BRAINE

BRAINE - Big data Processing and Artificial Intelligence at the Network Edge

Project Title:	**BRAINE - Big data Processing and Artificial Intelligence at the Network Edge**
Contract No:	876967 – BRAINE
Instrument:	ECSEL Research and Innovation Action
Call:	H2020-ECSEL-2019-2-RIA
Start of project:	1 May 2020
Duration:	43 months

Deliverable No: D5.8

BRAINE robotics use case

Due date of deliverable:	30 November 2023
Actual submission date:	25 November 2023
Version:	1.0

Project funded by the European Community under the H2020 Programme for Research and Innovation.

Project ref. number	876967
Project title	BRAINE - Big data Processing and Artificial Intelligence at the Network Edge

Deliverable title	BRAINE robotics use case
Deliverable number	D5.8
Deliverable version	Version 1.0
Contractual date of delivery	30 November 2023
Actual date of delivery	25 November 2023
Deliverable filename	D5.8 – BRAINE robotics use case
Nature of deliverable	Report
Dissemination level	PU
Number of pages	32
Work package	WP5
Task(s)	T5.4
Partner responsible	FS, CTU
Author(s)	Martin Ron (FS), Vojtech Janu (CTU)

Editor	Martin Ron (FS)

Abstract	This technical report, delivers the detailed information about the development of a robotics system designed to integrate and utilize the features available on the BRAINE platform to provide high performance real-time analytics and motif discovery of robot sensor data.
Keywords	Edge computing, Industry 4.0, AI, Motif Discovery

Copyright

Deliverable history

Version	Date	Reason	Revised by
0.1	01.03.2023	Table of Contents	Sean Ahearne
1.0	08.11.2023	Finalized content	Pavel Burget, Martin Ron, F. Cugini

List of abbreviations and Acronyms

Abbreviation	Meaning
AI	Artificial Intelligence
API	Application Programming Interface
CPU	Central Processing Unit
EEG	ElectroEncephaloGram
EMDC	Edge Mobile Data Center
EU	European Union
GDPR	General Data Protection Regulation
GPU	Graphics Processing Unit
IoT	Internet of Things
IT	Information Technology
KPI	Key Performance Indicator
PoC	Proof of Concept
QSD	Qualified Synthetic Data
TBC	To Be Confirmed
TBD	To Be Defined

Table of Contents

1. Executive summary

This section describes the BRAINE use case on Robotics 4.0. It includes:

- The overall objective of the use case
- The new techniques and technologies being used by the use case
- How the BRAINE platform integrates and enables these technologies
- The impact on KPI's compared to the state of the art
- The potential business impact of the use case and platform

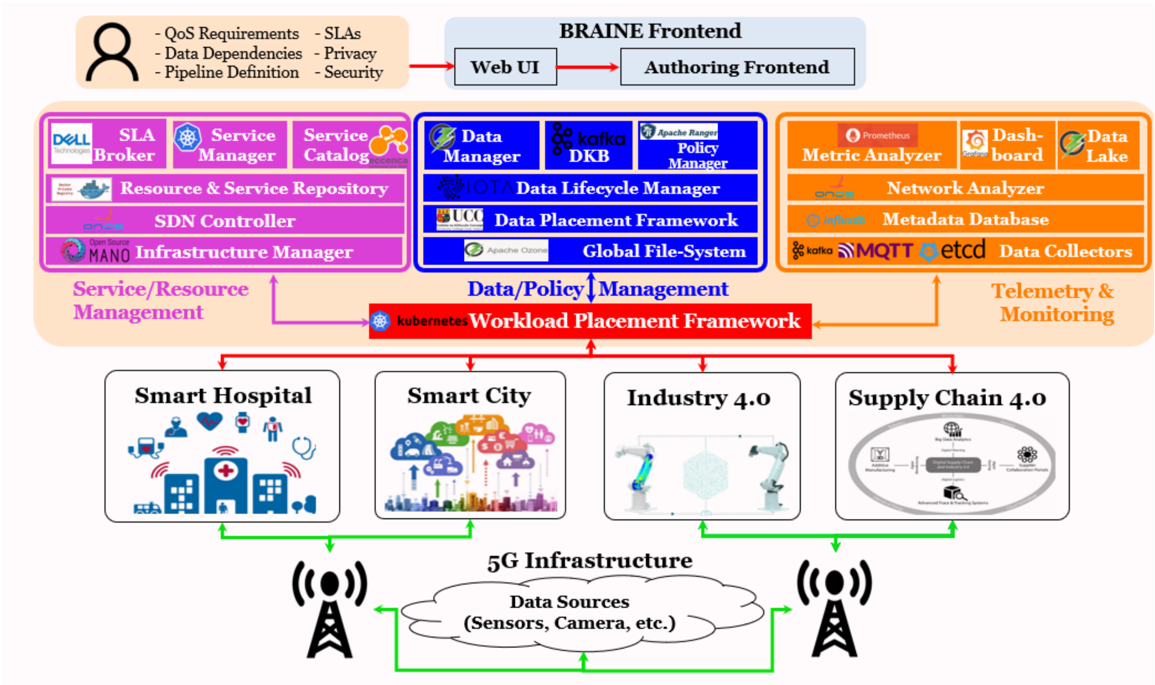

Fig. 1: BRAINE Architecture

2. Use case overview

In the era of Industry 4.0, where automation and connectivity are revolutionizing manufacturing processes, flexible production has emerged as a critical aspect of efficient and adaptive manufacturing systems. This use case explores employing of a multiagent platform and motif discovery tool to enhance flexibility in production processes.

A multiagent platform is a software framework that enables the coordination and communication of multiple autonomous agents. By utilizing a multiagent platform, manufacturers can establish a flexible production environment that adapts to changing demands and conditions in real-time. Each autonomous agent represents a specific element of the production system, such as machines, robots, or workstations. These agents can collaborate, negotiate, and make decentralized decisions to optimize production efficiency.

In addition to the multiagent platform, the use of a motif discovery tool further enhances the flexibility of production systems. The motif discovery tool analyses historical process data to identify recurring patterns or motifs. These motifs represent efficient process configurations or sequences that can be reused to optimize future production scenarios.

2.1. Background and Motivation

The adaptability and versatility required in modern production processes necessitate a novel approach that extends beyond the machines themselves and their local control systems. The scope of higher-level systems traditionally managed by IT systems, such as Manufacturing Execution Systems (MES) and Enterprise Resource Planning systems (ERP), must also be reevaluated. In this context, the focus of this use case is on the MES level, aiming to propose a multi-agent platform that enables the implementation of a distributed MES. This system will be responsible for overseeing the production process, conducting backward diagnostics, and providing supervision.

Drawing upon existing distributed production systems, a distributed algorithm for production planning will be employed. This algorithm will utilize agents that represent production machines and actively participate in the planning process. This approach allows for the handling of significantly more computationally complex tasks compared to centralized planning methods.

To ensure optimal performance, predictive supervision is crucial. To address this need, a machine learning tool developed by FS will be utilized for domain-agnostic motif/pattern discovery in multidimensional time series data (MOD). This tool is capable of detecting various production states of observed devices, identifying anomalies within production processes, and facilitating association analysis of operational and product parameters, such as product quality. MOD incorporates data preprocessing capabilities, serving as a supportive tool for data exploration. As part of the BRAINE project, modifications will be made to enable MOD to operate on edge devices and provide real-time inputs to an upper-level cloud-based model (Digital Twin) based on the current timing parameters of the observed devices. The Digital Twin will assume the role of supervising the production line, detecting higher-level anomalies within the production process.

Although the initial deployment of this system will occur on a testbed for an Industry 4.0 demonstration production line at CTU, it is important to highlight that the methods tested in this use case can also be applied to already operational industrial production lines.

2.2. Objective

Based on the benefits of the multiagent platform (decentralized decision-making, adaptive resource allocation, fault tolerance, redundancy, etc), and of motif discovery tool (process optimization, predictive maintenance, continuous improvement, etc.) were established nonfunctional objectives described below and functional objectives described in form of deployment topology and KPIs.

2.2.1. Nonfunctional objectives:

- **Scalability and Flexibility**: The ability to flexibly upgrade manufacturing processes as fast as possible can lower the time to market for new products and give companies a significant edge over competitors. Thanks to BRAINE's automatic workload management, rollouts of changes in the production systems or rollout of new additional components can happen immediately.
- **Heterogeneity**: Devices, software, and services of various brands and types usually accumulate in the factories during the time. Simultaneously, data from those services and devices shouldn't leave a manufacturing facility because of security. The powerful edge solution supporting the interaction of various services and devices would allow the integration of multiple data sources into a better-informed solution.

2.2.2. Functional objective:

One of the functional objective, that is based on the nonfunctional ones is deployment topology depicted in the following figure. In the figure, the blue-filled shapes symbolize agents, the orange ones symbolize the digital shadow, the purple ones symbolize the subpart recognition, the pink ones symbolize the position calibration, the red ones the motif discovery. Shapes with only bold borderline symbolize groups of units.

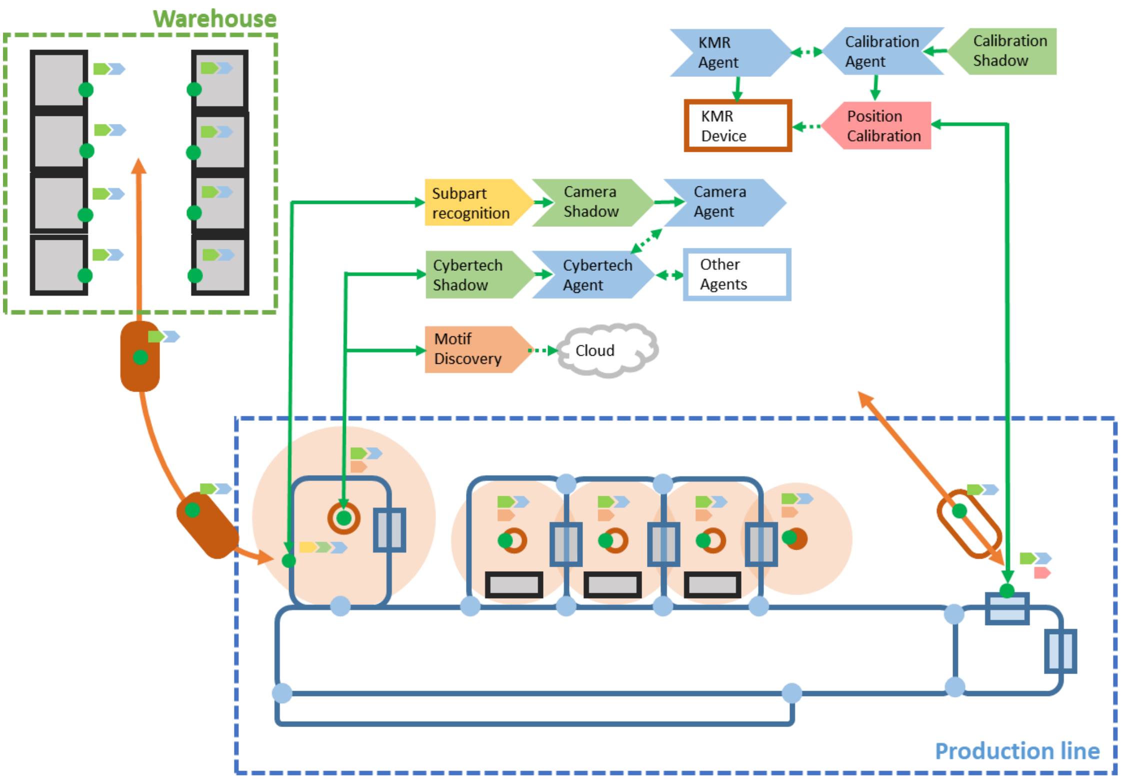

Fig. 2: scenario

2.3. Goals (KPI's)

All goals (KPIs) for the use case 3 are stated in the following table. Each KPI is provided with a name, brief description, and evaluation criteria, that precisely describes how the KPI can be evaluated.

KPI 1	Ratio of data processed on Edge to data sent to Cloud
Description	Expected reduction from 1:1 (baseline) to 1:5

Evaluation	There are 3 phases of KPI evaluation: 1) Preliminary synthetic dataset evaluated in Docker on standard PC 2) Manufacturing process dataset evaluated on Siemens Edge devices in CIIRC Testbed[9] 3) Manufacturing process dataset evaluated on BRAINE EMDC In case of remote evaluation, the data acquisition will be evaluated locally in CIIRC Testbed and the acquired dataset will be evaluated on EMDC remotely.
KPI 2	**Mean delay of deviation detection event**
Description	Baseline is mean delay of 500 ms + 50% of pattern duration, expected reduction is to 500 ms + 25% of pattern duration.
Evaluation	There are 3 phases of KPI evaluation: 1) Preliminary synthetic dataset evaluated in Docker on standard PC 2) Manufacturing process dataset evaluated on Siemens Edge devices in CIIRC Testbed 3) Manufacturing process dataset evaluated on BRAINE EMDC In case of remote evaluation, the data acquisition will be evaluated locally in CIIRC Testbed and the acquired dataset will be evaluated on EMDC remotely.
KPI 3	**Setup and commissioning time**
Description	One day setup and commissioning of EMDC, including secured registration into an edge device
Evaluation	Measure average time of deployment
KPI 4	**Commissioning time of robotic systems**
Description	In comparison to conventional/empirical approaches, the AI-enhanced process to identify and optimize the system parameters is expected to finish 10x faster.
Evaluation	Compare average time of multiple scenarios run by both approaches

KPI 5	Scalability and modularity of manufacturing lines
Description	By flexible reconfiguration, the production time reduced by 20%
Evaluation	Compare average time of multiple production scenarios run by MES unable of configure production for multiple various products vs. MES that is able to
KPI 6	**Image based subpart recognition**
Description	Minimum of 90% accuracy for all parameters
Evaluation	Compute average accuracy of 100 recognitions
KPI 7	**Image based calibration of position**
Description	More accurate than build in solution in KMP and KMR
Evaluation	Compute average accuracy of 100 recognitions

3. Implementation and Integration

The use case 3 takes into account three application scenarios:

- Warehouse (Fig. 3);
- Production line (Fig. 4);
- Diagnostic (Digital Twin).

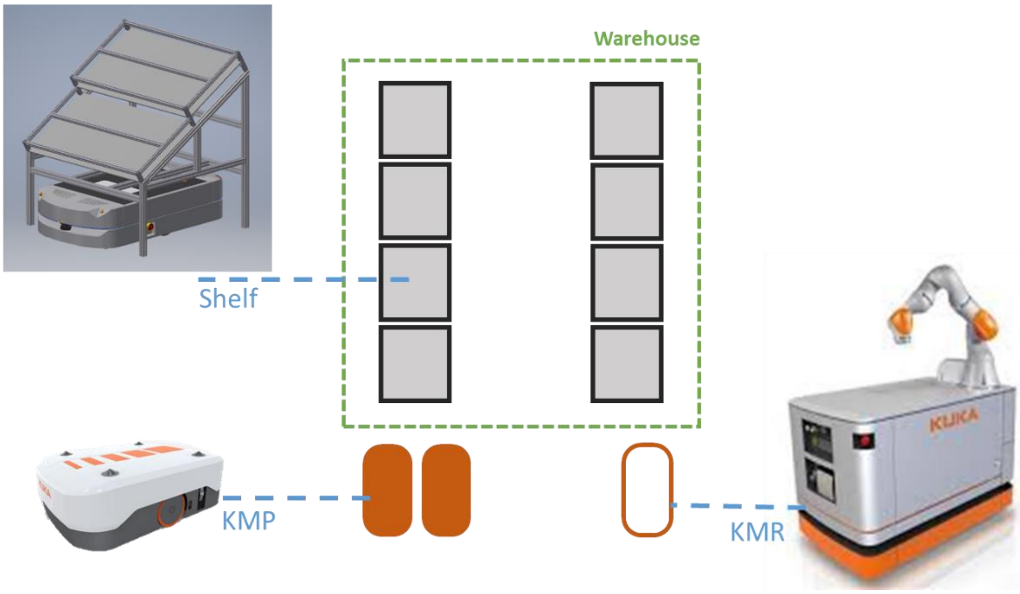

Fig. 3: Warehouse scenario

Item	Description
KMP Kuka	The KMP 1500 is an omnidirectional, mobile platform that navigates autonomously and flexibly. Combined with the latest KUKA Sunrise controller, it provides modular, versatile and above all mobile production concepts for the industry of the future.
KMR Iiwa	The autonomous KMR Iiwa robot is HRC-capable and mobile. It combines the strengths of the sensitive LBR iiwa lightweight robot with those of a mobile, autonomous platform. The KMR iiwa is location-independent and highly flexible – the perfect basis for meeting the requirements of Industrie 4.0.

Cybertech	The industrial robots of the KR CYBERTECH family represent the world's largest range of models in the low payload category with the greatest power density. They are ideally suited to space-saving cell concepts and provide top performance – with particularly low follow-up costs
Agilus	The KR AGILUS is our compact six-axis robot that is designed for particularly high working speeds. Different versions, installation positions, reaches and payloads transform the small robot into a precision artist.
Montrac Station	Montrac Shuttle - product details available at https://www.montratec.de/en/products/

Table 7: Testbed components

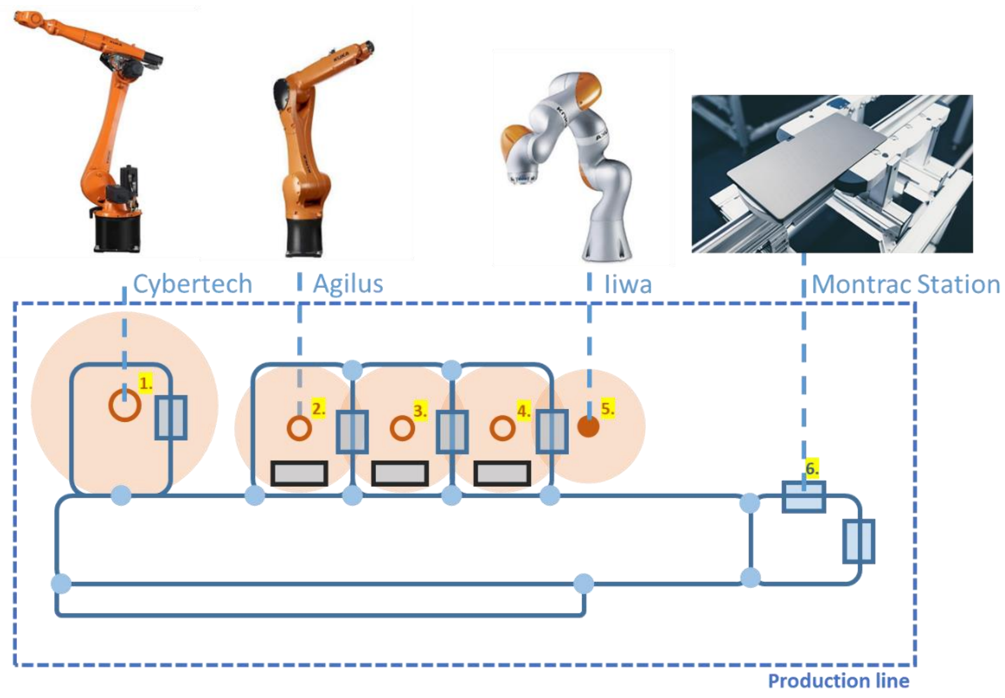

Fig. 4: Production Line scenario

3.1. Use case implementation

Use case production planning is based on a multiagent system. Particular agents represent specific hardware resources or product agents that communicate with each other to achieve the requested goal, i.e., producing the specific product.

3.1.1. Multi Agent platform

The multiagent approach enables flexibility of the planning and orchestration. The overall production plan is divided into smaller tasks (operations of the recipe), which are planned separately, assuming resources available at the time of planning. This makes the planning problem much more straightforward compared to the case when a full plan is calculated at the beginning of production. The multiagent approach facilitates the management of the resources and products at the runtime. New resources may be added/removed from the platform, and new products may be ordered during runtime without changing the platform's configuration or interrupting ongoing production.

The physical system and the products are described semantically in the ontology together with capabilities and properties (e.g., the machines' manufacturing activities, the composition of the product, and its production recipe). The recipe concept in the ontology specifies operations that must be performed during the production and their order. The overall plan splits into a sequence of jobs representing specific tasks performed by the machine. The top-level job represents a single operation of the production recipe. Within the job, the agent submits requests to other agents to satisfy prerequisites needed for the job (e.g., the robot may need to supply material and transport the product to its station before executing its task). Receiving the request from the agent invokes a new job, which may, in turn, submit requests to other agents.

The entire life cycle of production negotiation is based on the Plan-Commit-Execute protocol. The execution of a particular production recipe step is performed in three phases, the Planning phase, the Commit phase, and the Execute phase. The diagram of the negotiation flow is depicted in Fig. 5.

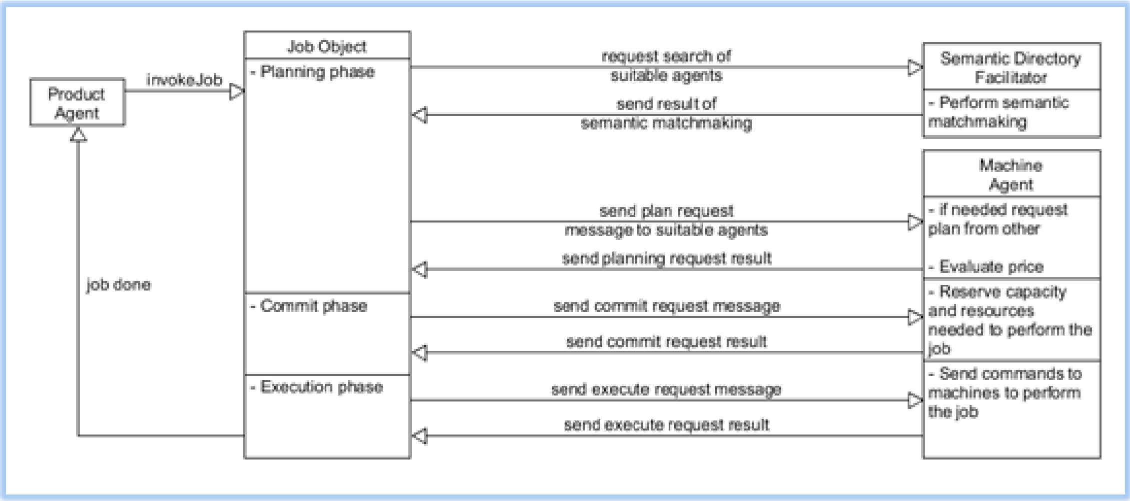

Fig. 5: Inter-agent negotiation life cycle

Within the Planning phase, a full plan of tasks is created. This phase splits into two steps. First, the request is sent to the directory service agent. The directory service agent performs semantic matchmaking and returns a list of agents capable of executing the job. The request from the product agent is then sent to the suitable agents.

Upon request from the product agent, the machine agent checks its resources, and if needed, the machine agent may submit requests to other agents to fulfill all requirements needed to perform the job. If the machine agent can satisfy all job requirements, it evaluates the job's overall price and sends success results back to the product agent. The product agent then evaluates responses obtained from all machine agents and selects the best option.

During the commit phase, the product agent requests all agents involved in the selected plan sequence to reserve their resources needed to perform the job. These agents are thus dedicated to the specific job and cannot accept other jobs.

Within the execute phase, the product agent commands the machine agents to perform the tasks in the sequence of the plan. The hardware agents send commands to the machines and monitor the execution of the tasks. Upon successful execution, the product agent is notified and may start following the operation of its production recipe.

The multiagent platform implements FIPA specifications. These specifications define standards for agent communication to facilitate understanding of agents and interoperability of different multiagent systems. FIPA specifications define elements of the message that determine the meaning of the message and how the message should be further processed. This includes, for example, the identification of the agent to whom the response should be sent. FIPA specifications also introduce the concept of so-called performative, which identifies the message's type, whether it is a request message, response message, reject the message, etc.

The agents themselves can be divided into three fundamental groups:

- Service agents;
- Product agents;
- Hardware agents.

The service agents provide the basic functionality of the platform, such as service discovery, monitoring, communication with external product order service. Product agents represent the Products to be manufactured. It is related to the specific product description in the ontology. The product agent sequentially negotiates the execution of particular operations composing the product recipe. The core of the platform includes the service agents and the product agents. The hardware agents may be running as separate applications outside the platform core. The architecture is displayed in Fig. 6.

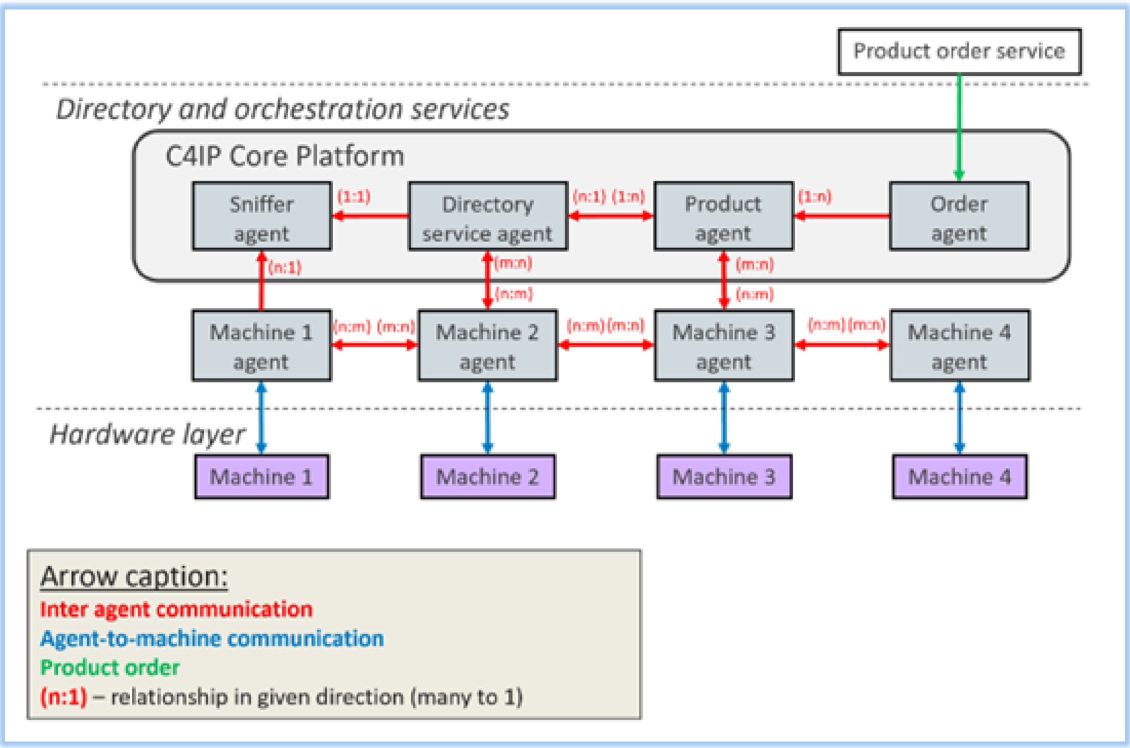

Fig. 6: Platform architecture

Registry is the key element of the platform. The registry provides three main capabilities:

1. manages running services.
2. registers available capabilities .
3. registers free resources

Order agent collects orders from external Product order service and creates and initiates corresponding Product agent..

Distributed logger collects and stores log messages from all agents.

Hardware agent – represents robots on the production line, both Kuka[8] Agilus robots, and Kuka IIWA. The manufacturing activity provided by these robots is Pick and Place. The KukaRobot agent provides two Job implementations, namely AssembleJob and SupplyMaterialJob. Following diagram Fig. 7 shows the workflow of the AssembleJob planning logic. The job is created upon request from the product agent. The job checks whether the requested material is available in the local warehouse and if not, the robot agent requests a supply of the material from other agents. Then it is checked whether the (partly assembled) product is already present at one of the stations used by the robot. If not, the robot agent requests reservation of the station and requests transport of the product or an empty shuttle to the reserved station. Finally, the robot's task, i.e., the Pick and place activity itself, is added to the plan. The response with the planning result and cost for the job is sent to the product agent.

Additional service agents can be implemented within the platform to facilitate monitoring of the performance, networking, etc

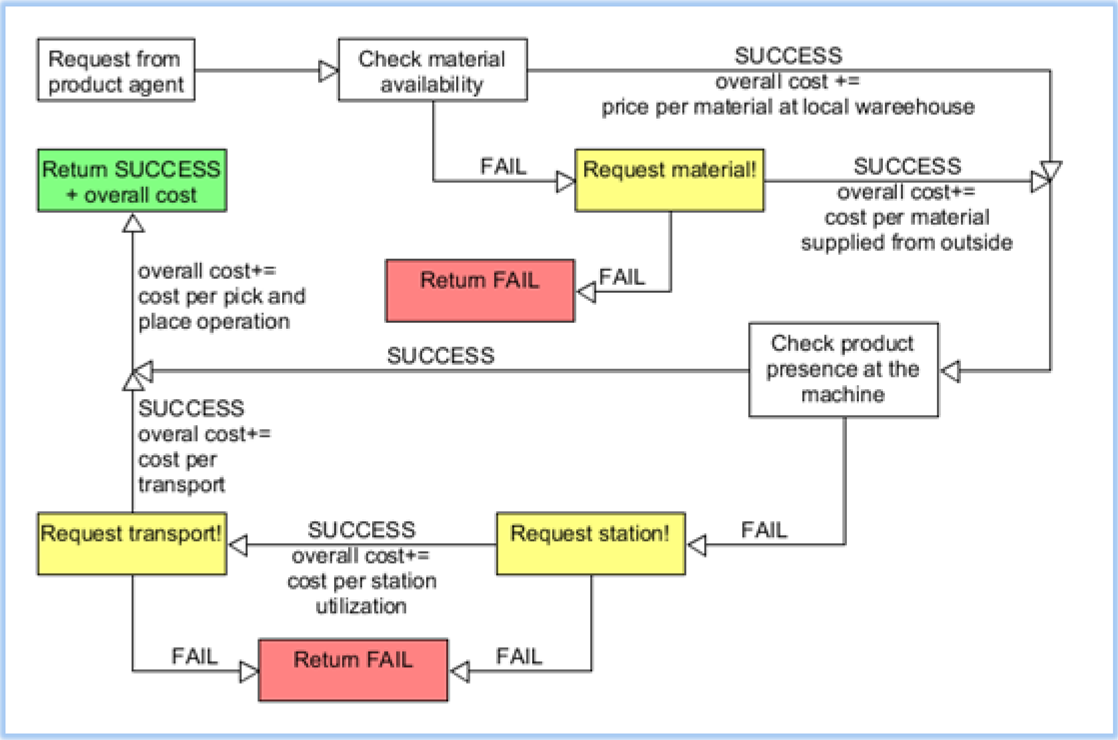

Fig. 7: Assemble job workflow

Station agent – since some of the stations on the production line are shared between two robots, the system must ensure that a given station is reserved for the job executed by a specific robot. For this reason, Station agents are implemented within the platform. These are passive agents. They don't actively negotiate with other agents but only commit to requesting plans if they are not already committed to others. The station agents also provide information about occupancy to other agents if requested.

Shuttle agent – the shuttle agent represents a particular Montrac shuttle on the production line. The manufacturing activity is transporting, and the shuttle agent provides two job implementations, TransportJob and ClearStationJob. The workflow of the transport job is displayed in Fig. 8.

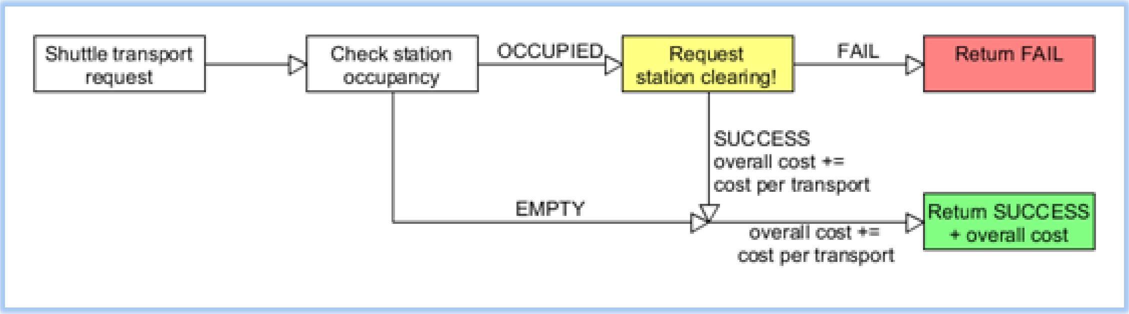

Fig. 8: Transport job workflow

3.1.2. Motif Discovery platform

The Motif Discovery tool (MOD) consists of three core building blocks. The Motif Discovery module, Motif Learning module, and the Online Detection module. MOD tool provides solution for the anomaly detection in observable manufacturing processes.To achieve this, any single instance of MOD analyzes data from a logical machine. A logical device is a collection of sensory data streams assumed to have some systematic interdependency (e.g., collection of temperature, speed, and position data from a motor). For analysis of a single logical device, a separate instance of MOD is started.

The Motif Discovery module consumes a batch of sensory data and produces a dataset of segments of the given time series. This process is both memory and computationally intensive and is done offline.

The Motif Learning module consumes an annotated dataset of time series segments and learns a detection model. This is also a resource-intensive task, and it is done offline.

The Online Detection module requires a detection model as an input at the beginning of its lifecycle. It subscribes to streams of sensory data of the modeled logical device and starts detecting learned motifs online.

When available, this module connects to a high-level model of the manufacturing process and communicates its findings to improve detection performance and to efficiently store a high-level discrete log of events in the cloud. The cloud connection is proxied through a pre-processing node, which compresses/decompresses the data sent to/from the cloud.

Additionally, a monitoring frontend application can subscribe to the MOD tool to fine-tune its behavior.

The overview is summarized in Fig. 9.

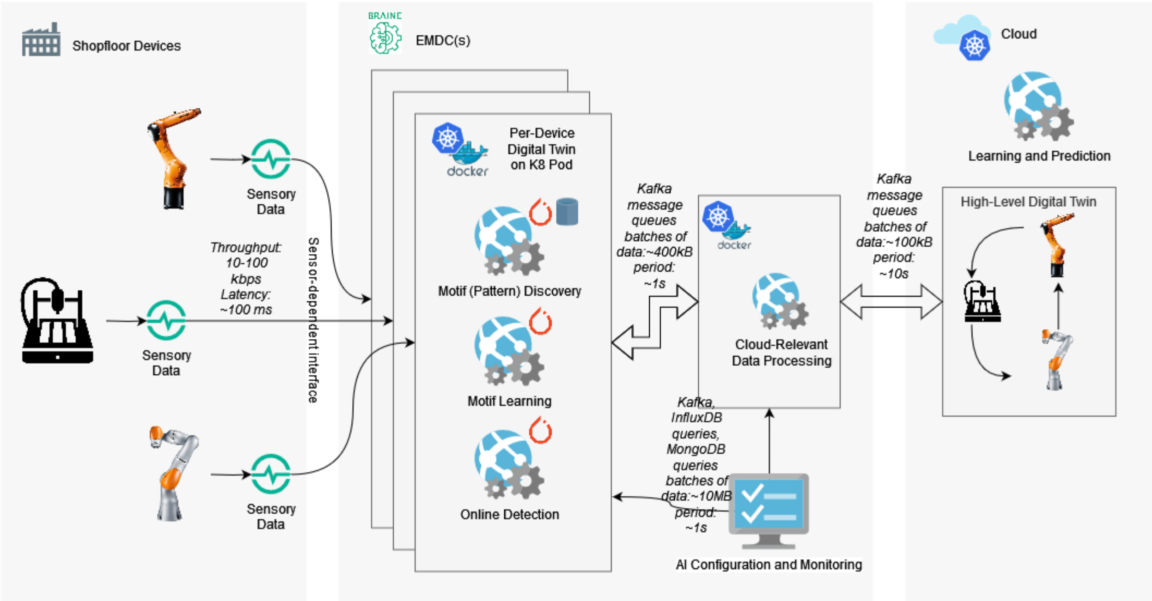

Fig. 9: MOD scheme - The deployment scheme for the MOD tool differentiated by logical devices and by task modules

Data Processing Pipelines Detail

The Motif discovery pipeline assumes that data is stored in the InfluxDB, from which all the time series are queried into memory, preprocessed and then searched for unknown repeating patterns, i.e., motifs. After the search, motifs are clustered according to similarity and stored in patterns container in MongoDB in a form of timestamp references to the raw original time series. This step ends by preparing the labeled training dataset ready for learning pipeline.

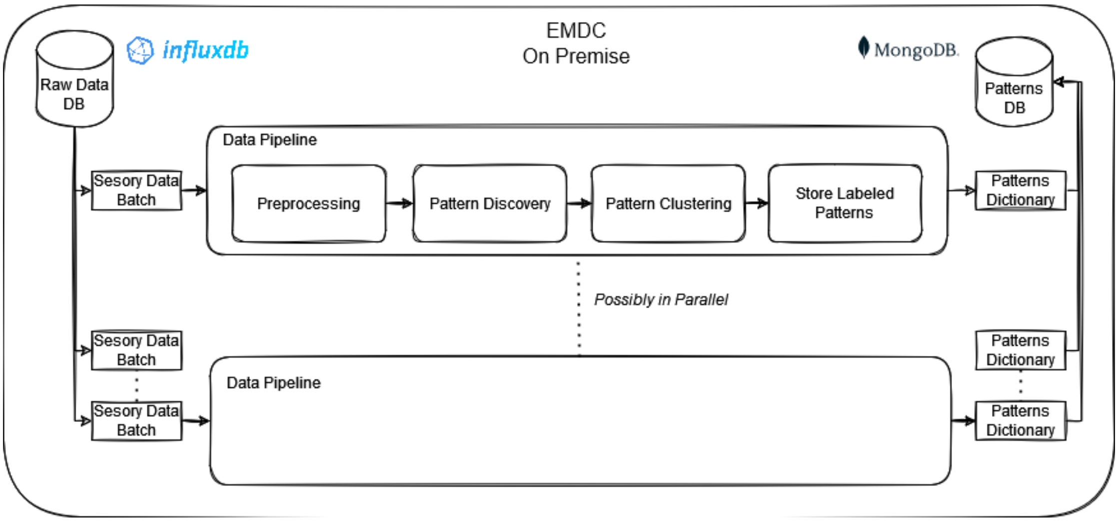

Fig. 10: Motif Discovery pipeline

The learning pipeline obtains the labelled dataset (Patterns Dictionary) and trains a set of probabilistic detection models (Models Dictionary) and stores them both in on-premises MongoDB and in the cloud MongoDB, if cloud is connected.

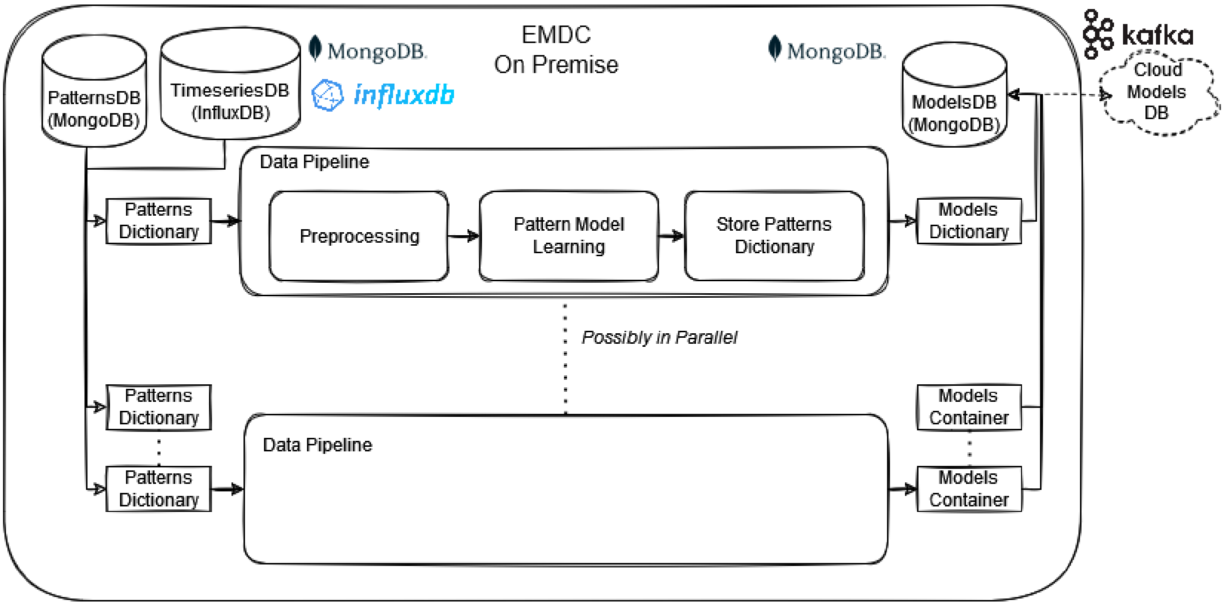

Fig. 11: Detection models learning pipeline

After learning, the detection can be started. Detection pipeline (Fig. 12) consumes data from Apache Kafka topics and feeds them into the detection steps, such as pre-processing, segmentation, pattern detection and finally the result presentation. The resulting detection is sent to the Digital Twin (DTwin) deployed in the cloud, if the cloud is connected.

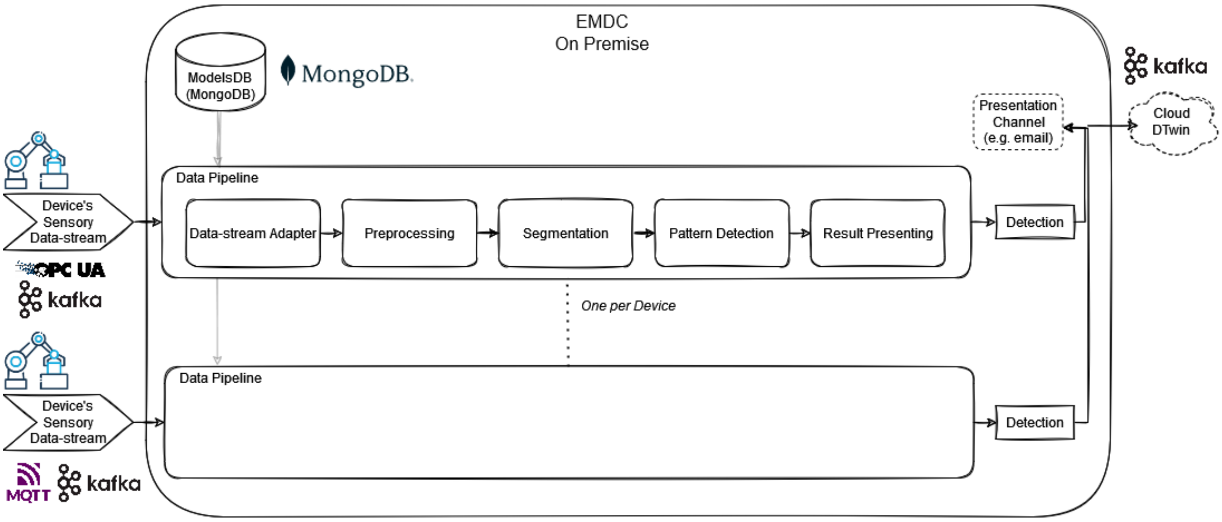

Fig. 12: Detection pipeline

There are several data pumps that can transfer live data streams from various protocols into Kafka topics. In our use case, we utilize the MQTT data collection from the robots in CIIRC Testbed and we utilize the Telegraf data pump to feed the online data into Kafka for detection and into the InfluxDB for motif discovery.

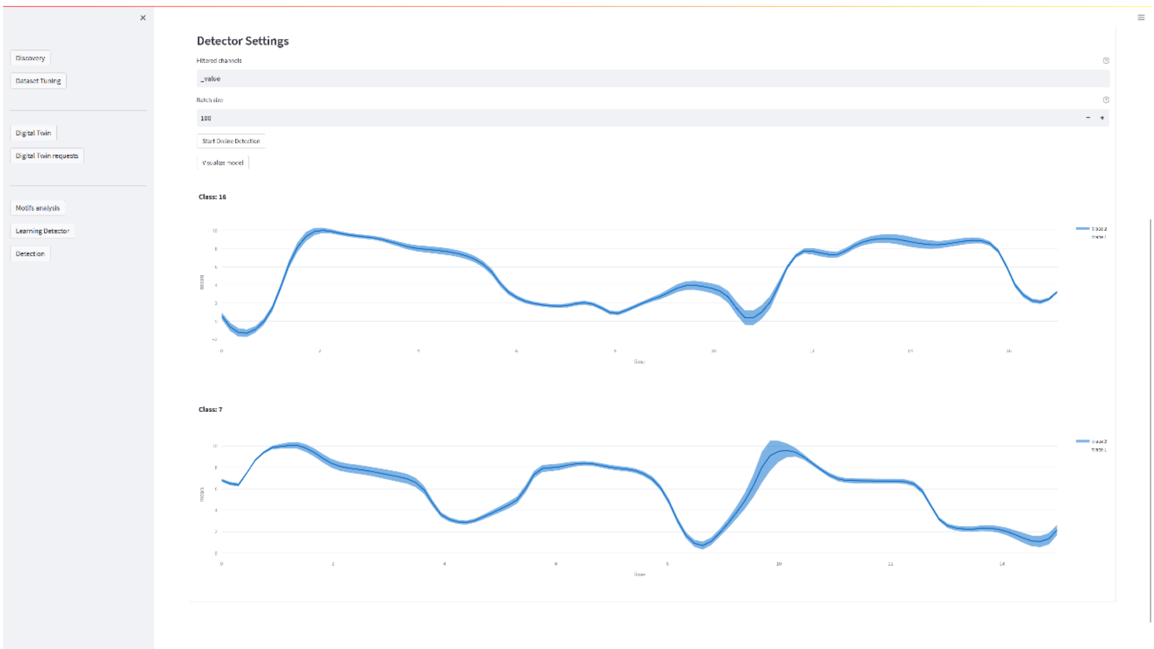

Fig. 13: Detection models example

In Fig. 13, we see an example of the frontend interface for approval of detection models with mean expected value and 2-sigma confidence interval.

3.2. Integration with the BRAINE platform

Each part of the use case, i.e. the multiagent platform and the motif discovery tool, is deployed in containers that run on the BRAINE platform. Each individual agent from the multiagent system is deployed in a separate container.

MOD application utilizes the platforms optimized scheduler through the SLA, where the main advantage is the latency guarantee for the online detection of operational states of machines. BRAINE’s telemetry is used to collect and present data about the state of the processing of the data during Motif Discovery and model Learning as well as health monitoring of the running online detection. The edge to cloud communication utilizes the BRAINE’s service mesh component.

4. Results

4.1. Data Collection

MOD Tool

The data was collected from 3 days of operation of the 3 industrial robots in the CIIRC Testbed. The mean sampling rate was 73 ms and the speed, position and current was measured per each of the 6 axis of each robot. The behaviour of robots showed significant repetitive patterns separated by periods of resting time. The resting periods are easily detectable, and they were removed by pre-processing step, so the KPI evaluation is done only on the meaningful active-time segments.

4.2. Data Analysis

MOD Tool

The testing CNIT Testbed cluster was distributed between CNIT premises, where the 3 worker nodes were located – databases and computing was done there. The CTU premises holds the detached 4th worker node that was used for data collection. This distribution of nodes also impacted the up-front latency in the online detection task as the data needs to be transferred over the internet through VPN tunnel to the CNIT virtual worker nodes. This obstacle is expected to be eliminated during the final testing on the BRAINE EMDC hardware.

4.3. KPIs

KPI 1	Ratio of data processed on Edge to data sent to Cloud	Status:
Result:	Reduction ratio on test dataset collected on robots in CIIRC Testbed show results ranging from **1:6 to , exceeding the target of 1:5**. This ratio was achieved while keeping acceptable reconstruction quality for business level of detail. The high-detailed raw data are stored on-premises in the InfluxDB.	**Achieved**
KPI 2	**Mean delay of deviation detection event**	Status:
Result:	Tests on Siemens devices show the mean latency of detection to be **500 ms + 25 %** of duration of the detection pattern. The up-front latency cost is expected to be lowered	**Achieved**

	by deploying the solution to the BRAINE EMDCs when those are available.	
KPI 3	**Setup and commissioning time**	Status:
Result:	After tests of multiagent platform on in IDE the system was deployed to the Kubernetes cluster running in Testbed for Industry 4.0. The whole process of creating containers for each agent and deploying them to the cluster took less than two hours. A test run of the multiagent platform production line took another hour. Altogether, setup and commissioning time was under three hours which is much less than the baseline. The KPI is therefore met.	**Achieved**
KPI 4	**Commissioning time of robotic systems**	Status:
Result:	During the reconstruction of the Testbed, the topology of the production line was changed. Commissioning of the new topology took two weeks (10 days). When simulating the same change in topology with the multiagent platform, we were able to commission the new topology in under a day (5 hours). The KPI is therefore met.	**Achieved**
KPI 5	**Scalability and modularity of manufacturing lines**	Status:
Result:	When testing the scalability the test scenarios were to add one to five robots to the original setup and measure reconfiguration time. When subtracting the time to spawn a new container the reconfiguration was every time instantaneous (under 1s). At the same time, the system was able to do the reconfiguration on the run which is something the traditional approach is not capable to do. The KPI was therefore met	**Achieved**
KPI 6	**Image based subpart recognition**	Status:
Result:	During evaluation of the detection accuracy, the camera was positioned within a distance range of 0.7m to 1.2m from the tags/objects being detected. The average accuracy obtained from subpart recognition was ±0.788 mm measured in XYZ	**Achieved**

	coordinates. The accuracy achieved for the yaw angle was within a range of ±0.023°. The KPI is met.	
KPI 7	**Image based calibration of position**	Status:
Result:	The KMR robot's initial positioning accuracy was determined to be within a range of ±1cm in the XY plane and ±2.5° in terms of angle. By utilizing a camera mounted on the robot's hand and employing a positioning algorithm that combines the transformation derived from subpart recognition with the known transformation from the camera to the center of the robot, the final accuracy of the robot was improved to be within a range of ±0.4cm in the xy plane and ±1.9% in terms of angle.	**Achieved**

5. Impact

Discuss the use case results in comparison to the state of the art, and also discuss the potential economic impact of the use case and overall BRAINE platform here.

5.1. Comparison to existing systems

In today's complex and dynamic world, efficient planning is crucial for businesses to stay competitive and achieve their goals. Traditional approaches like centralized planning or manual planning have their merits, but they often fall short when it comes to addressing the complexity and unpredictability of modern challenges.

- **Distributed Decision-Making:** Centralized planning relies on a single decision-maker or a small group of decision-makers who bear the burden of analyzing information, making decisions, and coordinating actions. This approach can quickly become overwhelmed when faced with large-scale or rapidly changing scenarios. In contrast, multiagent planning empowers multiple agents to make decisions autonomously based on their local knowledge and expertise. By distributing decision-making authority, multiagent planning leverages the collective intelligence and enables more efficient problem-solving.

- **Adaptability to Uncertainty:** In today's fast-paced business landscape, uncertainty is the new norm. Centralized planning and manual planning often struggle to adapt to unexpected events or changing conditions. Multiagent planning, on the other hand, embraces uncertainty as an inherent part of the system. Agents in a multiagent planning framework can react independently and rapidly to new information, adjust their plans accordingly, and collaborate with other agents in real-time. This adaptability allows businesses to respond effectively to unforeseen challenges and seize emerging opportunities.

- **Scalability and Resource Optimization:** Centralized planning often faces scalability limitations as decision-making becomes increasingly complex and resource-intensive. Multiagent planning overcomes this challenge by distributing the planning process across multiple agents, enabling parallel processing and resource optimization. Agents can independently allocate resources, balance workloads, and coordinate tasks, resulting in improved efficiency and reduced bottlenecks. The decentralized nature of multiagent planning facilitates scaling operations smoothly and accommodating growth without overburdening any single decision-maker.

5.2. Potential Impact to the robotics and manufacturing

The architecture developed for use case 3, with its emphasis on decentralized decision-making, adaptive resource allocation, and advanced analytics, brings a significant boost to production efficiency. `Industrial robots equipped with agents utilizing machine learning algorithms, and artificial intelligence capabilities with the use of advanced, even virtual, sensors can streamline repetitive tasks, eliminate errors, and optimize production workflows. This not only reduces costs but also ensures consistent quality and faster time-to-market for products.

Traditional manufacturing processes often struggle to respond swiftly to changing market demands. However, flexible production architecture provides the agility required for quick adaptations. With modular robotic systems and flexible automation cells, manufacturers can easily reconfigure production lines to accommodate new product variations or customization requests. This adaptability enables manufacturers to address evolving customer needs promptly and capitalize on emerging market opportunities without significant disruptions or retooling costs.

Overall manufacturing process generates vast amounts of data from various sensors, machines, and robots. Leveraging this data through advanced analytics and machine learning techniques empowers manufacturers to gain valuable insights into their production processes. Real-time monitoring, predictive analytics, and anomaly detection enable proactive maintenance, reducing downtime and optimizing equipment utilization. Moreover, data-driven decision-making enhances process optimization, identifies bottlenecks, and facilitates continuous improvement efforts. By harnessing the power of data and analytics, manufacturers can achieve higher levels of operational efficiency, quality control, and resource optimization.

The integration of multiagent production architecture with use of motif discovery provides manufacturers with a competitive edge in the global market. The ability to rapidly adapt to changing customer demands, introduce new product variations, and deliver personalized solutions gives manufacturers a unique selling proposition. Flexible production systems enable cost-effective small-batch manufacturing, reducing inventory costs and enabling just-in-time production. Moreover, by leveraging the platform, manufacturers can push the boundaries of innovation, explore new manufacturing techniques, and introduce cutting-edge products to the market.

5.3. Advantages of the BRAINE platform

There are multiple existing solutions. One of those are cloud platforms such as Azure, AWS or GCP. The cloud platforms brings multiple advantages such as easy scalability, efficient data management, and remote accessibility. They optimize operations, enable real-time analytics, promote collaboration, and ensure data protection, driving innovation in the industrial environment. Unfortunately, a lot of times data protection offered by cloud platforms is not enough for manipulating sensitive production data.

Because of the sensitivity of the data the data have to be anonymized to be able to leave shopfloor or are prohibited to leave shopfloor at all.

In this case only solution is custom on-demand solution that is hard to deploy and maintain, or edge platform (for example Siemens EDGE) which are in most cases not powerful enough and provide only small portion of services provided by cloud platforms.

This is where BRAINE platform can fill the gap. BRAINE provides a comprehensive suite of deployment, monitoring, and AI capabilities inspired by cloud platforms but modified for use on the edge. At the same time offering desired power such as server CPUs, GPUs, and so on.

5.4. Business solutions and economic advantages of BRAINE for robotics & manufacturing

In summary, the BRAINE platform's integrated advanced capabilities, customization options, real-time data processing, seamless integration, and advanced predictive analytics. Its unique features cater specifically to the challenges and requirements of the manufacturing industry, enabling manufacturers to achieve higher levels of efficiency, adaptability, and competitiveness.

In this use case, it has been validated experimentally that the BRAINE platform can perform computationally demanding tasks, which are not feasible for ordinary industrial edge device, which are typically equipped with an Intel i5 or i7 CPU, recently and in rare cases with a GPU. On the contrary, the BRAINE platform with its performance and scalability outperforms such systems.

This brings significant advantages to robotics and manufacturing as this use case also shows. In the scenario of Motiff Discovery as well as in the scenario of the multiagent platform, the performance and closely related low latency of the edge server allow performing computationally demanding tasks from the fields such as machine learning or

AI in real time and closing the feedback from the computation back to the machines. This fact enhances the capabilities of the machines significantly.

6. Conclusion

The use case demonstrated the BRAINE platform as being suitable for computationally demanding tasks, which need to be executed in real time. The solution also showed how flexible architecture of the manufacturing execution system can contribute to being able organize and command the production in a flexible way too.