OMP Use Case Collection: Condition Monitoring

Published
May 2022

www.open-manufacturing.org

info@open-manufacturing.org

Legal Disclaimers

"THESE MATERIALS ARE PROVIDED "AS IS." The parties expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title related to the materials. The entire risk as to implementing or otherwise using the materials is assumed by the implementer and user. IN NO EVENT WILL THE PARTIES BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS DELIVERABLE OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER MEMBER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

ACKNOWLEDGMENTS

This document is a work product of the Open Manufacturing Platform - Manufacturing Reference Architecture Working Group, chaired by Martina Pickhardt (Microsoft) and co-chaired by Nils Jensen (PwC).

AUTHORS

Lukas Birn	Capgemini
Nils Jensen	PricewaterhouseCoopers (PwC)
Ashish Joshi	Cognizant
Ivan Lai	Delta Electronics
Monica McDonnell	Teradata
Klaus Merk	Microsoft
Martina Pickhardt	Microsoft

Table of Contents

1. Introduction

Context

According to the OMP Whitepaper "Introduction to the OMP Manufacturing Reference Architecture" this publication contributes to the collection of dedicated use case whitepapers that focuses on particular manufacturing challenges.

Intended Audience

This document is primarily interesting for executors of the use case, that is, personnel who will plan, design, test, roll out and maintain the solution, but also benefiters of it, which are people who will work with the implemented solution.

For example, the executor group consists of solution/Cloud architects, IT/OT developers, data engineers/analysts, digital project managers, and process experts. This group gains insights into the design principles, blueprints, and rollout guidance of the different building parts for this use case.

The benefiter group consists mainly of plant and shop floor personnel who want to optimize their processes and reduce costs via downtime reduction and pro-active process parameter adaption. This group includes but is not limited to plant leaders, production planners, process/quality/maintenance engineers, and machine operators. Understanding the benefits, gaining new insights, and creating new ideas around this use case are the primary gains of these roles on the shop floor.

Approach

With references to Arc42, this paper describes the relevant requirements and the driving forces that software architects and development teams must consider for a specific use case. These include

- underlying business goals
- essential features and functional requirements for the system
- quality goals for the architecture
- relevant stakeholders and their expectations

General Information

Use case ID	**MRA UC-1 Condition Monitoring**
Version	**1.0**
Related to:	-

2. Introduction and Goals of the Use Case

Wikipedia defines condition monitoring (CM) as follows:

> *"**Condition monitoring** (colloquially, **CM**) is the process of monitoring a parameter of condition **in machinery** (vibration, temperature, etc.), in order to identify a significant change which is indicative of a developing fault"*

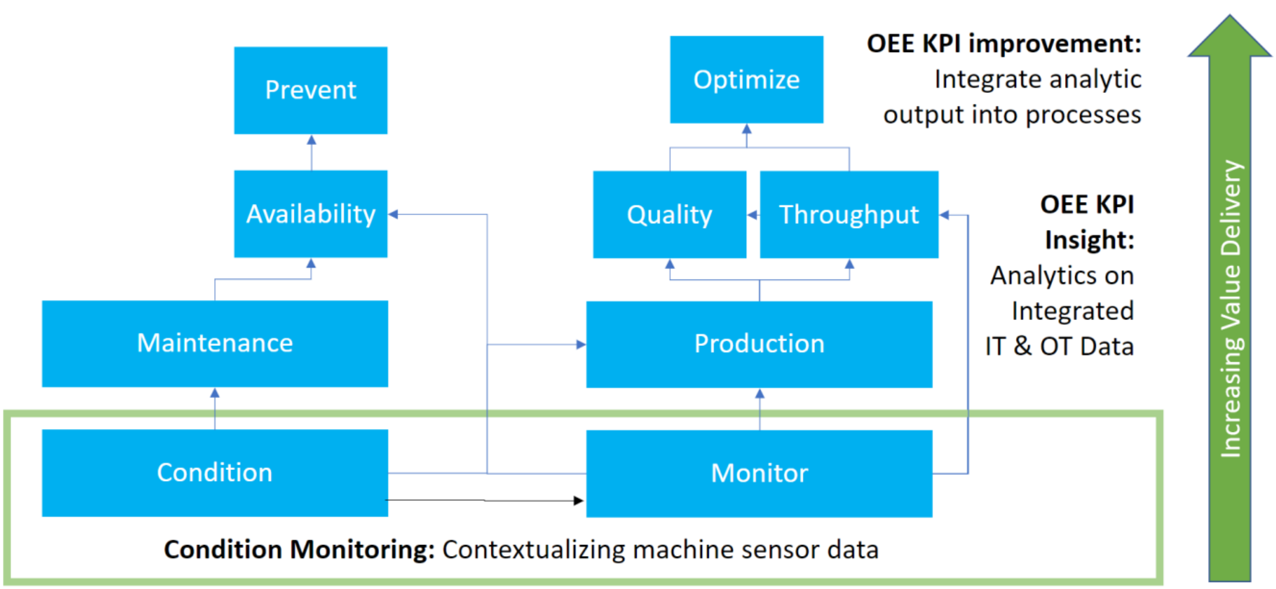


Fig 1: Demarcation between condition monitoring and other use cases.

Condition monitoring is specific to a single machine, capturing data from the machine to assess its condition.

The scope of the condition monitoring system *(as shown in Fig 1)* is limited to the collection, processing, and end-user consumption of machine sensor data. Although the condition of the machine is analyzed by using mainly data from the machine itself, other external data might also be used to determine and calculate the condition of the machine.

With this definition in mind, the goal of condition monitoring can be considered to **identify a change in the state of the machine that is significant in the context of the manufacturing process**. Depending on the manufacturing process, machines may be monitored for trends and/or specific values exceeding thresholds. The actual values and signals monitored can also vary and will be dependent on which characteristics of the machine are the most contextually significant.

Measurements of interest can include the level of vibration, imbalances in the electrical supply, lubricant health (e.g., Spectrographic oil analysis), unusual amounts of heat or sound, electrical elements of machinery, etc.

Problem Statement and Challenges

Condition Monitoring (CM) is the first step to understanding the relationship between the state of the machine and the impact on Overall Equipment Efficiency (OEE) metrics. It acts as a first step in the journey of running operations more efficiently and, as such, can be considered a 'foundational' but necessary use case.

Without monitoring the condition of a machine, the risk for failure or unplanned downtime of the machine is very high and will lead to additional production costs and other subsequent costs, like repairs and maintenance. Additionally, it will not be possible to reach the ambitious productivity improvement goals of Industry 4.0.

Context:

*OEE is a very important metric, which itself is derived from three key metrics: **Availability**, **Performance**, and **Quality**. We find that the major contributing factor making any or all the main or sub-metrics degrade is the state of the equipment used in the process steps.*

Examples:

- Machine failures or wrong/missed setup and adjustments can impact availability.
- Machines running with sub-optimal speeds or stopping in between running hours can impact performance.
- Quality can be impacted due to sub-optimal output because of process defects induced due to machine state and condition. One example is reduced speed or the machine deviating from set points because of faults or deviations in (raw) materials and processing aids.

Machine failures leading to unplanned downtimes have been one of the largest causes of loss of productivity, increase in delays, and loss of revenue. Traditionally this was solved by either doing maintenance more frequently (which has a tradeoff of increased planned downtimes as equipment must be taken offline), letting the machine run till a problem appears (which risks the overall life of the machine and its parts) or defining maintenance intervals based on experience (which still has the potential of missing machine failures in between the maintenance cycles).

Deploying CM comes with its own set of challenges that need to be effectively managed to gain the benefits. Some of the challenges are:

- There are various machines of different ages in operation on the factory floor in most cases. There are usually a mixture of "brownfield" (old machines) and "greenfield" (new machines). Along a production line, the first challenge is how to get the raw data from different machine classes as automatically as possible, which will help assess the current machine status. Many older machines do not easily provide access to this data or may require additional sensors to be installed. The IoT connectivity working group has published whitepapers specifically on the topic of connecting machinery, which can be read on the OMP Website.
- Due to legacy machines in place, deploying CM can involve high upfront costs as it requires installation/retrofit of additional condition monitoring sensors and connectors.
- In CM, the accuracy and performance of sensors and the data coming out of the machine also depend on operating conditions. There may be chances that the sensors may get damaged or not work optimally over time. Hence it is crucial that this aspect of the drift also be monitored and corrected
- Some of the CM sensors or the machines directly have the potential to generate data at a high frequency. Hence an effective strategy is needed to handle, process, store, and analyze this data at the edge and cloud.
- Effectively gaining knowledge from raw data, i.e., putting the raw data into context and thus using it for further analytical processing, needs a scalable and robust platform architecture. The semantic working group has published a whitepaper specifically on the topic of bringing context to the data, which can be read on the OMP Website.

In general, use cases should also lead to an organizational change management process to draw out the benefits of these initiatives. Hence effective training of users' needs to be planned.

Stakeholder

These stakeholders should be involved or participate in the condition monitoring process:

Role	Expectation
Production Line Expert	• Able to monitor and analyze the parts or monitor objects of the production line
Industrial & Process Engineers	• Enable machine and sensor connectivity to gateways • Able to capture meaningful data sets from a machine (or set of machines) in a consistent manner. These data sets can then form the basis of more advanced analytics, which will help improve production output. • Manage data acquisitions and develop routines to classify • Investigate and research sensing technology.
Maintenance Engineers	• Able to capture meaningful data sets from a machine (or set of machines) in a consistent manner with a specific focus on machine performance. The data sets can form the basis of more advanced analytics that will help reduce downtime (planned and unplanned) by fine-tuning maintenance practices.
IT Department	• Able to seamlessly onboard new sensors and machines on the platform with the least effort. • Manage the edge and cloud infrastructure centrally, helping reduce operational costs. • Able to quickly scale setup as the volume of assets and data grow to help optimize the TCO. • Develop and validate algorithms for the detection of specific events. • Validate the security requirements and conformance • Deploy Condition Monitoring use case-specific artifacts covering dashboards, rules, configurations, etc. • Enable gateway connectivity to the platform
Quality Engineer	• Design, prepare and run tests for CM relevant events to acquire data. • Define quality goals • Validate the data flow and data correctness and check for failure scenarios.

Production Planner	• making sure that everything is ready to create finished products according to the schedule specified. • Get noticed of any unusual machine activity as early as possible • Address issues when they arise, aiming for minimum disruption • Obtain output information (number of finished products, percentage of defectives, etc.) • Prepare and submit status and performance reports
Production Worker	• Get noticed of any unusual machine activity as early as possible • Receives directions on how to act to avoid a machine failure or reboot if necessary.
Data Scientist / Analyst	• Draw insights from data and so have a real stake in the quality & frequency of data collection.

OMP Guidance & Recommendations for best practices

OMP Member Siemens Healthineers:

The challenge is indeed to handle various machines. The combination of green and brownfield connectors is challenging. It's from our perspective even more important to understand which machines are currently in use, for which purpose, and which data they provide to which system. Going forward, we are recommending defining a strategy for how to connect machines. These machines should be connected to a single source of truth to reduce further variations after the initial machine connection. Machines shall be connected as much as possible by using a standard approach.

OMP Member ZF:

- Define in advance few but meaningful and preferably scalable standard signals
- Connect one machine after the other
- Close coordination with all parties involved with maintenance as lead
- Don't underestimate efforts and complexity

Quality Goals for the CM-Architecture

Quality goals help steer the planning and execution of the use case implementation in a sustainable and scalable direction. These act as guiding principles towards creating the proper evaluation criteria for accepting an architecture covering various facets critical to a use case. While there can be more goals that can be aligned, below are some of the key goals which are important.

Goals	Description
Functional Specification Compliance	Ensuring that the architecture complies with specifications of the core use case business requirements. For example: • The system should be able to log relevant data and report data and faults. • The system is able to monitor the machine, ensuring that it is running in safe, optimal operating ranges. • It is able to identify patterns for any impending issues leading to possible breakdown. • Help in the reduction of periodic testing to validate the correct functioning of the machine.
Reliability	Ensuring that the captured, processed, analyzed, and reported data can be trusted. This also mandates that the underlying infrastructure and architecture which is enabling this is also reliable to ensure that key information is not lost and there are necessary levers in place to identify anomalies and either proactively report them or mitigate them. For example: • The sensor and connectivity setup should be industry standard depending upon the operating conditions. • The system should be able to identify gaps in data and either fill those gaps or report the conditions. • The system should have a mechanism to check the quality of data. • The system should be able to recover from faults and failures, minimizing disruptions.
Safety & Security	Ensuring that the architecture brings in components that help capture the CM data **non-intrusively and does not impact the functioning and safety of the machine** while also ensuring the security of data (at rest and in motion) and the infrastructure which is enabling connectivity to the machine**.** For example, the edge infrastructure should be compliant with the working operating conditions (e.g., IP67, ATEX, etc.), and the connectivity to the data source should be secured.
Scalability	Ensuring that the architecture supports easy **onboarding of an additional number or variety of machines and sensors with the least effort** and no changes to the architecture or solution. It should also

	support easy scale up to cover shifts in data frequency and storage requirements. The architecture needs to ensure that the scale-up does not exponentially increase the maintenance and operating cost of the solution.

OMP Guidance & Recommendations for best practices
OMP Member Siemens Healthineers: From a functional point of view, there is at least the possibility to ensure quality by using a machine connection to control the activities of a machine to avoid errors. Other than that, it's again related to the comment mentioned above; there should be a clear strategy to cover all these points. Only with this strategy might it be possible to reach the goals. **OMP Member ZF:** The issue of infrastructure and security is a big one and should be carefully clarified and planned in advance. Many companies find it very difficult to upload machine data anywhere or to a cloud. This must receive appropriate attention.

3. Architecture Constraints

Designing and implementing a software system is a lot like constructing a building. If the foundation is not solid, structural problems can undermine the integrity and function of the building. Therefore, it is recommended to define a well-architected framework for your manufacturing platform. This framework is a set of guiding tenets that can be used to improve the quality of the diverse workloads. These architectural constraints define the characteristics and boundaries that an architecture must have to fit a particular context. Appreciating and being aware of these constraints helps to drive the right decisions towards defining an architecture that meets the business and quality goals.

Following are the key constraints to be considered in the broader architecture scope and do not focus exclusively on the "condition monitoring" use case.

- **Cost Optimization** needs to provide principles for balancing business goals with budget justification to create a cost-effective use case implementation. It is about looking at ways to reduce unnecessary expenses and improve operational efficiencies.
- **Organizational Policies** drive some of the key architectural decisions, for example, designing a system that aligns with the company's IT security guidelines, or a mandate on having an on-prem or hybrid solution to manage regulatory requirements.

- **Security** is an important aspect of any architecture. It provides confidentiality, integrity, and availability assurances against deliberate attacks and abuse of data and systems. Losing assurances can negatively impact your business operations and revenue, and the organization's reputation. It is important to correctly define the security boundary and controls (zero trust) aligning to the business context.
- **Operational Excellence** helps in aligning designed processes and roadmap, taking into account the organizational maturity towards maintainability of the system. The aim is to keep an application running and in production. Deployments must be reliable and predictable. Automated deployments (zero-touch) reduce the chance of human error. Fast and routine deployment processes won't slow down the release of new features or bug fixes. Equally important is the ability to quickly roll back or roll forward if an update has problems.
- **Governmental and Regulatory Policies** can have a strong influence on the architecture as they can mandate some of the controls to be placed in, for example, data exchange between companies, inter-country/continent data flow, and deployment decisions.
- **Condition and maturity of the target environment** play an important role when defining the target architecture as it has the potential to sometimes restrict some of the decisions or demand one to find out ways to overcome limitations. For example, the legacy OT systems in place can hinder the seamless extraction of data. Similarly, working environments (e.g., high heat environment, explosive conditions) can mandate the use of the required infrastructure matching that, and the availability of network and bandwidth at the plant can also influence how the data is processed and transferred.
- **Data Frequency** of the data being captured and processed has a strong influence on some of the architecture components, and it is necessary to ensure that the system is capable of handling high-frequency data if required.
- **Data Management** and its availability to internal and external stakeholders is crucial when architecting such a solution. In today's context of 'as-a-service' models and data monetization, sharing of data is gaining prominence. This may require the manufacturer to share the equipment data with the OEM. If these scenarios are to be fulfilled with the architecture, the necessary capabilities and controls should be planned for.
- **Repeatability** indicates the level of flexibility and extensibility that the architecture needs to support, making it easily cater to various scenarios. For example, on one end, the solution is designed to cater to only a single plant or use case, while on the other end, it may be required to cater to various manufacturing setups, processes, and use cases. Both these extreme scenarios demand the architecture to be designed aligning to the strategy so that it is cost optimized.
- **Performance** characteristics influence the architecture decisions in terms of the scale that the solution is supposed to handle. The scale could be in terms of the number of endpoints generating data, the amount of data being generated, the speed of data, the scale of operations, the variety of scenarios, data receipts, and the future roadmap. It is important to outline the performance requirements against which the architecture should be built.
- **Reliability** drives some of the important production operations affecting aspects to be managed in the architecture. On the one hand, it ensures that the architecture enables the solution to augment existing operations and not affect its functioning, while on the other hand, it also ensures the safety of the production operations. For example, if the solution enables a control loop functionality, then in the event

of a failure/system error, or in case the actual machine is performing a critical task, it should gracefully handle this scenario and ensure that it does not impact the actual production processes.

- **Production relevant** matrix helps in defining the scope. While laying down the architecture, it is important to define and segregate the architectural components which may be production critical versus those which may be nice to have. This distinction can help drive the architecture choices and make them more aligned to the critical aspect of the scenario versus the complexity and cost of the target solution. For example, the solution which is supposed to handle critical production assets may have to have architectural components which need to ensure data reliability and high uptime, while for non-critical assets (whose impact on production is not much), the solution can be simplified.

4. Use Case Solution Strategy

A solution strategy summarizes and explains the main requirements and the core decisions that influence the detailed solution architecture. The solution strategy needs to follow the architectural constraints and defines the fundamental solution concept and how to meet the quality goals.

Functional Capabilities

Firstly, a selection of monitoring aspects for the desired solution needs to be defined. Each production line might have different monitoring requirements and aspects to consider. Each of these categories can be seen independently and as building blocks to establish a comprehensive monitoring solution.

- **Vibration Analysis** includes various techniques, such as shock pulse and broadband vibration analysis.
- **Acoustic Analysis** can be sorted into the broad areas of sonic and ultrasonic categories.
- **Temperature Measurements** include simple temperature measurements, as well as more advanced techniques that involve passive or active thermography.
- **Lubricant / Oil Analysis** relies on testing lubricants and other fluids.
- **Motor Circuit Analysis** is used to determine the health of a motor.
- **Electrical Monitoring** includes tests for pulse and frequency response, capacitance, and resistance.
- **Electromagnetic Measurement** analyses distortions in the magnetic field.
- **Laser Interferometry** helps to identify defects both on surface and subsurface.

To enable a condition monitoring use case and achieve the "**Functional Specification Compliance**" quality goal (see Chapter 2), the following functional capabilities are required. All listed building blocks are required for several use cases enabled through a manufacturing platform. Therefore, it is essential to design them as generic capabilities to meet the architectural constraint of **"Repeatability".**

- **Monitoring Hierarchy Management:** This capability needs to enable the import of the existing machine hierarchy to adapt monitoring models and rules. The hierarchy management is enabled by a digital twin data structure, which also needs capabilities to ease the process and sync the physical truth with the digital truth, done by a process engineer. Furthermore, this capability needs to provide the possibility to create and manage custom monitoring objects configured by maintenance engineers.

- **Condition Monitoring Rule & Monitoring Model Library:** A customizable repository for defined monitoring models across different production lines and production sites to ease the process of create, re-use, adapt and deploy models and monitoring rules allow configuring and interfacing with existing data sources like historians. The catalog needs to provide and visualize necessary descriptive data for maintenance engineers.
- **Condition Monitoring Rule & Alert Execution:** Based on the ingested data, the solution needs to identify and report equipment conditions and measurable faults of the deployed condition monitoring models and rules. Dependent on the specific scenarios, the capability needs to be applied to the hot, warm, and cold paths. All alerts should be stored for later analysis and model optimizations. Required data might be alert, timestamp, alert initiation data set, and intensity.
- **Messaging & Notification to External Systems:** With the execution of an alert, an interface with external systems is required. The main intention of this capability is to provide an open and flexible environment to report equipment condition alarms to any users and external systems according to the configured condition monitoring rules. In some scenarios, alert acknowledgment might be required, which needs to be stored for process compliance and later analysis.
- **Data Visualization & Root Cause Analysis:** These interface with required data sources based on the scenario for hot, warm, and cold path reporting requirements. They also enable process and maintenance engineers to build customizable reports by selecting data sources and data sets, forming a data catalog, and enabling configurable views to visualize machine-level condition results, alarms, and KPIs.
- **Management KPI Reporting:** This provides daily, weekly, and monthly equipment condition and effectiveness KPIs to be used for maintenance planning and prioritization and enables the production planer to include data from additional sources via a data catalog.

OMP Guidance & Recommendations for best practices
OMP Member Siemens Healthineers: For us, it’s also important to offer the teams standardized but customizable dashboards to monitor the relevant values per machine. Going forward, it may be required to not only monitor the data which is relevant on first thought but also correlate further data from a machine with failures, etc. This may need support from deep learning techniques.

Design Strategy Considerations

Prior to starting the implementation, several areas of concern need to be assessed.

- Does the existing **technology stack** enable the required capabilities, and if not, what will be the chosen technology stack, and how does it influence the existing technical environment?
- What are the **success parameters** for each of the capabilities and the necessary implementation parameters? Furthermore, strategies for top-level **decomposition** of a capability and the approach to implementing capabilities to re-usable **patterns** must be developed.
- The more important a use case is, the more important become the decision decisions around **business continuity**. Hardware failure or even natural disasters, or data corruption might not be allowed to disrupt or stop the core processes.
- Data should become an asset, needs to be shared across the enterprise, and be easy to consume. A **Data as a Service** strategy should be considered. In fact, a service strategy should, in general, be considered. Therefore, building every capability as an internal and external consumable **service** with solid **API**s is recommended.
- Also important is the selection of the most relevant **non-functional requirements** and how to balance conflicting areas. For example, a "24/7" operating model might conflict with the "optimize cost" requirement. Therefore, it is important to choose carefully and describe in detail how to measure and what the purpose is of any in-scope non-functional requirement.

Furthermore, to achieve the quality goals mentioned in chapter 2, **"Quality Goals for the ",** the following architecture design decisions need to be considered:

- **Reliability** is a term including aspects of "Availability", "Robustness", "Stability", and "Resilience". All four need to be balanced along the different functional requirements to achieve a specific probability of failure-free software operation for a specified period in a specified environment. The required design decisions need to be considered for each solution layer or module independently. E.g., historical condition data analysis might have different requirements to availability as the near real-time machine monitoring aspects. With a bulkhead pattern in mind, the different experience elements can be isolated so that if one fails, the others will continue to work. In this matter, further patterns like "retry", "circuit breaker", "throttling", and "caching" should be considered.

- **Scalability** in the context of a condition monitoring solution has two different aspects. Firstly, the ability to handle increased load from an instance of a condition monitoring solution (vertical), and secondly, the ability to scale the solution to additional production lines or factories (horizontal).
  The first aspect requires automatic scaling services to match demand to accommodate the workload. These services ensure the availability of system capacity during workload peaks and return to normal operation when the peak drops. A horizontal scaling pattern for the core components needs to be applied. Another bottleneck might become the required data pipelines and the data preprocessing. Therefore, the end-to-end data flow needs to run through several optimization loops.
  Scalability across production lines and factories is a multi-discipline challenge. The platform architecture needs to support different brownfield situations, data harmonization across the enterprise, and ensure data quality. From a condition monitoring solution perspective, aspects like the data schema,

authentication, and authorization, or multi-tenant design patterns to support a multisite deployment need to be foreseen.

Solution Preconditions

The success of a condition monitoring solution is directly dependent on data quality and availability. In turn, data quality and availability are dependent on the installed sensors. Therefore, it is essential to identify the right condition monitoring use case to start with based on the availability of high-quality data as well as the procedural and asset know-how within the organization. Any shortfalls should be identified early, and steps to mitigate these put in place. This could include installing the required sensors, partnering with machine builders, or employing external consultants during the learning phase.

There are four broad categories that are important to be investigated: **Due Diligence & Business Alignment, Connectivity, Data Mapping, and Reference Architecture.** For three (**Connectivity**, **Data Mapping**, **Reference Architecture**), the OMP alliance has working groups dedicated to these topics. Working groups regularly publish their results and guidance, and where applicable, these have been referenced in this document.

Due Diligence & Business Alignment:
This is a step where the business objectives need to be identified and defined. It is important to clearly define the problem, anticipated benefits, time horizon to realize the benefits, approaches to solving it, and an initial idea of the use case ROI. In the case of condition monitoring, its status as a 'foundational use case' should be factored into the ROI. It is likely that condition monitoring alone will not deliver a positive ROI, but without the functionality provided by this use case, more valuable use cases covering maintenance, quality, throughput, and sustainability cannot be realized.

Discovery:
This is a crucial step in the technical discovery of the As-Is state and defining a To-Be state. Some of the important areas which are defined in this step and on which OMP is working are:

- Identify the machines that would be in the scope of the work.
- Validate and identify if you have all the manuals and documents in place for those machines and they are up to date.
- Identify and list the suppliers/OEM and what kind of agreements are in place for the machines. There may be scenarios where ports may be locked and may need additional development by OEM to get access to data.
- Create a map of data/tags for each machine we want to capture. Also, identify gaps if those data points are not available from the machine and may need additional sensorization or OEM intervention.
- Create a map of communication protocols through which each machine exposes data via Modbus (serial/ethernet), Profibus, Ether Cat, OPC-UA, etc.
- Create a map of the programmable logic controllers (PLC) with their name, type, protocol, and version type (e.g., Allen Bradley ControlLogix 5500, Siemens S7-300, etc.) for the machines to be connected.

- Identify the connectivity options to the industrial machines in terms of edge gateway based on communication protocol and controllers. There may be scenarios where additional software stack/drivers need to be procured to establish connectivity
- For greenfield scenarios, it is advisable to look for machines that give data in OPC-UA format. This would enable a seamless and short onboarding.
- For sensors (e.g., to measure vibration, temperature, acoustics, etc.) that are being planned to be bought for retrofitting, ensure that they support some of the standard industrial protocols for communication and do not have any sort of vendor locking to access data.

Overall Reference Architecture

The overall reference architecture can be detailed in stages/steps covering the short-term to the long-term roadmap, where it is incrementally built up. However, the core principles should ideally be defined early so that they can easily scale up and cater to the wide variety of use cases and machine ecosystem for which the organization has plans.

Finally, before starting the execution of the project, it is important to validate the findings based on the discovery step and firm up the plan and budget requirements. You can also look at reprioritizing the requirements in terms of benefits, cost, and time to market. Get an organizational agreement and commitment for aligning single points of contact for this initiative covering: IT, Maintenance Engineering, and Production Engineering.

OMP Guidance & Recommendations for best practices
OMP Member ZF: The connection of machines is a very complex topic and must be considered in a differentiated manner. There is no single connection or strategy that can be used to connect all technical systems. An exact consideration of the specific machinery in advance is absolutely necessary.

5. Use Case Building Blocks

This section contains a high-level description of the building blocks necessary to implement condition monitoring use cases and the implementation options. The Architecture Artifacts chapter will go into more detail as to the core capabilities which are required within the condition monitoring solution. At a high level, the use case architecture can be represented by the following diagram:

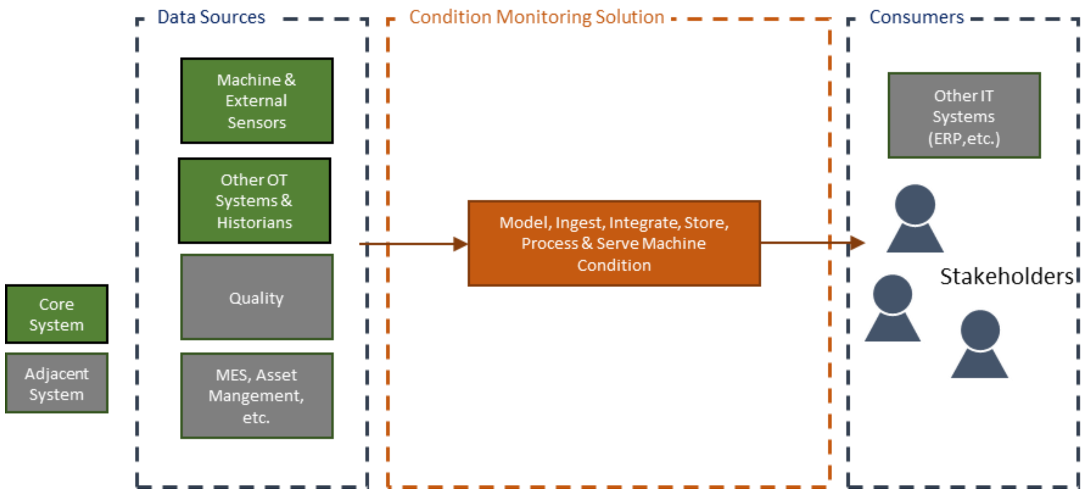


Fig 2: Overview Condition Monitoring Solution

As mentioned in Chapter 2, condition monitoring is the first step to understanding the relationship between the state of the machine and its impact on OEE metrics. It can be considered an initial use case for broader IIoT implementation or, in fact, as part of a more complex use case like predictive maintenance.

Implementation approaches for CM can be grouped into three primary categories: **Customer Build, Machine Builder Solution,** and **Third-Party Solution**. Note that in all cases, due consideration must be given to data privacy regulation if the data being captured for condition monitoring can be traced back to an individual person. For example, an employee may be required to log into a machine, and these details could then be captured as part of the current state of the machine.

Custom Build

In this approach, all responsibilities for defining, creating, deploying, and maintaining the condition monitoring solution will be undertaken by the manufacturer. This is usually done in a manner aligned to a corporate industrial data strategy as part of a larger strategic initiative. In practice, some aspects could be outsourced to third parties, but in principle, the manufacturer retains all responsibilities.

This approach is primarily practical for large manufacturers who have a strategic goal of implementing IIoT capabilities at scale. They require having the skills available to both build the original solution and expand the condition monitoring use case into other use cases to drive further value.

Benefits of Custom Build

Taking the custom build approach provides manufacturers with extreme flexibility in terms of what data is collected and how this data is used. Manufacturers can be in charge of their own roadmap and functionality deployment choices. All solutions will be uniquely adapted to their machines, production lines, and products – implying a better correlation to monitoring their specific environment. Data prepared for condition monitoring algorithms can readily be reused for other purposes, as can the output of the condition monitoring algorithms. These benefits will increase exponentially as the manufacturers build use cases beyond condition monitoring.

Considerations before embarking on condition monitoring custom build

The manufacturer must ensure that they have the resources and skills available to specify, build and maintain the condition monitoring use case. These resources should be available *and* be ensured for the medium to long term as part of the corporate strategy of building and maintaining IIoT models. Custom-built IIoT projects are, on the whole, complex to design and build, requiring considerable resources, and if implemented in isolation, may not deliver net value.

From a technical point of view, the technology choices for the core capabilities required to implement condition monitoring must be made with the end state in mind. Two key considerations are the scale of the solution required for the end state (number of machines, data volumes, analytical complexity, and required SLAs) and the strategy pertaining to the reuse of data and analytical models.
As different architectural components scale in different ways and have different licensing models – some components which are viable for one or two machines may not be viable at enterprise scale from a cost or capability point of view.

The desire to reuse data across multiple IIoT solutions implies significant investment in data management skills – starting with the data model and storage strategy. How data is stored has a significant impact on staff productivity, technical delivery of multiple solutions, and the total cost of ownership – and hence net value – of the environment.

As the owner of the end-to-end solution, the manufacturer will also take on all responsibility for data ownership, security, and access.

Machine Builder Solution

Many machine builders have started to offer a condition monitoring solution as a service offering with their machines. This is frequently tied into a servitization business model, where the manufacturer pays a predetermined rate to guarantee the uptime of their assets vs. maintaining the machines themselves. In these cases, and others, the use cases offered as a service by machine builders often extend beyond condition monitoring to predictive/preventative maintenance and some aspects of industrial inspection.

This approach is very attractive to smaller manufacturers without the skills or resources for a custom build. In addition, those companies that have not yet formulated a long-term strategic plan for deploying IIoT at scale can benefit immediately from several use cases, which can help inform their planning.

Benefits of Machine Builder Solution
The benefits are focused on immediacy and simplicity. These solutions are normally offered on a subscription basis, so they can be ramped up rapidly, shut down with a defined notice period, and require no investment on behalf of the manufacturer. As the machine builder can view their entire installed base across multiple customers, insights and subsequent recommendations are richer due to a larger data set that is being drawn on.

Considerations before selecting a machine builder solution
The primary consideration is, in effect, an ROI calculation. Are the insights which are made available worth the price being paid? For the purchase of newer machines, the availability of condition monitoring as a service may well factor into the vendor selection. However, there are some less obvious considerations that should be negotiated at the time of contract signing.

The first is about data ownership and use. Capturing data from machines and preparing this for analytical routines requires expert knowledge and needs to be actively managed. Manufacturers should ensure they have access to the data captured from their machines for additional purposes and be aware if this has an additional fee or not. The availability of data for additional use cases must be matched with the IIoT ambitions of the manufacturer.

A second consideration is data security and access. During contract negotiations, it should be clear who is responsible for data security along the entire use case value chain: extraction from machines, transmission across a network, data at rest, and data access via direct methods or through applications. Manufacturers should also consider how the machine builder can use the data extracted from machines in their production line and ensure any usage restrictions are included in the contracts if appropriate.

Finally, the number of different solutions should be considered. Suppose a manufacturer has multiple different machine builders as suppliers, all offering a variety of services. In that case, this could fragment the data and app environment, introducing unnecessary complexity and reducing worker mobility through the need for retraining.

Third Party Solutions

A final category of condition monitoring solutions is that of third parties. Third parties may be independent software vendors, consulting companies who will build and host the solution for a subscription rate, or machine builders who support a broad variety of machines, not just their own. From a deployment point of view, this is very similar to the Machine Builder solution but with different contracting parties.

As such, this option shares many of the benefits and considerations shared in section 6 and will be highly attractive to smaller manufacturers or those whose strategic plans for implementing IIoT involve outsourcing.

Benefits of Third-Party Solutions

Like the machine builder option, the benefits are both immediacy and simplicity. The attractiveness of this approach increases according to the number of machine brands and types that the third party supports and plans to support in the future. Depending on the provider, there will be options to customize the implementation to your production process and products, providing more specific insight than a more generic offering. Having one provider deliver condition monitoring functionality across a large number of machines also has clear benefits in simplicity of environment and operator training. There should also be the ability to create an integrated dashboard that shows the condition of an entire production line, factory floor, or across the enterprise.

Considerations before selecting a Third-Party Solution

Again, the considerations are very closely aligned to the machine builder solution. There are obvious additional considerations in terms of the number of machine types supported, the level of knowledge about these machines, and the richness of functionality offered across the manufacturer's installed base. Due to the increased capability, potential providers should be assessed not as a single solution provider but as a partner supporting industrial insights in the mid- to long-term. So, the vendors for future expansion of functionality beyond condition monitoring and current machines supported should be considered in the decision-making process.

Data ownership and security responsibilities are once again highly relevant, as are the ability and cost implications to reuse data beyond the suite of solutions provided by the vendor. In addition, contracts with machine builders must be considered. Many machine builders may prohibit data from their machines from being passed along to a competitor or third party, where this data may be used to gain a competitive advantage.

OMP Guidance & Recommendations for best practices
OMP Member ZF: Even if there are established providers on the market, high initial expenses and follow-up costs are to be expected. Investments and expenses are incurred with every variant. Requirements and expectations must be clarified in advance and compromises must be communicated.

6. Architecture Artifacts

The following diagram shows the necessary solution components for condition monitoring use cases by decomposing the solution into subsystems and their core modules, components, or capabilities. It also gives an overview of the relationship across the different subsystems and their associations.

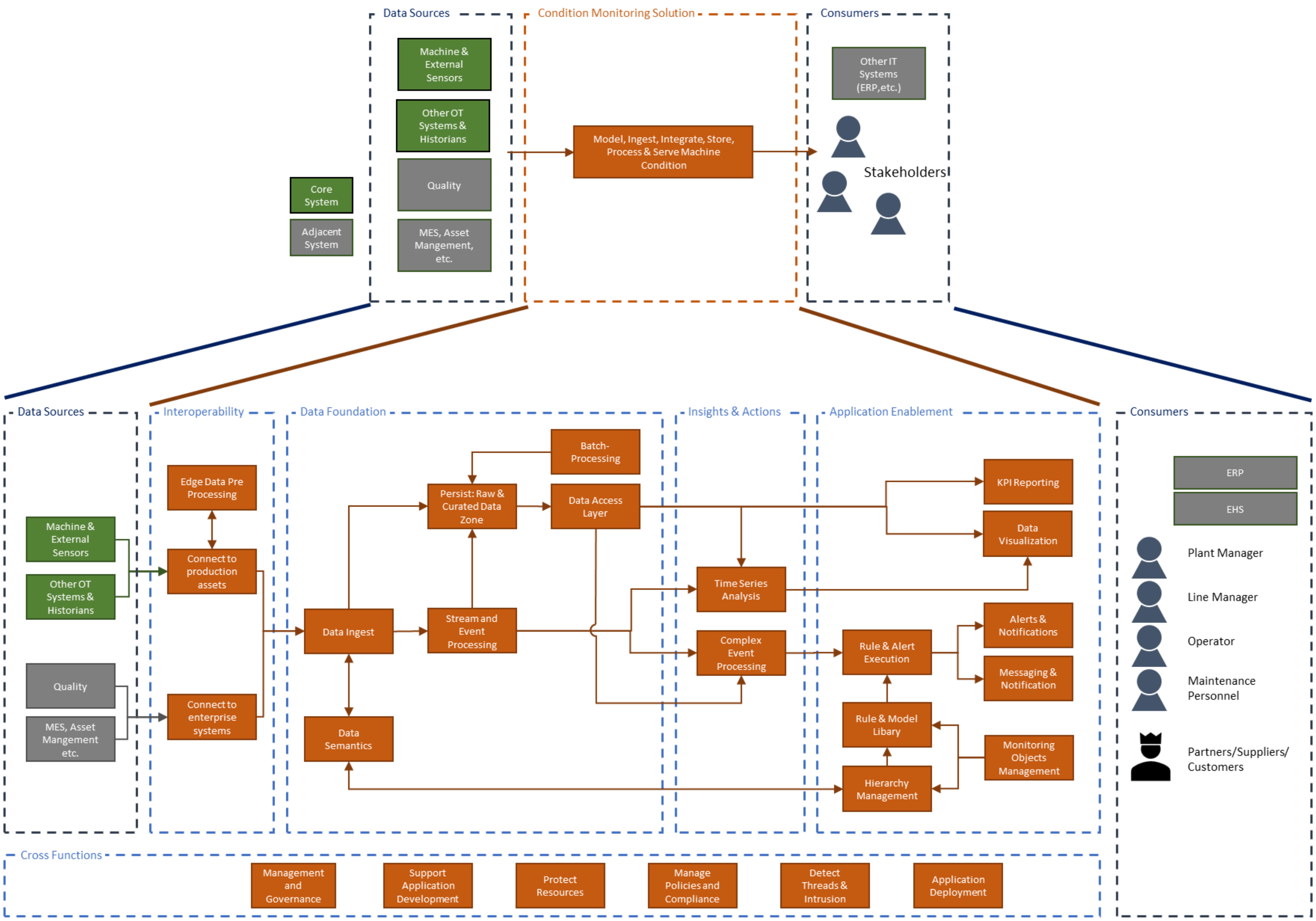
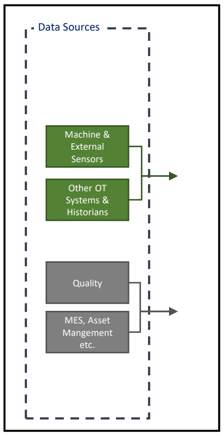


Fig3: Condition Monitoring Solution Components

A brief description of the role of the subsystems for CM use case development and the capabilities that drive the subsystems are given below. This may not be an exhaustive list of all the capabilities and sub capabilities; however, it covers most of the critical ones. When doing a specific architecture definition, there may be scenarios where more of the capabilities can get aligned.

Subsystem Data Sources



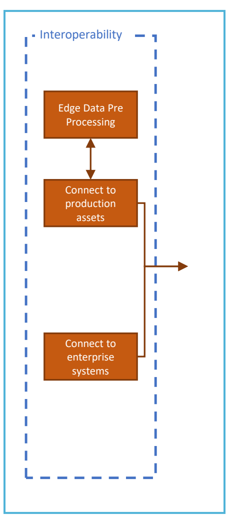
Data sources cover the endpoint from where the data needs to be integrated for condition monitoring. These sources are divided between "Core Systems" and "Adjacent Systems". Core systems are critical for the solution to function. Examples of core systems are the actual machines on the shop floor, sensors that are either part of such machines or are retrofitted, or systems like historians where the relevant data may already be made available in some of the scenarios.

Adjacent systems are systems that are not mandatory but may be required for any specific data added to the use case or when the condition monitoring use case is extended further, as outlined in section "General Information". For example, if there is a specific need to pull the asset hierarchy or asset maintenance details as well, then the Asset Management system can also be integrated.

Condition Monitoring Core Solution

Ingesting, processing, and visualizing the data from different source systems is the heart of the condition monitoring solution. Many of the required solution capabilities may already exist or can be reused for other scenarios as well. The main solution blocks are described below:

INTEROPERABILITY



Interoperability is a technical enabler to interact with other subsystems. In the scenario described throughout this document, this capability is used to connect and ingest data from either databases or software solutions, but it may also enable 3rd party software integration if it becomes necessary.

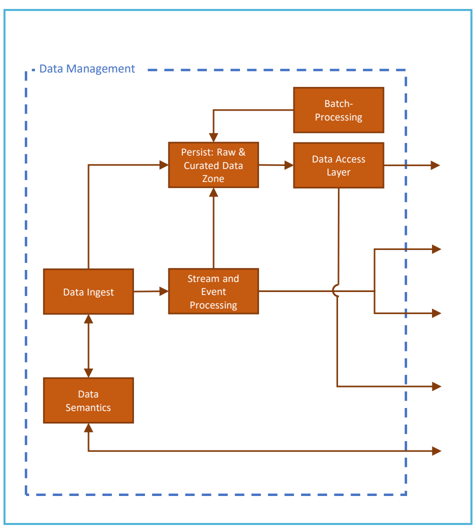
While the solution **connects to production assets** and the sensors which are deployed on the assets, several decisions might be considered. Master data, in comparison to telemetry data, requires different approaches and methods. It also needs to be considered if "all" available data or just scenario-specific data gets connected. It might be a good approach to collect training data early and ingest as much data as possible to support future scenarios. However, it also comes with cost. Therefore, the architectural constraint for cost optimization needs to be considered. From an architectural point of view, the solution needs to support industrial protocols (Modbus, ProfiNet, MQTT, OPC-UA, etc.) and ingest configurations (interval, single/group tag streaming, batch, data selection, etc.). Inbound data transport is sufficient for condition monitoring, but to be future-ready, in- and outbound data connections should be supported.

Depending upon the source of data collection, e.g., OT sources (like PLC, Historians), external sensors, there is a need to do some **edge-data preprocessing** before the data could be used for any further processing. Some of the requirements are filtering out some data to optimize the quality and size, aggregating high-frequency data, compressing data, contextualizing the data, normalizing the data, or adding some additional intelligence to the data. In certain scenarios, running a local rule engine may also be needed to generate quicker alerts or any other insights for local intelligence. These scenarios need this capability to be enabled, and this usually runs near the source systems, which can be at individual equipment or production line level or at a plant level.

In certain scenarios, the machine condition data needs to be **connected to enterprise systems** with some of the additional data which resides in the IT systems to drive more intelligence. Some examples are associating the equipment with a defined asset (available in an asset management system) and associating the health of the equipment with the past service record. Such scenarios need this capability to connect to some systems like MES, EAM, and QMS. The connectivity to these systems is usually over well-defined interfaces like REST, OData, or OPC-UA.

DATA FOUNDATION

Data management provides all necessary capabilities to ingest, transform, store, and provide access to the data. From a technical point of view, the data management architecture can be built based on reference architectures. However, the design should consider big data strategies and the new concepts of data mesh-orientated architectures to overcome data silos.



Depending upon if the data from external systems being received is coming in real/near-real-time or in batch, the **data ingest** capability needs to provide the ability to handle both streaming and batch data. It may include either standard pre-built connectors for known 3rd party systems or bespoke developed connectors to ingest the required data. **Data semantics** is an important sub-capability that is responsible for giving context to the data coming from various sources and meta data. The basis for doing so is a standard data model that enables that data from different sources can be understood. The different data sources may be in different dimensions (e.g., time dimension and transactions). This capability is also covered separately in detail in the OMP "Semantic Data Structuring" working group.

To **persist,** data requires segregation and storage depending on the speed, size, and context. This capability ensures the availability of data for any further reporting and analysis. Regardless of possible transformations to create curated data, data should always be kept in its raw format. For condition monitoring use cases, it should support the following types of storage mechanisms:

- Time Series: for storing all the time-series data coming from sensors, equipment, and other OT systems
- Unstructured: for storing all the log, configuration, image, text, and sound files

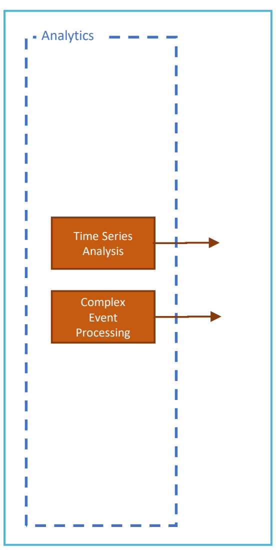
- Structured: for storing all the generated data from insights required for reporting and storing any configuration and meta data

Batch processing capability is usually invoked by batch reporting, historical trend analysis, root cause analysis, FMEA analysis, batch analytics, and model training kinds of requirements. It allows for processing large historical data sets and making the insights available back to the persist layer for enabling its usage towards other capabilities like reporting.

The opposite is **stream event processing**. It allows for real/near-real-time data in motion to be ingested. This can also allow for some intermediate processing like aggregation, filtering, etc., of data. This capability relies on messaging capabilities from integration services and should be able to handle data coming at high speed. Finally, the **data access** layer makes the data available to various subsystems via a well-defined and governed interface. This layer also interacts with data semantics capability to ensure that the data being made available contains the relevant context information for the calling subsystem. This also ensures that the calling subsystem has relevant authorization to access the data.

Analytics & Data Usability

The analytics and data usability pillar enables us to gain insights and make the data collected by the platform actionable. In the scenario of condition monitoring, the complexity of this pillar might be low, but in order to prepare for future scenarios, like predictive maintenance, the importance of this capability should not be underestimated.

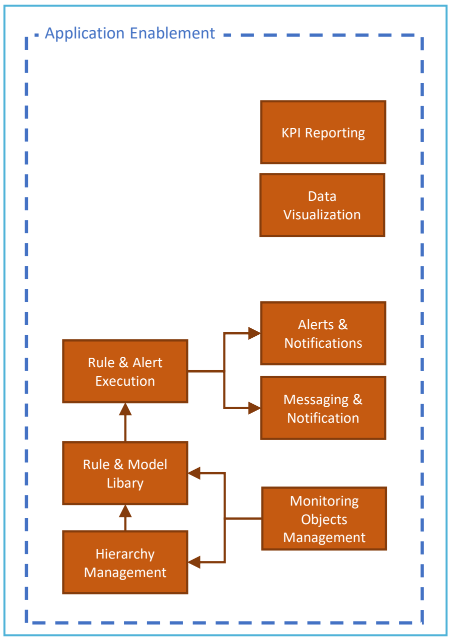


For any use case scenario in digital manufacturing, analyzing data within the context of time is important. **Time series analysis** provides this capability. Additionally, this also needs to be sometimes aligned to transactional data captured in other systems. For example, to identify in which shift the fault occurred, or if it has been a consistent fault in spite of the previous preventive maintenance activities. This gets used in the complex event processing to analyze the data.

The capability for **complex event processing** is typically required to enable an event-driven scenario where action needs to be taken on a real/near-real-time basis when the data is being streamed in via the data ingest capability. The event could, for example, be threshold breaches of a single or group of parameters, a sliding window aggregation insight indicating the presence of a condition for some time that needs action, or an event that is generated based on conditions derived from looking at data from various systems, etc.

Application Enablement

Within the application enablement cluster, all functional capabilities necessary to implement applications required for a condition monitoring scenario are grouped. Very often, different teams implement the core platform solution artifacts to integrate and manage data, and other teams implement the scenario-specific applications together with process engineers or other production-related roles. These applications require human interaction or present the data for human inference. They can enable various interaction technologies, from traditional PC, Industrial HMIs, Tablets, Mobile Phones, VR headsets, and AR. Web-based front-end applications should also cater to special requirements regarding usability, authentication, and authorization. These front-end applications can cater to various scenarios.



For regular monitoring, **data visualization** and **reporting** are one of the base sets of capability elements that should be present. These are standard web interfaces that provide meaningful data insights that may be the result of some algorithm (like FMEA analysis, root cause analysis) or could be trend analysis of one parameter or between parameters or could be visibility into a key defined KPI to name the few.

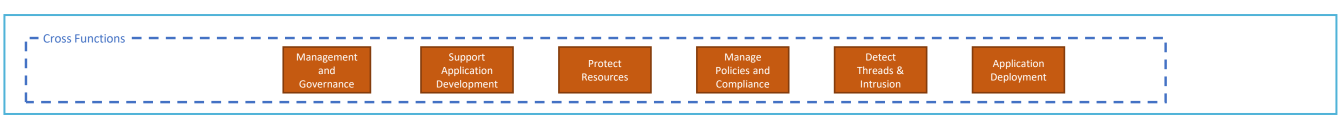
Based on the complex event processing capability, the **rule & alert execution** function combines the data stream with rule configuration or an algorithm's output, defined in a **rule & model library**, to send **notifications** to humans or **messages** to integrated consumer systems. With this, call-to-action scenarios can be implemented to help alert the users (e.g., operators, shift supervisors, plant managers, etc.) on conditions that have been triggered. For this scenario, the **rule & model library** acts as a repository of all the defined rules which need to be applied to the data. Such rules would be, for example, converting the serial number coming from data into a name, the definition of thresholds against a parameter (Hi, Lo, Hi-Hi, Low-Low), the definition of an action against a condition, etc.

To support operations, a **monitoring object management** capability should be considered. This solution artifact supports configuring monitoring rules based on the tag structure provided by the machine builder. Very often, especially in complex production lines, operators want to virtually modify for specific monitoring aspects the grouping of tags.

Lastly, administration dashboards are required to perform configurations like user onboarding & configuration, application-level configuration, plant hierarchy configuration, plant onboarding to name a few.

Cross Functions

Specific capabilities are the foundation of a manufacturing platform and need to be applied for layers across any manufacturing scenario. This document describes only the the core aspects. However, whilst designing a digital manufacturing platform, additional considerations include, how many environments are required (dev,



integration, test, staging, prod), how deployment scenarios are applied across the platform, the processes to have reasonable test data available, and how technology updates will be applied to the digital platform to name a few. For the condition monitoring core system, some solution management and security capabilities are the bare minimum.

This starts with the **management and governance** capability. It ensures the efficient and stable functioning of the CM solution and covers aspects like managing updates to deployment without downtime, managing configuration, and monitoring the overall system. Monitoring the entire landscape once the CM solution starts to get scaled up and rolled out across plants is also one of the important capabilities.

As for any solution, security, without a doubt, should be one of the capabilities looked at from the beginning to ensure that the critical systems being monitored are not impacted by any security breach, as these systems are core to the business. Almost all aspects covered under security, covered under **protect resources,** and **detect threads & intrusion** should be taken into consideration when designing and deploying such a solution.

Tools used for developing solution components like dashboards, algorithms, etc., need to be natively supported for application development. These could be traditional development environments for experts, as well as low/no-code environments. The latter will become more important over time to support process engineers and any other role needing to utilize the manufacturing data to optimize and make work more efficient.

Finally, processes and tools around **application development** need to be in scope for a condition monitoring solution. When the solution is deployed across the manufacturing landscape, any changes to the solution could create consistency issues with data and its usage if not planned properly. When a manufacturing customer foresees that the solution will undergo frequent changes, then investigating this capability is important. This capability is also important in the initial development cycles when the solution is being built and proven in some sample sites for enhancements and improvisations.

Subsystem Consumers

All the consumer elements which are interested in the data captured by the CM solution can be individual user personas within the manufacturer, suppliers, customers, or external systems.

7. Final Remarks

Condition Monitoring serves as the fundamental use case when it comes to the single point of truth for real time monitoring and therefore several use cases based on this, such as preventive, predictive, and prescriptive maintenance. It is therefore very important to implement this basic use case in such a way that it provides a good basis for all scenario implementations based on it.

Nowadays Companies monitor their machines in-house, but the industry sees an emerging trend to hand over the machine monitoring and analysis of gathered data to an external provider. In this context condition monitoring transforms into “Remote condition monitoring”.

Equipment machine builders are starting to provide this scenario as "Equipment as a service" (EaaS). These cloud-based subscription services for maintenance and machine monitoring often include or add machine leasing contracts or insurance packages to help the manufacturer moving from CapEx to OpEx purchasing models. The realization of new opportunities concerning investments (into new machines), cost reductions, and increased flexibility are the key drivers for these future models.