Cloud, Fog, and Dew Robotics: architectures for next generation applications

Alessio Botta*+, Luigi Gallo*, Giorgio Ventre*+
Department of Electrical Engineering and Information Technology
+*University of Napoli Federico II*, Napoli, Italy
**NM2 SRL, Napoli, Italy*
alessio.botta@unina.it, luigi.gallo10@studenti.unina.it, giorgio.ventre@unina.it

***Abstract*—Robots are rapidly moving from isolated to connected systems, with more and more operations offloaded to external systems. Cloud Robotics has indeed become a new important trend in this research field. In this paper we advocate the necessity to move a step forward in this direction considering the real needs of new generation robotics systems. Such systems have indeed very peculiar requirements with respect to the traditional applications that have progressively exploited Cloud infrastructures. Cloud is surely an important addition, but robots still require important computations to be performed closer to or onto them for, e.g., critical decisions involving human interaction. This motivates our work in which we delve into the possible alternatives to pure Cloud Robotics, considering Fog- and Dew-computing as better suited to next generation applications. We describe these complementary but alternative architectures and how they match the requirements of robotics applications. We identify Dew Robotics as the most promising architecture and motivate our choice. We present three use cases we have been working on, to better illustrate our proposal. We finally draw some conclusions for this emerging yet important research field.**

***Index Terms*—Cloud Robotics, Computer Networks, Network Performance**

I. Introduction

In the years, robotic systems have been equipped with a growing number of sensing units to be increasingly autonomous. The processing of data coming from such sensors can be very expensive in terms of energy (e.g. for the analysis of videos captured by cameras). Robots have originally been conceived with the necessary intelligence on-board. This was due to a number of historical reasons, including the scarce availability of proper network connections able to support the data transfer to and from the robot. The improvements of networking technologies made the use of external computing systems to offload the most resource-intensive tasks possible, while the increasing volume of data to be processed actually made it a necessity.

On the other hand, Cloud Computing has established as the de-facto standard for intensive data processing and storage, with its potentially unlimited resources, provided on demand, in a cost and power efficient fashion. Its wide availability of computational and storage resources appears to be almost unlimited and the necessary management effort of provision is minimal. Cloud is actually ubiquitous and hence accessible from almost everywhere. It also operates on demand, according to user requests, provides economies of scale, and facilitates sharing of data across systems and users.

Cloud Robotics was then naturally born from these premises, and it is now considered an important paradigm for current and future robotics applications. Today, several efforts are moving in this direction, including research projects, software frameworks, and commercial products already on the market. Among them, we can cite the Google self-driving car, a car referring to maps, images, and trajectories collected in the cloud to drive autonomously without human intervention [1]; the million object challenge, in which a robot tries several times to grasp a new object and, when it manages, it uploads the relative settings on a global library [2]; the verb surgical, a new system of robotic assisted surgery that improves typical performance connecting to the cloud [4]; and the decluttering robot, using an updated online dataset to put each encountered object in its correct location [5]. Cloud Robotics is also considered an important use case for 5G mobile technology which is currently being standardized by 3GPP and ITU [6], [7].

Robotics is expanding in several directions today and the applications envisioned are very numerous and diverse. We expect robotics to be more and more pervasive in the near future, like the Internet has been in last few decades. Robots are foreseen to be ever more present in our homes with cooking, decluttering, cleaning, and companion robots to come soon. They are also expected to spread out in our offices, hospitals, and streets. Offloading resource intensive tasks to external computers, e.g. the Cloud, is considered mandatory for most of these applications. However, robots interact more and more with humans for these applications with respect to traditional ones, such as, for example, industrial automation. Human proximity means much stricter requirements for safety. Robots must not collide and hurt humans, other living beings, or objects. Performing all computation in the Cloud means moving data to and from the Cloud, which implies latency and losses. These impairments can prevent the robot from respecting the strict safety requirements, which basically means that not all the computation can be moved to the Cloud. Cloud-based robots have also issues when network connection does not work properly: all actions depending on Cloud computation are blocked, which can also completely stall the robot. Summing up, today and tomorrow robotic applications require a proper trade-off regarding what to perform externally and what has to remain closer or local to the robotic system.

Different solutions have been proposed in literature, but a general approach and architecture is still missing. In this paper, we firstly describe current research efforts in Cloud Robotics and overview the different and diverse contributions provided. Afterwards, we introduce the main architectures available for next-generation robotics application, i.e., Cloud-, Fog-, and Dew-Computing. We analyze their pros and cons looking at these applications. We then present our proposal, called Dew Robotics, and three use cases to illustrate its application in real world scenarios. We finally conclude the paper in Sec. VI.

II. State of the Art

A. Current and Future Robotics Applications

Grasping an object, especially if unknown, is a typical, unsolved problem in robotics and the Cloud can be of great help for this important task. Google Cloud object recognition engine was recently used to empower a Willow Garage PR2 robot for this cloud-aided grasping and the results were much better than previous ones without the Cloud [8].

Simultaneous Localization and Mapping (SLAM) aims at building a map of the environment and finding the location of the robot inside the map. This is another important application in robotics. Several approaches have been proposed in literature and the possibility to exploit the Cloud for improving this task has also been considered. Besides the availability of increased computing power, Cloud also provides the possibility to share information among several robots to complete the task collaboratively [9]–[11].

The exploitation of **Big Data** repositories is considered a disrupting application of Cloud Robotics. With access to vast resources of data (trajectories, images, videos, maps etc.) that could not be maintained in on-board memories or in conventional database systems, the main benefit are several, among which we cite an easier process of machine learning. Online large datasets can be consulted by the robots or autonomous systems to accomplish better their tasks. In this way, some previously unattainable applications (computer vision, precise grasps etc.) are now doable.

Another important application is **Collective Learning**: to allow robots to create and collaboratively update joint knowledge bases, which are hosted in a shared storage infrastructure in the Cloud (see the example in Fig. 1) . Such knowledge repositories are ubiquitous and constantly available, offering a simple and powerful way for life-long robot learning. Collecting data from many instances of physical trials, robots can share data on the resulting outcomes of many operations.

Human in the Loop is also foreseen as an important application enabled by Cloud Robotics. In 2009 a project started called Heaphy with the aim of crowdsourced robotics. Heaphy proposers recognized that Artificial Intelligence (AI) was still not ready to control robots for every tasks at home and proposed to use real humans paid to remotely control the robots to do homeworks. The idea was to have the human controlling the robot for a while, until it learned how to do these tasks on its own. The project has been abandoned and one of the main problems spotted was related to network latency, preventing the human from properly controlling the robot [12].

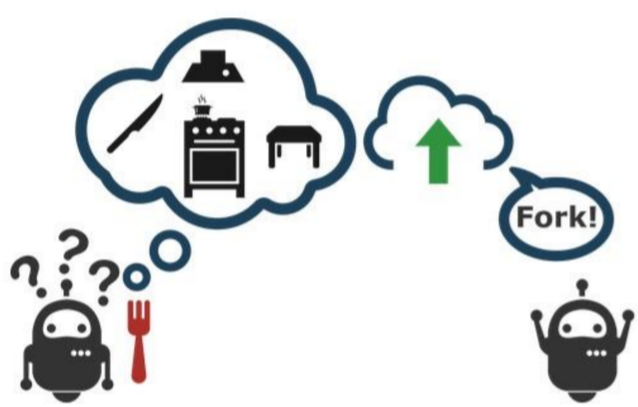

Fig. 1. Shared knowledge in the cloud

Besides the main application scenarios reported above, a number of new, emerging applications are also envisioned in the near future, ranging from autonomous cars to companion robots, from office assistants to home collaborators, from rescuing drones to self-driving bicycles. All these applications cannot be properly and safely implemented relying only on Cloud for computations and storage. For example, imagine a robot moving around in a home environment while using Cloud resources for recognizing objects seen through its video camera. At a given moment, the robot spots an animal that is moving towards it. We, as humans, would move away from the animal trajectory to avoid the impact, and would like the robots to do the same. The Cloud-enabled robot sends the video containing the animal to the Cloud. The network is congested and the video transfer takes a long time. The information regarding the animal does not arrive on time to correct the trajectory and the collision cannot be avoided. As another example, consider a companion robot using the Cloud to recognize the voice commands. If the network is down, the robot is basically down as well.

B. Scientific Literature

Hu et al. [13] propose an architecture for Cloud robotics in which they have two layers, one local among the robots, and another one remote with the Cloud. They make different proposals for the communication frameworks within and between the two layers. They also propose an elastic computing model for Cloud computations. They identify the need for local communications.

Arumugam et al. [14] propose a Cloud framework for FastSLAM using Hadoop on the Cloud and ROS on the robots, with proxy nodes for non ROS-compatible robots. Specific nodes collect data from robot sensors and send them to a HDFS in the Cloud on which Hadoop operates for creating the maps. They present results obtained using a public dataset and no real robots. They also recognize that in a real system network latency and losses can impact the performance.

A survey on cloud robotics and automation systems has been presented in 2015 [15]. The paper presents an interesting definition of Cloud Robot and Automation systems as follows: *Any robot or automation system that relies on either data or code from a network to support its operation, i.e., where not all sensing, computation, and memory is integrated into a single standalone system*. The authors also recognize that these systems systems often include local processing capacity for responses requiring low-latency and for periods or network unavailability or unreliability.

C. Projects

As of 2016, many research projects are pursuing for new cloud robotics developments, ranging from computing resources to systems architecture. A valid example is represented by the RoboEarth project, which envisioned "a World Wide Web for robots: a giant network and database repository where robots can share information and learn from each other about their behavior and environment" [16].

Barbosa et al. [17] developed a cheap robot based on arduino controlled by a smartphone with android. The whole system is managed through Robotic Operating System (ROS, described in the following) running also on a separate computer. The system also uses some Cloud services such as Google maps to locate the robot using the GPS information reported by the smartphone.

Google started a project in 2011 comprising the development of a Java implementation of ROS called rosjava. The project aims at connecting android devices, with cheap robots based on arduino and the Cloud. The rosjava also allows to exploit the sensors of the mobile device (camera, gps, etc.) for the robotic application [18].

D. Software Platforms

Projects aimed at developing software platforms for Robotics have also been presented in literature. In the following we report a few examples: ROS and Rapyuta.

ROS is an open-source, meta-operating system for robots. It runs on Unix-based platforms, so it provides all the operating system services, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. The main feature of ROS is to support code reuse in robotics research and development. It is a distributed framework of processes that enables executables to be individually designed and loosely coupled at runtime. These processes can be grouped into Packages and Stacks, which can be easily shared and distributed. ROS also supports a federated system of code Repositories that enables collaboration to be distributed as well. To make code written for ROS shareable and integrable with others robots software frameworks, ROS supports any modern programming language, and libraries have clean functional interfaces. ROS design, from the filesystem level to the community level, enables independent decisions about development and implementation, but all can be brought together with ROS infrastructure tools [3].

Rapyuta is an open source platform that allows robots to move their processing to commercial datacenters, actually realizing the cloud robotics paradigm. Robots do not to have to perform heavy processing on board, providing computational environments customizable and secure within the Cloud. These computing environments enable also easy access to RoboEarth knowledge repository. In addition, the environments can interconnect so that multiple robots can work together. It also provides a well-documented open source implementation that can be modified to cover a wide variety of robotics scenarios. Rapyuta supports outsourcing of more than 3000 ROS packages. With Rapyuta robots can authenticate on the platform, creating one or more computing environments in the Cloud, and launch the desired processes. Computing environments are private, secure, optimized for data transmission, and can be connected to build parallel architectures. Anyway, the performance is influenced by the latency and quality of the network and data center performance. [21]

E. The Need for a Common Architecture

As seen above, the requirements of a number of future applications cannot be satisfied using an architecture fully based on Cloud Computing. This has also been recognized in literature [19], in which efforts are being made in this direction, but most of them are application-specific with no general architecture. In current years, scientific papers, research projects, and software platforms each proposes its own solution which is scarcely reusable. In summary, for new robotics such applications, a new approach is necessary and a common architecture is needed, and this is still missing in literature. To fill this gap, in this paper we propose a new architecture for current and future robotic applications that takes into account their requirements. We start reviewing Cloud-, Fog-, and Dew-Computing. We then discuss how these computing architectures can be applied to the robotics field, and propose the use of Dew-Computing architecture to meet all the requirements of current and future robotics applications. We therefore propose the use of the term Dew Robotics for the application that is best suited to such requirements.

III. Possible architectures

In this section we review the computing architectures that are most suited for robotics applications, i.e. Cloud, Fog, and Dew Computing. Fig. 2 shows the three computing architectures, which are briefly described in the following of the section.

A. Cloud Computing

According to the NIST (American National Institute of Standards and Technology) definition, *Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications, services)*

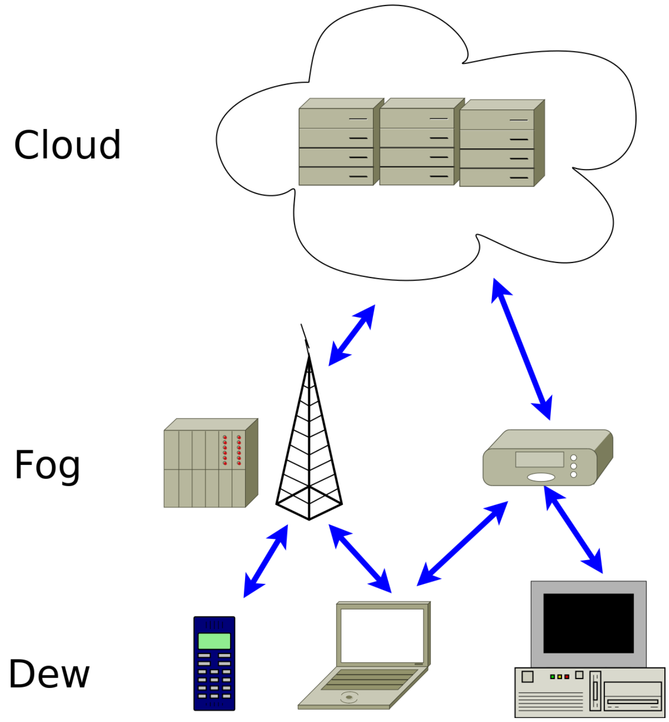

Fig. 2. Cloud, Fog, and Dew Computing

that can be rapidly provisioned and released with minimal management effort or service provider interaction [20].

Cloud Computing (or simply Cloud) has become really pervasive in a few years, so much so that it has been adopted for the delivery of several commercial services. The possibility of sharing datacenter resources has enabled a significant improvement in energy efficiency. The overcoming of hardware-based virtualization technologies has allowed performance isolation and elasticity on the part of customers to scale up and down the resources required (actually realizing the pay-as-you-go model). Managing these complex infrastructures has been made much easier thanks to powerful management frameworks, improving flexibility. In addition to the technical benefits, several economical benefits are also provided by the Cloud, which include lower CAPEX and OPEX with respect to installation on premise. These are just some of the causes and effects of the rise of Cloud Computing.

In the context of Cloud Robotics, Cloud Computing can be seen as the virtually unlimited brain robots can attach to, offloading all computational and memory intensive tasks as well as satisfying their large storage needs. Therefore, Cloud can provide a big boost in the Robotics field, as a possible solution to several problems that constituted a barrier to further development in the field.

B. Fog Computing

With the growth of IoT technologies, many applications with new requirements have emerged, and a new platform is needed to meet them. The Fog Computing paradigm (or simply Fog) is a distributed computing paradigm that provides data and services closer to end-users. It extends the Cloud paradigm to the edge of network, principally to make latency lower and more predictable. The key concept is the vicinity between the calculation nodes and the end-user applications (e.g. in the border routers or switches). For this reason fog computing is at an intermediate level in the hierarchy shown in Fig. 2. There are many types of applications that fit well with this new paradigm (e.g. latency-sensitive applications, large-scale distributed control systems, and geo-distributed applications) but there are still many others that work better with Cloud Computing paradigm. The Cloud is irreplaceable for economies of scale, energy efficiency, and flexibility of pay-as-you-go model. Fog Computing complements the Cloud, not replaces it, but interplays with it.

In the context of robotics, the Fog Computing paradigm is important to deploy the decision-making processes that require more reliability closer to the robots, to avoid that a condition of overload of the network or data centers prevents them from accomplish a task in time.

C. Dew Computing

The Dew Computing paradigm further expands the distribution of resources already seen with Fog Computing, and it is at the lower level in the hierarchy shown in Fig. 2. Dew computing is based on the concept of microservices, which are provided by the end-user devices (e.g. laptops, mobile devices, smart things, or robots) without the aid of centralized virtual nodes. The end-user devices, also defined on-site devices because they are geographically distributed, in the Cloud Computing paradigm simply run completely-online applications that use the Cloud services. In the Dew Computing paradigm, some features and data are moved (or replicated) on the on-site devices, fully realizing the potential of distributed devices and cloud services.

The main goal of Dew Computing is to improve scalability: the processing tasks are extremely distributed over a large number of devices, which are heterogeneous, ad-,hoc programmable, and self-adaptive. In this way, it is possible to realize highly distributed applications without the use of central nodes.

The Dew paradigm can coexist with the Cloud and the Fog paradigms: the on-site devices are able to collaborate with central computing nodes when the scenario allows it and when the Internet connection is available, but they are not dependent on them. The on-site devices are always able to provide a set of features, even without Cloud and Fog services, and without internet connection.

In the following we discuss how Dew Computing paradigm can be applied in the context of robotics.

IV. Dew Robotics

In Sec. III we have described the main architectures currently proposed for distributed computing. In this section we discuss their application to the robotics field and propose the use of Dew Computing for current and future robotic applications.

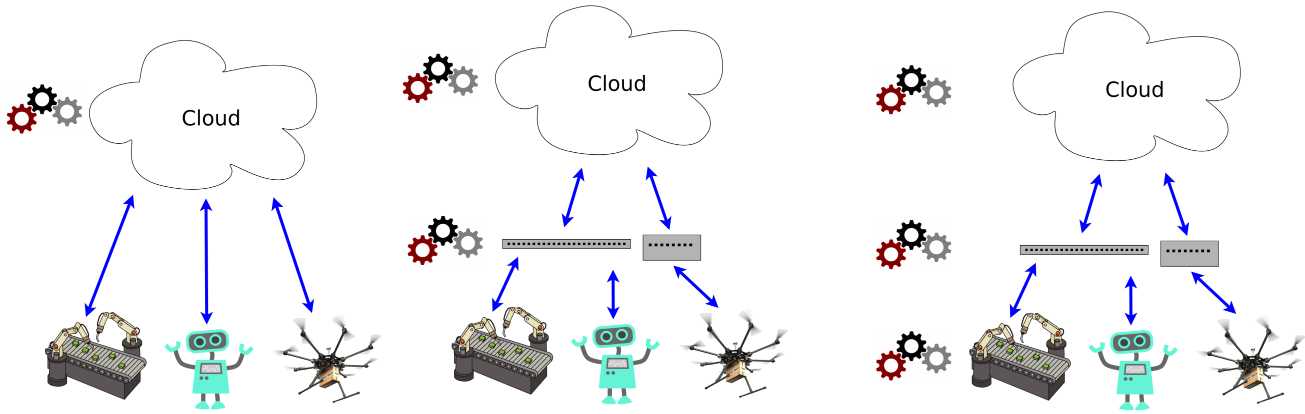

Fig. 3. Cloud (left), Fog (center) and Dew (right) Robotics

Fig. 3 shows the three alternatives we have for introducing the distributed computation in the field of robotics. The left part of Fig. 3 shows the Cloud Robotics one. In this case, the computation is completely offloaded to the Cloud. This has actually been the first proposal in literature to overcome the isolation of robots originally conceived. In this case, the robot can use the large amount of computational and storage capacity available in the data centers that provide the Cloud. This approach has several benefits, and some problems, mainly related to the latency and losses introduced and the necessity to have an always available, performing, and reliable Internet connection.

Fog Robotics is reported in the middle part of Fig. 3. In this case, we still have benefits because external nodes can have more computational and storage power with respect to the robot. Moreover, they can be attached to a power line, where the energy consumption is not a limiting factor. However, in Fog Robotics, there is still need to have a fast and reliable connection, at least up to the Fog nodes. Therefore, latency and network interruptions can still play a bad role and prevent the robots from behaving properly in situation of scarce (local) network connectivity.

Dew Robotics (reported in the right part of Fig. 3) seems naturally the best compromise. In this case, the computation and storage can be split in three parts: locally on the robots, on the Fog nodes, and in the Cloud. For example, critical computations needed in the presence of other human of living beings, can be kept locally so that the robot can always react properly and avoid damages. Less and less critical tasks can be moved to the Fog and Cloud, so to exploit their larger availability of computing, storage, and power supply. We therefore propose the use of this architecture for current and future robotics applications, and we believe that a common view on this important research problem is fundamental.

In Sec. VI, we draw conclusions and provide insights into possible research problems that have to be solved for the adoption of Dew Robotics. Before that, in Sec. V we report our experiences in this field with reference to three interesting use cases.

V. Use cases

A. *Self-driving car*

A self-driving car is a robotic vehicle designed to travel to a destination without any human operator. To be qualified as completely autonomous, the vehicle must be able to navigate without human intervention along roads without any special adaptation for its use. These technologies would eliminate human driver error, which is currently the main cause of accidents. In addition, using a collaborative approach, vehicles can be better distributed along the paths, minimizing traffic jams. Finally, these systems would allow people to do other things during the the journey, such as working, reading, or sleeping.
The Dew Robotics paradigm is very useful in this use case: for SLAM operations, the vehicle can reuse resources stored in the cloud to build the map of the surrounding environment and to calculate the shortest path. Some actions, however, such as avoiding an imminent obstacle (e.g. a pedestrian) require a sudden action, and the Cloud may not be able to process the decision on time, due to latency. In these scenarios the Dew paradigm may be much more appropriate. By processing input data from obstacle detection sensors on board, the decision of a sudden braking or steering can be always quick, even when the network is not working properly. This reduces the risk of causing personal injury or damage to property.

In this application domain, we designed a simple robotic vehicle, controlled with a voice control system through an android app. The vehicle is able to orient itself, avoid various obstacles and make decisions through visual feedback. Once the destination has been set with a vocal command, as shown in the Fig. 4 the robot is able to reach it, looking at the road junctions through the camera and recognizing the text of road directions. The user can see what the camera is shooting through the same app.

The Dew Robotics is implemented in the following way. The mobile application running on the mobile phone of the user records the voice command and sends it to a Cloud service for speech-to-text translation. The output is then provided to another Cloud service for semantic analysis so to recognize the intent of the user and extract precise commands to be sent to the car (e.g. “go to Naples”). Such commands are sent from the mobile application to the car. The car runs a local process to execute the command. Such process constantly sends the video stream to the Cloud for recognizing exits from parking lots, junctions, traffic lights, and the final destination. At the same time, such process constantly monitors a few (e.g. proximity) sensors to fire an alarm in case of risk of collision. If this last situation does not occur, the car uses the feedback from the Cloud (e.g. a sign on the junction is detected, the Cloud service recognizes the text on it and discovers that there is the intended destination, it then recognizes the shape next to such text and uncovers that a right turn is necessary) to move towards the destination. When a proximity alarm fires, the car immediately stops following the instructions from the Cloud and executes a local algorithm to avoid the obstacle. Afterwards, it can safely restart the destination-following algorithm.

Fig. 4. Robot behavior at a crossroads

B. Companion robot

The ability to access potentially infinite information in the Cloud, allows the creation of very sophisticated companion robots. Systems that are able to understand speech and process an answer, using resources in the Cloud, are already part of everyday life (Apple Siri, Microsoft Cortana, etc.). A companion robot, also defined as social robot, is a system with a physical form that extends these functions, also taking decisions autonomously. Once again, there are some important actions to be performed with the Dew Robotics paradigm, because they must always be available even with no connection. For example, the recognition of a dangerous situation, such as a fire or a person’s indisposition, must surely lead to the decision to act quickly. All conversation features, on the other hand, can reside in the Cloud.

We have created a simple companion robot that listens to user requests and reproduces the human social behavior. The robot records the audio tracks of the requests, uses a speech-to-text system, and then analyzes the text to understand the user intentions. We tested some features, such as playing a requested song, providing information about the weather, or telling a joke (main features are summarized in Fig. 6). In addition, the robot stores images of faces of people already seen, so that it can recognize people previously met (Fig. 5 shows an example of how it works).

The Dew Robotics framework allows this robot to provide a few services also in absence of connectivity to the Cloud. The robot makes a cache of lastly retrieved information (e.g. weather forecasts) and uses the cache in case the Cloud is not currently reachable. This is important because a companion robot has to be perceive as much human as possible to be accepted by humans. And a message such as “Internet connection not working” would strongly limit this aspect.

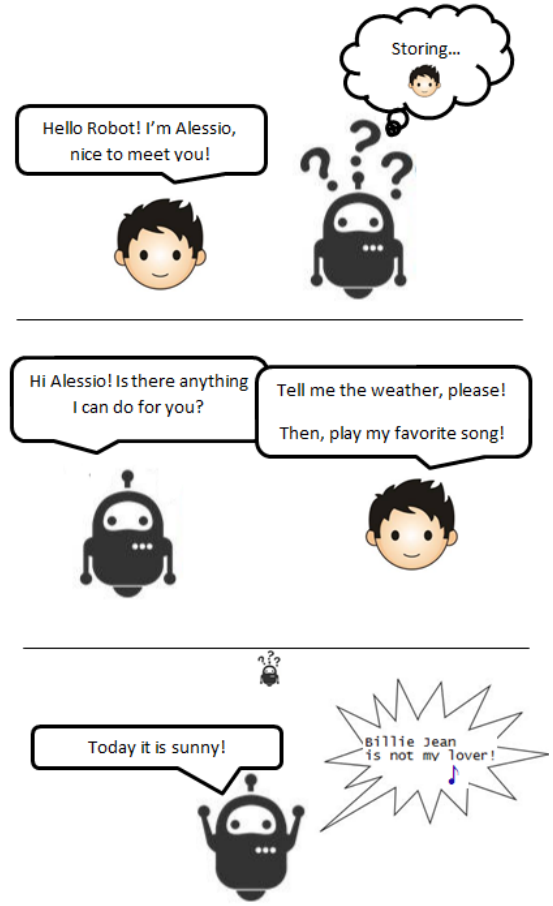

Fig. 5. Companion robot use case

C. Robots for human rescue in hostile environments

In this use case, we have worked on a fleet of robots that perform search and rescue operations in moutain scenarios (e.g. in case of avalances). In such environments, e.g. the Alps, network performance and reliability can be very scarce due to adverse terrain, remote areas, and weather conditions. Dew Robotics is therefore strongly necessary. The SHERPA project [22] addressed the problem of surveillance and rescuing in unfriendly and hazardous environments, like the ones usually operated by civil protection, alpine rescuers, and forest

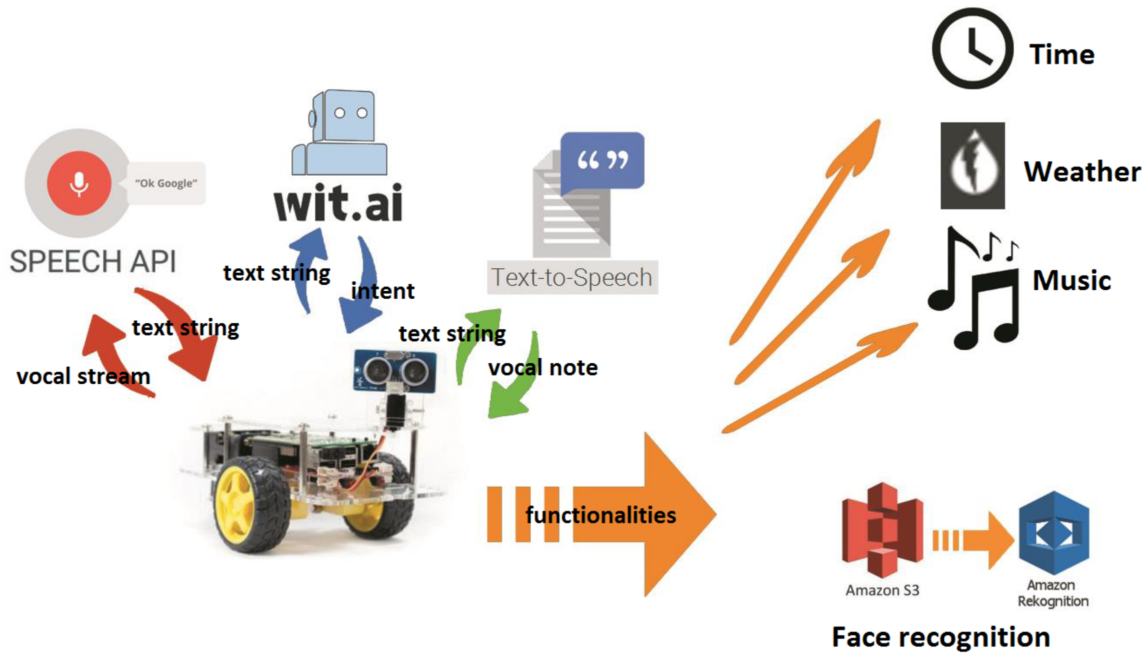

Fig. 6. Robot functions

guards. Within this context, the goal of SHERPA was to develop a robotic platform supporting the rescuers in their work and improving their ability to intervene promptly. Human operators could collaborate with a heterogeneous robotic system to find and rescue survivor victims after natural accidents, such as avalanches. In the context of SHERPA project there is a robotic team, mainly composed of a ground rover and several unmanned aerial vehicles (UAVs) with different characteristics and equipped with different types of sensors in order to retrieve information from the rescue scene and assist the rescuer during a mission. A fleet of many robotic teams might operate in parallel towards the achievement of a common task, like searching a missing person or dangerous situations.

In order to exchange information all the team members must be connected to a reliable network, the topology of which is shown in the Fig. 7. The processing of the data coming from the sensors (e.g. for the analysis of high-resolution videos captured by the HD cameras), can be very expensive in terms of energy and computational resources and the project originally imagined to make such computations on a more static component of the team (i.e. the ground rover reported in the top part of Fig. 7). The use of the Cloud to offload these heavy tasks has then been proposed [22] as an important extension for this system. Choosing the most suited type of network is crucial to meet the stringent network requirements of this scenario, in terms of reliability, coverage and latency. The network technology that best suit the project requirement is the satellite Internet. The ground rover deploys a local network to communicate with the rest of the team (e.g. WiFi) and the uses satellite Internet to reach the Cloud, behaving like a proxy for the other robots.

Dew Robotics is even more useful in this context. Robots can exploit the Cloud for video processing, but use on board computing equipment for following a flying path and record the video while flying in areas not covered by the network connection provided by the ground rover. This allows to enhance much more the coverage of the search area and the performance of the overall search and rescue system.

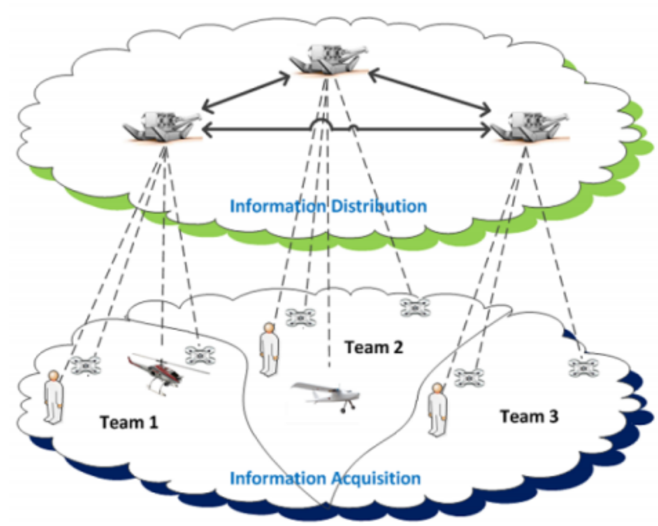

Fig. 7. Topology of the internal Sherpa network

VI. Conclusion

In this paper we have provided an architectural proposal for current and new generation robotics applications. We have proposed the use of Dew Computing and the term Dew Robotics for this aim. We believe this architecture allows to satisfy the requirements of modern robots, exploiting Cloud and Fog infrastructures when possible, and relying on local computation for important tasks that cannot be offloaded or may be offloaded but with a decrease of the overall system performance. We have also described three use cases we have realized to show the real benefits of Dew Robotics in real application scenarios.

We believe Dew Robotics can be an important step to follow. However, some research questions have to be answered for its application. It is necessary to understand how to differentiate the tasks that have to be performed at each of the layers of the Cloud, Fog, Dew computing architecture. The progressive offloading strategy has to take into account factors such as the volume of data to be exchanged, the delay deadline to complete the task, etc.. The decision has to also take into account whether it is more advantageous to execute the task further or closer to the robots. An important issue is also related to a common software platform to experiment with Dew Robotics. We believe that ROS has very interesting features and may be a suitable candidate to be enhanced to support Dew Robotics. Solving these research issues is an important step to make Dew Robotics a reality.

References

[1] "Google Self-Driving Car Project", [Online]. Available: https://www.google.com/selfdrivingcar/

[2] "Million Object Challenge", [Online], Available: http://h2r.cs.brown.edu/million-object-challenge/

[3] "Robot operating system", [Online], Available: http://wiki.ros.org/ROS/Introduction

[4] "Google and Johnson & Johnson Conjugate to Create Verb Surgical, Promise Fancy Medical Robots", [Online] Available: http://spectrum.ieee.org/automaton/robotics/medical-robots/google-verily-johnson-johnson-verb-surgical-medical-robots

[5] "Robots With Their Heads in the Clouds: The five elements of cloud robotics", [Online], Available: https://medium.com/aspen-ideas/robots-with-their-heads-in-the-clouds-88ac44def8a#.51rxoe48u

[6] 3GPP, "Spec 22.261", Available: http://www.3gpp.org/ftp//Specs/archive/22_series/22.261/

[7] ITU-R, "IMT Vision - Framework and overall objectives of the future development of IMT for 2020 and beyond", Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083-0-201509-I!!PDF-E.pdf

[8] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine", in Proc. Int. Conf. Robot. Autom. (ICRA), Karlsruhe, Germany, 2013, pp. 42634270

[9] K.M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment", Proc. Of Int. Conf. Intelligent Robots and Systems, 2008.

[10] Dieter Fox, "Distributed multi-robot exploration and mapping", In CRV '05: Proceedings of the 2nd Canadian conference on Computer and Robot Vision, Washington, DC, USA, 2005. IEEE Computer Society.

[11] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider, "Coordinated multi-robot exploration", IEEE Transactions on Robotics , 21(3):376-386, 2005.

[12] Mark Harris, "The Heaphy Project: Crowdsourced Robot Servants and the Willow Garage Spin-off That Never Was", IEEE Specrum, March 2015, [Online], Available: https://spectrum.ieee.org/automaton/robotics/robotics-software/the-heaphy-project

[13] G. Hu, W. P. Tay and Y. Wen, "Cloud robotics: architecture, challenges and applications", in IEEE Network, vol. 26, no. 3, pp. 21-28, May-June 2012.

[14] R. Arumugam et al., "DAvinCi: A cloud computing framework for service robots," 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, 2010, pp. 3084-3089.

[15] B. Kehoe, S. Patil, P. Abbeel and K. Goldberg, "A Survey of Research on Cloud Robotics and Automation", in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2, pp. 398-409, April 2015.

[16] "What is RoboEarth?" [Online]. Available: http://www.roboearth.org

[17] J. P. de A. Barbosa et al., "ROS, Android and cloud robotics: How to make a powerful low cost robot", 2015 International Conference on Advanced Robotics (ICAR), Istanbul, 2015, pp. 158-163

[18] Damon Kohler, Ryan Hickman, Ken Conley, Brian Gerkey, "Cloud Robotics", Presentation at Google IO 2011, [Online], Available: https://www.youtube.com/watch?v=FxXBUp-4800

[19] Wan, J., Tang, S., Yan, H., Li, D., Wang, S., Vasilakos, A. V., "Cloud robotics: Current status and open issues", IEEE Access, 4, 2797-2807, 2016.

[20] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, 2011. ⟨http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf⟩

[21] Mohanarajah, G., Hunziker, D., D'Andrea, R., Waibel, M. "Rapyuta: A cloud robotics platform" IEEE Transactions on Automation Science and Engineering, 2015.

[22] Alessio Botta, Jonathan Cacace, Vincenzo Lippiello, Bruno Siciliano, Giorgio Ventre, "Networking for Cloud Robotics: a case study based on the Sherpa Project", International Conference on Cloud and Robotics ICCR2017, November 22-23, 2017, Saint Quentin, France.