GeSCo: Exploring the Edge Beneath the Cloud in Decentralized Manufacturing

Badarinath Katti*, Christiane Plociennik†, and Michael Schweitzer‡
*Technische Universität Kaiserslautern
Kaiserslautern, Germany
Email: katti@rhrk.uni-kl.de
†DFKI GmbH
Kaiserslautern, Germany
Email: christiane.plociennik@dfki.de
‡SAP SE
Walldorf, Germany
Email: michael.schweitzer@sap.com

***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 the advantages and downsides of various MES offerings are explored in anticipation of changing production environments, the Cloud MES (CMES) emerges as the most flexible and affordable solution. However, when operated as a cloud based solution, the MES faces another big challenge: connectivity and network latency. To address these problems, we introduce an edge layer called Generic Shop-Floor Connector (GeSCo) 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. Finally, the detailed experimental evaluations suggest a marked decrease of the network latency after the introduction of GeSCo layer.**

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

I. INTRODUCTION

A prior version of this work has been published in [1].

Traditionally, the production was conceived to be a layered top-down approach and a corresponding architecture, known as *automation pyramid*, comprising of different layers such as Enterprise Resource Planning (ERP) [2], MES, Supervisory Control And Data Acquisition (SCADA) [3] and shop-floor. This architecture pattern supports the sensors, actuators, numerous software and hardware systems that perform the manufacturing operations on the one hand, and on the other hand also contain management or planning systems that provide access to the enterprise information. There are various flavors of automation pyramid proposed based on different research paradigms. For example, [4] proposes an evolved automation pyramid that supports networked and decentralized production. However, this research work considers the classical automation pyramid (see Figure 1 left) as its base working model.

With the advent of low-cost and smart sensors and subsequently Cyber Physical Systems (CPS), the sensors that are connected to the machines are now *reachable* as they have online capability. Thus, the manufacturing execution systems can directly co-ordinate with the plant machines. This development has given rise to the possibility of omitting the SCADA layer, the responsibilities of which can be taken over by the manufacturing execution. In many cases, SCADA systems and the connectivity solutions from the MES layer through the SCADA down to the shop floor have been characteristically vendor-specific. They do not follow industry standards and thus make it difficult to replace machines on the shop floor level. The trend of moving towards standardized communication protocols on all layers of the automation pyramid is also fostering this development of circumvention of the SCADA layer as illustrated in Figure 1.

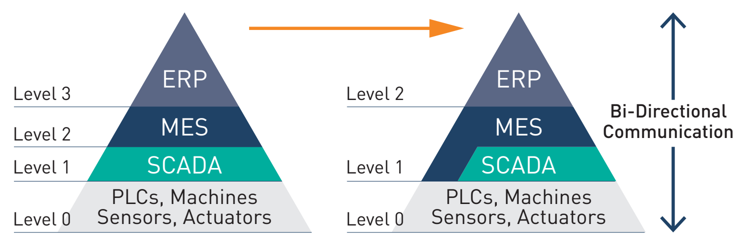

Figure 1: Evolution of classical Automation Pyramid.

In centralized manufacturing, a central entity is responsible for the system planning aimed at the optimization of the objectives of an entire organization [5]. The centralized system is often complex in design and tailor-made to solve a specific class of problems. Centralized systems have slower response times since they employ complex algorithms and analyze more data. In cases of unexpected events and product customizations, centralized systems have proven to be inflexible [5]. 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. This necessitates the coordination of the supply chain where the operational decisions and activities are shared accurately and in time bound manner with all the entities to avoid uncertainties. However, the solution quality of decentralized systems may be lower since they are based on local information. Furthermore, they require more communication effort.

Both the above-mentioned modes of control in manufacturing: centralized and decentralized, are viable depending on the manufacturing circumstances. For instance, the centralized manufacturing is preferred when the manufacturing process involves complex but static procedures [6] such as the determination of the best locations for a set of warehouses and cross-docks. The reasoning behind this choice is that a centralized decision-maker arrives at a decision not based on a certain local maximum, but on a global solution.

When the manufacturing process expects dynamic changes during production, decentralized systems are preferred. 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 [6]. These arguments support the adoption of decentralized control in manufacturing.

The topic of this paper is cloud MES. The MES historically has been a vendor and industry specific solution and hence, is also called by other names such as Collaborative Production Management (CPM) and Manufacturing Operations Management (MOM) [7]. The IEC 62264-3:2016 standard [8] divides the entire MES activities into four functional areas namely production, maintenance, quality and inventory management. Typical functionalities of production management in MES include sequencing the operations, monitoring the production and to determine the states of different entities involved in production with respect to real time. The focus of this research work is the production management aspect of MES. Although decentralization is the norm of smart manufacturing, MES is conceptually a centralized control system. This does not auger well for current trend of production automation and data exchange methodologies. Therefore, MES should also evolve to adopt the new innovative manufacturing techniques. In the context of manufacturing controlled by centrally managed cloud MES, the research target is to designate the task of production control to the edge layer near the production site. Subsequently, this edge layer enables decentralized decision-making in production control.

This paper explores the various flavors of MES and lists the advantages and the downsides of each of the MES types. It argues that among the various MES, the cloud based MES offers a range of affordable functionality without the problem of vendor lock-in. However, when the MES shifts from on-premise to cloud, it faces the challenge of remote resource management and production control. To enable decentralized production, we propose to introduce an edge layer called *GeSCo* between CMES and shop-floor that coordinates with the shop level entities to perform the task of Production Order (PO) execution. Since modern industries increasingly make decisions by coordinating with business systems, this results in higher network load and latency. To counter these cloud related performance issues, GeSCo also caches the routing details and other production related data of CMES.

The outline of the paper is as follows: Section II lists related work. Section III explores various MES adoptions across the industries to give the audience sufficient background before introducing the research question. Section IV describes the use case of the paper and Section V highlights the problem of network latency in the context of high speed manufacturing. Section VI introduces an edge layer that acts as production control delegate. It then also describes the responsibilities such an edge layer should fulfil in Section VII. Section VIII presents a system architecture that addresses these challenges. Section IX presents implementation details based on the proposed system design and some simulation results. Section X provides the conclusion and an outlook on the future work.

II. Related Work

This section is divided into two parts. The subsection II-1 catalogues the previous and contemporary research in the area of decentralized manufacturing, where as the subsection II-2 details the specialized work in the domain of cloud manufacturing.

1) Edge Analytics and Decentralized Manufacturing: Edge computing is in practice since two decades and is also known 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 PO within short time. Moreover, [10] declares that owing to the rigidity and low receptiveness to changes in the manufacturing, centralized manufacturing practices were replaced by decentralized manufacturing models. [11] 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 [12]. 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 [13] and holonic manufacturing [14] introduced the concept of artificial intelligence in manufacturing with an aim to respond promptly and correctly to changes in PO. [15] 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. [16] recognizes the issues relating to cloud computing such as latency and low Quality of Service (QoS), and argues that edge computing is the solution. It also proposes an extensible edge server architecture as an ongoing work. [17] 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. [18] discusses the autonomous MES that generates alternative schedules when given schedule is infeasible. However, [19] argues that localization of decision-making with an obligation to decentralize has the risk of losing the global vision of the network. [20][21] 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. [22] 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 [23][24], in the domain of cloud manufacturing, that combine the emerging advanced technologies, such as cloud computing, virtualization, internet of things and service oriented architecture. In a broad categorization, two types of cloud computing adoptions in the manufacturing are proposed, namely, direct adoption of cloud computing in manufacturing and centralized management of distributed resources that are encapsulated as cloud services [25]. The latter categorization is also known as distributed manufacturing. The potentials and relationships among cloud computing, internet of things and cloud manufacturing is investigated in [26]. [27][28] illustrate the concept of centrally managed CMES, but its application area is distributed manufacturing, which is outside the purview of this paper. [15] also argues that even though cloud datacenters provide cheaper and unlimited computing power, the fundamental practice of storing the manufacturing control and associated exception handling data necessary to successfully carry out production in the cloud datacenter is increasingly being challenged due to rapidly growing requirement of making production decisions with minimum data processing delays and data transfer to facilitate smart manufacturing. There is no research work that focuses on enabling the edge computing when CMES is in control of production to counter the problem of connectivity and network latency. In general, the research focus in the domain of manufacturing 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. MES Taxonomy

This section explores the various available generic MES offerings that are adopted across the manufacturing industries. In this industry-neutral MES study, the advantages and the downsides of each of the MES type are weighed up in order to make a applicability assessment from a manufacturer point of view.

1) In-House MES solution: This approach involves implementation of a customized MES that fits to the specific needs of the manufacturer. It involves direct interaction of the users and developers. The manufacturer should own a group of business analysts and developers with a common reporting line to facilitate smooth coordination between the teams. These human resources should have vast experience in IT implementation and a thorough understanding of the business processes. The home-grown MES allows the user to have an in-depth knowledge of the system functionality and hence, complete control over the manufacturing processes. The organization has proprietary rights of the software and also possesses the knowledge that was gained during software development. The responsibility of software maintenance over a long period of time rests with the manufacturer. At the same time, the organization should adapt to changing business requirements and newer technologies. A MES is inherently difficult to own and maintain and even more rigid to evolve owing to the tight coupling of IT infrastructure to the manufacturing operations [12]. This characteristic aversion of MES to change quickly also hinders the implementation of a streamlined production process. The inability of early adoption of new innovative technologies adversely impacts the revenue generation of the manufacturer.

2) Proprietary production control system: The proprietary production control system that is part of the automation hardware is another facet of MES. These production control systems convert the ERP orders to technical production orders for the assembly lines. This allows the manufacturer to do away with the development of the software and hence, lessen the cost burden. It also enables the manufacturer to immediately focus on the production. However, such production control systems are tightly coupled to the machinery and hence, even a small change in production creates a ripple effect across the automation layers. When the hardware and subsequently the production control software is discontinued, future manufacturing maintenance is not safeguarded. This compromises the flexibility and future security of the entire plant.

3) Third Party Vendor MES: A third classification is the third-party vendor MES which is built according to the functional model specified by the manufacturer. The vendor guarantees long term maintenance and further development of MES modules, and integrate future customer requirements in the product design and development. The selection of MES generally results in long term relationship with the MES vendor in the interest of protection of investment.

The vendor provides a proven and off-the-shelf solution that incorporates industry best practices, along with professional support and training to the work force of the manufacturing organization. On the other hand, the continuation of the status-quo after successful installation of MES is expensive since it involves upgradation of hardware components and IT solutions owing to their short innovation cycles. The additional difficulties such as platform dependency, license model and work force that needs to be trained to use the software come to the fore and further increase the cost pressure on the manufacturers. Moreover, these custom built MES command a high price which is difficult to justify for some of the manufacturers. To that end, a detailed analysis of investment is necessary taking into account the life cycle and cost of maintenance during the feasibility evaluation of an MES vendor.

4) Cloud based MES: To address the above described difficulties, the traditional MES should be replaced by a comprehensive MES setup that can quickly adapt to newer innovative technologies and offer significant cost benefits to the manufacturer at the same time. The cloud based MES [24] is one such solution. The cloud based MES is a blend of various IT technologies such as distributed computing, internet technology, hardware virtualization and open source software. To be more precise, Internet of things (IoT), which enables the perception, internet connection, acquisition and automatic control of various manufacturing resources and capabilities, is the core enabling technology for the implementation of cloud based MES [24]. Cloud based solutions, in general, are best described as web based solutions that run on remote servers and accessed over the internet via standard web browsers [2]. Cloud MES solutions are offered as IaaS (Infrastructure as a service), PaaS (Platform as a service) and SaaS (Software as a service) layers in the cloud architecture that are demand driven and charged as per usage [29].

The services in cloud based MES are generated by virtualizing and encapsulating the perceived manufacturing resources and capabilities [30]. These MES solutions are mostly assembled from configurable software components. The generic set of functionalities is built as per the customers requirements and typically, the functionalities provided by cloud based MES are richer than on-premise counterparts [26] and are also simple, fast and cheap [31]. Another main benefit of the cloud based MES is that it requires nearly no IT resource investment [2]. This lowers the entry costs for smaller firms that try to benefit from compute-intensive business analytics that were previously available only to large corporations. This also lowers the IT barriers to innovation in the manufacturing processes [26]. The cloud based MES helps smoothly face peak production demand without additional investment on on-premise IT resources [32]. This is made possible with the virtualization principle of cloud computing technology. The cloud virtualization facilitates multiplexing of a physical equipment by a privileged hypervisor kernel, thereby providing the end-users separate environments to execute their applications. The argument of virtualization holds true also in cases of redundancy or upgrade costs of the on-premise resources. The dearth of skilled resources that are acquainted with MES technology, achieving the ROI and technology compatibility are no longer the problems in the cloud scenario. Since the cloud servers are run as per the necessity, licenses can be increased or decreased accordingly. This decision need not be made upfront.

IV. Use Case

The communication between the both on-premise and cloud MES can roughly be described as follows. 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 or conflicting goals, 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.

V. Challenges in Cloud based MES

Section III-4 has presented the cloud based MES and its advantages over the classical MES. Nevertheless, there are certain challenges in the cloud MES, or cloud computing technology in general. The cloud downtime and network latency are critical concerns for the manufacturer. This latency becomes even more challenging in high speed manufacturing scenarios where the right information is required at the right time. It is a difficult proposition to measure the exact latency of the network. Historically, it depends on the number of router hops between the client application and the target machine. The network latency is fairly measurable or predictable when the intervening routers are governed by the same corporation. The situation changes when the business migrates to cloud and the issue of latency becomes increasingly complex. The notion of enterprise data centers is no longer followed. The nature of applications is changing from being contained in a local infrastructure within an organization to distributed across the world. Since these applications are deployed across the world, they have varying degrees of latency that are based on each of the internet connections. Hence, the location of data centers plays a significant role in determining the network latency. Furthermore, the network latency is a function of internet traffic that undergoes random fluctuation for the same bandwidth and infrastructure.

The loss of governance is perceived as another biggest impediment to acceptance of cloud based manufacturing solutions [33]. When business applications are moved to cloud, it forces the organizations to accept the control of the service provider on several important issues and areas of business and manufacturing data. As a result, the cloud solution provider will have overarching influence on the business processes. The fact that the valuable enterprise data resides outside the company firewall raises serious security and privacy concerns. However, the security concerns of the cloud based MES are outside the purview of this research work.

The communication between traditional MES and shop-floor takes place over WAN, which means that the transmission delay is not bounded [34]. 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. The request and response data travel through the source and destination entities in the network via a series of routers. These data packets suffer several types of delays at each node along the network path. A node can be a source, destination or an intermittent router. The following are significant delays encountered by the data packets:

- Nodal processing delay
- Queueing delay
- Transmission delay
- Propagation delay

The throughput of the network is greatly affected by these network delays, particularly the nodal queueing delay and the propagation delay [35]. The 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. When the data packet arrives at a router, the router examines the packet header for redirection to the appropriate destination. 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. Normally, the processing delay is of the order of few tens of microseconds in most of the high processing routers in case of forwarding a simple packet [35]. But this delay can be up to a millisecond in case the router undertakes the task of performing encryption algorithms aimed at examining or

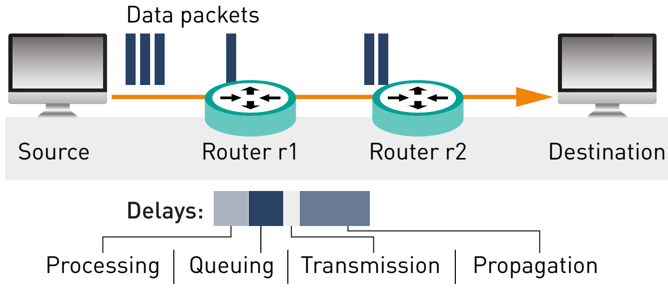

Figure 2: Illustration of network delays.

modifying a packet designed for security and legal aspects [35]. 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. 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.

To gain some insight, let A denote the average number of packets that arrive at the router queue per unit time. Let R be the transmission rate of the router; that is, it is the rate at which the bits are pushed out of router queue. For the sake of simplicity, suppose all the packets consist of B number of bits. Then, the average number of bits that arrive per unit time at the router queue is $(A * B)$. The ratio of $(A * B)/R$, called network traffic intensity, plays an important role in determining the queueing delay. If network traffic intensity is less than 1, the nature of arriving data packets influences the queueing delay. If a data packet arrives every A/R units of time, each of these packets then arrives at an empty queue and will not encounter the queueing delay. Conversely, if the packets arrive in bursts due to traffic congestion, it then results in substantial average queueing delay. For example, assume P packets arrive simultaneously every $(A/R) * P$ units of time. The first packet that is transmitted will encounter no queueing delay. Nonetheless, second packet encounters a queueing delay of (A/R) units of time. Similarly, the third packet experiences a queueing delay of $2*(A/R)$. In general, the n^{th} data packet will experience a queueing delay of $(n-1)*(A/R)$ units of time. However, the packet queueing does not follow a pattern in practical situations and the packets are spaced apart by an arbitrary amount of time. Therefore, the above quantity alone is not adequate to fully characterize the queueing delay. Nevertheless, it is a useful tool in the estimation of queueing delay. If the traffic intensity tends to 0, approximation is then few packets arrive and they are spaced far apart in time. The probability that each packet encounters a non-empty queue is close to 0. Conversely, when the traffic intensity tends to 1, there will be intervals of time when the packet arrival rate is greater than the network transmission capacity. The packet queue at the router site builds and grows open-endedly when the packet arrival rate is greater than the router transmission rate. On the contrary, when the packet arrival rate is less than

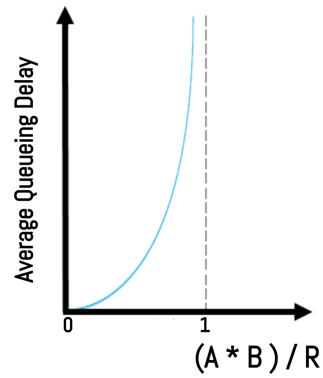

Figure 3: Dependence of average queueing delay d_{queue} on traffic intensity [36].

the router transmission rate, the size of the queue shrinks.

The qualitative dependence of average 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 ratio of lost packets depends on the router queue capacity, network traffic intensity and the nature of traffic arriving at the queue. In general, the queue capacity greatly depends on the router design and cost. From the above discussion, it can be inferred that the network latency not only depends on the delay during transmission, but also on the packet loss.

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. It is directly proportional to the number of bits in the data packet and independent of queue length and distance between the two nodes. It is of the order of microseconds [35].

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 propagation delay varies directly with the distance between the adjacent nodes and is of the order of tens of microseconds to milliseconds [35]. It can vary significantly between few microseconds for a link connecting two routers within the same intranet to millisecond for a link joining two routers thousands of kilometers apart.

The total nodal delay d_{nodal} is then given by [35]

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \tag{1}$$

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

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

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 [36].

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 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.

Figure 4 illustrates this situation where there are three operations - welding, color spraying and quality check, which 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 are instances in the life cycle of cloud solutions where the services are disrupted due to many reasons such as electric failure, hardware failure, cascading failure on routers and cloud downtime arising out of data center migration, server update against vulnerability et cetera. These incidences, on an average, reduce the availability to 99.91%, which in other words a non-availability of 7.884 hours per year [37]. Such network outages are not acceptable in the event of manufacturing a priority order.

VI. Introducing an Edge Layer

To realize the decentralized production, this research paper proposes introducing an edge layer called *Generic Shop-Floor Connector (GeSCo)* between CMES and shop-floor. However, as explained in Section V, 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. With this view, the production control data of CMES is cached in proximity to the shop-floor can reduce the problem of network latency. To that end, GeSCo is an ideal place to store the cached data.

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 produce* 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 V. The cached data constitutes production control data of part of/complete/multiple PO(s). Such information empowers GeSCo to take decisions with regard to production control without the consulting the centralized CMES and hence, it facilitates the implementation of *decentralization* of the production execution.

In its basic conception, the GeSCo is a web service and other numerous industrial communication protocols framework. It 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. Instead, 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.

VII. Requirements for GeSCo

Subsequent to the caching of the production control data, 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 manufacturing resource breakdown is one such case in point which is a highly disruptive occurrence in an automated production environment. Even as preventive maintenance or repair is a preferable way to increase the system reliability and significant system cost reduction, [38] claims that, in real-life manufacturing systems the machine breakdowns are inevitable. The GeSCo should anticipate such an eventuality and must be well equipped to take appropriate course of action.

The current manufacturing operation cannot be swapped to another manufacturing resource when there are no alternative manufacturing resources in the shop floor cell. In such a case, the GeSCo should preempt all the other steps of the routing and retain its state. Under such an abort/resume policy in case of random manufacturing resource breakdown, production should resume with the processing of the preempted step of the routing after the breakdown is fixed. When GeSCo has started execution of another PO of different product variant with no dependency on the resource which has broken down, it should resume the execution of aborted PO after completion of the current PO.

In a job shop environment, the presence of multi purpose manufacturing resources enables to execute multiple operations on several alternative resources. In such a scenario, GeSCo must reschedule the production routing by replacing the disrupted resource with an alternative resource. In the event of

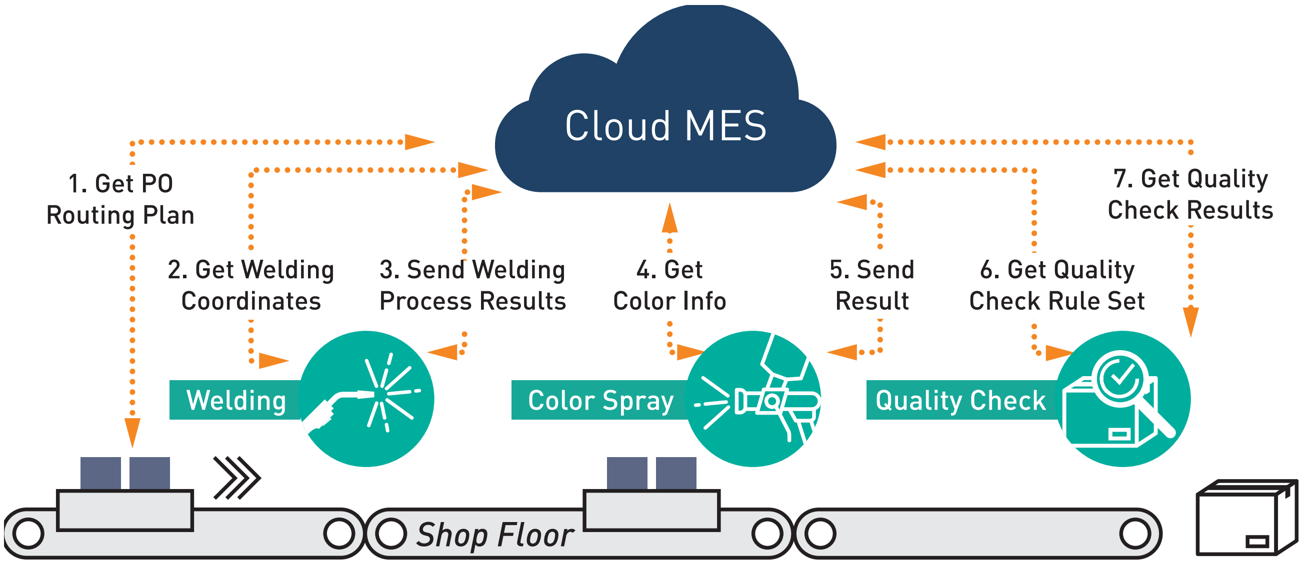

Figure 4: CMES - Shop floor connectivity in production.

Figure 5: Evolution of CMES - Shop Floor Connectivity.

manufacturing resource replacement, the new manufacturing resource should be introduced easily and quickly into the manufacturing system without reprogramming or reconfiguration of the production setup. The GeSCo should be resilient to such *plug and produce* concepts.

When the PO is changed at ERP during the execution, the CMES should deliver the necessary information promptly to GeSCo. The GeSCo should check the feasibility of the changed PO, take appropriate measures and convey the same to the upper layers of automation pyramid. Under normal circumstances, the GeSCO should adopt the First-In-First-Out (FIFO) policy for the execution of a PO. However, when the production routing consists of manufacturing operations of different lead times, the priority order in the pipeline should finish the execution at the earliest. Therefore, the provision should be made in GeSCo by defining a priority policy to put a non-priority order on *hold* state in order to expedite the execution of the priority order.

The traditional MES creates static production routing where manufacturing resources and operations are coupled together and pushed down to shop-floor execution. This approach does not allow the edge component the freedom to make decisions at the shop-floor. In case of deviation from the production planning, the edge component seeks the directions from CMES to recover from the path of deviation to successfully perform all the activities of production. However, in order to provide more autonomy to the GeSCo, the CMES should only create the abstract production planning without tying the manufacturing operations to resources. This process should be performed in GeSCo. The GeSCo should possess local intelligence during the dispatch of manufacturing operations to manufacturing resources. In addition to their *reachable* property, the modern manufacturing resources also known as Cyber Physical Production Systems (CPPS), have more computing power to complement large number of embedded sensors and actuators. These resources can track their state, PO buffer, and are *aware* of their various configurations to manufacture products with unique characteristics. In state-of-the-art factories, the shop-floor is considered to be a *service market place* where different manufacturing requirements are matched against the corresponding services offered by the resources to produce a tailored product defined by the customer.

It should be the responsibility of GeSCo to associate each operation of a PO to a particular manufacturing resource also called CPPS in order to process a semi-finished assembly also called CPS. The dispatched operation characterizes the logical binding between the CPS and CPPS. The changes brought about by this combination of the CPS, CPPS and GeSCo that has the relevant contextual information of the current POs drive changes in manufacturing production and control, and actuate the remodeling of centralized to truly decentralized production decision-making systems.

The idea is not to store complete informational and operational technology information in the GeSCo to make these runtime decisions. The provision should be made where the manufacturing resources publish their capabilities to the GeSCo. The GeSCo should utilize this information to assign a routing step to one of the manufacturing resources. The PO should also push the required abstract services in case of quality non-conformance along with the non-conformance codes. In the event of quality non-conformance, the GeSCo only looks-up the non-conformance code and seeks the corresponding services from the manufacturing resources. Another major challenge is to make feasible decisions taking into the account the physical configuration of the cells of the shop-floor.

In all the above described exceptional situations, the GeSCo should either resolve or find an alternative course of actions. 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 [13]
6) Dealing with the change of the PO [13]
7) Handling the POs of high priority [13]
8) Course of action in the event of quality defects
9) Resumption of production after a disruption due to unforeseen circumstances
10) Course of action in the event of unavailability of raw materials
11) Distributed manufacturing where components are being manufactured at different sites

VIII. Proposed System Architecture

The solution architecture should be designed taking into account the challenges mentioned in Section VII-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.

Figure 6: Resource Virtualization.

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 concept of enriching the digital plant models by making the virtual copies of the manufacturing resources with near real time data from sensors also makes the information flow more transparent. 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.

This service interface of virtual resource enables GeSCo to store the resource relevant data in a realistic resource model, also called as resource digital twin [39]. The GeSCo collects the machine data from resource periodically and pushes it to CMES resource model. This assists in real time monitoring of the manufacturing resource for the purpose of tracking the status and understanding its behavior in interaction with other manufacturing systems, and also to calculate the equipment effectiveness. Further, the data is 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) [40], 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 historic manufacturing reports, quality investigations and process improvements.

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. To this end, this module has internally a sub-module known as Capability Discovery Repository (CDR), which stores the capabilities of the various manufacturing resources. The manufacturing resources are also allowed to directly announce all their capabilities semantically to the CDR. However, a formal explicit specification of shared concepts [41] and relationship among those concepts, also called ontology, needs to be modeled at the organization level in order to realize the semantic publishing of the capabilities. A static service which provides access to the created ontologies enables referencing and dereferencing of the semantic concepts. The decentralization facilitator exploits this semantic information from CDR to arrive at the decisions at run-time. To this end, the authors extended OWL-S and SAWSDL specifications to the OPC-UA application specific methods in order to automate the process of method discovery and subsequent method composition in [42] [43].

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 VII-A.

3) Decentralization Facilitator: This entity enables the decentralization in 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 semantically searching for 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 VII-A, numbers 5 to 9. The exception handler either attempts to find an 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 production engine also assigns the unique PO identifier to the smart product or the product carrier at the start of the PO, so that the carrier can be identified and tracked any time during production. During the dispatch of each routing step of a PO, the manufacturing operation harnesses the unique CPS identifier and binds the product to the manufacturing resource. 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

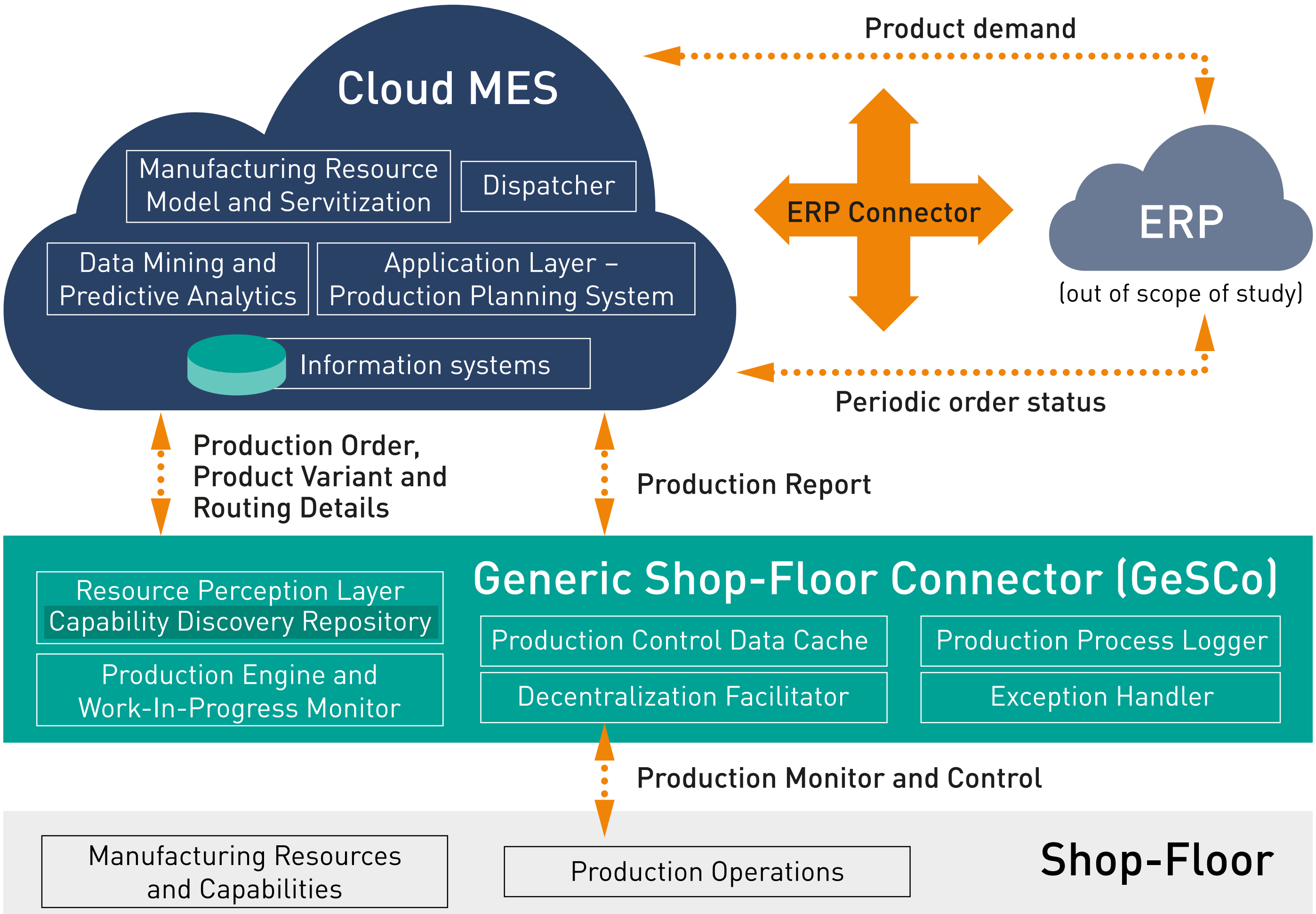

Figure 7: Integration of GeSCo with CMES.

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.

IX. Implementation

In order to demonstrate the feasibility of the concepts introduced above, the author simulated the shop-floor behavior by implementing a prototype of the architecture shown in the Figure 7. Existing MES solutions proved to be inflexible to experiment since they are passive in behavior and hence, do not voluntarily react to the conditions of the shop-floor and also percolate the changes in the PO to GeSCo. In general, the MES solutions provide the directions to the events of the shop-floor only when the information is sought.

In order to engineer a seamless change in PO and to have more control over the simulation, a CMES was developed that mocks the real CMES in the context of production planning and execution (see Figure 8). The SAP Plant Connectivity (SAP-PCo) [44], which is a framework of set of services and management tools was chosen as a basis for GeSCo.

SOAP, REST and an ODATA based web servers, and OPC-UA servers were implemented inside the PCo. During the research, the PCo was architecturally enhanced to accomodate all the modules of GeSCo (refer Figure 7). These modules were developed inside a Dynamically Linked Library (DLL) along with a set of wrapper operations that were exposed as both web service operations and OPC-UA application specific methods (see Figure 9) that contain the production execution logic. This concept is also called the Enhanced Method Processing (EMP) [45]. The EMP concept enables to cache the production control and routing data, and also embed the orchestration plan algorithms. Furthermore, the EMP implementation assists in behavior specification of the edge component by allowing flexible definition of the actions that need to be executed when invoked by web/OPC-UA client. The EMP DLL is implemented independently by inheriting the API class of the PCo and freely configure the actions that need to be executed during the production. This DLL is imported into the PCo agent instance at design time and the resulting loaded operations/methods are hooked onto the PCo SOAP/REST/OData Webserver(s) and/or OPC-UA server(s).

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. The experimental simulation setup with reference to the running use case presented in the paper (refer Figure 4) is shown in Figure 10. The simulation

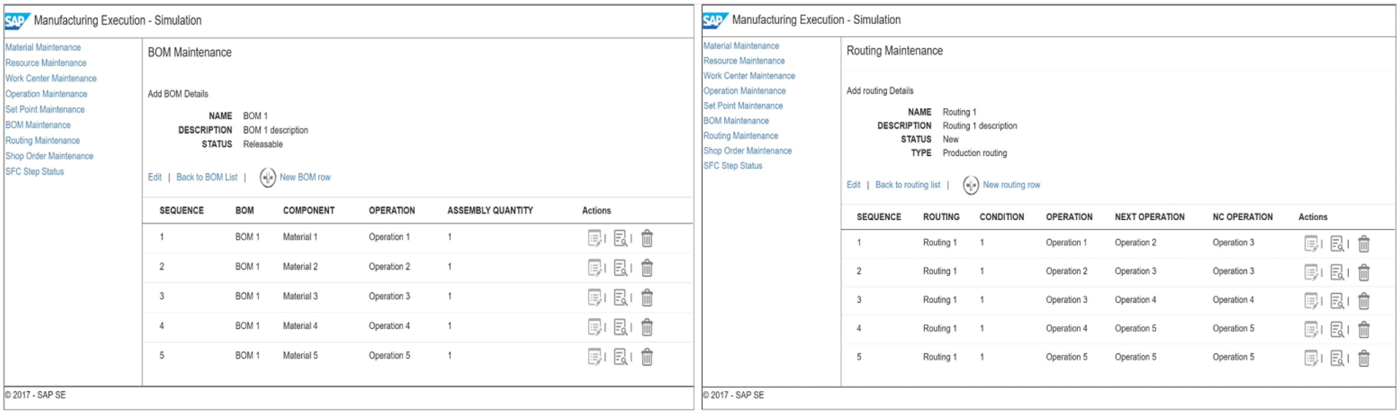

Figure 8: Screen shots of mock CMES user interfaces.

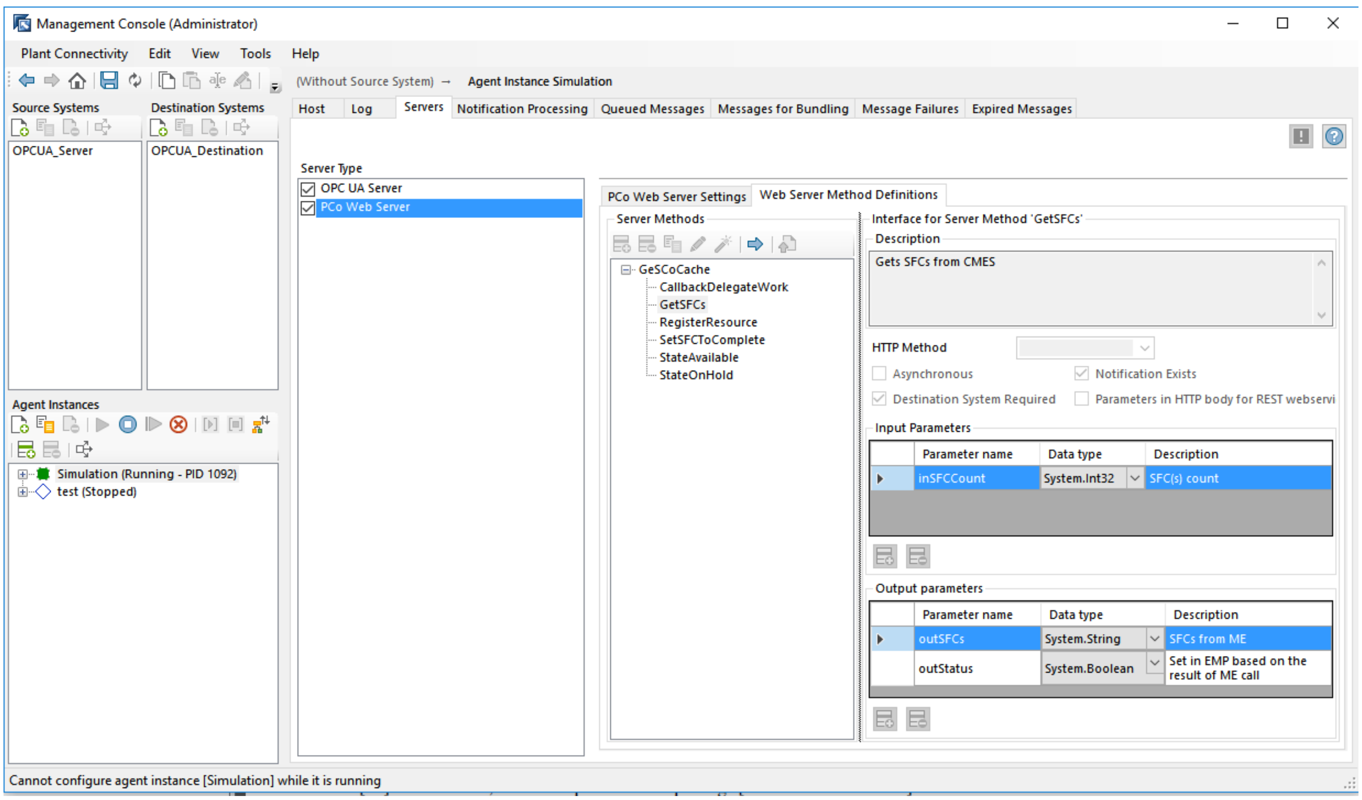

Figure 9: Caching concept implemented as Enhanced Method Processing in PCo.

manufacturing resources supported two important industrial communication protocols: REST based WS-* and OPC-UA. These manufacturing resources acted both as WS-* and OPC-UA client and servers. The capabilities of the manufacturing resources were exposed as a set of web server operations in case of web server and application specific methods in case of OPC-UA server. During the dispatch of the manufacturing operations to the manufacturing resources by the GeSCo, the manufacturing resources act as web or OPC-UA servers. These manufacturing resources operate as WS-* or OPC-UA clients to the GeSCo server during the operation completion acknowledgement step. 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 involved with the cloud solutions, where as the GeSCo and resource work stations were deployed on the same Local Area Network (LAN). A production process without exceptional scenarios that corresponds to the use case illustrated in Figure 4 was simulated to address the challenges 1 and 4 from Section VII-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

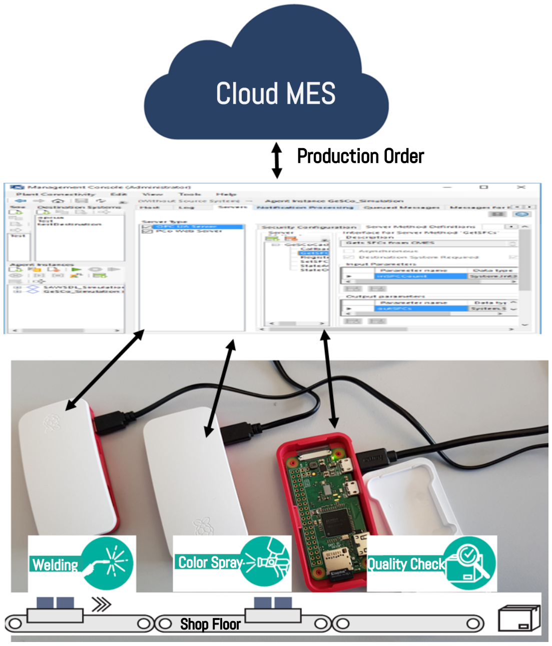

Figure 10: Simulation Setup.

Figure 11: Open Integrated Factory - Generation 2017.

SOAP UI tool [46]. Tables I and II provide the simulation results w.r.t. the network latency encountered without and with GeSCo, respectively. The total latency showed a marked decrease in simulation with the edge layer.

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

The research concept was also implemented in the *Open Integrated Factory - Generation 2017* (see Figure 11) that SAP along with other technology partners showcased in *Hannover Industrial Fair - 2017*, which verifies the assumption that the result of simulations is valid under real manufacturing conditions.

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	

X. CONCLUSION AND FUTURE WORK

This paper describes the various facets of MES solutions and compares the benefits and drawbacks. Based on the arguments, the paper contends that the CMES is better suited in changing production environments than traditional on-premise MES solutions. The most important challenges of cloud solutions are twofold: connectivity and network latency, and security issues. The latter is not the research focus of this work. The former challenge is elucidated and an empirical study is carried out with the aid of an elementary use case. This challenge requires to be addressed in order to make CMES viable in the context of high speed manufacturing.

To overcome the problem of network latency and connectivity associated with CMES, an edge layer called GeSCo that caches the production control data is introduced and a comprehensive architecture is designed to integrate this edge layer with the CMES. The decentralization of the decision-making process in manufacturing was also taken into consideration during the design of the edge component.

Future work includes further refinement in realization of decentralization, development of a semantic data model for GeSCo, research on the extent of caching under given conditions and handling of exceptional scenarios such as quality non-conformance, machine breakdown, priority orders and changes in PO.

ACKNOWLEDGMENT

This research work is supported by a doctoral grant from SAP SE. A non-provisional patent, with ID 15/926,856, on the research of CMES controlled edge manufacturing has been filed in USA on 20-Mar-2018 with the authors as the inventors.

REFERENCES

[1] B. Katti, M. Schweitzer, and C. Plociennik, "Gesco: Introducing an edge layer between cloud mes and shop-floor in decentralized manufacturing," in 11th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM-2017), Special Session on Edge Computing in factories of the future (ECFoF), 2017, pp. 153–158.

[2] A. Lenart, "Erp in the cloud–benefits and challenges," Research in systems analysis and design: Models and methods, 2011, pp. 39–50.

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

[4] K. D. Bettenhausen and S. Kowalewski, "Cyber-physical systems: Chancen und nutzen aus sicht der automation," Thesen und Handlungsfelder; VDI/VDE-Gesellschaft Mess-und Automatisierungstechnik: Düsseldorf, Germany, 2013.

[5] G. K. Saharidis, Y. Dallery, and F. Karaesmen, "Centralized versus decentralized production planning," RAIRO-Operations Research, vol. 40, no. 2, 2006, pp. 113–128.

[6] C. Anderson and J. J. Bartholdi, "Centralized versus decentralized control in manufacturing: lessons from social insects," Complexity and complex systems in industry, 2000, pp. 92–105.

[7] "Manufacturing Operations Management from Siemens," URL: https://www.plm.automation.siemens.com/mom/ [Accessed Date: 2018-02-15].

[8] "IEC 62264-3:2016 Standard," URL: https://www.iso.org/standard/67480.html [Accessed Date: 2018-02-15].

[9] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: Focusing on mobile users at the edge," CoRR arXiv preprint arXiv:1502.01815, 2015.

[10] K. Ueda, A. Lengyel, and I. Hatono, "Emergent synthesis approaches to control and planning in make to order manufacturing environments," CIRP Annals-Manufacturing Technology, vol. 53, no. 1, 2004, pp. 385–388.

[11] H. Lödding, "Decentralized wip oriented manufacturing control," in Handbook of Manufacturing Control. Springer, 2013, pp. 435–452.

[12] R. Y. Zhong, Q. Dai, T. Qu, G. Hu, and G. Q. Huang, "Rfid-enabled real-time manufacturing execution system for mass-customization production," Robotics and Computer-Integrated Manufacturing, vol. 29, no. 2, 2013, pp. 283–292.

[13] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," Engineering Applications of Artificial Intelligence, vol. 22, no. 7, 2009, pp. 979–991.

[14] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," IEEE Transactions on Industrial Electronics, vol. 53, no. 1, 2006, pp. 322–337.

[15] D. Georgakopoulos, P. P. Jayaraman, M. Fazia, M. Villari, and R. Ranjan, "Internet of things and edge cloud computing roadmap for manufacturing," IEEE Cloud Computing, vol. 3, no. 4, 2016, pp. 66–73.

[16] V. Gezer, J. Um, and M. Ruskowski, "An extensible edge computing architecture: Definition, requirements and enablers," in 11th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM-2017), Special Session on Edge Computing in factories of the future (ECFoF), 2017, pp. 148–152.

[17] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, "Orca-fms: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," Computers in Industry, vol. 65, no. 4, 2014, pp. 706–720.

[18] P. Valckenaers, H. Van Brussel, P. Verstraete, B. Saint Germain et al., "Schedule execution in autonomic manufacturing execution systems," Journal of manufacturing systems, vol. 26, no. 2, 2007, pp. 75–84.

[19] B. Montreuil, J.-M. Frayret, and S. D'Amours, "A strategic framework for networked manufacturing," Computers in industry, vol. 42, no. 2, 2000, pp. 299–317.

[20] M. Marques, C. Agostinho, G. Zacharewicz, and R. Jardim-Gonçalves, "Decentralized decision support for intelligent manufacturing in industry 4.0," Journal of Ambient Intelligence and Smart Environments, vol. 9, no. 3, 2017, pp. 299–313.

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

[22] D. Mourtzis and M. Doukas, "Decentralized manufacturing systems review: challenges and outlook," Logistics Research, vol. 5, no. 3-4, 2012, pp. 113–121.

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

[24] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, and B. H. Li, "Cciot-cmfg: cloud computing and internet of things-based cloud manufacturing service system," IEEE Transactions on Industrial Informatics, vol. 10, no. 2, 2014, pp. 1435–1442.

[25] X. Xu, "From cloud computing to cloud manufacturing," Robotics and computer-integrated manufacturing, vol. 28, no. 1, 2012, pp. 75–86.

[26] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computingthe business perspective," Decision support systems, vol. 51, no. 1, 2011, pp. 176–189.

[27] P. Helo, M. Suorsa, Y. Hao, and P. Anussornnitisarn, "Toward a cloud-based manufacturing execution system for distributed manufacturing," Computers in Industry, vol. 65, no. 4, 2014, pp. 646–656.

[28] L. Zhang, H. Guo, F. Tao, Y. Luo, and N. Si, "Flexible management of resource service composition in cloud manufacturing," in International Conference on Industrial Engineering and Engineering Management (IEEM), 2010 IEEE,. IEEE, 2010, pp. 2278–2282.

[29] J.-J. Hwang, H.-K. Chuang, Y.-C. Hsu, and C.-H. Wu, "A business model for cloud computing based on a separate encryption and decryption service," in Information Science and Applications (ICISA), 2011 International Conference on. IEEE, 2011, pp. 1–7.

[30] F. Tao, L. Zhang, Y. Liu, Y. Cheng, L. Wang, and X. Xu, "Manufacturing service management in cloud manufacturing: overview and future research directions," Journal of Manufacturing Science and Engineering, vol. 137, no. 4, 2015, p. 040912.

[31] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to cloud computing," Cloud computing: Principles and paradigms, 2011, pp. 1–41.

[32] T. Wood, P. J. Shenoy, A. Gerber, J. E. van der Merwe, and K. K. Ramakrishnan, "The case for enterprise-ready virtual private clouds." in HotCloud, 2009.

[33] D. M. Mangiuc, "Enterprise 2.0-is the market ready?" Accounting and Management Information Systems, vol. 10, no. 4, 2011, p. 516.

[34] "CISCO Wiki," URL: https://docwiki.cisco.com/wiki/Introduction_to_WLAN_Technologies [Accessed Date: 2018-02-15].

[35] R. Ramaswamy, N. Weng, and T. Wolf, "Characterizing network processing delay," in Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE, vol. 3. IEEE, 2004, pp. 1629–1634.

[36] J. F. Kurose and K. W. Ross, "Computer networking: a top-down approach," Addison Wesley, vol. 4, 2007, p. 8.

[37] M. Gagnaire, F. Diaz, C. Coti, C. Cerin, K. Shiozaki, Y. Xu, P. Delort, J.-P. Smets, J. Le Lous, S. Lubiarz et al., "Downtime statistics of current cloud solutions," International Working Group on Cloud Computing Resiliency, Tech. Rep, 2014.

[38] Y.-S. P. Chiu, K.-K. Chen, F.-T. Cheng, and M.-F. Wu, "Optimization of the finite production rate model with scrap, rework and stochastic machine breakdown," Computers & mathematics with applications, vol. 59, no. 2, 2010, pp. 919–932.

[39] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," IFAC-PapersOnLine, vol. 48, no. 3, 2015, pp. 567–572.

[40] J. Haupert, "Domeman: Reprasentation, verwaltung und nutzung von digitalen objektgedachtnissen," Ph.D. dissertation, Saarbrücken, Universität des Saarlandes, Dissertation., 2013, 2013.

[41] T. R. Gruber, "A translation approach to portable ontology specifications," Knowledge acquisition, vol. 5, no. 2, 1993, pp. 199–220.

[42] B. Katti, C. Plociennik, and M. Schweitzer, "SemOPC-UA: Introducing Semantics to OPC-UA Application Specific Methods," in 16th IFAC INCOM 2018 full proceedings, 2018, p. To Appear.

[43] B. Katti, C. Plociennik, M. Schweitzer, and M. Ruskowski, "SA-OPC-UA: Introducing Semantics to OPC-UA Application Methods," in 14th IEEE CASE 2018 full proceedings, 2018, p. To Appear.

[44] "SAP Plant Connectivity," 2017, URL: https://help.sap.com/viewer/p/SAP_PLANT_CONNECTIVITY [Accessed Date: 2018-02-15].

[45] "SAP Plant Connectivity, Enhanced Method Processing," 2017, URL: https://help.sap.com/doc/saphelp_mii151sp03/15.1.3/en-US/75/819a57c209ab6be10000000a4450e5/frameset.htm [Accessed Date: 2018-02-15].

[46] "SOAP UI," 2017, URL: https://www.soapui.org/ [Accessed Date: 2018-02-15].