

# Architecting a Hybrid Cross Layer Dew-Fog-Cloud Stack for Future Data-Driven Cyber-Physical Systems

Marc Frincu

*e-Austria Research Institute and West University of Timisoara  
Department of Computer Science  
Timisoara, Romania  
Email: marc.frincu@e-uvt.ro*

**Abstract**—The Internet of Things is gaining traction due to the emergence of smart devices surrounding our daily lives. These cyber-physical systems (CPS) are highly distributed, communicate over wi-fi or wireless and generate massive amounts of data. In addition, many of these systems require near real-time control (RTC). In this context, future IT platforms will have to adapt to the Big Data challenge by bringing intelligence to the edge of the network (dew computing) for low latency fast local decisions while keeping at the same time a centralized control based on well-established scalable and fault tolerant technologies brought to life by cloud computing.

In this paper we address this challenge by proposing a hybrid cross layer dew-fog-cloud architecture tailored for large scale data-driven CPSs. Our solution will help catalyze the next generation of computational platforms where mobile and dynamic IoT platforms with energy and computational constraints will be used on demand for storing and computing Big Data located nearby in near real-time for local decisions and extend to cloud systems for fast orchestrated centralized decisions. The proposed architecture aims to leverage the advantages of both cloud and dew systems to overcome the challenges and limitations of modern communication networks.

We also discuss two real-life life solutions from the field of smart grids and smart transportation systems.

## 1. Introduction

The Internet of Things (IoT) is growing at a fast pace especially with the increasing presence of networked interacting components present in everyday activities such as transportation, distributed robotics, medical monitoring, automatic pilot avionics, and the emerging smart city. In 2016, 6.4 billion connected devices were, according to Gartner [1], present around the world with forecasts predicting up to 30 billion by 2020. Many of these devices are part of autonomous cyber-physical systems (CPS), e.g., smart grids, self-driving cars, intelligent traffic management systems, and require near real-time control (RTC) for coordination and critical decision making. Thus, IoT is already impacting our society in ways that will revolutionize our lives and how we interact with each other and with the CPSs themselves from a cultural and socio-economic perspective. However,

migrating towards a fully interconnected world poses both scientific and technological challenges which need to be met in the short term by novel solutions if the vision of an IoT world is to become reality. These systems will no longer be controllable through complex linear and nonlinear systems of equations but through **data-driven machine learning algorithms**. In particular, the Big Data problem together with the low energy requirements and computational resources of IoT devices need to be mitigated through novel platforms which enable the *extension* of these systems to large scale processing environments such as the clouds which already possess the necessary software stack for data analytics of heterogeneous data streams.

The large amount of high speed sensor data arriving in variable data rates poses serious design and cost challenges to existing on-edge cyber-infrastructures. The stress of having tens of thousands of devices communicating by using wi-fi or wireless over vast distances with centralized data centers placed in the cloud is already limiting the ability of existing smart grids to send real-time data (e.g., most smart grids can only send aggregated data every 8 hours or even daily which means that decisions at finer granularity are hard if not impossible to achieve). This greatly limits the ability to efficiently control these large distributed CPS in near RTC. The problem is that the cost of upgrading the data transmission medium (e.g., wi-fi, wireless, PLC, bluetooth) to handle city wide CPS infrastructures of millions of devices makes the current approaches which centralize data and control inefficient. Thus, cloud computing alone will not be able to address the issue of CPS generated Big Data.

One solution would be to move the processing towards the edge of the network closer to the data source. We identify here three layers of computation [2]: edge or dew (on device), fog (local area with partial view of the CPS), cloud (global view of the CPS). *Combining these three layers in a single coherent scalable data-driven solution is the main challenge that IoT will have to overcome to truly become ubiquitous*. While clouds offer the premise for on-demand scalable computing they need the data delivered to them before proceeding. Battery powered edge devices already have some processing power and storage capacity to preprocess local data before sending them up the hierar-

chy for centralized control and optimizations. In this setup the network itself becomes the bottleneck. *A hierarchical approach where intelligence is brought near the edge and combined with the advantages of using clouds is therefore required.* In this context, the limitations of the network become a problem especially for near RTC based CPSs.

Fundamental research is warranted in several core areas ranging from hardware infrastructure, networking and communication to programming models, scalable and resilient distributed execution frameworks as well as domain specific software optimizations. This paper addresses the latter end of the spectrum. The objectives are twofold: 1) propose an architecture for generic near RTC based CPS to leverage fog and cloud computing, and 2) map the proposed architecture on two use cases with different requirements, demonstrating the wider applicability of our solution.

The rest of the paper is structured as follows: Section 2 details existing solutions, Sect. 3 describes the proposed cross layer architecture, Sect. 4 describes the software stack envisioned to be deployed on top of the architecture, Sect. 5 describes two possible use cases, and finally Sect. 6 outlines the key points of the paper.

## 2. Related Work

While much research has been conducted on designing secure and scalable CPSs [3], [4], [5], [6], [7] few focus on utilizing cloud infrastructure to enable scalability and RTC [8], [9], [10]. Most of the existing work is geared towards challenges in specific CPSs such as vehicular networks [8] or smart grids [11] and employ several domain specific strategies to achieve RTC.

A framework to integrate IoT and CPS is proposed in [12]. However, it does not consider the resources of clouds to enable fast coordinated control over large scale distributed CPSs.

We propose to abstract the features common across an array of CPSs and develop a generic hierarchical architecture and middleware with scalability, reliability, and adaptability as its fundamental principles.

Finally, studies [9], [13], [14] have been conducted on evaluating the feasibility of using clouds in the context of CPS and on identifying challenges, performance issues and bottlenecks. By building on these results we develop an architecture that will be beneficial not only in the context of CPSs but also in other domains where near RTC is crucial.

## 3. Proposed Architecture

We assume CPSs to comprise of low energy and memory distributed devices which need to share their information in order to globally optimize the entire CPS. Examples of such systems include smart grids where each smart meter and sensor needs to send its information to the utility which then decides on the critical areas of the power grid. However, clients can still optimize their consumption by taking decisions based on local information.

Orchestrating large scale CPSs is challenging due to the data deluge coming from their sensors which needs to be timely processed and analyzed for decision making. To avoid flooding the network with data several choices exist. First, the sampling interval can be decreased but this can lead to crucial loss of information which cannot be accurately reconstructed through interpolation techniques. Second, data could be preprocessed in situ before sending *meta-information* to the *central controller* for the heavy processing part of global decision making. The *local processing* could act as a local optimization where decisions requiring a local or partial view of the entire CPS are taken without involving the central controller.

Focusing on the second approach we argue that in order to make the process efficient the central controller could be employed only when localized control fails to keep the system within predefined parameters as given by the *control model*. The control model is a data-driven model used by machine learning algorithms to take local or global optimizations of the CPS. In a smart grid context this means that while certain users could optimize their consumption reducing the stress on the power grid in demanding days (i.e., hot days when A/C is responsible for a large portion of the total energy consumption) they do not sum up to form the critical mass required to reduce the demand below the utility generation capacity.

Figure 1 depicts the envisioned architecture for such a hybrid system. The entire system is event based where the execution of each component is triggered by the occurrence of one or more events. There are three key components we discuss next in relation with the events used for triggering them.

**Big Data preprocessing and local control** component which enables dew computing. Data is generated locally on the edge devices. Based on the recorded data data pruning takes place and historical data is stored within the limited device memory of each device. Then, data energy efficient and low memory footprint machine learning algorithms can be used to predict the future state of each device assuming no outside world interaction. This simple model can be extended by allowing nearby devices – in the fog – to share their information and feed their data-driven algorithms with fog data. Section 5 will further discuss the implications of this approach on two different use cases. This preprocessed data (either by each device or by a group of nearby devices) is sent to the RTC control component. By sending only partial data we can relieve some of the stress on the network and enable better horizontal scalability of the CPS. Depending on the CPS the preprocessed data could be sent only when a global deterioration of the overall system is noticed. This situation could be triggered by an event from a module monitoring the control accuracy in the RTC control component.

**RTC component for fast cloud processing.** To efficiently control large scale CPSs global information is required. This information can be exchanged through a P2P approach among edge devices or can be handled centrally. It is obvious that for near RTC this information can easily saturate,

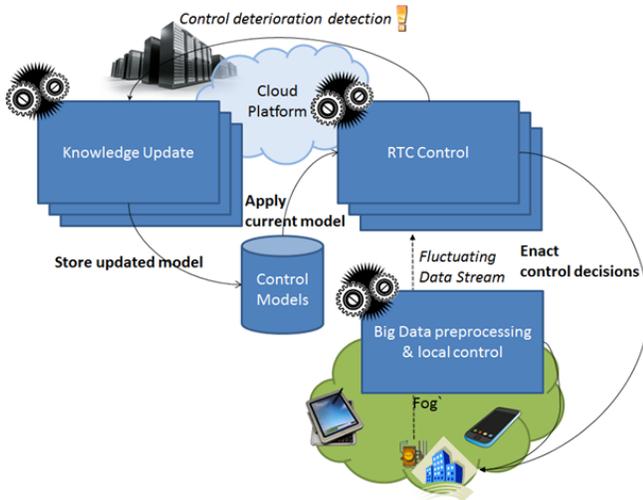


Figure 1. Envisioned hybrid cross layer architecture for dynamic data-driven CPSs.

in both cases, the network pushing the need for alternative methods which do not require constant near RTC data delivery. In our case we argue for a cloud based approach due to the elasticity, reliability, and support (i.e., software like Apache Storm, Spark, MapReduce) of such systems to process Big Data. This component is responsible for the management of the CPS and acts the central controller. Based on the control model and recent data from the edge devices it updates the global view of the CPS and takes decisions to globally optimize the system. Recent works [15] have also shown that it is possible to not rely on information from all edge devices but rather use causality to reduce the search space and still be able to take accurate decisions.

**Knowledge update** component. This component is responsible for updating the control model used by the RTC control component. The control model can refer to a pair  $\langle \text{machine learning algorithm, data} \rangle$  where it has been shown [16] that the efficiency of a machine learning algorithm depends on the input data. Hence, based on the characteristics of the data arriving from the edge devices different algorithms could be used and changed online. The update in the control model could be triggered by the RTC component when significant changes in the data properties are noticed or when a control deterioration is observed despite using up to date data from all edge devices. the entire collection of control models is computed offline based on benchmarks on real-life CPS data.

## 4. Envisioned Software Stack

Based on the proposed architecture, a hybrid dew-fog-cloud control system for near RTC of CPSs would have to implement the following software stack comprising of on-edge algorithms, middleware, and elastic machine learning on clouds.

### 4.1. Low Energy and Memory On-Edge Algorithms

Most edge devices have limited memory and energy consumption limitations due to battery lifetime. As example, the current state of the art smart meters have a memory of only 92KB on which they store temporary data and perform minimal data preprocessing. The remaining memory, while extremely low could be used for data pruning and simple data-driven predictions. The predictions could be done either on the smart meter or at the data concentrator (gateway to a set of customers) in the fog. Alternatively, some of the data could be sent to customers mobile devices – if in the local area – for processing.

Performing simple machine learning on these edge devices will require analysis and redesign of existing algorithms to tailor them for memory and energy constrained environments without significantly impacting their accuracy. Furthermore, the algorithms will have to consider data scarcity and veracity.

### 4.2. Dew-Fog-Cloud Middleware

Bridging fog and clouds requires a middleware capable of leveraging the benefits of clouds to improve the control of edge devices. The three-tier middleware will enable:

- Dew level:
  - Local data preprocessing and control decisions by using on on-edge algorithms for low energy and low memory environments;
- Fog level:
  - Preprocessed data aggregation and processing to reduce network overhead when communicating updates to the cloud;
- Cloud level:
  - Periodic control model updates to train the machine learning analysis on incoming data;
  - Near RTC centralized decisions based on the current control model.

The middleware consists of three key pluggable components as depicted in Fig. 1, each acting as a plug-in for easy customization. Inter-component communication needs to be reliable and light to reduce chances for network congestion and to enable horizontal device scalability to support large scale distributed CPSs. Hence, some tasks will have to be migrated towards the edge either directly on the device (dew computing) or in a local area aggregator (fog computing).

Since two key components, the RTC and the Knowledge update, are deployed on clouds, several key aspects will have to be overcome. These include reliability, variability in cloud resource performance, data privacy, and timely delivery of the control decisions to meet the near RTC requirements. In [13] several cloud performance issues of a community seismic application running on virtual machines were identified. These include variable load and deadline misses for

processing requests. While the analysis was done on a non elastic cloud environment running on Google AppEngine it does outline some key problems encountered by applications requiring near RTC which require the redesign of applications and algorithms to support elastic environments for fast processing. Building such an infrastructure requires taking into account the cost impact. The cost model will have to link virtual machine performance fluctuations, infrastructure reliability, network latency, and cloud elasticity with the time needed to perform knowledge training, behavior forecasting, and control decision and enactment. Cost will therefore measure the penalty for performance deterioration due to late decisions in soft and hard RTC CPSs. The performance model will also have to be tightly linked to the economics of clouds which allows significant cost reductions through their pay-per-use approach since it may be more affordable to deploy the middleware on public clouds rather than on private infrastructures.

### 4.3. Elastic Machine Learning on Clouds

An important part of the middleware is represented by the machine learning algorithms used by the central controller and selected based on the current control model. These algorithms will have to be adapted for elastic cloud infrastructures and suitable for deployment in streaming software such as Storm and Spark. While public providers such as Amazon and Azure already offer machine learning tools, some of the algorithms in use by CPSs may be newly designed for the specific use cases, improved versions, or ensemble models of several existing algorithms. In these cases, these algorithms which are not normally designed for elastic and environments will have to be re-engineered.

## 5. Use Cases

In this section we focus on describing two use cases in relation to our proposed architecture by detailing functional aspects and particularities in each of them. The objective is to demonstrate the generality of our proposed solution.

The first use case deals with smart grids which rely on homogeneous data transmitted periodically using limited network transmission environments for global consumption optimization. The second use case is a smart transportation system where the route of a car must be optimized by considering heterogeneous sources of information as well as the behavior of other drivers. Heterogeneity is not the only difference between the two. Smart grids are inherently static while vehicles are constantly moving. A generic cross layer architecture must therefore cope with both cases efficiently. In the smart transportation case processing data in the nearby fog could lead to data inconsistency and wrong decisions as a car which shared information might leave the fog after a decision is taken but before it is transmitted to the drivers.

### 5.1. Smart Grids

An increasing number of utilities migrate their traditional power grids to automated smart grids. These systems

rely on smart meters to receive consumption data. As seen most of these meters have inbuilt limited memory allowing them to store historical data and to run simple algorithms. Information from smart grids can be used both locally by clients to reduce their consumption and globally by utilities to balance the supply-demand of energy during peak hours. In both cases there is a need for constant monitoring and adaptation based on latest data. The problem is challenging especially for the utility which needs to constantly monitor clients and select those that are most likely to respond to energy curtailment actions without impacting too much their comfort. Customer selection algorithms and machine learning algorithms for predicting the energy curtailment of each selected customer are needed and during peak hours they need to execute within the sampling period of the smart meters. Obviously to maximize efficiency the sampling rate must be as high as possible.

When mapping this use case on our architecture we have the smart meters acting as edge devices and the utility which relies on cloud computing to process the incoming data. As seen in Sect. 1 data networks currently limit the transmission rate of smart meters. Therefore, bringing some of the centralized decision toward the edge could greatly impact the efficiency of the smart grid for near RTC control. Such localized control (by the Big Data preprocessing and local control component running on the smart meter) could include optimization of individual customers' consumption based on their own historical usage and behavior. Furthermore, these predictions for the consumption of each customer could be sent to the cloud for taking global control actions and to target customers most likely to take part in the energy curtailment event (i.e., demand response). Sending predictions to the RTC component instead of raw data based on which the RTC component would make its own predictions enables us to remove some of the stress in both the network and the component itself. In addition, each smart meter could be remotely programmed with a different prediction method tailored to each customer's behavior. The RTC component would then use these predictions to select the most suited customers and to monitor the impact on the supply-demand balance. When an increasing negative impact (in terms of predicting the aggregated consumption in the next time frame for instance) would be noticed the component would adapt by changing both the customer selection and the consumption prediction algorithms by calling the knowledge update component.

### 5.2. Smart Transportation

Recent years have witnessed the installation of intelligent traffic lights and speed control systems in major cities. Combined with GPS information and smartphone applications which constantly monitor traffic and cars that exhibit increasingly intelligence when it comes to driving efficiency it becomes clear that integrating the three systems (traffic, car, and smartphone) can reduce CO2 emissions and traffic congestion. Such an integrated system would consider traffic speed, fuel consumption, traffic lights, and

driver preferences to pick the optimal route to a given point by leveraging information from other cars, drivers, and the traffic management system.

When mapping this use to our architecture we have the sensors on cars, the smartphones, and the sensors on the street as edge devices. This heterogeneous CPS will have to share data and coordinate to reduce CO2 emissions, improve traffic speed and reduce congestion. In this scenario, each car would take local decisions on how to adapt speed and path based on the preferences of the driver, nearby vehicles, and traffic speed and situation taken from Google Maps for instance. This decision could be taken in a car controller or by the smartphone application and then presented visually to the driver both playing the role of the by the Big Data preprocessing and local control component. Then, by relying on information from other cars it would further optimize its path by considering the direction (and possibly the destination) and speed of other drivers. In this way traffic jams and average speed can be improved by predicting where congestions are likely to happen. Given the limited processing and storage power of car sensors and controllers, the energy constraints of smart phones, and mobility of nearby vehicles this global optimization based on swarm intelligence can be taken centrally on a cloud system through the RTC component. Information from the CPS is then processed and analyzed and path optimization algorithms based on traffic and driver behavior can be applied to suggest the best routes. In this case the knowledge update component could pick from the control model between various algorithms for stream based route selection and prediction algorithms based on incoming data properties.

## 6. Conclusion

In this paper we have argued that due to the complexity of IoT systems cross layer dew-fog-cloud systems should be developed. However, since the data transmission network limits both the scalability and data rate of edge devices some of the processing usually done on clouds needs to be migrated towards the fog by leveraging the storage and processing capabilities of edge devices. Based on these we have proposed a hybrid cross layer architecture and a possible software stack consisting of low-energy low-memory on-edge machine learning algorithms, middleware, and elastic machine learning algorithms for clouds. Two use cases have been investigated with references to the proposed architecture.

Future work will include building a prototype and implementing some machine learning algorithms for edge and cloud environments.

## Acknowledgments

This work has been partially funded by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI - UEFISCDI, project number PN-III-P3-3.6-H2020-2016-0005, within PNCDI III, and by the EU H2020 CloudLightning project under grant no. 643946.

## References

- [1] A. Nordrum. (2017) Popular internet of things forecast of 50 billion devices by 2020 is outdated. (accessed January 3, 2017). [Online]. Available: <http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>
- [2] K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," *Open Journal of Cloud Computing (OJCC)*, vol. 2, no. 1, pp. 16–24, 2015. [Online]. Available: [https://www.ronpub.com/OJCC\\_2015v2i1n03\\_Skala.pdf](https://www.ronpub.com/OJCC_2015v2i1n03_Skala.pdf)
- [3] Y. Xin, I. Baldine, J. Chase, T. Beyene, B. Parkhurst, and A. Chakraborty, "Virtual smart grid architecture and control framework," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2011, pp. 1–6.
- [4] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. S. Gupta, "Ensuring safety, security, and sustainability of mission-critical cyber-physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012.
- [5] P. Bogdan and R. Marculescu, "Towards a science of cyber-physical systems design," in *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*, 2011, pp. 99–108.
- [6] S. Karnouskos, "Cyber-physical systems in the smartgrid," pp. 20–23, 2011.
- [7] Z. Wang, Y. Zhang, and K. Du, *Cyber-Physical Traffic Systems: Architecture and Implementation Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 490–500.
- [8] H. Abid, L. T. T. Phuong, J. Wang, S. Lee, and S. Qaisar, "V-cloud: Vehicular cyber-physical systems and cloud computing," in *Proceedings of the 4th International Symposium on Applied Sciences in Biomedical and Communication Technologies*, ser. ISABEL '11, 2011, pp. 165:1–165:5.
- [9] M. Kim, M. O. Stehr, J. Kim, and S. Ha, "An application framework for loosely coupled networked cyber-physical systems," in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010, pp. 144–153.
- [10] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09, 2009, pp. 85–98.
- [11] A. J. Conejo, J. M. Morales, and L. Baringo, "Real-time demand response model," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec 2010.
- [12] T. S. Dillon, H. Zhuge, C. Wu, J. Singh, and E. Chang, "Web-of-things framework for cyber-physical systems," *Concurr. Comput. : Pract. Exper.*, vol. 23, no. 9, pp. 905–923, Jun. 2011.
- [13] M. Olson and K. M. Chandy, "Performance issues in cloud computing for cyber-physical applications," in *2011 IEEE 4th International Conference on Cloud Computing*, July 2011, pp. 742–743.
- [14] Y. X. Junhua Zhao, FushuanWen and Z. Lin, "Implementing an essential computing platform for future power systems," *Automation of Electric Power Systems*, vol. 34, no. 15, pp. 1–8, 2010.
- [15] S. Aman, C. Chelmiss, and V. K. Prasanna, "Influence-driven model for time series prediction from partial observations," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 601–607.
- [16] S. Aman, M. Frincu, C. Chelmiss, M. Noor, Y. Simmhan, and V. K. Prasanna, "Prediction models for dynamic demand response: Requirements, challenges, and insights," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 338–343.