GeSCo: Introducing an Edge Layer Between Cloud MES and Shop-Floor in Decentralized Manufacturing

Badarinath Katti
TU Kaiserslautern
Kaiserslautern, Germany
email:katti@rhrk.uni-kl.de

Michael Schweitzer
SAP SE
Walldorf, Germany
email:Michael.Schweitzer@sap.com

Christiane Plociennik
DFKI GmbH
Kaiserslautern, Germany
email:Christiane.Plociennik@dfki.de

***Abstract*—Decentralized manufacturing is an active research topic in current smart and open integrated factories, and is probably also the future state of practice in both the process and manufacturing industries. The Manufacturing Execution System (MES) is a comprehensive automation software solution that coordinates all the responsibilities of modern production systems. However, the MES solution is essentially designed as a centralized manufacturing control unit, which goes against the principle of the decentralized manufacturing paradigm. When operated as a cloud based solution, the MES faces another big challenge: connectivity and network latency. This paper addresses the problem of network latency experienced when the Cloud MES (CMES) is in charge of production control by introducing an edge layer near the shop-floor. In other words, the CMES delegates the responsibility of manufacturing control to this edge layer which consequently facilitates decentralization in manufacturing.**

***Keywords*–Decentralized Manufacturing; Edge Computing; Cloud MES; Generic Shop-Floor Connector.**

I. INTRODUCTION

Traditionally, the production was conceived to be a top-down approach comprising of different layers such as Enterprise Resource Planning (ERP) [1], MES, Supervisory Control And Data Acquisition (SCADA) [2] and shop-floor (see Figure 1 left). However, with the advent of low-cost and smart sensors, the MES can directly coordinate with the plant machines. The trend of moving towards standardized communication protocols on all layers of the automation pyramid is fostering the development of circumvention of the vendor-specific SCADA layer as illustrated in Figure 1. In centralized manufacturing, a central entity is responsible for the system planning aimed at the optimization of the objectives of an entire organization [3]. Centralized systems are often complex in design and hence inflexible in cases of unexpected events and product customizations [3]. Decentralized manufacturing systems are based on distributed control in which the local decision-making bodies react to conditions of the shop-floor at real time. Centralized systems have slower response times since they employ complex algorithms and analyze more data. However, the solution quality of decentralized systems may be lower since they are based on local information. Furthermore, they require more communication effort. In terms of robustness, decentralized systems perform better: The failure of the machines at the lower level of the automation pyramid does not cause the whole system to fail. In a typical centralized system, a failure of central entity can cause the catastrophic failure of the entire system [4]. These arguments support the adoption of decentralized control in manufacturing.

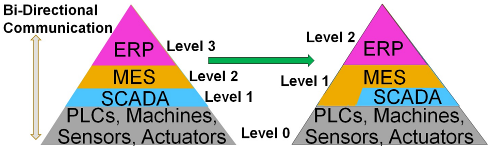

Figure 1: Evolution of classical Automation Pyramid.

The IEC 62264-3:2016 standard [5] divides the entire MES activities into four functional areas namely production, maintenance, quality and inventory management. This paper focuses on the production management aspect of MES. The MES is inherently difficult to own, maintain and evolve owing to the tight coupling of IT infrastructure to the manufacturing operations [6]. In the interest of protection of investment, a detailed feasibility evaluation is necessary as the selection of MES generally results in long term relationship with the MES vendor. Therefore, a recent trend is to move MES to the cloud. Cloud MES (CMES) can quickly adapt to the newer innovative technologies and offer significant cost benefits to the manufacturer. The generic set of functionalities provided by CMES are richer than on-premise counterparts [7]. Another main benefit of the CMES is that it requires nearly no IT resource investment [1] and hence, lowers the IT barriers to innovation in manufacturing processes [7]. The CMES helps face the challenge of peak production demand without additional investment on on-premise resources [8]. Since the cloud servers are run as per the necessity, licenses can be increased or decreased accordingly. However, when the MES shifts from on-premise to cloud, it faces the challenge of the remote resource management and production control. Since modern industries increasingly make decisions by coordinating with business systems, it results in higher network load and latency. To tackle this problem, we propose to introduce an edge layer called *Generic Shop-Floor Connector (GeSCo)* between CMES and shop-floor that caches the routing details and other production related data and hence supports the decentralization in manufacturing. The outline of the paper is as follows: Section II lists related work. Section III highlights the problem of network latency in the context of high speed manufacturing. Section IV introduces an edge layer that acts as production control delegate to tackle the network latency and lists the challenges faced by the edge layer. Section V presents a system architecture that addresses these challenges. Section VI presents implementation details based on the proposed system design and some simulation results. Section VII provides the conclusion and an outlook on the future work.

II. RELATED WORK

1) Edge Analytics and Decentralized Manufacturing: Edge computing is in practice since two decades and is also known

ISBN: 978-1-61208-598-2

by other names such as fog computing, mobile edge computing, cloudlets and cyber foraging [9]. Edge analytics applied to the domain of manufacturing addresses the problem of network latency and enables to take decisions at runtime in production and thus, can adopt to changes in the Production Order (PO) within short time. [10] proposes decentralized work-in-progress manufacturing control that serves as an alternative to the centralized manufacturing systems. The RFID-enabled MES was introduced for mass-customization in manufacturing that faced challenges of manual and paper-based data collection, production plans and schedules [11]. However, the assumption was that machines in the factory shop-floor are at best partially connected and the decision-making rests entirely on employees on the shop-floor. Agent-based manufacturing [6] and holonic manufacturing [12] introduced the concept of artificial intelligence in manufacturing with an aim to respond promptly and correctly to changes in PO. [13] professes the idea of edge datacenters that process the data on behalf of IoT devices and delegate to the cloud only when more complex analysis is required. [14] proposes a Centralized Scheduling System (CSS) and decentralized MES, where the latter follows a fixed global schedule and turns to CSS in case of perturbation. [15] discusses the autonomous MES that generates alternative schedules when given schedule is infeasible. However, [16] argues that localization of decision-making with an obligation to decentralize has the risk of losing the global vision of the network. [17][18] argue that even though the decentralization of manufacturing is the norm in the future, there are cases where a centralized entity is obligatory to overwrite the lower level decisions, e.g., in the event of redefinition of production processes at higher levels of automation pyramid. [19] also contends that the absence of a central decision-making body necessitates continuous harmonization of objectives among the agents leading to high coordinative complexity. Therefore, there is a renewed interest in incorporating centralized production control concepts to manufacturing.

2) Cloud Manufacturing: There have been several works, for example [20][21], in the domain of cloud manufacturing, that combine the emerging advanced technologies, such as cloud computing, virtualization, internet of things and service oriented architecture. The potentials and relationships among cloud computing, internet of things and cloud manufacturing is investigated in [7]. [22][23] illustrate the concept of centrally managed CMES, but its application area is distributed manufacturing which is outside the purview of this paper. In general, the focus has shifted from centralized manufacturing systems - and MES in particular - to the decentralized paradigm of manufacturing. This research paper is novel in the aspect that it focuses on the adaptation of CMES, which is traditionally linked to the centralized paradigm, to the context of decentralized manufacturing. In other words, it attempts to retain a degree of centralized aspects of manufacturing to strike the right balance.

III. CMES Use Case and Network Latency

During production execution, the shop-floor constantly seeks information from MES. The work stations at the shop-floor request MES for routing details at every stage of the production. Each work station collects the operation, Bill Of Materials (BOM), machine parameters and other resource configuration details. Once this information is collected the machine is instructed on how to proceed with that step of the production process. Once that step of the production is completed, the work station informs MES the same along with the generated results. The MES then processes the results and accordingly sets the next operation of the production. This process continues until all the planned operations are executed to manufacture the planned component. During exceptional cases if the need arises, the routing path is changed, as instructed by MES, to accommodate the exceptional situations. For example, the work in progress is diverted to rework station if the concerns regarding the quality of the products are raised.

The communication between MES and shop-floor takes place over WAN, which means that the transmission delay is not bounded [24]. When moving from MES to CMES, network latency becomes an even bigger challenge as the geographical distance and, consequently, the number of intermittent routers increase. Hence, direct client and server communication between the CMES and shop-floor over WAN encounters network latency due to a variety of factors such as nodal processing delay, queueing delay, transmission delay, propagation delay and packet loss, and thus affect the throughput of the network. These delays are explained in the context of Figure 2. The data packets are sent from source to destination via routers r_1 and r_2. Each router has an incoming queue and an outbound link to each of the connected routers. The packet arriving at a router goes through the queue and the router determines the outbound link after examination of the packet header. An incoming data packet is immediately bound to outbound link if the router queue is empty and there are no packets being sent on the outbound link at the time. If the router queue is non-empty or the corresponding outbound link is busy, the incoming packet joins the router queue. This causes a delay which is known as Processing delay d_{proc} and is the key component of network delay. The node also checks for bit level errors in the packet arising while transmitting from the previous node. After this nodal processing, the router directs the packet to a queue that precedes the outbound link. The time a packet spends in the queue while earlier packets are transmitted at the node is called queueing delay d_{queue}. The incoming packet experiences zero queueing delay when the router queue is empty and no other packet is being transmitted by the router. Alternatively, the incoming packet experiences a queueing delay in direct accordance with the length of the router queue. The router transmits the data at a rate known as transmission rate R. When the data packets arrive for a sustained period at a given router at a rate more than its transmission rate, these data packets will queue in at the router. The ratio of $(A*B)/R$, called network traffic intensity, plays an important role in determining the queueing delay, where A denotes the average number of packets that arrive at the router queue per unit time and B is the average number of bits in each of these packets. The qualitative dependence of average

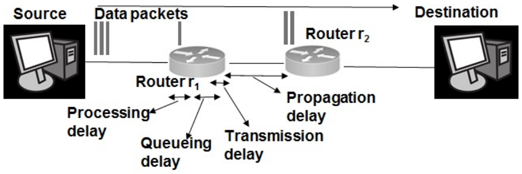

Figure 2: Illustration of network delays.

queueing delay on the network traffic intensity is demonstrated in Figure 3. It can be observed from Figure 3 that as the traffic intensity tends to 1, the average queueing delay grows exponentially. When the packet arrival rate is greater than router transmission rate, the size of packet queue grows at the router. However, this cannot continue indefinitely due to the finite capacity of the router queue. Therefore, the router drops the packet when it finds no place at its queue. Such a dropped packet is lost and this phenomenon is called Packet loss. At this juncture, the client that transmitted the packet to the network core expecting the delivery acknowledgement from the server re-transmits the packet after waiting for a specified amount of time. This reduces the throughput of the network connection. The router takes a finite time to transfer the bits of a data packet onto the outbound link. This time is known as transmission delay d_{trans} and mathematically, it is defined as B/R. The packet on the outbound link propagates to the next node in a time known as the propagation delay. If l is the length of the physical link and v is the propagation speed of the data packet in the physical link, the propagation delay d_{prop} is then given by l/v. The total nodal delay d_{nodal} is then given by,

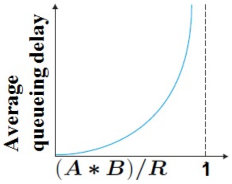

Figure 3: Dependence of average d_{queue} on traffic intensity [25].

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \quad [26] \qquad (1)$$

If there are N number of similar routers between the source and destination spaced apart at equal distances, then the end-to-end delay $d_{end-to-end}$ is measured as,

$$d_{end-to-end} = N*(d_{proc}+d_{trans}+d_{prop})+\sum_{n=1}^{N} d_{queue_n} \qquad (2)$$

where the last part of the above equation is sum of the queueing delays experienced at each of the routers. The network delays are directly proportional to the distance and consequently, the number of intermittent routers, between the client and the server. In practice, with the exception of d_{proc}, which is on the order of microseconds, all other above-mentioned delays are on the order of milliseconds [26][25]. It is not possible to accurately determine the latency between two fixed points since the data packets encapsulated at the network layer of OSI model need to pass through several proprietary routers of the internet before reaching the destination. Each of these routers has unpredictable traffic which is dependent on variety of factors and hence, the network latency is a function of internet traffic that undergoes random fluctuation for the same bandwidth and infrastructure. Therefore, instead of imposing the hard real-time constraints, the practical unit of measurement should be average time for the network latency. The virtualization principle of cloud computing that can be applied at different levels such as computer hardware, operating system, storage and network also introduces its own series of packet delays and causes further performance degradation. The Figure 4 illustrates this situation where there are three operations - welding, color spraying and quality check, that are required to be performed to produce the planned component. In the state of the art industries, the work stations constantly communicate with CMES to seek process parameters, recipe, machine configuration values and push the results during production control. The problem of network latency which is encountered each time the request is created to fetch the next operation details from CMES does not auger well in high speed manufacturing scenarios. In addition, although cloud providers claim near 100% availability, there is an average non-availability of 7.884 hours per year [27]. Such network outages are not acceptable in the event of manufacturing a priority order.

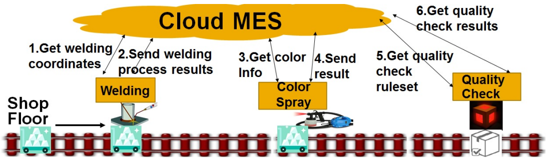

Figure 4: CMES - Shop floor connectivity in production.

IV. INTRODUCING AN EDGE LAYER

As explained in Section III, the network latency is directly proportional to the geographic distance. The MES in cloud is not guaranteed to be close to the site of production. Hence, caching the production control data in proximity to the shop-floor can reduce the problem of network latency. To this end, this research paper proposes introducing an edge layer called *Generic Shop-Floor Connector (GeSCo)* between CMES and shop-floor.

GeSCos are close to, but not tightly coupled to the shop-floor. They control the production processes and collect the data to and from the shop-floor and enterprise software. GeSCos also help in enabling the 'plug and work' feature of today's smart factory, since they can connect to wide variety of industry specific data sources of diverse manufacturers, such as OPC UA, classical OPC and http based web services. Due to the physical proximity of GeSCos and shop-floor, the data communication latency is short as data packets need not cross multiple routers. GeSCos also alleviate the problem of latency introduced by the virtualization layer of cloud infrastructure explained in Section III. Furthermore, the caching strategy facilitates the implementation of *decentralization* of the production execution. In its basic conception, the GeSCo is a web service framework that collaborates with enterprise software and diverse industrial data sources to execute a PO by performing division of labor in the shop-floor under the supervision of CMES, i.e., it distributes the production operations to resources on the shop floor based on the production recipe at run-time. The introduction of GeSCo in the shop-floor is not to take over the role of SCADA. It should just serve as a thin client to CMES server. Based on these arguments, the CMES and the shop-floor communication evolution can be illustrated as in Figure 5.

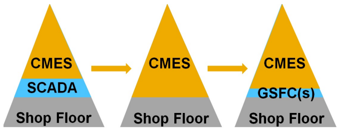

Figure 5: Evolution of CMES - Shop Floor Connectivity.

After the production control data is cached, the intention is to reduce the communication between the GeSCo and CMES as far as possible. Several exceptional situations may arise in the shop-floor while the GeSCo is in control of the production execution. The GeSCo should either resolve or find an alternative course of actions to the prevailing exceptional situations. The objective of this exercise is the successful completion of the production execution. The CMES should support this goal by sending meaningful data at the right time.

A. *Challenges of Integration of GeSCo: A Survey*

The GeSCo should assume the role of the CMES after the PO is transferred to its cache. The transfer of production control to the GeSCo is smooth under normal circumstances when the production encounters no problems. However, the system should be designed such that it should be robust against production fluctuations and should mitigate or solve the problems that may arise under exceptional circumstances.

In order to determine which responsibilities such a system must fulfill, several experts in the field of manufacturing were asked to prioritize the challenges for GeSCo during the execution of shop orders. The results of this survey are, in descending order of their weighted average:

1) Determination of next routing step since business rules that govern the routing decisions are present in the CMES
2) Semantic translation of data arriving from CMES to technology and business agnostic solution such as GeSCo
3) Adaptation in GeSCo in the event of change of the data model in centralized CMES
4) Determination of the suitable resources to perform the current operation
5) Routing-path substitution in the event of machine breakdown [6]
6) Dealing with the change of the PO [6]
7) Handling the POs of high priority [6]
8) Course of action in the event of quality defects
9) Course of action in the event of unavailability of raw materials
10) Distributed manufacturing where components are being manufactured at different sites

V. PROPOSED SYSTEM ARCHITECTURE

The solution architecture should be designed taking into account the challenges mentioned in Section IV-A. It should enable the CMES to exercise control over the production process while at the same time ensuring a smooth integration of the GeSCo for providing flexibility in exceptional cases. Hence, the architecture should incorporate both centralized and decentralized aspects.

A. *Design of CMES*

This section describes the proposed set of building blocks and services that are required in the CMES. The overall architecture is depicted in Figure 7.

1) Production Planning System: This application layer enables the human production planner to plan the production sequence in a generic way. To this end, it has different maintenance user interfaces that help define the plant and product definition, operation planning and production execution aspects. This master data facilitates the design of BOM and the shop-floor routing for a product variant. This unit also enables the human to create and release the PO to the shop-floor.

2) Manufacturing Resource Model and Servitization: Remote resource sharing and management is a challenge to CMES since it is geographically separated from the shop-floor. The resource virtualization is the key idea behind building the cloud services in the context of manufacturing. The resource model is the transformation of a real manufacturing resource to a virtual or logical resource. Each manufacturing resource is modeled formally with a set of inputs and outputs according to its main functionality. The functional and non-functional capabilities of the resource can be semantically modeled. The model is then subjected to real-to-virtual mapping methods to map to a logical resource as illustrated in Figure 6. The virtual resource servitization is the transformation of abstract concepts of capabilities provided by these resources into formal services that are understandable by the cloud platform. This process involves several aspects such as definition of the service model, message model, ports and protocols. The service model includes the template for the service offered by cloud platform. The reception of inputs and generation of outputs of the service is defined in the message modeling process. The port modeling involves the definition of functional operation port used to accomplish the operation target. The protocol binding specifies the different protocols that are supported by the service.

Figure 6: Resource Virtualization.

This service interface of resource enables GeSCo to store the resource relevant data in resource model, which is resource digital twin. The GeSCo collects the machine data from resource periodically and pushes it to CMES resource model, which is required for real time resource monitoring and calculate the equipment effectiveness. The data is also archived and the aggregated historical data is fed to the predictive analytics tool to find the insights into the resource behavior.

3) Dispatcher: The PO created and released by the production planner is transferred from the CMES to the shop-floor by the dispatcher. The logic of transferring the priority order(s) is pre-loaded into the dispatcher. The parameters that expedite the release and subsequent transfer to the shop-floor are production end date, priority customer, and inventory and manufacturing resource availability. The GeSCo, introduced in this paper, is a technology and business agnostic solution. Therefore, the dispatcher should send the unambiguous data, for example, a collaborative product definition and operations semantic model to the GeSCo. The GeSCo translates this information to its compatible data model for further processing.

4) Data mining and predictive analytics: Instead of relying on human expertise alone, there is an increasing inclination towards aggregating and processing a large amount of data at the shop-floor, which in turn enables to train better models for classification, clustering and prediction. This component analyzes the current and past semi-structured or unstructured data and extracts useful patterns and transfers this knowledge to GeSCo. This knowledge of past experience is then helpful for GeSCo to take run-time decisions that solve or mitigate the problems arising in the shop-floor during production. This information is also helpful to achieve optimization of the production processes in the shop-floor.

5) Information systems: This constituent stores the product genealogy including complete work instructions, components and phantom assemblies, operation flow and routing, manufacturing resources and work centers employed, bill of materials, activities on the shop-floor, rework instructions and the discrepancies. This is realized using the Digital Object Memory (DOMe) [28] which maintains all the information about a product instance over its production lifecycle, where each product is identified and tracked using RFID tag that contains the unique shop-floor control number. Since DOMe is centrally accessible to all the involved entities of production, it enables production coordination among these entities, compilation of the historic manufacturing report, quality investigations and process improvements.

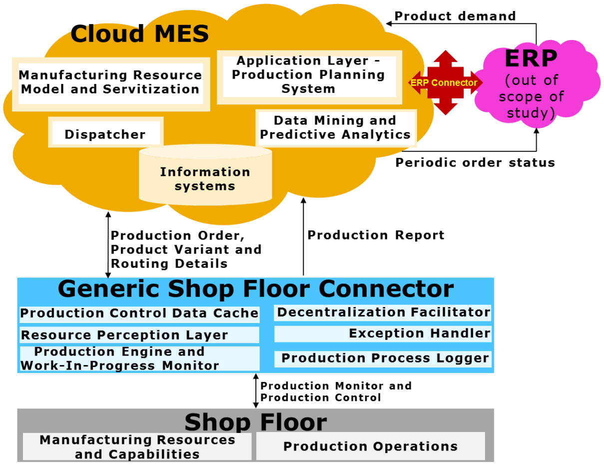

Figure 7: Integration of GeSCo with CMES.

B. Design of GeSCo

The GeSCo should consist of the following components with dedicated responsibilities (see also Figure 7):

1) Manufacturing Resource Perception Layer: To achieve harmonization among various manufacturing resources, they need to be coupled together. The perception layer undertakes this responsibility of loose coupling of different resources on the shop-floor. The different manufacturing resources at the site also register themselves to this layer. The registration can take place either with the resource meta-data or the resource endpoint that permits the perception layer to browse the resource data structures to extract the meta-data of the resource. This data is transferred to decentralization facilitator component which enables it to take decisions at run-time. The perception layer should support the standard industrial communication protocols, such as OPC UA, classic OPC and HTTP based data sources. These IoT protocols are employed to perceive different manufacturing resources with an intent to enable intelligent identification, detection, communication, tracking, monitoring and management. The effectiveness of this exercise hinges on the ability of this layer to extract the key information from the real resources.

2) Production Control Data Cache: This component stores the data delivered by the CMES. It contains the blueprint of the production execution on the shop-floor, which is the detailed routing information in the case of discrete manufacturing. Various entities of GeSCo such as decentralization facilitator and production engine base their decisions and actions on this cached production execution data. This unit is designed to address the first three challenges listed in Section IV-A.

3) Decentralization Facilitator: This entity enables the decentralization in the manufacturing by coordinating with various manufacturing resources and CMES, and thus helps address the challenge of determining the suitable resources for a particular operation. The layer maintains the virtual resource pool consisting of a collection of virtual manufacturing resources. It is used in run-time classification of resources that aids in on-demand resource capability matching. The virtual resource management helps GeSCo identify capabilities intelligently by semantic searching of suitable services and the manufacturing resources on the shop-floor to meet the production requirement.

4) Exception Handler: This block of the GeSCo is accountable for overcoming any shortcomings that arise in the production environment. These shortcomings are explained in Section IV-A, numbering from 5 to 9. The exception handler either attempts to find alternate course of action by local coordination or seeks further instructions from the centralized entity which has global picture of the system.

5) Production Engine and Work-In-Progress Monitor: The production engine is the heart of the GeSCo that collaborates with all the other components of GeSCo to achieve the end goal of successful completion of the PO. It fetches the PO information and routing details from the production control data cache and delegates the responsibility of matching the manufacturing resources for the given operation to the decentralization facilitator. After the decision-making process, the production engine delegates the job to the perception layer that assigns the operation to the real resources after the necessary configuration. The PO is put on hold in the event of non-availability of default and alternate resources, and is only resumed after the required resource registers to the perception layer. To ensure the production is running as expected, it is necessary to monitor run-time status and respond to changes. In case of changes and exceptions, this layer coordinates with decentralization facilitator and exception handler to solve or mitigate the contingency. The production engine also has the intelligence to recognize the situations where GeSCo cannot take the optimal decision based on local information. In such scenarios, it seeks the master data, the singular source of truth, stored in centralized CMES.

6) Production Process Logger: This component uploads the variety of knowledge it gathers during the production onto the CMES. This unstructured data is subjected to analysis and an effort is made by CMES to find patterns and transform it into a structured data. This knowledge in turn can be channeled as a feedback to the closed loop system in order to optimize the production in the long run.

VI. IMPLEMENTATION

In order to evaluate the above mentioned findings, the author simulated the shop-floor behavior by implementing the prototype of the architecture shown in the Figure 7. A CMES was developed that mocks the real CMES in the context of production planning and execution. The SAP Plant Connectivity (SAP-PCo) [29], which is a framework of set of services and management tools was chosen to act as GeSCo. During the research, the PCo was architecturally enhanced to cache the production control and routing data, which is also known as Enhanced Method Processing (EMP). A web server was designed inside a SAP-PCo agent instance and its operations were hooked on to the Dynamically Linked Libraries (DLLs) embedded with the production control logic. The shop-floor is simulated via a series of Raspberry Pi3 units that act as resources that receive the control instructions from the PCo during production. For the purpose of this simulation, the CMES was geographically separated by approximately 1000km from the GeSCo and mock resource work station deployments to reproduce the typical network latency, where as the GeSCo and resource work stations were deployed on the same Local Area Network (LAN). A production process without exceptions was simulated to address the challenges 1 and 4 from Section IV-A with different product types of lot size 1, where production routing contained operations that were distributed to resources in a random manner. Two POs with 5 and 3 operations respectively in their routing plan were created in CMES in order to measure the network latency encountered during the production execution. The latency times were measured in the SOAP UI tool [30]. Tables I and II provide the simulation results w.r.t. the network latency encountered by POs without and with GeSCo, respectively. The total latency encountered by the PO showed a marked decrease in simulation with the edge layer. The research concept was also implemented in the open integrated factory that SAP along with other partners showcased in *Hannover Industrial Fair - 2017*, which verifies the assumption that the result of simulations is valid under real manufacturing conditions.

TABLE I: SIMULATION RESULTS WITHOUT GeSCo

Number of Operations in PO	5	3
Client - Server Entities	Resource - CMES	Resource - CMES
Network Latency Per Call	∼400 ms	∼400 ms
Client-Server calls	10	6
Total Network Latency suffered by PO	∼4000 ms	∼2400 ms

TABLE II: SIMULATION RESULTS WITH GeSCo

Number of Operations in PO	5		3	
Client - Server Entities	GeSCo-CMES	GeSCo-Resource	GeSCo-CMES	GeSCo-Resource
Network Latency Per Call	∼400 ms	∼30 ms	∼400 ms	∼30 ms
Client-Server calls	2	10	2	6
Total Network Latency	∼800 ms	∼300 ms	∼800 ms	∼180 ms
Total Network Latency suffered by PO	∼1100 ms		∼980 ms	

VII. CONCLUSION AND FUTURE WORK

This paper argues that the CMES is better suited in changing production environments than traditional MES solutions. To overcome the problem of network latency associated with CMES and also achieve decentralization in manufacturing, an edge layer called GeSCo is introduced and a comprehensive architecture is designed to integrate this edge layer with the CMES. Future work includes further refinement in realization of decentralization, development of semantic data model for GeSCo, research on the extent of caching under given conditions and handling of priority orders.

ACKNOWLEDGMENT

This work was supported by a doctoral grant from SAP SE. A patent, with ID 62/566551, on the research of this paper has been filed in USA with the authors as the inventors.

REFERENCES

[1] A. Lenart, *ERP in the Cloud – Benefits and Challenges*. Springer Berlin Heidelberg, 2011, pp. 39–50.

[2] D. Wu, M. J. Greer *et al.*, "Cloud manufacturing: Strategic vision and state-of-the-art," *Journal of Manufacturing Systems*, pp. 564–579, 2013.

[3] G. Saharidis, Y. Dallery *et al.*, "Centralized versus decentralized production planning," *RAIRO-Oper.Res., 2006*, vol. 40, no. 2, p. 113128.

[4] C. Anderson and J. Bartholdi, "Centralized vs Decentralized control in manufacturing: lessons from social insects," in *Complexity and Complex Systems in Industry,2000*, pp. 92–105.

[5] "IEC 62264-3:2016," URL: https://www.iso.org/standard/67480.html [accessed: 2017-10-17].

[6] P. Leitao, "Agent-based distributed manufacturing control: A state-of-the-art survey," in *Engineering Applications of Artificial Intelligence, 2009*, pp. 979–991.

[7] S. Marston, Z. Li *et al.*, "Cloud computing, The business perspective," in *Decision Support Systems, 51, 2011*, pp. 176–189.

[8] T. Wood, A. Gerber *et al.*, "The case for enterprise-ready virtual private clouds," in *Proceedings of the HotCloud 2009*.

[9] T. Luana, L. Gao, Z. Li *et al.*, "Fog computing: Focusing on mobile users at the edge, 2015," *CoRR*, vol. abs/1502.01815.

[10] H.Loedding, K.W.Yu *et al.*, "Decentralized wip-oriented manufacturing control," *Production Planning & Control, 2003*, pp. 42–54.

[11] R. Zhong, Q.Y.Dai *et al.*, "Rfid-enabled real-time manufacturing execution system for mass-customization production," *Robotics and Computer-Integrated Manufacturing, 2013*, pp. 283–292.

[12] A. W. Colombo and R. Neubert, "An Agent-Based Intelligent Control Platform for Industrial Holonic Manufacturing Systems," 2006.

[13] D. Georgakopoulos, P. Jayaraman *et al.*, "Internet of things and edge cloud computing roadmap for manufacturing, 2016," pp. 66–73.

[14] C. Pach, T. Berger *et al.*, "Orca-fms: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," *Computers in Industry, 2013*, pp. 706 – 720, 2014.

[15] P. Valckenaers, H. V. Brussel *et al.*, "Schedule execution in autonomic manufacturing execution systems," *Journal of Manufacturing Systems, 2007*, vol. 26, no. 2, pp. 75–84.

[16] B. Montreuil, J.-M. Frayret *et al.*, "A strategic framework for networked manufacturing," *Computers in Industry, 2000*, vol. 42, pp. 299 – 317.

[17] M. Marquesa, C. Agostinho *et al.*, "Decentralized decision support for intelligent manufacturing in Industry," *JAISE, 2017*, pp. 299–313.

[18] B.Hubanks, *Self-organizing military logistics*. Cambridge,Mass., 1998.

[19] D. Mourtzis and M. Doukas, "Decentralized manufacturing systems review:challenges and outlook," *Logistics Research,2012*, pp. 113–121.

[20] D. Wu, M. J. Greer *et al.*, "Cloud manufacturing: Strategic vision and state-of-the-art," *Journal of Manufacturing Systems, 2013*, vol. 32, no. 4, pp. 564–579.

[21] F. Tao, Y. Cheng *et al.*, "Cciot-cmfg: Cloud computing and internet of things-based cloud manufacturing service system," *IEEE Transactions on Industrial Informatics, 2014*, vol. 10, no. 2, pp. 1435–1442.

[22] P. Helo, M. Suorsa *et al.*, "Toward a cloud-based manufacturing execution system for distributed manufacturing," *Computers in Industry, 2014*, vol. 65, no. 4, pp. 646 – 656.

[23] L. Zhang, H. Guo *et al.*, "Flexible management of resource service composition in cloud manufacturing," in *International Conference on Industrial Engineering and Engineering Management, 2010*.

[24] "CISCO Wiki," URL: https://docwiki.cisco.com/wiki/Introduction_to_WLAN_Technologies [accessed: 2017-10-17].

[25] J. Kurose and K. Ross, *Computer Networking: A Top-Down Approach*. New York: Addison-Wesley. p. 35-42., 2013.

[26] N. Weng and T. Wolf, "Characterizing Network Processing Delay," 2004.

[27] C. Cerin *et al.*, "Downtime Statistics of Current Cloud Solutions," 2014.

[28] J. Haupert, "Domeman : Repraesentation, verwaltung und nutzung von digitalen objektgedaechtnissen," Ph.D. dissertation, 2013.

[29] "SAP PCo," URL: http://help.sap.com/pco [accessed: 2017-10-17].

[30] "SOAP UI," URL: https://www.soapui.org/ [accessed: 2017-10-17].