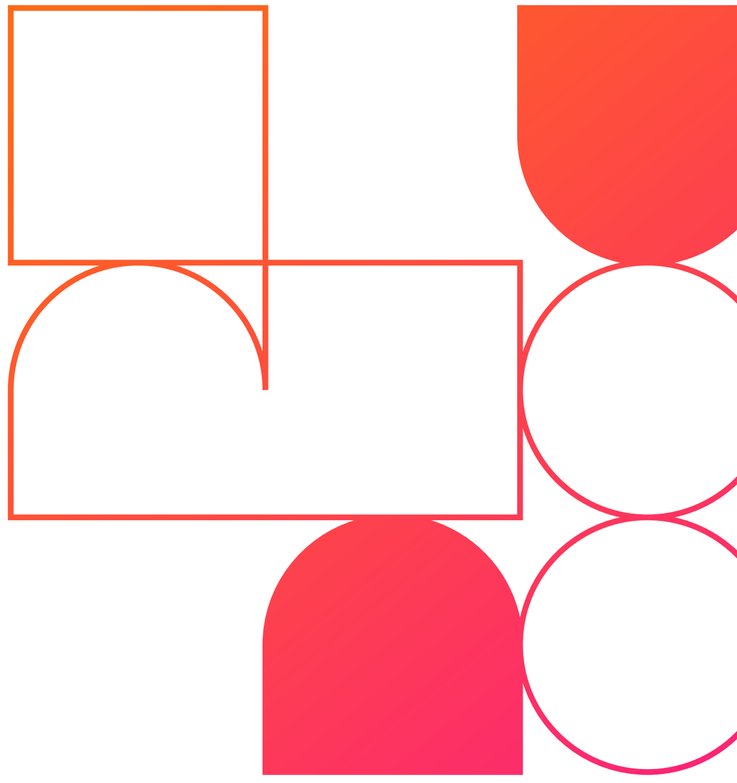
↘ The Origins of "Data as a Product"

Solving the most valuable and complex problems in the industry is particularly challenging because it requires both industrial domain experts to understand the problem and data engineering experts to provide simple access and governance of industrial data.

These two domains have traditionally not collaborated closely. And unfortunately, industrial data is not readily available directly from the sources and cannot be easily consumed by domain experts. The data may not have been cleaned, verified, approved, guaranteed, or transformed. Additionally, it may not even be understandable without operational context. All these steps are needed for domain experts to solve business problems, but they don't have the tools to complete these steps with the data on their own. Additionally, it is often challenging for domain experts to communicate their needs to software and data experts.

The often-overlooked problem of making reliable, clean data available for analysis is complex, even for a marketplace like eBay. Who bought what? To be shipped where? Is the shipping address valid? Can we see purchasing trends by region? How can we predict how big the Black Friday sales will be? Just like industrial organizations, companies like eBay, Über, and Netflix have a size and complexity that makes such questions hard to answer. To scale data management and dynamically answer new questions, early ideas of “Data as a Product” were pioneered by the aforementioned companies to allow finance, sales, and other internal operational departments to quickly answer new questions by combining streams of data coming from operations.

The internal users of the data used tools like Tableau and custom-made dashboards. The data experts who offered the data as a product built more powerful and flexible tools to cater to an increasing number of requests for running an analysis and presenting the results. The business side was the "buyers" of the data product, and the data experts packaged the data and made it self-service. This Data as a Product concept is also applicable in our complex industrial environments.



DataOps Evolves

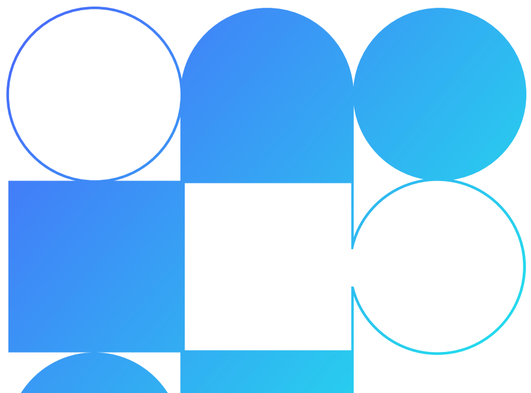
The data experts started to evolve a set of practices to continuously improve the data products and make small, incremental changes as they learned what the business needed.

For example, an error in a service handling movie recommendations could result in a loss in the data that showed how many times a certain movie was recommended per hour, per day, to which region, etc. Fixing that error could dramatically improve the data quality and thus every decision where you needed to know these specific statistics. Solving challenges like this led to the DataOps practice, named as such because DataOps merges the operations of supplying the data with the understanding of the data and how it should be improved as one feedback loop to iterate and improve quickly. In this way, DataOps does for data what DevOps does for development – merging operations and creation, for rapid feedback loops and frequent iterations to drive improvement.

Industrial Data Products Are Complex

In the traditional technology domains mentioned above, the data experts can understand the data and determine whether the data makes sense or has issues. They also partially understand the problems that the finance or operations people want to answer. In an industrial context, data is harder to control and understand.

To give an example, values from a thermometer can be in Celsius, Kelvin, or Fahrenheit, and in one context a value above 1,000 can be within an acceptable range, but in another context the value does not make sense or indicates a fault. Another example, two pieces of the same equipment can be slightly different within the same factory floor. A maintenance operation done on one piece of equipment can impact the calibration of another.



All these physical world realities lead to two major implications:

1. The interactions and iterations needed between the data expert and the domain expert are higher.

2. Determining the usefulness and quality of a specific piece of data requires a lot more context.

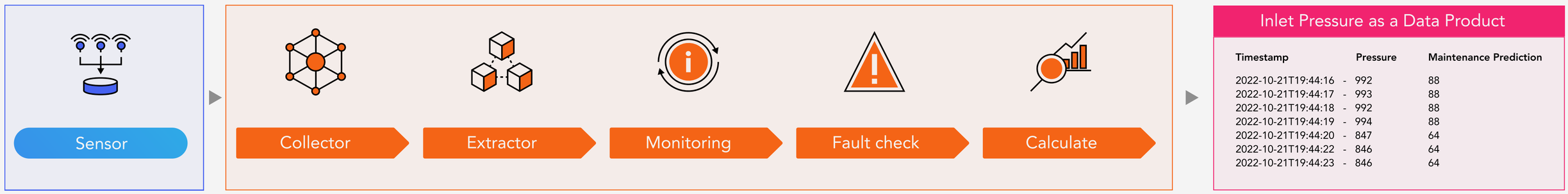
This context can be found by looking at the total spectrum of information related to the equipment you want to study. Pieces of information exist within PDF documents, databases storing work orders for maintenance, machine-generated events that have been registered, imagery data from the last inspection, the 3D model of the system, and a myriad of other data.

To be useful for solving problems, industrial data products require data from many different source systems to be connected to reflect the physical world. It must include historical and operational

data of a physical asset or system that you need to study to extract the full value in order to solve particular process problems.

Below is a simple example of a data product where the inlet pressure from the centrifugal pump is presented together with a (theoretical) calculated maintenance prediction estimating the likelihood of failure from 0 to 100. The prediction may be based on the inlet pressure, how that compares to the vendor's recommended range, the maintenance record, and other factors (to simplify, not everything that is part of the data product is shown in the diagram below). Of course, if the pump is replaced with a version with a slightly different specification, this should not mean that you have to create a new prediction model. The model should automatically pick up the latest specifications and calculate an appropriate maintenance prediction. This is easier said than done and DataOps and Data as a Product alone is not enough.

How do data products relate to the digital twin? Conceptually (and visually), you can think about live data "streaming" through the digital twin, each set of streams governed and managed so you can trust the data. Imagine operational failures lighting up in color in your 3D model or seeing live sensor values in your engineering diagram while you quickly click through to related sensor values or data. Using your data product, you run simulations and analytics and create data science models that predict upcoming failures or challenges. Then these predictions "light up areas" in the 3D model or list components where you may want to investigate maintenance or further analysis, creating insights. What glues this together is a set of data products making up the continuously updated digital twin. But how do you represent all these variations of structured data elements, and how can you ensure that you can handle the replacement of a pump without breaking your maintenance prediction?



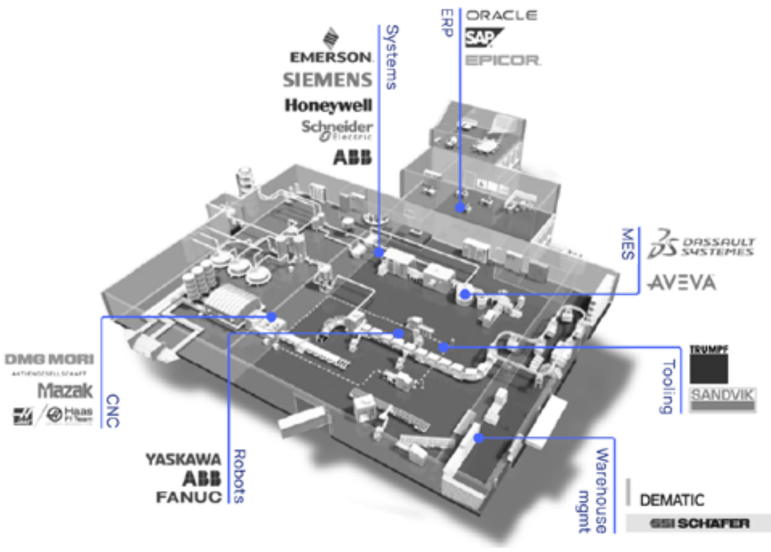
Data Models to The Rescue

Physical, industrial systems are complex to represent, and there is no single representation that applies to all all the different ways you need to consume the data. In oil and gas, drilling a well is very different from operating a well, and drilling a well offshore is very different from drilling a well onshore.

Many well bores are drilled for one well, but not all will be put into production, and once in production, you need operational data, but data gathered as part of drilling may still be relevant. In manufacturing, understanding how an individual piece of equipment is performing is very different from measuring how materials flow through the production line or process.

. Individually, all pieces of equipment perform within tolerable levels, but the final product quality as the raw materials move through the process may be off-specification. The complexity tof representing equipment, assets, and processes is true for refining and chemical, food and beverage, metals and mining, and most asset-heavy industries.

The solution to this complexity is standardizing on a set of data models containing subsets of the same data, allowing you to view the data from different perspectives and then add other additional data depending on your goals. Although there are industry standard data models, most companies have their own adaptations or extensions. Addtionally, they would also benefit from sharing specific subsets of data with their suppliers. Reusability of subsets of the data in a data model thus becomes an issue because, for example, when you refer to a pressure value on a pump, it may be called different things in different data models (and languages). Still, you want to accurately and consistently refer to the same thing



Domain Data Models

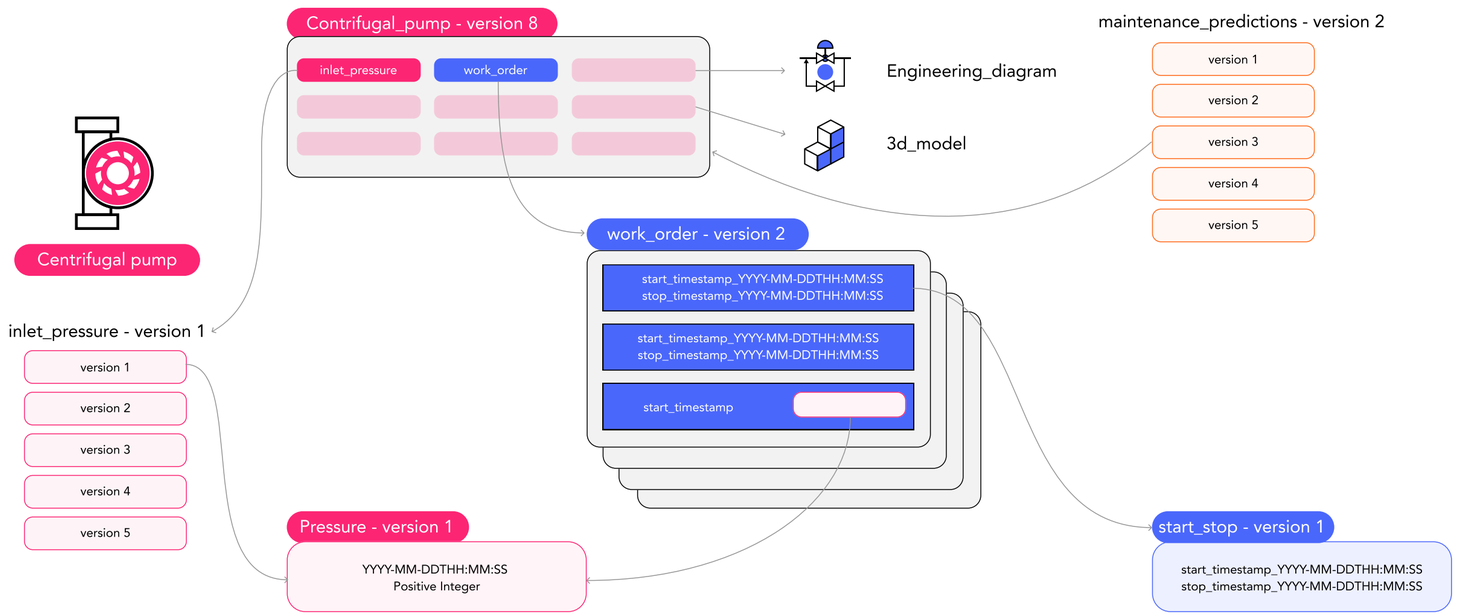
In Cognite Data Fusion®, predefined, configurable domain data models are available for every customer. These domain data models are best practice data models for the industry, so while an oil & gas customer may want to use the asset hierarchy (and the CFIHOS standard), a manufacturing customer can follow ISA:95, and a power and utilities customer can use CIM. Other domain and purpose-oriented data models (e.g., ISO 14224 Collection and exchange of reliability and maintenance data for equipment) can also be added to the same project, thus offering a different view of some or all of the data.

A domain data model is built up the way you would expect: you start with primitive types or properties like integers, floats, and strings, but also include references to Cognite Data Fusion® industrial data types like 3D nodes, time series, or other customer-defined specified types. You can see this data as classic database tables. However, you can group some properties, like a start date and a stop date, and reuse this group of properties across your data models. Or you can use the definition of a "work order" as a foundational type that you can reuse in other data models. This reusability is also extended with versioning in such a way that your data products can be consumed while also evolving the data models. This protects you from breaking running dashboards, calculations, and predictions.

The diagram to the right shows this reusability. Even if you update your contrifugal_pump model to version 9, the maintenance_predictions model will continue to reference version 8. Since inlet_pressure uses pressure - version 1, the maintenance_predictions model can use pressure - version 1 in it's calculations. You can thus continue to use the maintenance_predictions in your dashboard to monitor your pump without fearing that it will break.

The data models shown in the diagram below are members of a bigger domain data model, and a complete domain data model that represents all needed data aspects of a physical system is the data foundation for an industrial digital twin. Cognite Data Fusion® supports 2D and 3D models to visualize each digital twin but represents the underlying data in a meaningful way for you to use that data. The shown example is simple but still illustrates the complexity. But what if you want milliseconds in your start_stop model in your work_orders? All your work orders up to this point will have recorded only seconds, and all other applications expect this format. You need to support new uses of start_stop with milliseconds while ensuring that old uses of the seconds-only model do not break. The classic approach from the IT world is to upgrade everything to use milliseconds and migrate the old data to use 00 milliseconds. In the industrial world, such changes happen continuously, and since you need your non-IT domain experts to be able to build dashboards, solutions, and models, this old migration model does not work.

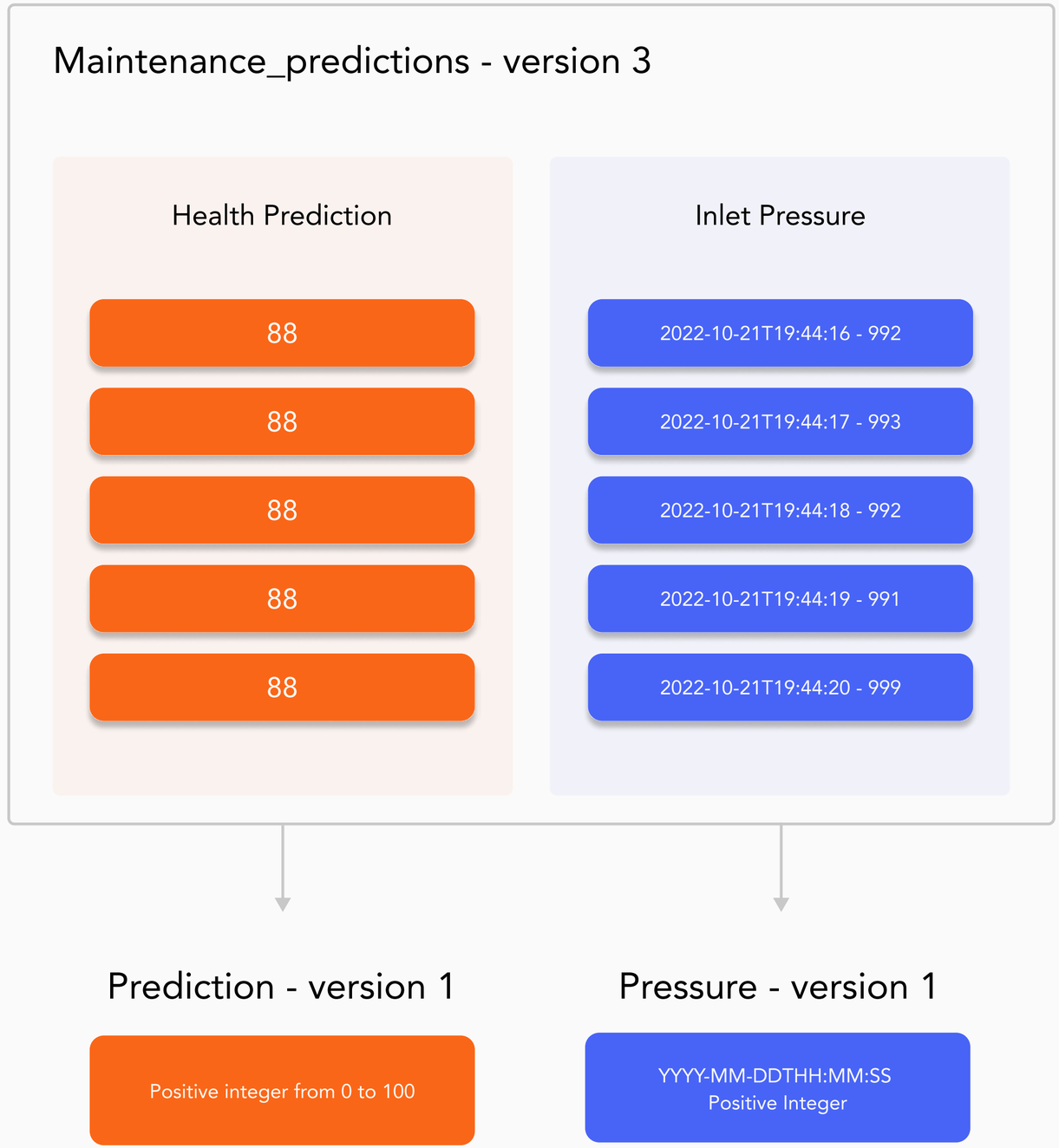
Manage versioning with data contextualized into a domain data model



Consuming Data

As an application developer or data scientist building a dashboard to visualize data, you need to find the necessary data to analyze and solve your problem. And to iterate quickly, you must be able to consume data in a manageable way.

Let's say you can find the data using the domain data models that have the data represented in full context. However, when you have found the data, you realize your work would be greatly simplified if you could create your own view – a subset of the larger data model – creating a simpler structure where only the relevant data elements are visible. The maintenance_predictions model from the previous section is an example of such a view. Maybe in addition to creating your 0-100 health prediction, you also want to access the specific values that were used to calculate the prediction. Instead of navigating into the bigger contrifugal_pump data model to find that data for a specific timestamp, you can expose the pressure values directly from your maintenance_predictions data model.



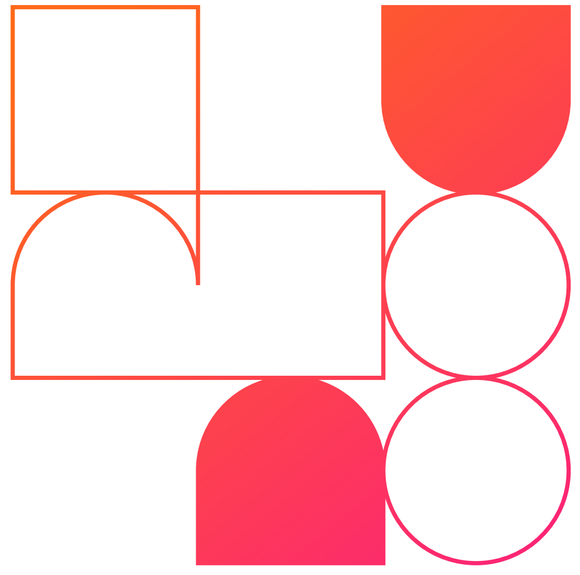
Cognite Data Fusion® optimizes for the consumption of data. Part of this optimization occurs by storing the data based on typical consumption patterns. Further optimization is achieved through caching, pre-aggregations and data replication to ensure that data is structured in a way that will allow incredibly performant queries and data retrieval. While modern databases have similar optimizations, industrial data like engineering diagrams, time series, and 3D models have special usage patterns that are harder to optimize and usually not supported. Also, although you can use so-called “materialized views” in a database to speed up queries of that data, your typical domain expert or a data scientist will not be able to operate on such a granular database level. You thus need a system that allows consumption optimization to happen invisible to the builders and consumers of the data.

A dashboard built by a data scientist or an application that monitors operations and shows predictions from a running data science model will likely need data criss-crossing types and data models (as shown in the maintenance_predictions example). To simplify building and reasoning around the data for the builder and to allow using meaningful names on the data for the particular problem to be solved, Cognite Data Fusion® supports solution data models. Solution data models offer a mirror into the data specifically tailored to the solution or the application, and can evolve with new versions of the solution or application.

Of course, the domain data models may evolve and be versioned as well, but the solution data models can evolve on a separate cadence. Solution models also allow for the scaled rollout of applications that are built on top of Cognite Data Fusion®.

The solution model acts as a stable data interface for the application, and underlying domain variations across sites and customers can be compensated for by changing the mapping of data into the solution data model.

This functionality allows Cognite Data Fusion® to support scaling of applications and dashboards across factories and assets with similar problems, but where data is structured and named differently.



Onboarding Data

Up to this point, we have assumed that data is already loaded into the domain data models, structured and put into context with rich connections across the data. However, onboarding the data into a structure that can be understood and consumed is non-trivial and typically the task of the data engineer.

Data you need to solve complex industrial problems come from various source systems. Some are traditional IT systems from large vendors like SAP, others are operational and local, or even only accessible locally on the equipment itself. Some data may have been consolidated into a data warehouse and may have lost some of its original context. Either way, you cannot assume that the data is in a state where it can be easily used. If you have multiple systems with the same type of data, which one is authoritative? Generally, companies have come further in consolidating IT data into data warehouses than what they have achieved for their operational data, 3D models, and imagery data that typically is scattered across operational systems.

All this means that you need a system to support onboarding, cleaning, and contextualizing the data in such a way that you can easily and quickly find the information you are looking for, regardless of its location or type (unstructured documents, scanned papers, engineering diagrams, vendor specifications, work orders, or maintenance reports).

The primary tool for capturing as much of the original source context and semantics as possible is through the use of source data models. If the world was perfect, you could load your data into the domain data model(s) directly through a data pipeline e.g., using OPC-UA that supports source model semantics. The typical ETL (Extract, Transform, Load) pipeline approach is built around the idea of a start state and an "ideal" end state. Unfortunately, industrial data is messy, and sometimes you need to look at the raw data to understand what might have happened, other times you want the cleaned data that removes spikes or flattens out and merges gaps.

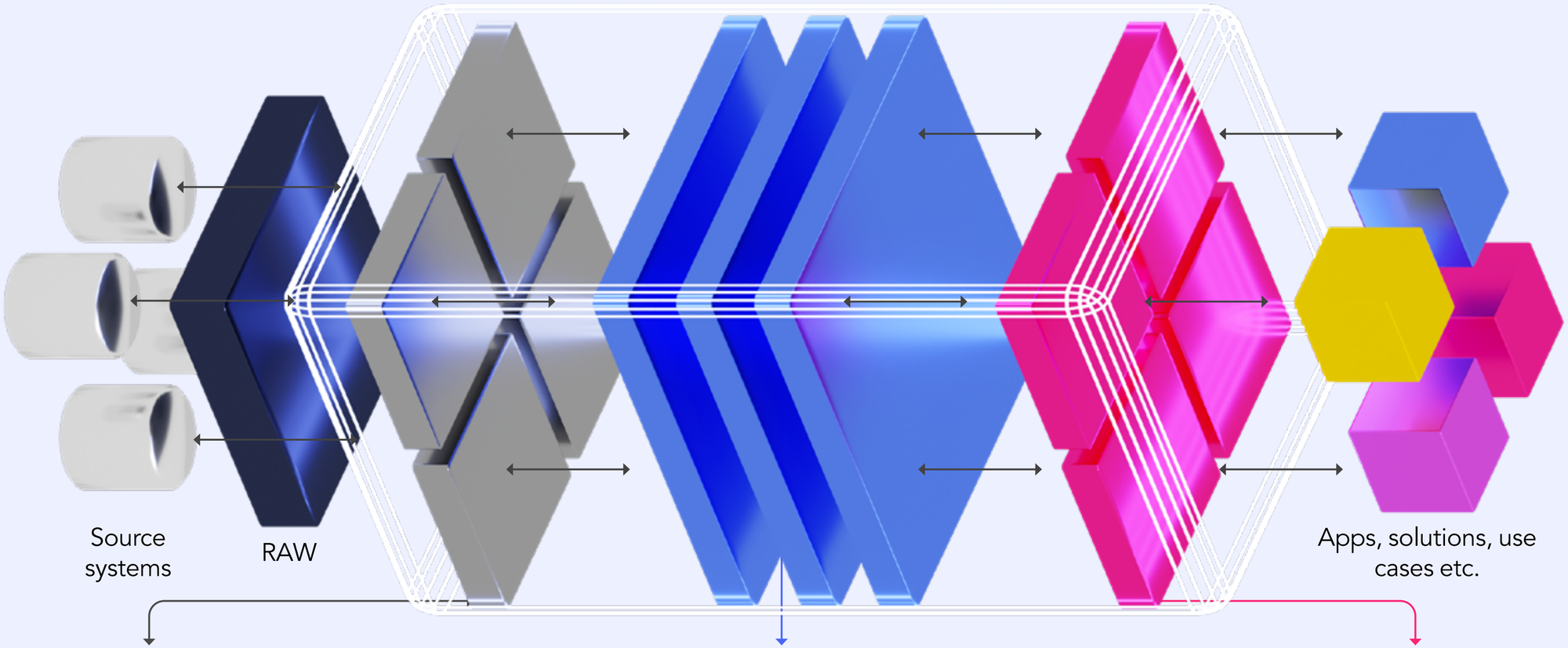
As a data engineer, you thus build your source data model to reflect each source systems' data you want to bring into Cognite Data Fusion®. You do this by first bringing the data from the source system into the staging areas. Depending on the data, you may want to use a NoSQL staging area with no structure, a file storage to store a CSV file, or a more traditional SQL schema-based staging area. From there, you can explore the data and verify your source data model, figure out where you have uniqueness and profile the data.

Once you have a clear view of how you want to bring the source data into the source data models, you need to look at how to map the source data model onto your domain data models. Cognite Data Fusion® comes with predefined mappings for common source systems that can be tweaked to fit the specific implementation. The below diagram illustrates this data onboarding process. (For simplicity, it does not show the contextualization and transformation processes that need to be running continuously and that are set up as part of the data onboarding.)

Very often, there is a gap between the physical world and your digital representations due to incompleteness, physical changes, and errors. Cognite Data Fusion's contextualization services can be used in this process to connect and build your industrial knowledge graph and allow

you to expand your domain data model using contextualized documents, diagrams, 3D models, and on-boarded data.

Below the process is shown as linear, but in reality, onboarding, structuring, and consuming industrial data needs to be an iterative process. Cognite Data Fusion® thus allows updates and changes to the entire process and through this establish sound DataOps practices where the experts on data, data engineers and architects can collaborate with the experts on interpreting the data, the domain engineers and data scientists.



Source data model

Asset hierarchy from SAP ■ OPC from Kepware ■ BOP/BOM from MES

Liberate data in source systems by having them modeled and queryable in Cognite Data Fusion® through the same API interfaces.

Domain data model

Asset or process centric ■ Follow standards (e.g. OPC, ISA95) ■ Custom semantic types (CNC)

Domain data models are optimized for reuse between application developers, data scientists and solution developers. We offer a set of industry standard OOTB data models that can be extended.

Solution data model

Data model targeting data consumption for use cases

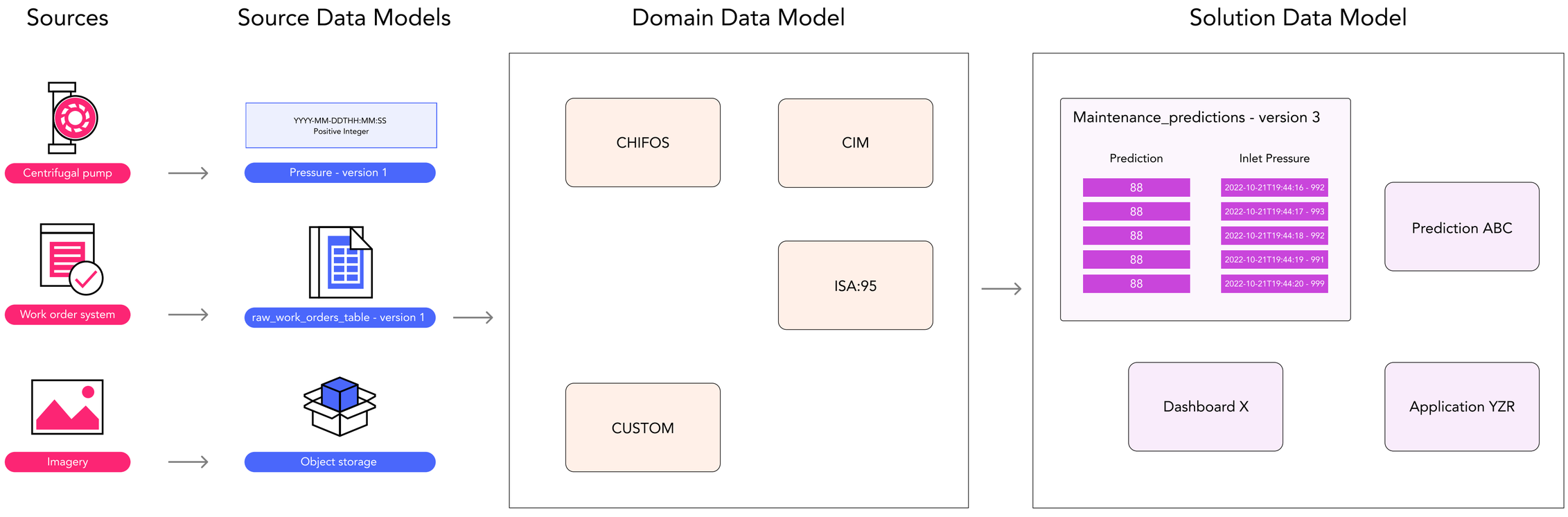
Optimized data models for applications, data scientists and other solutions. We provide a GraphQL interface with auto generated SDKs for minimal code. This provides Backend as a Service (BaaS) to reduce cost of development and maintenance of solutions.

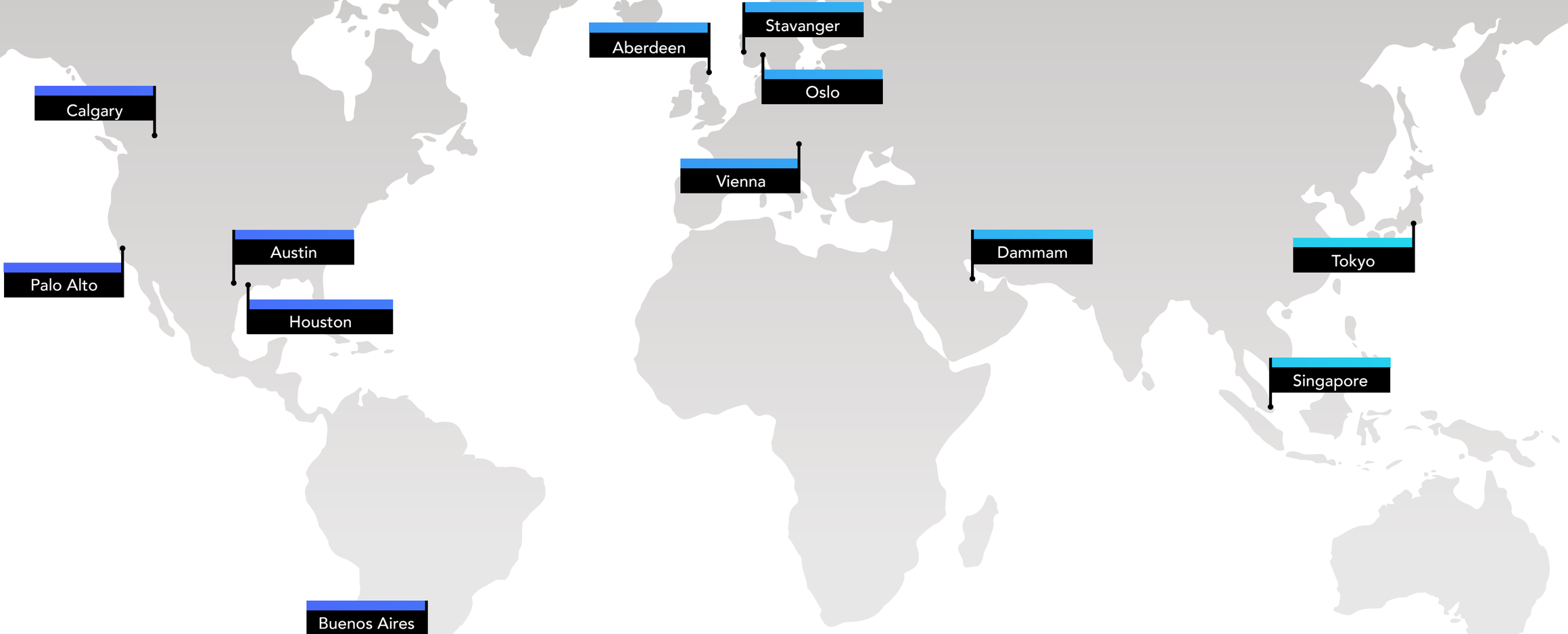
Cognite Data Fusion®'s Data Modeling Architecture

Putting it all together, the below diagram illustrates the high-level view of how data originating from existing OT, IT and engineering source systems are on-boarded into Cognite Data Fusion® through the use of source data models that are then stored in the organization's domain data models such as CHIFOS or ISA:95.

These domain data models are the reference models, or digital twin models, used throughout the entire organization for data discovery and communication. Application developers and data scientists can build solution data models for their specific needs that will make an organized view of the relevant data from the domain data model(s).

Cognite Data Fusion® has support for a rich set of industrial data types, and while the most common domain data model has been the asset hierarchy, Cognite's new digital twin modeling capabilities are expanding the ease and speed at which new data models are being developed. If you are interested in learning more about how Cognite is enabling industrial solution development, news on releases and upcoming changes can be found on Cognite Hub!





All About Cognite

Cognite is a global industrial SaaS company that supports the full-scale digital transformation of asset-heavy industries around the world. Our core Industrial DataOps platform, **Cognite Data Fusion®**, enables data and domain users to collaborate to quickly and safely develop, operationalize, and scale industrial AI solutions and applications.

Cognite Data Fusion® codifies industrial domain knowledge into software that fits into your existing ecosystem and enables scale from proofs of concepts to truly data-driven operations to deliver both profitability and sustainability.

Visit us at www.cognite.com and follow us on Twitter or LinkedIn.

Contact us

A unique global team that combines software with deep industry expertise

10+ International Olympiad in Informatics medalists
15% Ph.D.s

Microsoft Google

Software and data science

COGNITE

Industry expertise

AkerBP SIEMENS ABB
Schlumberger ConocoPhillips

BCG Deloitte. McKinsey & Company
pwc

COGNITE