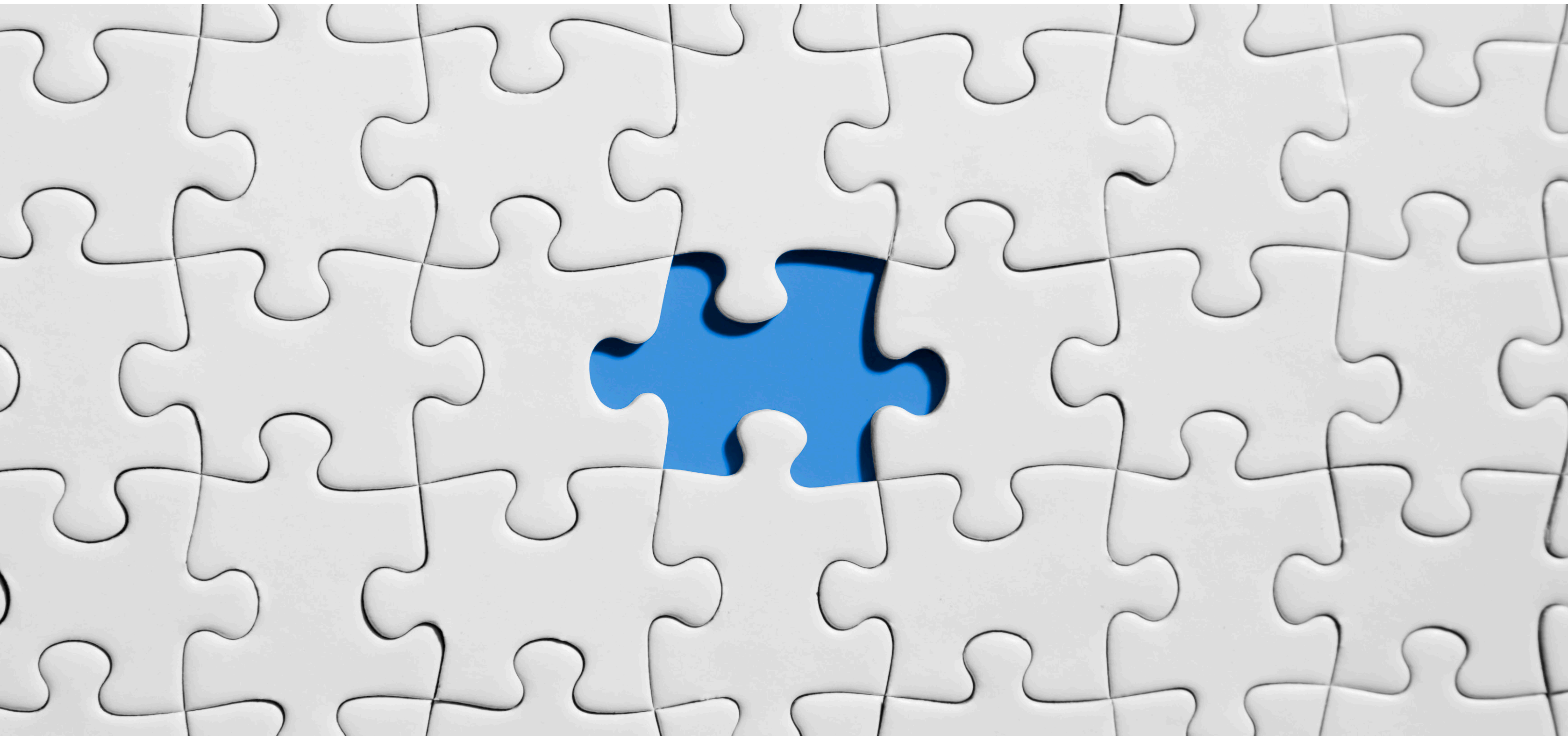
FLUMMOX AROUND OSMOTIC COMPUTING WITH DELL TECHNOLOGIES



Shadaab Mohiuddin
Principal Systems Development Engineer
Dell Technologies
Shadaab_mohiuddin@dell.com

Sreedevi Katabattuni
Senior Systems Development Engineer
Dell Technologies
Sreedevi_Katabattuni@Dell.com

Tabraiz Amman
Principal Systems Development Engineer
Dell Technologies
Tabraiz_amman@dell.com

Table of Contents

I. ABSTRACT

There is the edge, where data is processed roughly where it is actually generated; then there are data centers, where clusters of server store, distribute, and process data. Somewhere in the middle sit low-overhead containers, which support better edge/data center data integration, where microservices can be offloaded to/from the edge/data center, without scheduling delays or extensive storage or memory requirements. Welcome to the world of ‘osmotic computing.’

The phrase ‘osmotic computing’ is derived from the word ‘osmosis’, which means the process of gradual or unconscious assimilation of ideas, knowledge, etc. What makes osmotic computing ‘osmotic’, is that developers don’t actually have to worry about service migrations from the edge/fog/core/cloud – the transfer can happen in the background, without active human intervention.

What osmotic computing is can be hard to define. No one owns the term, so everyone uses it in different ways. However, everyone from service producers to consumers agree on one thing - there would be many functional services running code on computing nodes somewhere. What makes something “osmotic" is that developers don’t have to worry about those services migrations from edge, fog to core or in cloud, for the availability and its use by service consumers. That’s why osmotic computing is more than just a technology paradigm. It’s also a movement.

Barriers to entry are never more apparent than when a new technology comes along and knocks them down. Osmotic computing is doing just that. Osmotic computing enables DevOps teams to build new software services by using polyglot programs for latency-aware, data-intensive services hosted on smart computing environments. Think about how functional services data might have to flow through various Global Sensor Networks (GSN), actuators, gateways, edge servers, Network Function Virtualization (NFV), cloudlets, Micro Data Centers (MDC) and cloud environments based on required latency and allocation of optimized compute, network, storage provisioned by Software-Defined Virtual Infrastructure. Sorting out permissions and securing data from various public/private partnership hosted fog/cloud functional services, might mean bringing your Ops team into the planning stages of application development or refactoring, instead of just tossing the finished package over the fence for them to deploy.

In short, implementing Osmotic Computing would simplify dynamic management of functional services or microservices across an edge/fog/cloud-based architecture, built and maintained by a DevOps team. Put another way: Osmotic computing means adopting all those cutting-edge “best practices” that you’ve been reading about for the past decade or more. And there are plenty of organizations that are already leading the way. For this paper, we will walk you through what Dell Technologies offers.

Pivotal and Dell Boomi are pioneers in enterprise innovation, assisting companies across the globe to improve efficiencies and, in the process, become both more agile and gain significant business advantages over the competition with measurable savings and ROI.

Osmotic computing-enabled migration of “functional services” brings emerging heterogeneous communication technologies such as fog, or edge computing together to substantially reduce latency and energy consumption of Internet of Everything (IoE) applicable services running on various platforms.

The key feature that distinguishes osmotic computing from existing cloud computing paradigms is that it spreads communication and computing resources over the wired/wireless access network (e.g. location proximity access points and spectrum for WiFi and LiFi base-stations) to provide augmented resources (i.e. for cyber foraging) for resource- and energy-limited wired/wireless (mostly mobile) things.

Motivated by these considerations, this paper presents osmotic computing as the promise of providing potentially better services on unreliable computing nodes. Devices such as mobiles, smart watches and various other edge devices which are power and resource constrained coupled with limited scalability and unreliable network availability options would need to interact with services hosted on cloud computing (especially Infrastructure-as-a-service [IaaS]). In the Internet of Things (IoT) world, osmotic computing solutions can improve quality of service (QoS), fostering new business opportunities in multiple domains, such as healthcare, finance, traffic management, and disaster management.

II. INTRODUCTION

Recent advancement in technology is disrupting legacy-centralized computing or client-server models, moving cloud resources closer to users, or even the edge. The unprecedented deployment of mobile and sensor devices has created a new environment, namely the Internet of Everything (IoE) that enables a wide range of future Internet applications.

Rapid growth in usage of Internet-enabled mobile and sensor devices has led to creation of a wide range of novel, large-scale, latency-sensitive applications (often classified as ‘Future Internet’ applications). Smartwatches can analyze heart rate data and identify an atrial fibrillation anomaly in the heart or detect if the wearer has fallen. There are AI-based applications sitting on wearable devices that can go a step further and predict if a person is having a heart attack or an opiate overdose. Based on such actionable insights, the app can subsequently alert and involve real-time decision support or proximity systems to assist in emergencies. Smart surveillance applications can support law enforcement officers by displaying video streams on their smartphones or tablet devices for camera feeds.

At the network edge, innovation in new and existing data transfer protocols and maturity of heterogeneous resources has enabled edge based resources to seamlessly interact with datacenter-based services or cloud-hosted services in public/private, hybrid or multi-cloud. Osmotic computing is a new paradigm that supports this and enables automatic deployment of lightweight microservices on resource-constrained edge devices as well as datacenter based services. Especially when considering latency-aware, real-time data-intensive application services, which aim at highly distributed and federated environments and enable automatic deployment of micro-services that are composed and interconnected over edge/fog, core and cloud infrastructures.

From the memories of our chemistry classes in high school, the word “osmosis” represents the seamless diffusion of molecules from a higher concentration to a lower concentration solution. The process of enabling services to be migrated across on-premises and cloud data centers to the network edges represents osmotic computing in distributed systems. It helps in dynamic management of microservices, addressing issues related to deployment, networking, and security, thus providing reliable IoE support with specified levels of QoS. The elasticity-related challenges in cloud data centers remain unchanged in osmotic computing. In reality it multiplies with several features due to the heterogeneous nature of edge/ (proximity aware) micro-cloud data centers and traditional cloud data centers. In addition, various stakeholders (cloud providers, edge providers, application providers, and so on) are supposed to contribute to provisioning IoE service and applications in a federated environment.

III. MOTIVATION

The falling costs of the available heterogeneous IoE devices, inbuilt sensors from smartphones and inclusion of home automation devices like Alexa and Google, as well as wearable devices, has a tremendous impact on our lives, fueling the development of critical next-generation services. Efficiently capturing, analyzing and acting on data produced from IoE devices is still an unresolved challenge because relying on large-scale cloud computing systems is a bottleneck that is yet to be solved by new innovation at fog and edge computing. However, a key limitation of current cloud-centric IoE programming models (such as Amazon IoT and Google Cloud Dataflow) is its inherited inability to transfer large data tuples to such centralized cloud data center environments, especially in latency-aware mission critical systems, or with enhanced privacy mode requirements. Currently, most IoE programming models fall short and are inefficient in the context of osmotic computing.

On one hand, Platform-as-a-Service (PaaS) offerings similar to those offered by Pivotal Software are attractive for developing privacy critical, context aware-sensitive large-scale mission-critical applications with pre-requisite scalability and high-level programming design models that simplify developing large-scale web services.

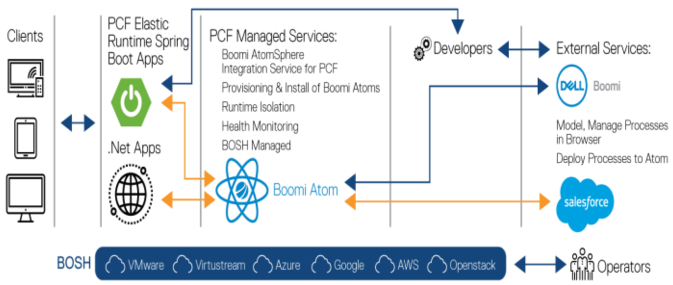


Fig. 1. **Dell Boomi as an Integration Layer empowered by Pivotal for Smart Application (Cloud-Native)**

However, most existing PaaS programming models are designed for traditional cloud-native applications, rather than Future Internet applications running on various mobile and sensor devices. Moreover, public clouds are far from the idealized utility computing model. Applications are developed on a vendor-agnostic and custom-designed infrastructure from the provider's platform and run in data centers that exist at singular points in space. This directly impacts highly latency-sensitive applications utilized by many users from networks that are too far away in destination points.

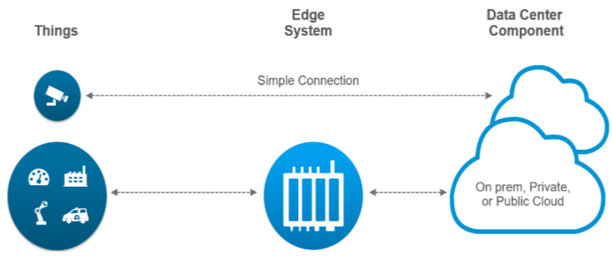


Fig. 2. **Edge and cloud computing for the Internet of Things (Federated Environment)**

Hence, to implement complex IoE-oriented computing systems, both cloud and edge resources must be exploited while setting up a hybrid infrastructure, as Figure 2. shows. Cloud and edge data centers will be managed in a federated environment, where participating providers share their computing, network and storage resources for IoE services and application support. The exercise of data upload moving towards data centers leads to inefficient use of communication bandwidth and energy consumption. A recent study by Cisco(https://newsroom.cisco.com/press-release-content?articleId=1804748) shows cloud traffic is expected to rise 3.7-fold, up from 3.9 zettabytes (ZB) per year in 2015 to 14.1 ZB per year by 2020, worsening the situation further.

The Open Fog consortium defines Fog computing as, “A horizontal system level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum,”. Allowing computing functions to be distributed between different platforms and industries in fog computing is provided by the “horizontal” platform. Unlike horizontal, while a vertical platform may provide strong support for a single type of application (silo), it does not account for platform to-platform interaction in other vertically focused platforms. A vertical platform promotes siloed applications. Hoard-and-Forward-later approaches, which can save network bandwidth, undermine real-time decision making, which is often a necessary requirement behind IoE applications in the domains of critical support management and healthcare. On the other hand, edge/fog computing proposes to lay computing needs on resource-constrained edge devices, as Figure 3 shows. Usually edge/fog applications are highly time-sensitive (for example, tactical warning applications for natural disaster management for weather conditions such as storms) and are required to act immediately on analysis, or response to accumulated sensing data. Nevertheless, even if cloud-based programming models can’t support the desired degree of sensitivity for IoE applications, based on necessity they can strongly increase computation and storage availability. This usually occurs and follows the most prevailing cloud-centric IoE programming models provoking them to be revised into something that’s more adaptable and decentralized to meet the needs of emerging IoE applications.

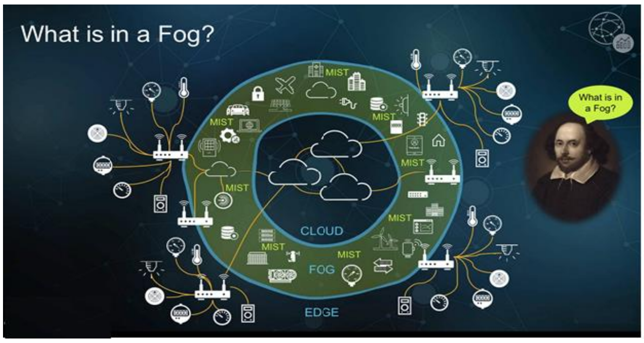


Fig. 3. **Osmotic computing in Edge, Fog and Edge datacenters**

IV. OSMOTIC COMPUTING

Decomposing applications into micro-services, and dynamic allocation and routing micro-services in smart environments exploiting resources in edge, fog and cloud infrastructures is the fundamental principle of osmotic computing. **A crucial component of achieving this, is dependent on network protocols**. Over the past few decades, many network protocols have been introduced to manage WSN (Wireless Sensing Networks). Most of these protocols are classified into three categories based on routing table availability, i.e. proactive, reactive and hybrid protocols. Proactive routing is based on the fact that all the routes from the source to destination will be added to the routing table in advance where all nodes will be periodically exchanging information about the network topology. Examples of proactive protocols are: Fisheye State Routing (FSR) Optimized Link State Routing (OLSR), and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Meanwhile, Reactive protocol uses on-demand routing, i.e. route identification is done based on the need to forward a packet from source A to destination B. Examples of reactive protocols are: Ad-hoc On-Demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Temporary-Ordered Routing Algorithm (TORA). The third type – hybrid protocol – is a mix of both, with the aim of reducing overhead of the proactive protocol and speed up route discovery in the reactive protocol. Zone Routing Protocol (ZRP) and Hybrid Ad Hoc Routing Protocol (HARP) are examples of hybrid protocols. The efficiency and effectiveness of each routing protocol is out-of-scope for this paper.

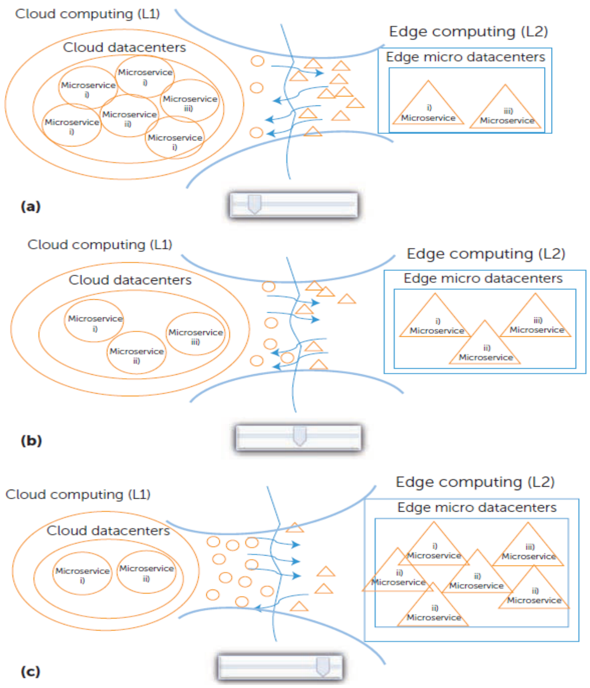
Stakeholders of the osmotic computing solution design keep getting flashbacks of the three-layer density experiment learned in high school.

What happens when we pour honey, water, and oil in a jar?

The liquids appear to have mixed when we stir the mixture but settles into three discrete layers when left alone. The oil floats on the water and honey sinks to the bottom.

Why does this happen?

This happens because each of the three liquids has different densities. Honey which has the maximum density settles at the bottom whereas the oil which has the least density floats on the water. Like the above experiment, data computing density behavior when applied in functional services can help maintain the balance of dynamic management of applications running inside containers. This model helps balanced deployment of services to meet the latency sensitive requirements at the edge/fog or satisfy high level needs in the cloud.



PROF. MASSIMO VILLARI HEAD OF FUTURE COMPUTING RES LAB (FCRLAB) UNIVERSITÀ DI MESSINA. Osmotic computing in cloud and edge datacenters: (a) movement of microservices from edge to cloud, (b) optimal balance of microservices across the edge and the cloud, and (c) movement of microservices from cloud to the edge.

By applying methods such as Exploration versus Exploitation, the decision to match the computing requirements with regard to specific workloads for the deployed functional services is based on data weight and time-sensitivity. One of the key distinctions of osmotic computing, unlike the chemical osmotic process osmotic computing allows a tunable configuration of the resource involvement factoring resource availability and application requirements (see Figure 4 below). Additionally, available resource battery power on the edge computing node in context of movable wireless devices should also be taken into consideration for opportunistic computing. The difference in configuration (very much application and infrastructure dependent) can determine whether microservices should migrate from edge/fog to cloud or vice versa.

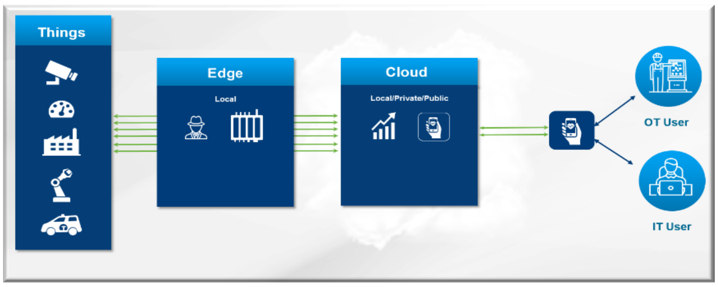


Fig. 4. **The Exploration Vs Exploitation techniques for Edge/Fog and Cloud (micro services for users)**

In addition to simple elastic management of deployed resources, the osmotic computing environment must consider deployment strategies related to requirements of both applications (such as sensing/actuation capabilities, context-awareness, proximity, and QoS) and infrastructure (such as load balancing, reliability, and availability) as they change over time. Due to the high heterogeneity of physical resources, any microservice deployments need to be in sync with the virtual environment for available hardware. Hence, a bidirectional flow of adapted microservices from edge to cloud (and vice versa) must be managed. Moreover, the migration of microservices in the edge/cloud system propels the requirement for efficient management of application breakdowns and proactive avoidance of degradation in QoS because of the virtual network and infrastructure required for Future Internet applications.

For monitoring and management purpose both the users (IT, OT) use same interface. The service is forwarded based on availability to run in local or cloud server to provide results for IT and OT users alike.

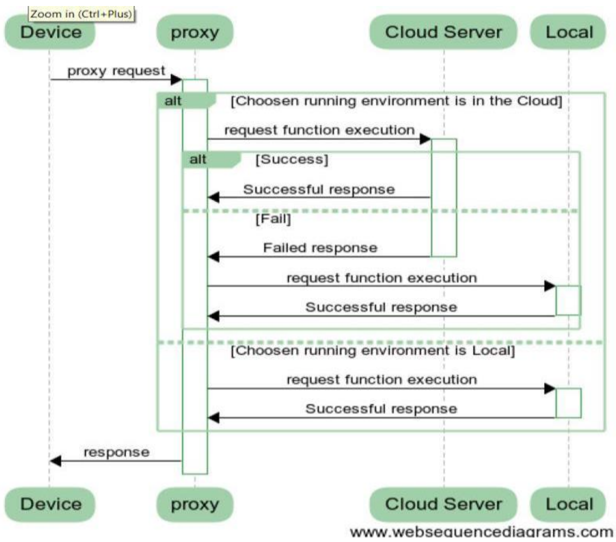


Fig. 5. **Open Faas – Ref [16] Dynamic allocation of Serverless Functions in IoT Environment : Request Function Execution**

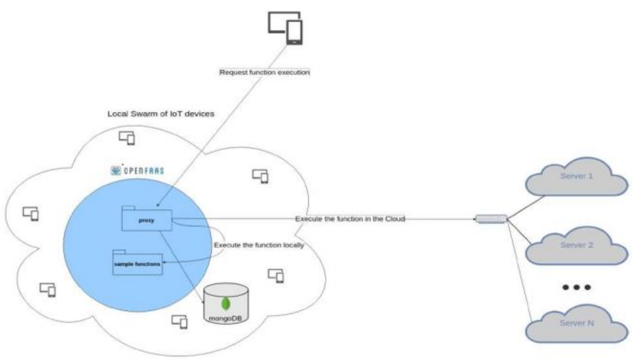


Fig. 6. **Open Faas – Ref [16] Dynamic allocation of Serverless Functions in IoT Environment - Interaction of the different components**

For instance, to store the function service metrics OpenFaas uses a MongoDB database. Beyond the local network of devices, on the cloud, the servers contain an exact copy of the osmotic functions, whose execution could be requested remotely.

Dell Technologies has open-sourced its internal project OpenFuse, which is now known as EdgeXFoundry and released it under Linux Foundation’s open source industrial IoT group license backed by Intel and other partners. In our scenario it would provide enterprise-scale interoperability to implement the same for the dynamic allocation of server-less functions in a dynamic environment with heterogenous protocols, devices and integration of edge, core and cloud IoT solutions.

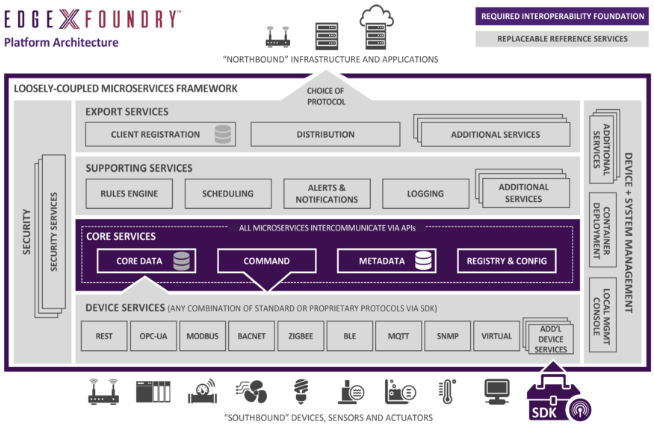


Fig. 7. **[16] EDGEXFOUNDARY Architecture**

The breakthrough approaches of Exploration versus Exploitation methodologies do not address the issues of decoupling the management of user data and applications from the management of networking and security services. Osmotic computing moves forward in this direction, offering an automatic and secure microservices deployment solution by providing a flexible infrastructure. What is included in osmotic computing is to act on the innovative application-agnostic approach by exploiting lightweight container-based virtualization technologies (such as Docker and Kubernetes) for deployment of micro-services in heterogeneous edge and cloud data centers.

V. Ecosystem for Osmotic Computing

There are three main infrastructure layers in osmotic computing; cloud layer (CL), fog Layer (FL) and edge layer (EL). CL consists of cloud data centers, which provide several types of services and micro services. In regard to osmotic computing purposes, micro-services are composed at this layer, based on the users’ high-level requirements. At the FL layer, several other computing environments can overlap with the edge layer (EL), that identifies the fog nodes and edge server computing environment, along with Global Sensor Networks (GSN) aggregators to perform operations (min, max, filtering, aggregation, average, and so on) on local data. This is depicted in Figure 6.

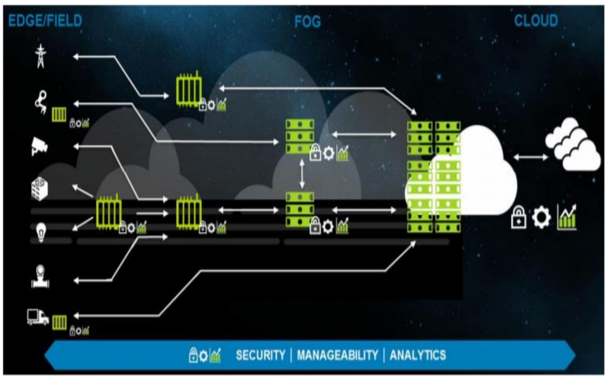


Fig. 8. **Osmotic Computing in Edge/Field, Fog and Cloud Data Centers**

GSN aggregators such as Axeda (www.axeda.com) and Xively (xively.com/whats_xively/) capture data with a predefined frequency (the rate of change is often dictated by phenomenon being observed). The device's capacity to record or collect data is dependent on specific system requirements that are to be satisfied. EL devices can perform different complex operations on the raw data accumulated from the ecosystem, such that incoming data tuples are encrypted, and operational encoding/transcoding is performed before forwarding this data for subsequent analysis to CL or FL, based on availability.

Like academicians, technology industry experts across Dell Technologies envision different types of resources at each of these three layers to be in some distributed heterogeneous clouds, complemented by fog and edge computing environments. A key academic research and industry challenge on distributed systems is to understand how a microservice hosted on a fog node at FL can interact and coordinate with a microservice in CL. The objective functionalities at each level has its own influence on the types of operations performed at the desired layers. For instance, EL generally consists of resource-constrained devices (limited battery power, network range, and so on) and network elements, without overloading the available resources they need to perform the tasks allocated.

Data centers at CL and micro data centers at FL and GSN sensing aggregators at EL would involve participation of multiple providers. To increase business opportunities in a federated scenario with the respective three layers above, providers can establish relationships and cooperate to share resources and services. In an osmotic computing ecosystem, microservices are trifurcated into three categories; generic-purpose, network management, and security management as shown in Figure 7.

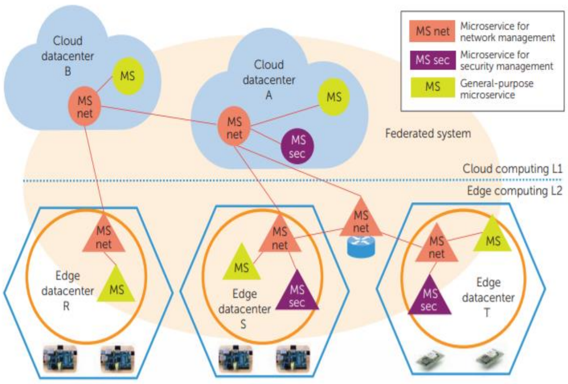


Fig. 9. **A federated cloud environment in cloud computing from Professor: MASSIMO VILLARI HEAD OF FUTURE COMPUTING RES LAB (FCRLAB) UNIVERSITÀ DI MESSINA**

In a general-purpose microservices scenario, the osmotic framework would enable user applications which are related to the specific applicative goal to be orchestrated and deployed in the cloud, fog or edge infrastructure. For setting up virtual networks among microservices deployed in the distributed and federated cloud/edge system, network management microservices are used. Further, to support cross-platform development of security-enabled microservices, security management microservices will be used.

In CL, FL and EL deployment, environments can benefit by running functional services based on microservices aggregated on different types of resources. The perceived knowledge on how these systems are to be aggregated to support application requirements (specifically nonfunctional requirements- latency, throughput, security, and budget) is an important challenge yet to be concluded. We intend to use containerization technology as a key differentiating approach, where microservices are opportunistically deployed in virtual components as an alternative to hypervisor-based approaches (such as Xen and Microsoft Hyper-V) used in the cloud. The lightweight evolution of container-based virtualization technologies (for example zLinux containers, Docker, Preboot Execution Environment, Google containers, and Amazon Compute Cloud containers) are best suited in Future Internet application environments. Compared to hypervisors, which consume higher instance-density on a single device, a container-based technology encapsulates well-defined software components (such as a database server) reducing deployment overhead drastically. In addition, deployment of lightweight microservices on resource-constrained and programmable smart devices on the network edge – i.e. routers (Cisco IOx), network switches (HPOpenFlow), and gateways (Raspberry Pi and Arduino) – on top of increasing performance in the dynamic management of microservices in multi-cloud datacenter scenarios is a supported functionality in container-based virtualization.

To be adaptable to the deployment sites, the composed microservice owners should consider deployment location and their dependencies on the physical and virtual infrastructure capabilities where containers are to be spun-off. Some of these decisions are influenced by constraints identified by the infrastructure provider, unique application requirements owned by different stakeholders and managing mapping decision of the microservices to the relevant location. Further, generating revenue by utilizing the unique workload computing resources (such as a GPU cluster) and improving overhead management (i.e. system administration and/or energy costs) also influence how a decision maker must map microservices to the relevant location. During execution, generic-microservices must also be able to adapt to fluctuations in the computing environment over a period of time. To detect changes in infrastructure performance and QoS metrics a feedback-driven orchestration would be required.

VI. INDUSTRY RESEARCH IN PROGRESS FOR OSMOTIC COMPUTING

We propose the following research directions to make the most effective use of the osmotic computing paradigm required for Future Internet applications. Analytics and deep learning will play a critical role on the three spheres of edge, core and cloud as well as applying machine learning to pull insights from the cloud and push insights from edge analytics.

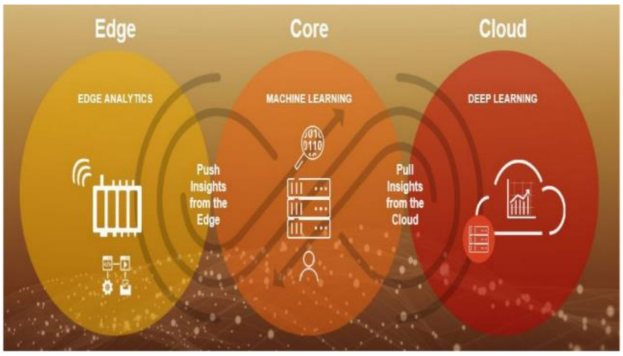


Fig. 10. **The Edge, Core and Cloud Infrastructure and learning methodology for Smart Applications**

Microservices configuration: In order to select resources from the offerings of different technology provider's cloud data centers and be able to deploy virtual machine (VM) images based on performance measurement techniques, there is lack of quality data on categorizing microservices suitable for edge, fog and micro-cloud data centers. Additionally, vendor lock-in evaluated methods are most commonly found in industry-specific vertical solutions. Defining the processes where techniques are used for selecting existing configuration and requirements of VM images have largely been blind-folded during the migration processes. The ideal configuration would require transparent decision support and adaptability for the custom criteria. In addition, to meet the objectives of deployed microservices with selective constraints there are lack of industry standards to model configurations at fog and edge assigned resources. Moreover, considering the configurations and QoS at the network edge, VM deployment on cloud data centers are different from selecting and ranking microservices on edge data center resources. Therefore, it is necessary to automate configuration and selection of resources in cloud and edge data centers to meet QoS constraints. Further, techniques such as multi-criteria decision must be applied on analytic network processes. For instance, on genetic algorithms, multi-criteria optimization techniques are to be applied.

Microservices Networking: Smart applications deployed on heterogeneous infrastructure desire certain performance over communication among microservices based on an abstraction of networks that spawn from cloud, fog to edge and vice versa. To dynamically adjust the overall microservices behavior based on user requirements, network plays a critical role. Moreover, to provide network management abstraction independent of the underlying technology to support in-network/in-transit processing of data (between edge, fog and data center to evaluate and innovate solutions based on software-defined networking (SDN) and network function virtualization (NFV)), there is ongoing research in progress at both commercial industry levels, and by the open-source community.

Fig. 11. **VMware IoT Pulse from Dell Technologies**

The missing piece of the puzzle from the scientific literature community and traditional IT industry in regard to osmotic computing is the characterization of federated networks in the domain of cloud and edge needs for specific metadata ontology. Enabling API-first strategy for network management and a common framework on interoperability of heterogeneous edge devices (ex: improvising NFV capabilities and advancement in SDN) would be the main concern moving forward in the next decade.

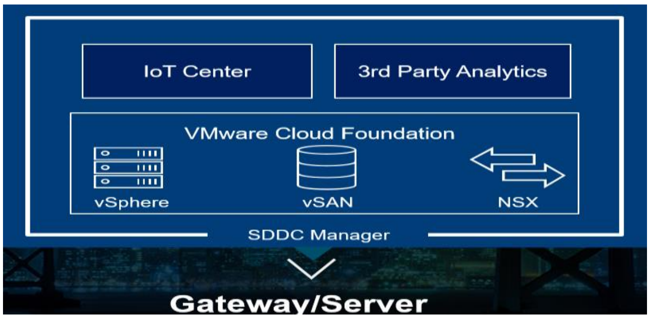


Fig. 12. **Project Fire from Dell Technologies**

Project Fire is an on-going research project from Dell Technologies on a ready-to-deploy, pre-validated and pre-integrated solution that supports the new distributed IoT architecture by enabling analytics at the edge using technologies you use and trust [See Figure 7] above empowered by VMware IoT Pulse.

Microservices Security: The challenges around security accompanied by the threats of integrating edge computing devices and fog nodes (like, edge servers, or IoT devices emitting data in GSN networks) with a cloud data center must be based on execution and migration microservice enablement. To facilitate a wide adoption of osmotic computing technology, deployment of microservices with the desired security features would create self-identification processes ensuring unique observation of challenges inside cloud and edge devices.

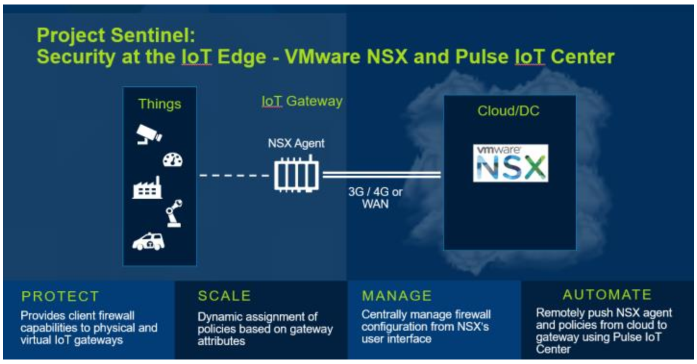


Fig. 13. **Project Sentinel from VMware**

Moreover, one needs to enable the secure deployment of containers including micro-services on IoE devices which is based on security capabilities to the container engine. By means of a transversal security process, which requires creating trust on edge devices and cloud systems, developers must be allowed to build chains on distributed ledger (blockchain) technology which is another parallel objective of osmotic computing. Project Sentinel empowered by Velo-cloud (VMware) and Project IRIS are on-going research projects from Dell Technologies for security at Edge utilizing NFV (Network Function Virtualization) and a Security solution empowered by RSA.

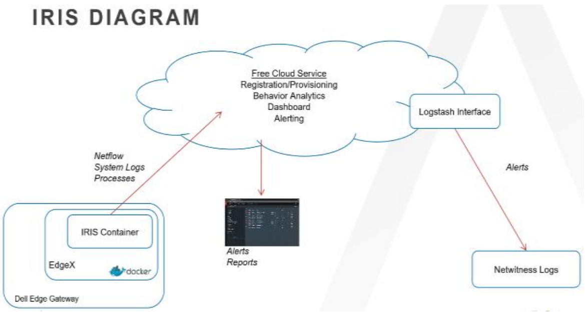


Fig. 14. **Project IRIS from RSA** (https://iris.rsa.com)

VII. EDGE COMPUTING

Smart Applications ranging from intelligent buildings to home automation are introduced to Lifi technology. Unlike Wi-Fi, where data transfer and consumption are based on confined radio spectrum for the IoE devices to communicate to one-another, Lifi opens a new frontier with unlimited bandwidth up to 1 Gbps. Veimenni, a startup from India is one of the few to offer HTTP based REST API commercial solutions on Lifi technology. Xively, Open Sense and Think Speak offer commercial solutions on Wi-Fi technology for edge computing. In addition, there has been growing interest from the academic community and efforts from the European Union sponsored Open IoT project proposal to create an open source “IoTCloud” (providing Sensors-as-a-service) and middleware GSN solutions. This also aligns with the fog computing efforts involving cloudlets (from Cisco) and micro-datacenter (from Microsoft and Amazon) which act as small datacenters at the network edge and are geographically scattered across a network to support scatter-nets for Mobile-Edge Cloud and Mobile-Adhoc Edge/Cloud opportunistic computing use cases.

Fig. 15. **The evolution of IoT Actuators and commercial Gateway from Dell Technologies (2016)**

To reduce delays caused by intermittent network connectivity and battery power consumption on smart application, there is need to offload complex and long-running tasks from mobile devices to cloud-based datacenters. Assigned tasks from mobile devices which generally have lower computation and storage capabilities than a datacenter would need to have periodic synchronization between the edge device and the datacenter. An extensive research approach to achieve the same outcome based on a VM is studied from the prism of Clone Cloud and Moitree(Middleware for Cloud-Assisted Mobile Distributed App) which involves creating a mobile device clone within a data center.

The desire to connect mobile offloading with data center offloading is recommended in osmotic computing. we are able to offload computation initially carried out within a datacenter to a mobile device. This “reverse” offloading enables computation to be undertaken closer to the edge of sensing networks (overcoming latency and data transfer costs). Further, osmotic computing research focuses on understanding the types of microservices that would be more relevant to execute at the edge than within a data center environment, and vice versa. Dell Technologies has evolved from pioneering hardware edge-gateways and embedded

PC's for Industrial IoT to edge resource management Software Defined Solutions from its evolution since 2016.

VIII. MICROSERVICE INTERFERENCE EVALUATION FOR WORKLOAD CONTENTION

Advances in research have propelled digital transformation in the industrial world toward virtualization as an alternative to VMs in the cloud. The cloud-based solutions for IoT and edge devices are presented in most of the academic research activities. The rise in popularity is largely driven by open source systems for management of clusters running containerized microservices enabling automated deployments, operations on cloud data center and on edge devices and fog servers. For instance, Docker Swarm and Kubernetes provide a native orchestration framework (container engine) for multiple Docker deployments. It is critical to decide which microservices can be deployed together by studying the nature of their composition based on the workload contention which leads us toward the deployed, containerized micro-services with respect to consumed resource and QoS, to minimize hosted applications on cloud data centers.

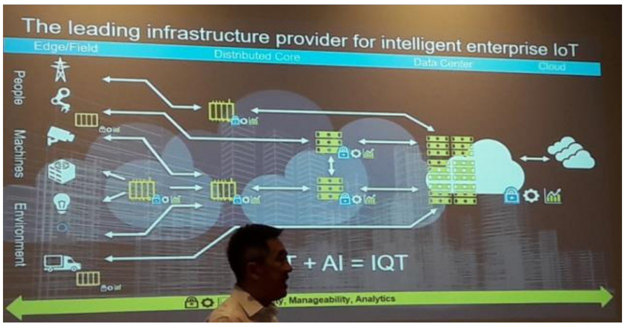


Fig. 16. **One of the leading infrastructure providers for enterprise IoT**

It gets difficult to manage workload contention over time with hardware-based approaches as they add complexity to the processor architecture. A scheme to quantify the effects of cache contention between consolidated workloads was developed by Sriram Govindan [14] and his colleagues. Although, the contention issues of other resource type were ignored, their techniques focused only one hardware resource type (that is, cache). A control theory-based approach for coexisting workloads which require consolidation of cache, memory, and hardware prefetching contention while managing effects was proposed by Mohammad Nathuji and his colleagues for CPU-bound or compute-intensive applications.

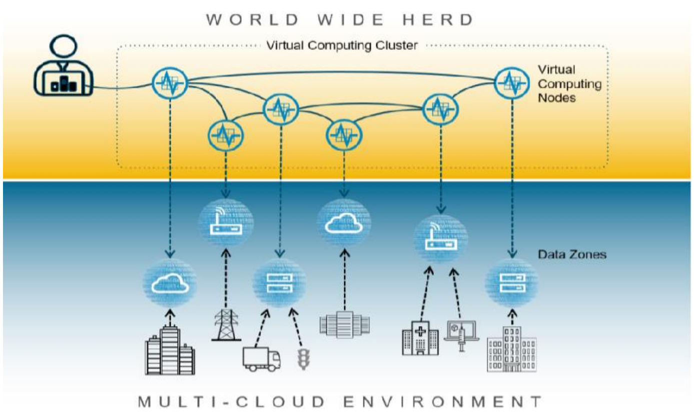


Fig. 17. **Project World Wide Heard (Solving Federated Analytics)**

Though there has been adoption of technologies such as Open-Shift Origin, Amazon EC2 Container Service, Docker Swarm, and Kubernetes for orchestrating and management of containerized workloads, there is still a need to automatically detect and handle resource contentions among deployed microservices across cloud and edge data centers. Presently, the techniques that can dynamically detect and resolve resource contention via microservice performance characterization, workload prioritization and coordinated deployment is the focus area of research in osmotic computing.

Dell Technologies is conducting research on Federated Analytics through Project WWH. The capability of joining various, distributed data sources and performing analytics as if they were a single data source is known as Federated analytics, a core part of the osmotic computing analytics paradigm.

As one of the infrastructure providers for Enterprise IoT for, Edge, Fog and Cloud, they have divided the world into two fabrics as shown in Figure 17. The important layer of infrastructure fabric – what we call the data-zone layer. At the top is the analytics fabric that is exposed to the data scientist. The data scientist views the world as a gigantic virtual computing cluster. That is the abstraction we want the data scientist to know – to be aware that the data is distributed across the data zones, but that is the extent to which they need to be exposed. The data scientist initiates the computation in any of the nodes – there is no notion of hierarchy or a notion of master and slave. Rather, they connect to a virtual computing node, any node, and start the computation and that computation gets distributed.

IX. MONITORING

The complexity of the infrastructure and the inherent responsibility to monitor activities at the origins of scale for the deployed and allocated microservices is based on osmotic computing framework, where the infrastructure requirements are based on hardware resources in the data center (CPU, storage, and network), in-transit network (SDN/NFV-enabled routers and switches), and resources on the network edge (i.e. gateways). It is difficult in such complex systems with microservice-based deployment scenarios to detect problems (for example, in end-to-end request processing latency) and pinpointing the key-source culprit components (micro-service or data center resources or in-transit network).

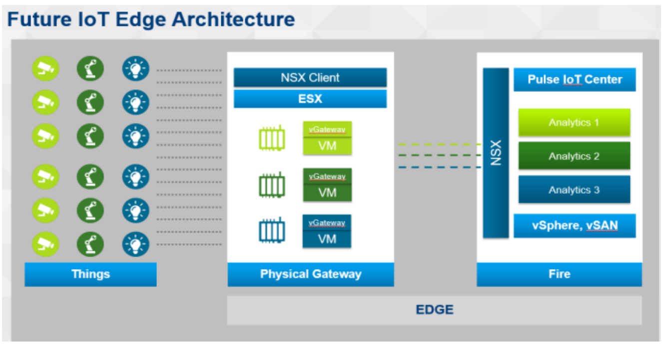


Fig. 18. **The Future IoT Edge Architecture from Dell Technologies**

It becomes extremely difficult to implement a robust monitoring technique to diagnose the root cause of QoS degradation with the present scale of microservices and infrastructure resources (data center, in-transit, and network edge) in a heterogeneous ecosystem. While present techniques and monitoring frameworks provided by Amazon Container Service (Amazon CloudWatch) and Kubernetes (Heapster) typically monitor CPU, memory, filesystem, and network usage statistics, they are unable to monitor microservice-level QoS metrics (query processing latency of database microserver, throughput of data compression microserver, and so on).

Researchers are investigating scalable methods (based on self-balanced trees) to monitor QoS and security metrics across multiple levels of osmotic computing, for the deployment of microservices in cloud and edge data centers. Presently, to the best of our knowledge, none of the approaches proposed in academic literature and commercial monitoring tools/frameworks can monitor and instrument data to detect root causes of QoS violations and failures across the infrastructure based on workload and QoS metrics logs (workload input and QoS metrics, disruptive event) across microservices, cloud data center, in-transit network, and edge datacenters. Dell Technologies' strategically aligned businesses are developing a Future IoT Edge Architecture. The ongoing research and industrial use case and implementations would determine change in product and services according, as shown in Figure 16.

X. MICROSERVICE ORCHESTRATION AND ELASTICITY-CONTROL

It is difficult to estimate the number of users connecting to different types of microservices in a scalable fog/edge/cloud system. Runtime orchestration of micro-services to characterize behavior of micro-service workload in terms of data volume to be analyzed, data arrival rate, query types, data processing time distributions, query processing time distributions, I/O system behavior is a well-known research problem. At any given time, it is difficult to make decisions about the types and scale of cloud and edge data center resources to be provisioned to microservices without a clear understanding of the workload behaviors of micro-services. For instance, Amazon's autoscaling service (https://aws.amazon.com/ autoscaling) employs simple threshold-based rules or scheduled actions based on a timetable to regulate infrastructure resources (for example, if the average CPU usage is above 40 percent, use an additional microservice container). OpenShift Origin (www.openshift.org) provides a microservice container reconfiguration feature, which scales by observing CPU usage ("scaling is agnostic to the workload behavior and QoS targets of a microservice") along with Kubernetes. There is on-going investigation by academic and industry researchers to use machine learning techniques for developing predictive models to forecast workload input and performance metrics across multiple, co-located microservices on cloud and edge data center resources for the Osmotic Computing framework. Current techniques for runtime control and reconfiguration that only considers resources hosted in cloud datacenters needs to be extended to resources that are deployed and available at the edge.

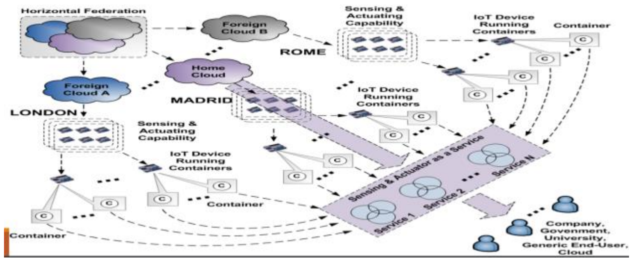


Fig. 19. **The evolution of Smart to Osmotic Cities from Professor:** MASSIMO VILLARI HEAD OF FUTURE COMPUTING RES LAB (FCRLAB) UNIVERSITÀ DI MESSINA

Additionally, intelligent, QoS-aware, and contention-aware resource orchestration algorithms should be developed based on the described models, monitoring systems, and configuration selection techniques. Osmotic computing enables a unifying paradigm for latency-sensitive applications to minimize data sizes that must be transferred over a network. There is an emphasis on research that would showcase reverse offloading – that is, movement of functionality from the cloud to the edge devices. However, all the available academic and industry research has been placed on (mobile) cloud offloading, whereby software applications can be offloaded from a mobile device to a data center.

XI. TOOLS TO MOVE MICROSERVICES

Tools or engines can be used to move microservices around seamlessly. Wikipedia references a number of cloud vendors, including:

Apache jclouds®: https://jclouds.apache.org/ is an open source multi-cloud toolkit for the Java platform, that gives you the freedom to create applications that are portable across clouds while giving you full control to use cloud-specific features. Clouds offers several API abstractions as Java and Clojure libraries. The most mature of these are BlobStore and ComputeService.

ComputeService: ComputeService streamlines the task of managing instances in the cloud by enabling you to start multiple machines at once and install software on them.

BlobStore: BlobStore is a simplified and portable means of managing your key-value storage providers. BlobStore presents a straightforward Map view of a container to access data.

LoadBalancer: The Load Balancer abstraction provides a common interface to configure the load balancers in any cloud that supports them. Define the load balancer and the nodes that should join it, and it will be ready for the action.

The Simple Cloud API: An API that enables users to access cloud application services written in the PHP programming language across different cloud computing platforms; Simple Cloud API is a common API for accessing cloud application services offered by multiple vendors.

The following services are supported:

- Storage with adapters for services such as Amazon S3 and Nirvanix [6].
- Document with adapters for services such as Azure Table Storage and Amazon SimpleDB.
- Queue with adapters for services such as Amazon SQS and Azure Queue Storage.

Deltacloud API: An application programming interface developed by Red Hat and the Apache Software Foundation that abstracts differences between cloud computing implementations. Deltacloud enables management of resources in different clouds by the use of one of three supported APIs; Deltacloud classic API, DMTF CIMI API or EC2 API.

This enables you to start an instance on an internal cloud and with the same code start another on EC2 or RHEV-M.

XII. CONCLUSION

In the end we can state abstraction of existing Computing (IoT/Edge/Fog/Cloud) is Osmotic Computing indeed. The evolution from Smart to Osmotic Cities can be seen from Figure 19.

XIII. REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In Workshop on Mobile cloud Computing, MCC '12, 2012.

[2] K. Hong, S. Smaldone, J. Shin, D. J. Lillethun, L. Iftode, and U. Ramachandran. Target container: A target-centric parallel programming abstraction for video-based surveillance. In International Conferenceon Distributed Smart Cameras, ICDSC '11, 2011.

[3] B. Koldehofe, F. Dürr, M. A. Tariq, and K. Rothermel. The power of software-defined networking: line-rate content-based routing using openflow. In Workshop on Middleware for Next Generation Internet Computing, MW4NG '12, 2012.

[4] B. Koldehofe, B. Ottenwälder, K. Rothermel, and U. Ramachandran. Moving range queries in distributed complex event processing. In Distributed Event-Based Systems, DEBS '12, 2012.

[5] A. Celesti et al., "Characterizing Cloud Federation in IoT," Proc. 30th Int'l Conf. Advanced Information Networking and Applications Workshops (WAINA), 2016, pp. 93–98.

[6] M.K. Qureshi and Y.N. Patt, "Utility-Based Cache Partitioning: A Low-Overhead, High-Performance, Runtime Mechanism to Partition Shared Caches," Proc. 39th Ann. IEEE/ACM Int'l Symp. Microarchitecture (MICRO 06), 2006, pp. 423–432.

[7] Q. Zhu and T. Tung, "A Performance Interference Model for Managing Consolidated Workloads in QoS-Aware Clouds," Proc. 5th IEEE Int'l Conf. Cloud Computing (CLOUD), 2012, pp. 170–179.

[8] S. Jain et al., "B4: Experience with a Globally-Deployed Software Defined WAN," Proc. ACM SIGCOMM, 2013, pp. 3–14.

[9] D. Puthal et al., "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," IEEE Cloud Computing, vol. 3, no. 3, 2016, pp. 64–71.

[10] M. Satyanarayanan et al., "Edge Analytics in the Internet of Things," IEEE Pervasive Computing, vol. 14, Apr. 2015, pp. 24–31.

[11] S. Abolfazli et al., "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges," IEEE Comm. Surveys Tutorials, vol. 16, First 2014, pp. 337–368.

[12] B.-G. Chun et al., "CloneCloud: Elastic Execution between Mobile Device and Cloud," Proc. 6th Conf. Computer Systems (EuroSys 11), 2011, pp. 301–314.

[13] W. Felter et al., "An Updated Performance Comparison of Virtual Machines and Linux Containers," Proc. IEEE Int'l Symp. Performance Analysis of Systems and Software (ISPASS), 2015, pp. 171–172.

[14] S. Govindan et al., "Cuanta: Quantifying Effects of Shared On-Chip Resource Interference for Consolidated Virtual Machines," Proc. 2nd ACM Symp. Cloud Computing (SOCC 11), 2011, pp. 22:1–22:14.

[15] R. Nathuji and A. Kansal, "Q-Clouds: Managing Performance Interference Effects for QoSAware Clouds," Proc. 5th European Conf. Computer Systems (EuroSys 10), 2010, pp. 237–250.

[16] Dynamic allocation of Serverless Functions in IoT Environment [Duarte Pinto] , Universidae Do Porto.

[17] Mobile Fog: A Programming Model for Large–Scale Applications on the Internet of Things [Kirak Hong, David Lillethun, Umakishore Ramachandran College of Computing Georgia Institute of Technology Atlanta, Georgia, USA]

[18] P.G.V. Naranjo, et al., FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments, J. Parallel Distrib. Comput. (2018), https://doi.org/10.1016/j.jpdc.2018.07.003

[19] Open Fog reference architecture https://www.openfogconsortium.org/ra/