

A novel fluid architecture for cyber-physical production systems

R. Beregi, G. Pedone & I. Mezgár

To cite this article: R. Beregi, G. Pedone & I. Mezgár (2019): A novel fluid architecture for cyber-physical production systems, International Journal of Computer Integrated Manufacturing, DOI: [10.1080/0951192X.2019.1571239](https://doi.org/10.1080/0951192X.2019.1571239)

To link to this article: <https://doi.org/10.1080/0951192X.2019.1571239>



Published online: 17 Feb 2019.



Submit your article to this journal [↗](#)



Article views: 102



View Crossmark data [↗](#)

ARTICLE



A novel fluid architecture for cyber-physical production systems

R. Beregi, G. Pedone and I. Mezgár

Centre of Excellence in Production Informatics and Control, Hungarian Academy of Sciences Institute for Computer Science and Control, Budapest, Hungary

ABSTRACT

Cloud computing has revolutionised the conceptualisation of IT environments in most fields of the economy. Nevertheless, it still requires adequate pre-requisites both in the business strategy and the computational architecture to be adopted successfully and profitably. This is especially true in the context of manufacturing where production services are very sensitive with respect to their distribution and localisation within the boundaries of the computing constellation. The paper aims at clarifying the concept and the application of Fluid Computing in Cyber-Physical Production Systems, connecting the idea of computational sedimentation to the novel Fluid Manufacturing Architecture (FMA). The FMA encompasses the computing layers of cloud, fog, edge and mist, and extends them with a new one called *Dew*. The *Dew* represents the entry point where manufacturing legacy devices are converted into network-able, embedded components for the distributed production scenarios of Industry 4.0. The application of the FMA in a pilot factory has been investigated. The preliminary results of an agent-based simulation case-study for collaborative manufacturing services within the FMA are finally presented.

ARTICLE HISTORY

Received 11 June 2018
Accepted 22 November 2018

KEYWORDS

Fluid computing; dew computing; fluid manufacturing architecture; cyber-physical production system

Introduction

The complexity of modern distributed production architectures has reached a point where the boundaries of the computational layers are not always containable into rigid structures. Manufacturing strategies dictated by Industry 4.0 scenarios necessitate an extreme level of reactivity, adaptivity and pro-activity, which indicate the need of a more *fluid* approach in assembling production services based on Cyber-Physical Production Systems (CPPS), as well.

The Internet of Things (IoT) is generating an unprecedented volume and variety of data also in distributed factories. By the time data is propagated, for example, to the cloud for analysis, the opportunity to act on it might be already lost, due to issues related to i) latency and response time, ii) bandwidth cost and capacity, iii) security and privacy, iv) power consumption, v) data obesity and vi) off-line versus only-on-line usages. Also, privacy and security issues in the IoT devices have a crucial role, too, as discussed, for example, by Alrawais et al. (2017). In this context, *Fog computing* provides upgraded security, reduced data transfer bandwidth capacity and decreased latency. These characteristics potentially qualify fog computing an ideal technology for IoT applications.

The more decentralised approach of Fog computing is sometimes called also *Edge computing*; they both aim at bringing intelligence and processing closer to the creation of data, but while Fog environment places intelligence at the local area networks (processing data in a fog node or IoT gateway), Edge computing places them directly into devices (like embedded automation controllers). For a comprehensive survey on Fog computing architectures, technologies, applications and open issues, please refer to Hu et al. (2017).

In addition to this, *Mist computing* devices are Edge/Fog nodes with functionality usually dedicated to local analytics and decision-making processes; redundant, resilient and highly robust data and applications; root of trust; data access control mechanism to enforce privacy consent at a local level; and default security and privacy by design. A schematic comparison between cloud, fog, edge and mist computing can be found in Malik (2017). The NIST presents a conceptual model of Fog and Mist computing, and places these concepts in relation to Cloud computing and Edge computing (NIST 2018).

The *acatech* paper (National Academy of Science and Engineering 2011) started a discussion in Europe about Cyber-physical Systems (CPS) in 2011. The main goal was to adapt the hardware manufacturer's Networked Embedded System (NES) based bottom-up CPS approach with the IT technology experts top-down IoT concept. According to their definition, every CPS consists of one or multiple NES, as a missing layer between the IoT and the low-level electronics, connecting them over a unified protocol. This bridging layer creates the opportunity to move previously dedicated functionalities to higher or lower layers. This freedom requires a new architecture, which define the trade-off of each structural choice. This paper aims at providing a contribution in this direction, as well, as introduced in the next section.

Ideally, time-sensitive decisions should be made closer to the system that is producing and acting on the data, so factories can move computations and analytics closer to the data, saving costs on network and storage, too. Data that can wait minutes to be actionable can be passed along to the above aggregation nodes for analysis and actions. Finally, data that are less time-sensitive can be sent to the Cloud for historical analysis and long-term storage, as an example. According to this view, the lower is a

layer, the more adherent are the computing capabilities of the physical components to their original functionality (close to the shop-floor); on the contrary, the higher is a layer, the more aggregated and artificial is the computational outcome of its elements (close to the public cloud). Simply put, if data are generated in some remote field and decision needs to be made quickly, sending data to the Cloud for processing will not represent, in many cases, a feasible solution.

The concept of Fluid computing (FC) proposed by the authors in this paper is both a computing approach and an architectural proposal, which hierarchically embraces, reorganises and extends the computational paradigms of Cloud, Fog, Mist and Edge computing with a new 'interfacing' layer, an entry-point for computational capabilities of the physical devices. Authors called this layer *Dew computing*, as a logical reduction of computational complexity and infrastructural requirements necessary to establish it in legacy manufacturing systems. The Fluid architecture proposed by the authors locates (public and private) Cloud and *Dew computing* at the upper and lower boundary of the CPPS computational architecture.

The rest of the paper is organised as follows: the following two sections introduce the motivations for this research work and investigate the existence of related works. Then, the model of the proposed architecture is presented, showing how it meets the identified requirements, as well as explaining how it is designed and constructed. In the next section authors illustrate the preliminary realisation steps, organised into i) working environment, ii) agent-based production service simulation and iii) evaluation of the Fluid CPPS performance, together with a case study scenario highlighting the simulated functionalities. Conclusions and future work close the paper.

Motivations

One of the main objectives of this paper is to adapt and extend the metaphor of meteorological dynamics in the definition of a *Fluid CPPS* architectural layers, by re-organising and disambiguating, where possible, existing concepts of Cloud, Fog, Edge and Mist computing. These levels have been extended with a new one, called *Dew computing*, as explained in the following of this paper.

Authors consider Fluid computing (FC) as an architectural principle whose infrastructural abstraction provides an end-to-end mechanism that can be used to seamlessly provide, deploy, manage and monitor applications, regardless of whether the underlying resource is provided by the Cloud, the Fog, the Edge, the Mist or by Things, intended as the physical devices on the shop-floor.

In FC layers do not have well-defined boundaries. On the contrary, due to this 'non rigid' nature, each computational categorisation might eventually overlap to some extent, as illustrated in the schema of [Figure 1](#). Authors consider the FC architectural stack as the proper deployment environment for the introduction of the Computational Sedimentation concept.

Authors herein intend to define the mechanism of *Computational Sedimentation* as the tendency of resources (such as generated data or computations) to separate from the CPPS over-all information service flow, and remain located on a more 'proper' hosting and storing layer. This is due to key requirements influencing their provision: latency-time, data propagation and resource replication minimisation. A practical estimation of latency-time in large-scale and distributed event processing systems can be found in Szymaniak et al. (2008) and Chandramouli et al. (2011).

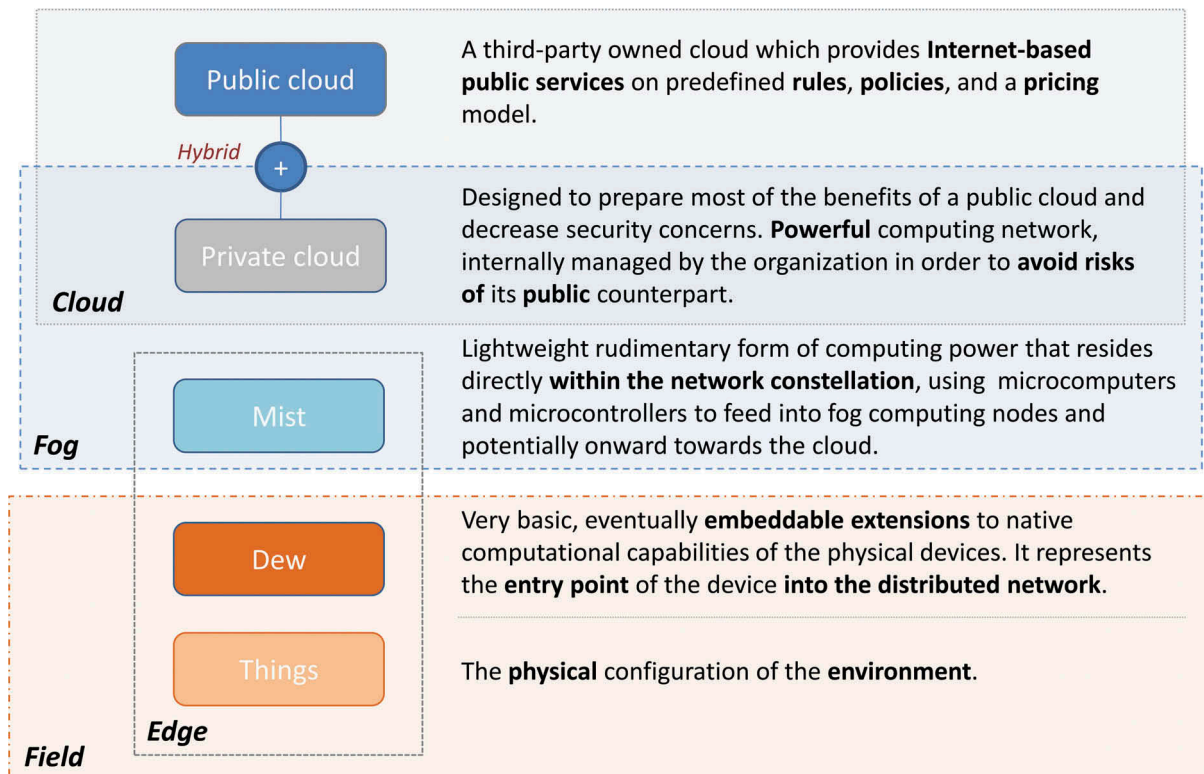


Figure 1. Fluid computing layers.

The basic motivation for the research work was to solve a communication and data handling problem in the control field of CPPS, as explained hereafter. Because of the increasing spread of IoT technology machines in production are equipped in a great extend with sensors that generate and send data with different frequencies and quantities to higher levels of the hierarchy. The usage of these data is also different on the different layers of the control architecture. Cloud-based architectures are too general in CPPS and data sharing and cooperation is of vital importance in these new type of systems. Different layers use different types of data with different content, so it is important to have the right data (in quantity, frequency and speed) on the right level.

The main guidelines or directives in smart factory operation using cloud architecture can be summarised as:

- ‘software service to the data’, instead of ‘data to the cloud’
- data processing directly at the source
- hard real-time system demands for devices operating at the Edge of the network
- higher requirements concerning infrastructure (latency, bandwidth, etc.)

It is not effective nor profitable to have data on a level where is no actionable, indeed. There are data used frequently on a certain level of hierarchy but not necessary to handle on higher levels.

The basic goal was to develop a control architecture for smart manufacturing where the proper data is on the right place (level) with the needed frequency (daily or every second), at the needed time (real time) and the needed quantity. This architecture filters the data needed on the lowest level that will not go up, only the needed data will reach the highest level. In this way the data processing evaluation analysis will be simpler, the data flow will be quicker the whole communication in the system will be smooth.

The basic motivation has been extended with experiences gained during the implementation of a control system that has been designed, developed and implemented for a prototype smart factory system in a project. During the implementation, a gap has been detected in the control hierarchy and in data control flow. The first special results for bridging this gap have been published in Kemény et al. (2016). Based on this special solution a generalised version has been developed and inserted into the present Fluid Manufacturing Architecture (FMA). It can be stated that, in this way, the first version of the *Dew* layer and units has been produced. Based on these findings the definition and the development of the *Dew* layer and units have been started.

Related works

Authors have investigated resources available in the literature for works and ongoing projects with similarities in relation to the two main concepts in this paper (Fluid computing and *Dew* extension applied to Cyber-Physical Production Systems) and

have found no work directly related to FC applied to CPPS to-date. At the same time, authors have realised a possible disambiguation issue with the terminology of *Dew* computing existing in the literature, indeed, which is used for a different characterisation of the computational environment. The two expressions, although sharing the common intention of defining a computational paradigm in distributed architectures, they convey different characteristics and have a specific service identity and mission. *Dew* computing, as found in literature, is considered as a paradigm that inherits a hybrid approach to provide personal information to users independent of internetwork connectivity. The main idea behind this is to minimise the dependency over existing internetwork backhaul, thus reducing network traffic, indirect overall power consumption of the network system as well as data dependency over Cloud, Fog and Edge services (Sunny Wang 2015; Ray 2018). The concept of *Dew* computing presented in this paper incorporates design attributes in terms of minimal hardware and software capability extensions. The aim is the creation of the necessary preconditions for legacy physical resources through which they become Industry 4.0 ready components in a hybrid peer-to-peer manufacturing network. The *Dew* layer proposed by the authors is intended to be as the fundamental element of service specification for an I4.0 CPPS (otherwise not considerable as such) and not a stand-alone layer for bridging complex service provision between the physical layer and the IIoT.

Other types of architectures in the field of CPPS and Smart Factories have been proposed in the literature, evaluating, for example, the consequence propagation of security attacks in CPSs (Orojloo and Azgomi 2017). This is one of the hot-topics, together with interoperability, of the Industry 4.0 scenarios.

In Sunny, Liu, and Shahriar (2018) authors present an agent-based communication method of manufacturing services in a cyber-physical manufacturing cloud, trying to enable manufacturing with various physically connected machines from geographically distributed locations over the Internet. In research work of Lin et al. (2018), authors present an IoT-enabled, real-time synchronisation, whose main goal is to provide consistency in production orders, real-time data presentation and planning. For a collection of papers presenting the latest research on various aspects of smart manufacturing (from interoperability, communication and data exchange to planning, monitoring) and decision-taking in a cyber-physical smart manufacturing environment, refer to Nassehi et al. (2018).

As mentioned, in literature different concepts and meanings have been given to Fluid Computing since its early origin and the term fluid has received different definitions in accordance to the specific system, like in Graf (2003), where fluidity is merely seen as the application graphical user interface smoothly adaptation to the user’s situation and the capability of the device being used. A preliminary introduction to FC general characterisation, as intended in this paper, can be found in Corsaro (2016). The work presented in Bourges-Waldegg et al. (2005) proposes a ‘fluid computing’ middleware aiming at bringing application fluidity to the mobile Internet.

The generation of Big Data deriving from IoT capillary diffusion has fostered the need for scalable and cost-efficient

platforms which can enable the distribution of data analytics, optimise the allocation of resources and minimise response times. Example of such attempts can be found in crowdsensing supporting applications (Arkian, Diyanat, and Pourkhalili 2017), and in secure and scalable architecture for smart and connected health (Barik et al. 2018). A messaging-based gateway platform for the Fog of Things, which enables clustering and abstraction of peripheral communication protocols is presented in Verba et al. (2017). An example of Fog computing application in cyber-manufacturing for process monitoring and prognosis can be found in Wu et al. (2017). In the research work of O'Donovan et al. (2018,) authors present a simplified Fog computing CPS for embedded low-latency machine learning Industry 4.0 applications. The initial findings highlight and confirm the Fog's potential to deliver consistent and reliable cyber-physical interactions in real-time engineering scenarios, while Cloud computing can support such scenarios and be tolerant to occasional failures.

An example of Hybrid Manufacturing architecture is presented in Lu, Xu, and Xu (2014), in which companies are enabled to deploy different cloud modes for their periodic business goals, by combining the three typical cloud deployment modes (private cloud, community cloud and public cloud). Cloud-manufacturing interoperability and service model affinity in Industry 4.0 standardization technologies are presented in Pedone and Mezgár (2018).

In Linthicum (2017) the data flow from devices to public cloud through a fog node has been analysed taking into consideration time latency issues. The conclusion, in most of these works, is that the only application of (public) clouds in fog scenarios is not effective, other solutions have to be developed.

Architectural model

In decentralised applications, management and data analytics are included in the network, so they require the use of distributed and federated computing models. This is also the case in Cloud, Fog and Mist computing. These paradigms are named based on meteorological phenomena as the characteristics and content of the meteorological names cover well the peculiarities and content of the computational systems.

The authors realised the existence of a gap, a missing interface in the flow of information and data from the Edge devices to the Cloud: this interface layer has been named as *Dew*. Functionality and characteristics of the *Dew* fit well into the logic of the meteorological analogy as well, as introduced in Table 1.

The relationships, the overlapping of the different Fluid computing components are graphically illustrated in Figure 2 and find a practical instantiation proposal in the physical layers of the Fluid Manufacturing Architecture (FMA).

Purpose of the FMA is to enable continuity in the computational flow of production services, fundamental aspect for a smooth re-organisation of factory assets to newly emerged production requirements. In addition, it balances the computational load of the system by conveying the correct data and information towards the correct computing layer. The FMA for CPPSs presented in this paper aims at embracing concepts and resources related to cloud, fog and mist computing,

Table 1. Meteorological analogies in computational paradigms and their application to ISA/MESA manufacturing levels.

Paradigm	Meteorological terminology (UK Met Office, 2017)	Computational terminology	ISA/MESA
Cloud	Cloud is a visible mass of condensed water droplets that are formed in the atmosphere typically high above of the Earth, so clouds exist in the sky	Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable (virtual) computing resources	Level 4 and Level 5
Fog	Fog is a thick cloud that appears near the ground or touches the surface of the ground, so fog sits near the ground	Fog computing pushes intelligence down to the local area network (LAN) level of the network architecture, extending the traditional cloud-based computing model	Level 3 and Level 4
Mist	Mist is a thin fog that appears close to the ground. Thus, both can be defined as the clouds that touch the Earth. Mist is what actually sits on the ground	Mist computing is a lightweight form of fog computing that resides directly within the network nodes at the edge of the network, bringing the fog computing layer closer to the smart end-devices	Level 3 and Level 4
Dew	Dew is condensed moisture that appears in the form of small drops on objects near the Earth's surface, so dew sits on the surface of individual, separate objects	Very basic, embeddable extensions of the capabilities of individual, separate physical devices (this is the computational entry level proposed by the authors in the present paper)	Level 1 and Level 2

harmonising and hierarchically organising them for a manufacturing context deployment (Figure 4).

The meteorological analogy

Each level of the meteorological analogies onto computational paradigms expresses the same characteristics in a computational environment as the meteorological expression in the description of the phenomena.

The similarity between the meteorological and computational layer structure is that several of the physical and functional characteristics of the components on the layers are very similar. The layers in the meteorological and the computational terminology show similarities in characteristics on their vertical positions, relations to each other, distance from the

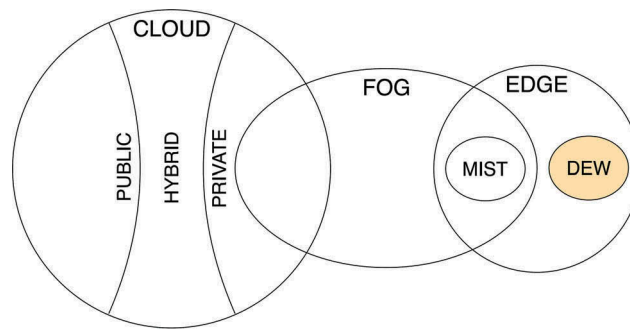


Figure 2. Fluid computing domains and eventual service overlapping.

lowest level (in Table 1 these are considered from a vertical network layer structure). From Cloud to Mist these layers have been already defined and the authors added to this structure the *Dew*. Like the meteorological 'dew' sits on the surface of individual, separate objects, the computational 'dew' is a minimal (drop-like) extension of capabilities of individual separate physical legacy devices of the production environment.

Another aspect in developing a hierarchy is the application of existing standards. In this research work, authors referred to the ISA-95 MESA classification framework for manufacturing systems integration (MESA 2007), which describes the control layer structure of an enterprise in terms of those levels that can fit into the control architecture, as reported in the list below:

Level 5 (L5) In this level business management activities (assets, customers, employees), as well as plant schedules (production, material use, delivery and shipping), are planned, generated and executed (ERP, PLM, CRM, etc.). System response times are in months, weeks and days.

Level 4 (L4) Work unit definition and control are given. A typical system on this level is the Manufacturing Execution Systems (MES, APS, etc.) – maintain records and optimise the production process. System Time Frame is in days, shifts, hours, minutes and seconds.

Level 3 (L3) Automated control of the production process is going on here. Response time is in hours, minutes, seconds and micro seconds or less (SCADA, HMI).

Level 2 (L2) Systems monitoring and supervisory control of the production process is going on here. Response time is in minutes, seconds and micro seconds or less (PLC, DSC, etc.).

Level 1 (L1) Sensing and manipulating of production work process. Device time is real-time; in micro seconds or less.

Level 0 (L0) Containing the actual production physical processes (shop-floor).

The positioning of manufacturing automation levels in relation to the FMA computing layers are illustrated later on in Figure 7. In this sense, Edge computing differs from any meteorological analogy, as it can be seen as a logical re-organisation of connections among Mist, Dew and the physical layer (Things), as also described forward in this section and highlighted in Figures 1, 4 and 7.

CPS core

Authors' definition of CPS relies on the core elements identified in NES: *sensing&acting*, computing and network capabilities (Figure 3). These three components are essential for a system to gain the functionalities defined by CPS. *Sensing&acting* means the ability to interact with the physical world, while computing is the capacity to analyse and react to the information gained. These two elements define every low-level controller used in automated industrial environments for the last four decades. The availability of complex communication networks and the opportunities arouse with them are the real novelties nowadays.

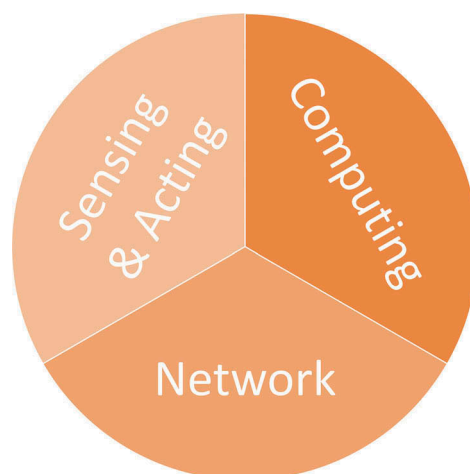


Figure 3. Core capabilities of a cyber-physical system.

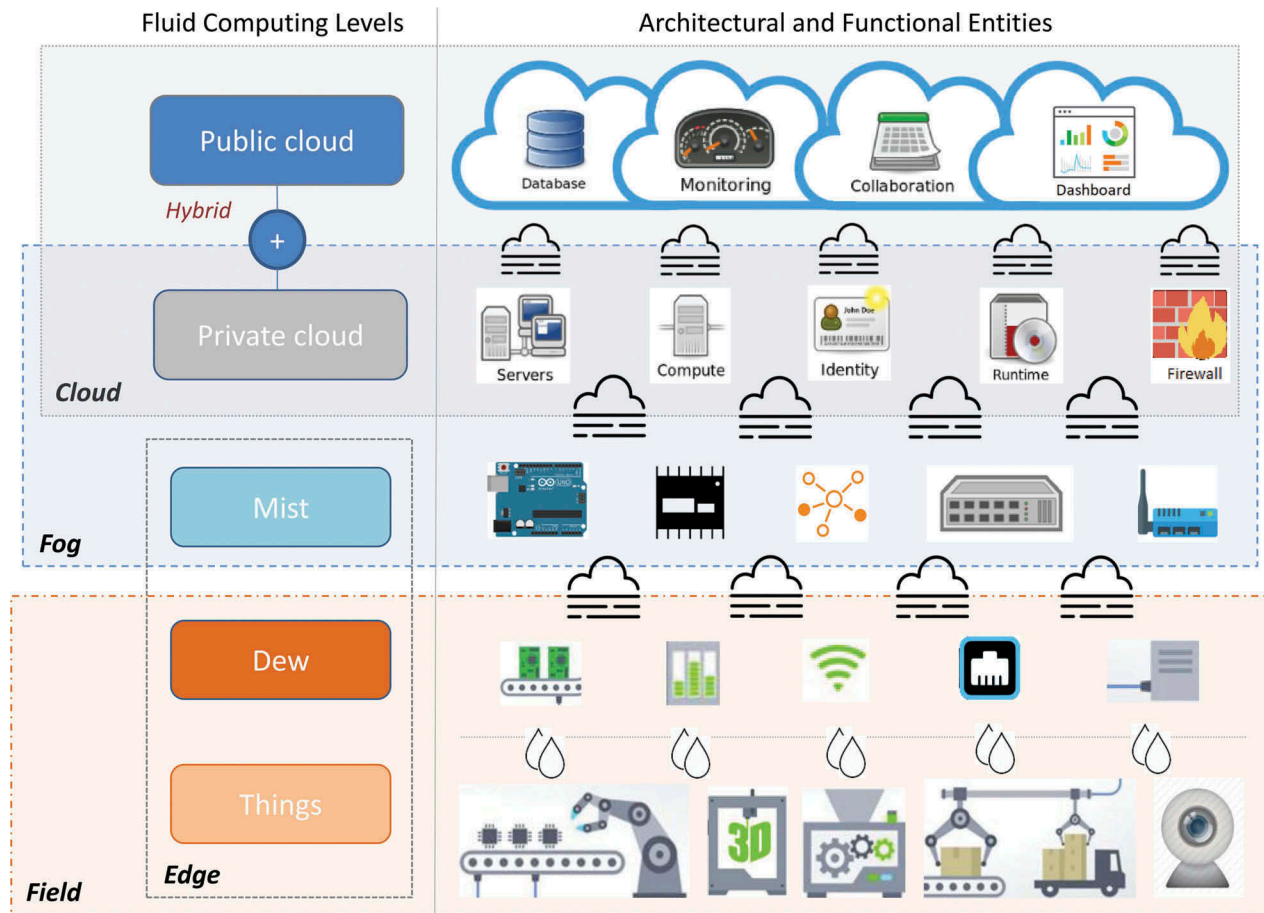


Figure 4. Fluid CPPS Architectural Layers and Typical Components.

To specify the identities of each network hub and their responsibilities the FMA have been proposed.

FMA layers

FMA layers represent the actual instantiation of the Fluid computing levels onto the manufacturing CPPS and have been organised as follows (Figure 4):

Cloud

IT architectural model where computing services (both hardware and software) are abstracted and delivered to customers over the Internet, on-demand, in a self-service fashion, independent of device and location. Eventually divided into:

- **Public Cloud:** a third-party owned cloud infrastructure which provides Internet-based public services on predefined rules, policies and a pricing model;
- **Private Cloud:** designed to prepare most of the benefits of a public cloud and decrease security concerns. Internally managed computing network by the organisation in order to avoid issues related mainly to information confidentiality, response time and legal

involvements inherent geographic distribution of data storage.

Fog

Initially coined by CISCO in 2014 as a term to indicate the extension of Cloud towards the edge of an enterprise (this is also the reason why it is sometimes referred to as *Edge* computing), it became much more than that, giving computational identity to all those infrastructural components positioned between the public Cloud and the Things in a manufacturing company (Field). In these sense, Fog computing can be seen (following the terminology metaphor) as the computational level which encompasses all the computation, network and functionality IT components from the private Cloud to the *Dew*. The positioning between Field and the private Cloud of the different Fog computing levels components has been identified in accordance with the criteria reported in Figure 1. This layer eventually overlaps the private Cloud and comprises:

- **Mist:** lightweight rudimentary form of computing power that resides directly within the network constellation, using microcomputers and micro-controllers to feed into fog computing nodes and potentially onward towards the CC. A mist device can eventually also have powers similar to the ones

found in cloud servers. But anyway mist and cloud are complementary to each other due to their offline and online usages.

Field

the physical configuration of *Things*, such as robot, programmable logic controller (PLC), 3D printers, warehouses and so forth.

- *Dew*: as introduced in the previous section, authors propose a very basic, eventually embeddable extensions to native, if any, computational capabilities of the physical devices (the role of *Dew* is presented in details in the next section).

Edge

it logically encompasses and organises *Mist*, *Dew* and *Things*.

The Dew of a fluid CPPS

Dew layer represents the novelty in the FMA of a CPPS; the layer where manufacturing physical components are transformed into NES components by providing new (where not existing) or extending their original (where not sufficient) capabilities, be these computational, networking, or related to sensing and/or acting in the manufacturing environment, in accordance to the CPS Core model presented in Figure 3.

During the implementation of a control system for a CPPS, the lack of an appropriate 'interface' functionality has been detected. It became clearer that the future data exchange and communications would become smoother and faster in presence of an additional 'layer' to be introduced as a conjunction, between the *Mist* and the *Things*, in the FMA. This element would extend the capabilities of the individual, separate physical devices on a basic level, usually not provided with any computing knowledge. A special, on-site solution has been developed, but it was obvious that a general solution for this problem would raise the efficiency of communication between devices and software on higher levels. As this basic, embeddable extensions of the capabilities of the separate physical devices, the solution could focus on the individual physical object of the control system. From these findings originated the name for the new layer, *Dew*, for instance, consequently fitting to the meteorological analogy used so far also for *Cloud*, *Fog* and *Mist*. In nature the dew appears as a separate drop on the surface of individual objects; in FMA environment the *Dew* appears in the separate devices as an embedded element for extending its software and hardware capabilities, and so generating NES components. The *Dew* operates in preparation of connection in case of a legacy system; the role and characteristic of this element have been demonstrated with some examples, as illustrated in a separate section.

In the case, for instance, when human agents operate by smartphone a conventional turning machine, they usually scan the work-piece and forwards it to a higher level for further processing (input), and receives in return some commands on the phone to execute the task on the machine. This case can be seen as *implicit Dew*, as the

extension of individual physical capabilities, as well. A *Dew*-related unit can be handled as the basic element or component of a CPS (intended as a networked embedded unit) and considered as a unifying layer where the over-all manufacturing services are to be provided also in lack of certain capabilities: e.g. in case of no Wi-Fi capability, the *Dew* can complete the component by adding to the device this single requirement. Capability interface can be extended both in the form of add-ons or plug-ins available from the device manufacturer or in general as an ad-hoc implementation (like in the case of a customised 3D-printed extension or tool for a robot).

The fundamental aspect of the *Dew* is that its functionality is realised by very basic, embeddable extensions (both at hardware and/or software level) of the native capabilities of individual physical devices, providing them with the minimal computing, networking and acting&sensing capability required by Industry 4.0 CPPS. *Dew* and FMA computing layers' peculiarities have been summarised in Table 2.

Experimental realisation

For the acceptance of the novel architecture presented in the previous section, it is necessary to support with experimental results as presented in the next section.

Working environment

The realisation of previously described concepts started parallel with two different approaches: in physical systems (two locally separated laboratory with similar functionalities) and in a cloud-based simulation environment.

Physical systems

The research institute has two laboratories, one situated near an automotive industrial plant (the Centre of Excellence in Industry 4.0, CEI – in Figure 5(a)) and the so-called SmartFactory (Cyber-Physical Production System sample – Figure 5(b)) in the head-quarter of the institute. The figures give a visual overview of these systems. Both labs contain multiple kinds of low-level controllers with a modularised instruction set which are reachable and invocable over a standardised protocol. There are PLC controlled workstations with a conveyor system to demonstrate a traditional linear production line, while the industrial robots are approachable by autonomous guided vehicles (AGV) as an example for job shop production.

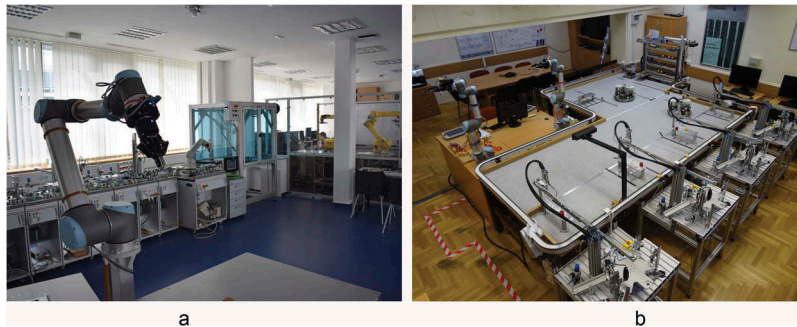
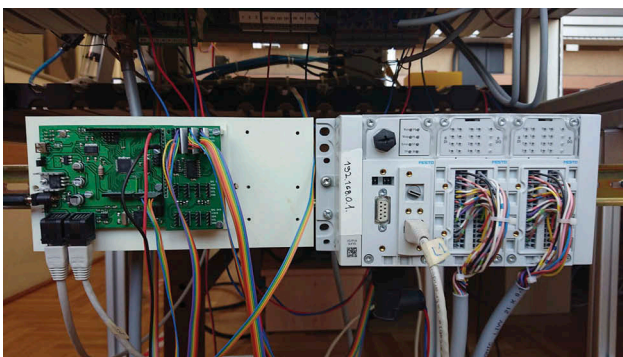
The implementation of the *Dew* layer has been realised in the SmartFactory, as well, and its physical instantiation is illustrated in Figure 6. On the left side of the figure, a new component is visible which improves the PLC's (right side of the figure) capabilities with a converting functionality from LAN TCP/IP protocol to CANOpen protocol.

Simulation environment

For the experiments, authors used the opportunities offered by the authors institute's Cloud, which is an IaaS cloud service for researchers. The range of services starts from simple desktop

Table 2. Dew and other FMA computing layers properties: architecture, key technologies, applications and open issues.

Characteristic	FMA layers				
	Dew	Mist	Edge	Fog	Cloud
Geographical topology	Connected to device	Distributed	Distributed	Decentralised and distributed	Decentralised and unsupervised distribution
Location of service	Connected to individual device	At the boarder on local network	At the edge of local network	At the edge of local network	Within the network
Service Internal structure	Direct Simple, individual	Virtualisation Decentralised, distributed	Virtualisation Decentralised, distributed	Virtualisation Decentralised, distributed	Virtualisation Hierarchical, decentralised, distributed
Relation in the architecture	Core element of the CPS	Not a mandatory layer, may not require cloud	Can work without cloud or fog	Extending cloud to the edge of the network	Service-driven element of the network
Network connection	Customer development, plug-in	Wireless or wired	Wireless or wired	Wireless or wired, works well with wireless IoT, mobile devices	IP network
Number of nodes	Few to many	Limited	Large	Large	Few
Computational capabilities	Very limited	Limited	More limited	Limited	High
Storage capacity	Very limited	Very limited	Limited	Limited	Unlimited
Duration of data storage	Transient	Transient	Minutes	Short duration: hours, days or weeks	Months or years
Location of data collection, processing and storage	Field devices	Network edge	Network edge, edge devices	Near-edge and core networking, network edge devices	In the LAN network and Internet
Data heterogeneity	Device only	Data collection from limited sources	Support collection data from different sources	Support collection data from different sources	Data collection from network
Latency	Low	Low	Low	Medium	High
Response time	Sub-seconds	Milliseconds	Milliseconds	Seconds to minutes	Minutes, days, weeks
End-to-end security	Can be well defined	Can be well defined	Same as in other parts of IT system	Same as in other parts of IT system	Cannot be exactly defined or controlled
Scalability	Medium	Medium	High	High, supports elastic computing	High – supports, resource pooling, variable network conditions

**Figure 5.** (a) Physical environment of experiment – CEI Lab. (b) Physical environment of experiment – SmartFactory Lab.**Figure 6.** An example of Dew application.

computers up to computation clusters with the power of supercomputers.

The specification for all the 100 cloud nodes instantiated in this project was the same: 3 GHz CPU, 1 GB RAM and 10 GB MSU. Collaborative agents were developed in Python 2.7, while the authors chose to use a Linux-based operating system: an Ubuntu distribution with minimal additional features installed. The platform provided by the cloud virtual machines for the agent scripts is reachable over a main dedicated cloud node running the coordinating script. This Controller program represents the user interface. The user gains the ability to initialise the agents, start or stop the simulation experiment and collect the final reports and statistics.

Table 3. Agent roles description in the cloud simulation architecture.

Agent role	Capability
Controller	<p>is aware/acknowledged of every agent acting in the cloud infrastructure</p> <p>instantiates and activates its own behavioural model</p> <p>assigns roles to node agents</p> <p>decides on collaboration network (manager, peer, subordinate)</p> <p>informs interested parties about the existence other agents</p> <p>acknowledges every agent about the end of the simulation</p> <p>receives final statistics</p> <p>receives lists related to tasks executed during the simulation period</p>
Basic peer	<p>receives lists related to messages exchanged</p> <p>instantiates and activates its own behavioural model</p> <p>discovers agents relevant to its execution logics</p> <p>accepts tasks assigned to it</p> <p>executes tasks</p> <p>in case of errors, it delegates its task to neighbour-agent</p> <p>if delegation not feasible, it will inform its reference agent</p> <p>randomly generates further task for sub-divisions (2 exp N)</p> <p>reports to its supervisor agent about the state of the ongoing process</p> <p>receives report on the process by other peer agents</p> <p>collects responses from peer agents acting on a lower level</p> <p>stores operation specific messages</p> <p>stores operation specific task details</p>

Agent-based simulation

The cloud simulation constellation has been populated with agents addressing two main behavioural roles: the *controller* (user) and the *collaborator* (peer-agent). Their main functional and technical peculiarities are summarised in the following subsections.

Agent functionality

In Table 3 the capabilities and tasks assigned to each roles are enumerated.

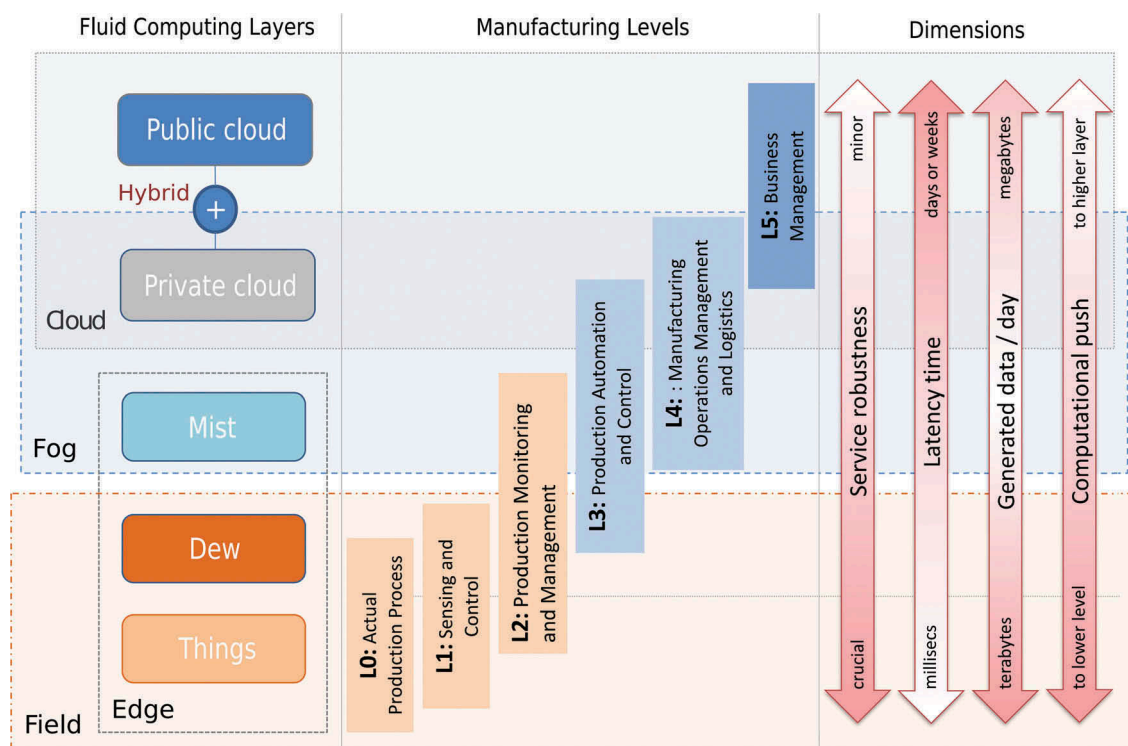
Agent connectivity

Agent connectivity is defined between so-called *Basic Peers*. Each *Basic Peer* is assigned furthermore to one specific level, as defined in Figure 7 for a clearer role. These *Agent Types* are in accordance with the industrial terminology but as visually represented in Figure 8(a,b) the *Plant* and *Shop Floor* are on the same level because their control mechanisms are the same from the FMA perspective while their cycle time is different. As an addition Figure 7 illustrates the functional positioning of all traditional production levels and selected operation scaling factors over FMA layers.

The agent network is managed by the controller and is susceptible to significant changes at initialisation. On Figure 8(a) the traditional centralised production hierarchy is displayed. Every agent reports to the top, while they gain their tasks from their direct superior nodes. In Figure 8(b), in contrast, a distributed network has been drawn, which is consistent with the FMA. Each and every participant in the collaborating network communicate with its subordinates, neighbours and manager, thus the decision-making process is brought to a lower level in the hierarchy.

Agent-exchanged message specification

Messages exchanged between node agents of the simulated collaboration platform are composed of the following parts: a *header* (subject of the message), a *body* (contents of the message), the *sender* (the agent sending the message) and the *receiver* (the recipient agent of the message).

**Figure 7.** Functional positioning of traditional production levels and scaling factors over FMA layers.

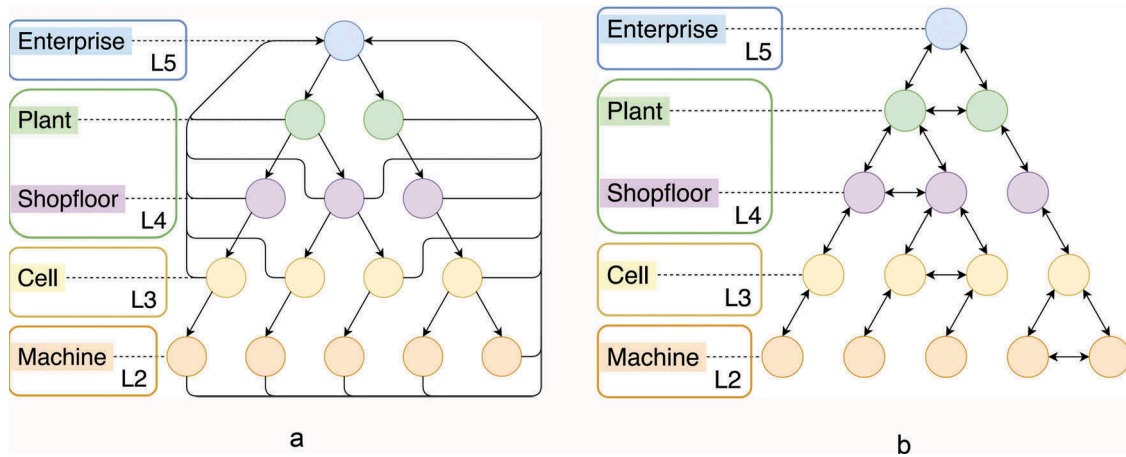


Figure 8. (a) Agent connection topology of a *centralised* control chain. (b) Agent connection topology of a *distributed* control chain.

Table 4. Agent message elements description.

Property	Description	Value type/Possible values
id	Universal Unique Identifier	int
name	Agent Name	string
com	Communication Properties	
ip	TCP/IP Network Address	string
port	TCP/IP Network Port	int
timeout	Communication Timeout	int
type	Agent Role	see Table 6 for details
state	Current Agent Status	see Table 7 for details

The header is the actual name of the function to run by the receiver agent. At first, the body contains the function's arguments, while in the reply message it consists of the possible returning values. Refer to Table 4 for further details on the possible parameters of a message header and body.

Half of the message headers are for initialising the simulation environment, while the second half consists of runtime commands issued by the user or sent out according to the predefined scenario in an automatised way. In most of the cases, the *Controller* acts as the highest authority in the hierarchy (*Enterprise*).

In Table 5 the properties of an agent are enumerated. The possible agent *types* and agent *statuses* are explained in more details in Tables 6 and 7. These attributes are referenced by an *Id* in the agent object but they consistently represent the same types or statuses with the same properties in every instance. The *name* generally consists of a concatenation of the agent type and id, for an easier reference. The *com* object contains the network configuration for the agent.

Preliminary results

The FMA has to be compared with the traditional communication topology to be justified. The simulation environment can be shaped as explained in subsection related to agents' connectivity. At first, the authors implemented the FMA topology in an agent-based simulation environment. In Figure 9 data packages coming from the *Enterprise* agent level and moving down to the *Machine* agent level are graphically represented.

In Figure 10 a focused node is presented. On the left side of the figure, the connections of the node are partially shown, while on its right side the characteristics of the node are introduced. The numbers on top of the coloured circles represent the target tuple $\langle \text{Plant} \mid \text{Shopfloor} \mid \text{Cell} \mid \text{Machine} \rangle$, whereas the number at the bottom indicates the value of the work time in sec.

Preliminary tests with this environment show a decrease in the number of communication messages and the improvement of the system's reactivity (decrease in time to perceive events). This finding implicates the possibility to actually improve system performance with the implementation of FMA. The next step will be the connection of the cloud

Table 5. Agent properties.

Property	Description	Value type/Possible values
id	Universal Unique Identifier	int
name	Agent Name	string
com	Communication Properties	
ip	TCP/IP Network Address	string
port	TCP/IP Network Port	int
timeout	Communication Timeout	int
type	Agent Role	see Table 6 for details
state	Current Agent Status	see Table 7 for details

Table 6. Agent role.

ID	Name	Operation	Cycle time
0	Device	Act & Sense	10 ms
1	Machine	Process	1 s
2	Cell	Produce Products	1 min
3	Shop floor	Engineer Production System	1 h
4	Plant	Engineer Manufacturing of Product	1 day
5	Enterprise	Design Product	20 days
10	Controller	Control Simulation Process	NA

Table 7. Agent status.

ID	Name	Operation
0	Idle	Idle
1	Setup	Pre-execution
2	Operate	In progress
3	Error	Status of error

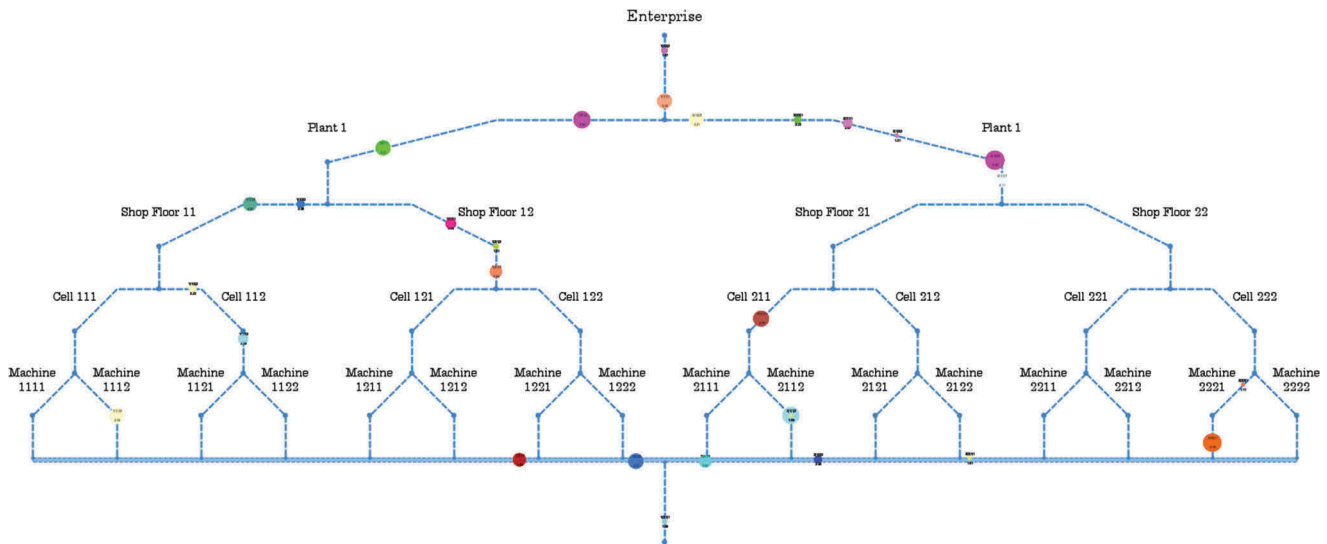


Figure 9. Visualisation of simulation.

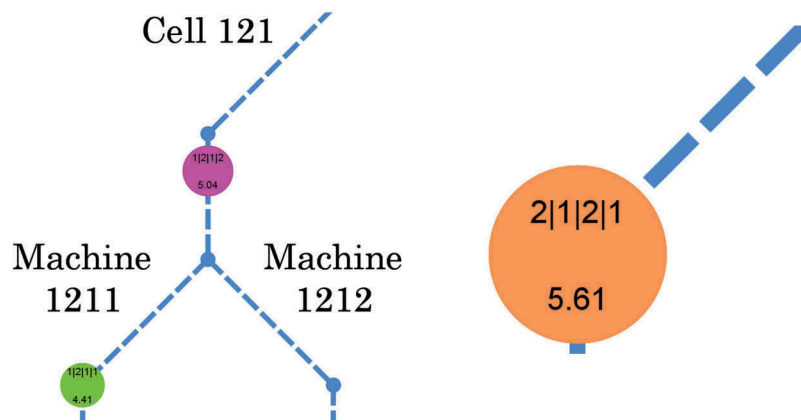


Figure 10. Enlarged section of visualisation.

agent system with the physical systems in order to test the overall architecture.

Conclusions

Today the applications of CPS paradigm can be found in all fields of economy. In industry, the instantiation of CPS is called CPPS and its implementation is realised through the concept of Industry 4.0. As the IoT/IIoT market evolves, Cloud, Fog and Mist computing platforms are converging into Fluid computing platforms for production systems.

In order to provide a seamless flow of expected data between the different layers of the control architecture, the authors have re-organised and applied the concept of Fluid computing to CPPS, giving form to the FMA. In the FMA, Cloud, Fog and Mist have been unified under a single abstraction model, taking into account the crucial aspects typical of the manufacturing service provision: decreased latency, faster decisions and optimal data distribution (service to the data).

Based on practical realisation experiences and theoretical considerations, a fluid computing architecture has been extended with a new layer called *Dew*, whose elements are positioned under the Mist (as part of the Edge) within the field level. The *Dew* is the layer where manufacturing physical devices are transformed into networked and embedded components of the Industry 4.0 scenarios. This has been achieved by augmenting the CPS with all of those capabilities which are the core of a collaborative, distributed manufacturing scenario: computing, networking, sensing and/or acting.

Simulation experiments have been conducted by deploying a constellation of cloud nodes, representing the interactions of physical devices at lower levels. Preliminary evaluations indicate that advantages of such Fluid Manufacturing Architecture derive not only from the introduction of the *Dew* but also from the reduction of communications and data propagation. Results of service simulations underline the expectations regarding the improvement of quality parameters in communications

within the fluid architecture. Main challenges for the implementation of the FMA in real industrial environments may derive from two major areas: business interest of IIoT providers and the lack of trust toward 'novel' approaches. Industrial controller suppliers, at a global scale, usually try to impose their own quasi-standardised solutions and protocols for strategic reasons, and this may threaten the collaborative approach demanded by the FMA.

In case of future introduction of 5G networks, the speed of data flow will be significantly higher so the reaction of control devices will be faster (nearly real-time). The introduction of the Dew layer will result in a smoother, more precise control mechanisms as data and signals frequency on the lowest level will be very high and close to control requirements in critical production scenarios. This will enable a stronger digitalisation and composition of manufacturing services for real-time control of production and transportation robots.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research has been supported partially by the European Commission through the EU H2020 project EXCELL [691829] and by the GINOP-2.3.2-15-2016-00002 grant on an "Industry 4.0 research and innovation center of excellence".

References

- Alraway, A., A. Alhothaily, C. Hu, and C. Xiuzhen. 2017. "Fog Computing for the Internet of Things: Security and Privacy Issues." *IEEE Internet Computing* 21 (2): 34–42. doi:10.1109/MIC.2017.37.
- Arkian, H. R., A. Diyanat, and A. Pourkhalili. 2017. "MIST: Fog-Based Data Analytics Scheme with Cost-Efficient Resource Provisioning for IoT Crowdsensing Applications." *Journal of Network and Computer Applications* 82 (Supplement C): 152–165. doi:10.1016/j.jnca.2017.01.012.
- Barik, R. K., A. C. Dubey, A. Tripathi, T. Pratik, S. Sasane, R. K. Lenka, H. Dubey, K. Mankodiya, and V. Kumar. 2018. "Mist Data: Leveraging Mist Computing for Secure and Scalable Architecture for Smart and Connected Health." *Procedia Computer Science* 125: 647–653. doi:10.1016/j.procs.2017.12.083.
- Bourges-Waldegg, D., Y. Duponchel, M. Graf, and M. Moser. 2005. "The Fluid Computing Middleware: Bringing Application Fluidity to the Mobile Internet." In *The 2005 Symposium on Applications and the Internet*, 54–63. IEEE, Trento, Italy.
- Chandramouli, B., J. Goldstein, R. Barga, M. Riedewald, and I. Santos. 2011. "Accurate Latency Estimation in a Distributed Event Processing System." In *Proceedings - International Conference on Data Engineering*, 255–266. IEEE, Hannover, Germany.
- Corsaro, A., ed. 2016. *Cloudy, Foggy and Misty Internet of Things*. ICPE '16. New York, NY: ACM.
- Graf, M. 2003. "Fluid Computing. ERCIM News SPECIAL THEME: Applications and Service Platforms for the Mobile User, No. 54." Accessed June 7 2018. <https://medium.com/@YogeshMalik/fog-computing-edge-computing-mist-computing-cloud-computing-fluid-computing-ed965617d8f3>
- Hu, P., S. Dhelim, H. Ning, and T. Qiu. 2017. "Survey on Fog Computing: Architecture, Key Technologies, Applications and Open Issues." *Journal of Network and Computer Applications* 98: 27–42. doi:10.1016/j.jnca.2017.09.002.
- Kemény, Z., R. J. Beregi, G. Erdős, and J. Nacsá. 2016. "The MTA SZTAKI Smart Factory: Platform for Research and Project-Oriented Skill Development in Higher Education." *Procedia CIRP* 54: 53–58. doi:10.1016/j.procir.2016.05.060.
- Lin, P., M. Li, X. Kong, J. Chen, G. Q. Huang, and M. Wang. 2018. "Synchronisation for Smart Factory - Towards IoT-enabled Mechanisms." *International Journal of Computer Integrated Manufacturing* 31 (7): 624–635. doi:10.1080/0951192X.2017.1407445.
- Linthicum, D. S. 2017. "Connecting Fog and Cloud Computing." *IEEE Cloud Computing* 4 (2): 18–20. doi:10.1109/MCC.2017.37.
- Lu, Y., X. Xu, and J. Xu. 2014. "Development of a Hybrid Manufacturing Cloud." *Journal of Manufacturing Systems* 33 (4): 551–566. doi:10.1016/j.jmsy.2014.05.003.
- Malik, Y. 2017. "Internet of Things Bringing Fog, Edge & Mist Computing." Accessed June 7 2018. <https://medium.com/@YogeshMalik/fog-computing-edge-computing-mist-computing-cloud-computing-fluid-computing-ed965617d8f3>
- MESA. 2007. "An Overview and Comparison of ISA-95 and OAGIS." Accessed June 07 2018. <https://services.mesa.org/ResourceLibrary/ShowResource/85c47b67-e6fc-4cfb-a5f9-ff71dce607e2>
- Nassehi, A., D. Schaefer, D. Wu, X. Xu, and M. Zaeh. 2018. "Special Issue on 'Cyber-Physical Product Creation for Industry 4.0.'" *International Journal of Computer Integrated Manufacturing* 31 (7): 611. doi:10.1080/0951192X.2018.1482106.
- National Academy of Science and Engineering, acatech. 2011. *Cyber-Physical Systems: Driving Force for Innovation in Mobility, Health, Energy and Production*. Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-29090-9
- NIST. 2018. "Fog Computing Conceptual Model." Accessed June 7 2018. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf>
- O'Donovan, P., C. Gallagher, K. Bruton, and D. T. J. O'Sullivan. 2018. "A Fog Computing Industrial Cyber-Physical System for Embedded Low-Latency Machine Learning Industry 4.0 Applications." *Manufacturing Letters* 15: 139–142. doi:10.1016/j.mfglet.2018.01.005.
- Orojloo, H., and M. A. Azgomi. 2017. "A Method for Evaluating the Consequence Propagation of Security Attacks in Cyber-Physical Systems." *Future Generation Computer Systems* 67: 57–71. doi:10.1016/j.future.2016.07.016.
- Pedone, G., and I. Mezgár. 2018. "Model Similarity Evidence and Interoperability Affinity in Cloud-Ready Industry 4.0 Technologies." *Computers in Industry* 100 (C): 278–286. doi:10.1016/j.compind.2018.05.003.
- Ray, P.P. 2018. "An Introduction to Dew Computing: Definition, Concept and Implications." *IEEE Access* 6: 723–737. doi:10.1109/ACCESS.2017.2775042
- Sunny, S. M. N. A., X. F. Liu, and M. R. Shahriar. 2018. "Communication Method for Manufacturing Services in a Cyber-Physical Manufacturing Cloud." *International Journal of Computer Integrated Manufacturing* 31 (7): 636–652. doi:10.1080/0951192X.2017.1407446.
- Szymaniak, M., D. Presotto, G. Pierre, and M. van Steen. 2008. "Practical Large-Scale Latency Estimation." *Computer Networks* 52 (7): 1343–1364. doi:10.1016/j.comnet.2007.11.022.
- UK Met Office. 2017. "What Is the Difference between Mist and Fog and Haze?" Accessed November 7 2018. <https://www.metoffice.gov.uk/learning/clouds/fog/difference-mist-and-fog>
- Verba, N., K.-M. Chao, A. James, D. Goldsmith, X. Fei, and S.-D. Stan. 2017. "Platform as a Service Gateway for the Fog of Things." *Advanced Engineering Informatics* 33: 243–257. doi:10.1016/j.aei.2016.11.003.
- Wang, Y. 2015. "Cloud-Dew Architecture." *Journal of Network and Computer Applications* 4 (3): 190–210. doi:10.1504/IJCC.2015.071717.
- Wu, D., S. Liu, L. Zhang, J. Terpenney, R. X. Gao, T. Kurfess, and J. A. Guzzo. 2017. "A Fog Computing-Based Framework for Process Monitoring and Prognosis in Cyber-Manufacturing." *Journal of Manufacturing Systems* 43: 25–34. doi:10.1016/j.jmsy.2017.02.011.