mimik

MEC & MIMIK, ONE PLUS ONE EQUALS ELEVEN

let's edgifi the world together

DOCUMENT NAME

MEC & MIMIK; one plus one equals eleven

AUTHORS

Fay Arjomandi
Michel Burger
Siavash Alamouti

ABSTRACT

The purpose of this white paper is to describe an enhanced approach to Multi-access Edge Computing (MEC). The new approach enables end-devices to act as server nodes in close collaboration with MEC.

We first review the client/server software architecture and its evolution to microservice architecture. We will then illustrate the benefits of the new approach where all connected nodes on the network can act as cloud servers. This new approach keeps the separation between network, service, and application layers while providing each layer with contextual awareness of other layers for efficiency and optimization.

For the purpose of this document, we refer to the existing ETSI's MEC and 3GPP's SA6 architecture [4] [5] [6] [10] as "network-MEC". We also introduce the concept of "hybrid-MEC" where end-devices are included as server nodes as an integral part of MEC.

We will show that hybrid-MEC makes MEC more scalable, efficient, and financially feasible. We will also show that it can help improve data privacy and user experience when building cloud-native applications.

OVERVIEW

MEC and 3GPP SA6 promise to bring cloud functionalities closer to where “end-devices” run applications. This is done by embedding cloud functionalities in network elements that are customarily used purely for network functionalities today.

MEC targets a series of objectives [8] [11], including:

Reducing the latency between end-devices by hosting some backend functions on Base Stations and gateways,

Preserving network bandwidth by reducing the number of round trips to the cloud,

Offloading application functionalities from end-devices to the network-MEC reduces the need for computing resources on end-devices. This is, in effect, reverting to a thin client approach.

The first two objectives have rather obvious benefits. However, offloading device functions to the network embodied in the third objective can go against the first two principles. This could potentially increase end-to-end latency and waste bandwidth in many use cases and scenarios.

It also goes against major trends of increased device battery capacity, storage, memory, CPU, and GPU power. Furthermore, specialized integrated silicon capabilities in end-devices such as AI and security are being added.

Most end-devices such as smartphones are highly under-utilized, most of the time, when considering CPU and memory usage. They're designed for typical user-centric functions such as rendering rich graphics and quick real-time responses during sporadic periods of usage. These functions typically require very high levels of processing and memory when active but have low average requirements.

History has shown that end-devices have a much shorter lifetime than telco network appliances. Therefore, if we offload functions from end-devices to network appliances, we need to continuously add more network appliances to manage new end-devices.

To reduce latency and increase system-level efficiency, MEC can leverage end-devices as part of the network. We can reverse the role of end-devices from burdens to the network to collaborative server nodes that improve scalability and efficiency.

Moreover, processes (task/app) running on end-devices must be contextually aware of resources and services available within the network. They must invoke resources optimally in close coordination with all other network and server nodes. MEC can leverage the computing resources of underutilized connected devices to create an aggregate computing infrastructure that is orders of magnitude larger. The resulting platform is more bandwidth and energy-efficient with less end-to-end latency and better data privacy.

Moreover, Hybrid-MEC can improve Opex and Capex for the telcos. We can leverage the rapidly increasing computing resources in end-devices to increase the lifetime of MEC server resources in telco infrastructure. Otherwise, telcos will have to constantly add to their infrastructure as new devices and solutions are added.

EVOLUTION OF CLOUD COMPUTING

Offloading from central Cloud to end devices

The first generation of mobile internet applications were web browsers (aka thin clients) communicating with their backend hosted in the Cloud. This led to the birth of web service enablers over time. This was primarily driven by constraints in computing resources in end-devices.

Moreover, most use cases at the time were designed to consume content. Therefore, the traffic flow was mostly made up of media content from the Cloud to end-devices. However, in the last decade, since the inception of MEC, some underlying trends have changed the landscape:

End-devices have become much more powerful. For instance, smartphones today had more computing resources than servers just a decade ago.

Network traffic patterns changed as applications have evolved towards a more balanced uplink/downlink from a highly downlink-heavy usage pattern. Many applications today generate more data on end-devices than consume data on the downlink. [3]

Applications have much richer features and a higher level of integration with on-device hardware capabilities (microphone, Siri, payment services, camera, etc.).

The maturity of AI/ML has made real-time decisioning on devices feasible and will push more computing resources to end-devices.

As a result, he app platforms have been progressively moving from browser to native apps (aka fat clients). The transition to native apps in the majority of cases is driven by the following:

- Many application providers such as Facebook moved from browser to native apps to reduce their cost of Cloud and enable offline capabilities.
- To ensure a better user experience, they relegated many of the backend functions to the frontend. The native app uses the computing resources of end-devices to do as much as possible offline and on the device itself.
- Utilize and integrate with device peripherals and utilities such as microphones, cameras, sensors, Siri, etc.

Whilst offloading to the device has proven somewhat beneficial, it doesn't go far enough. Cloud-native apps remain dependent on many intermediaries for communications with other nodes. As a result, the usage of central cloud resources and intermediary nodes is not optimized. We have an opportunity to improve latency and efficiency by adding end-devices to the available pool of server nodes.

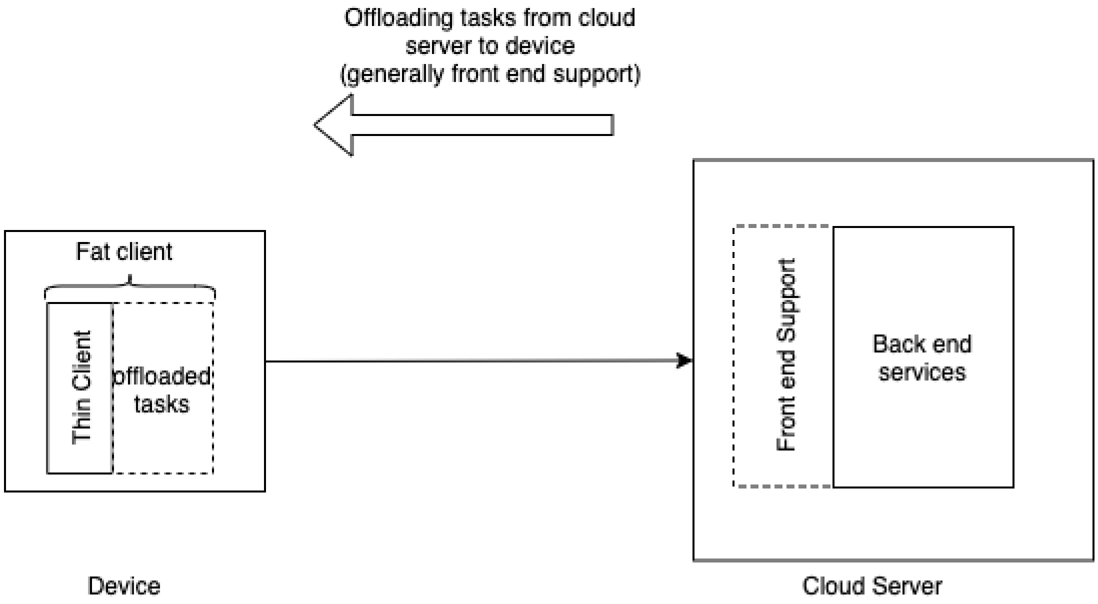

Figure 1: Cloud to device offloading

Offloading from the device and central cloud to the access network

Telcos are modernizing their network to modular and software-driven Network as a Service (NaaS) [1]. They plan to offer highly personalized services with full automation and agility of updates both internally and for their Enterprise customers. This has led to the introduction of MEC for service deployment. Similarly, there are ongoing efforts for orchestration and other similar architectures like 3GPP SA6, Software Defined Network (SDN), Network Function Virtualization (NFV) [2] [7].

As a result, networks are transforming from black boxes into distributed mini data centers [9], also known as "edge cloud computing nodes". They offer both connectivity and computing resources to devices attached to the network.

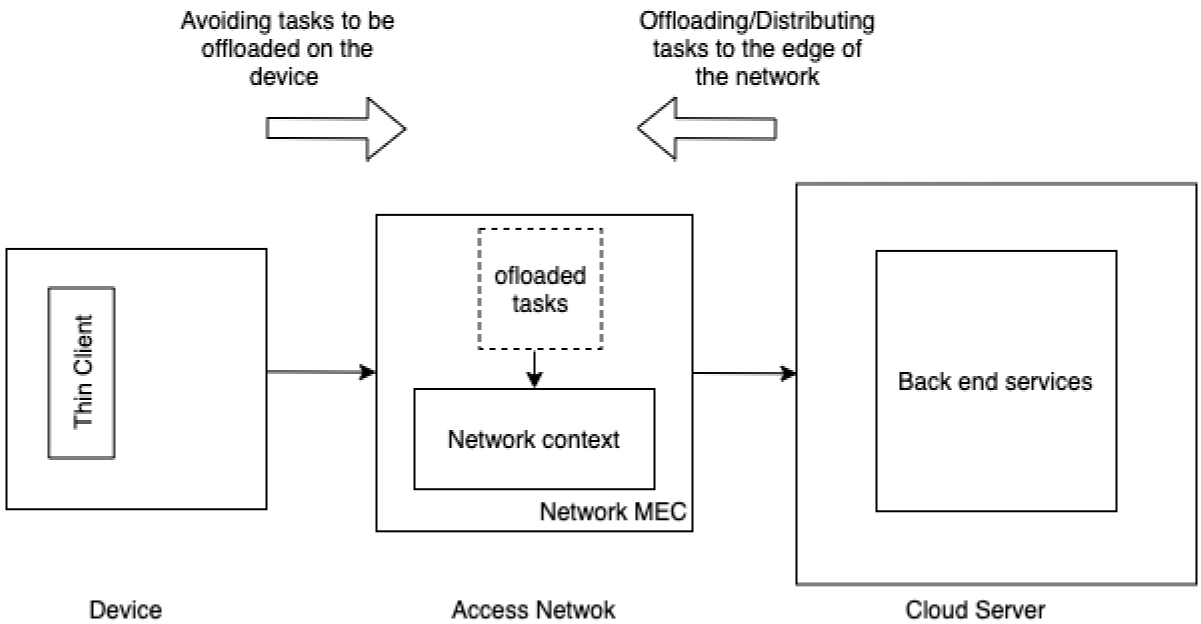

Figure 2: Offloading from the device and the cloud to the access network

To further expand on this, with network-MEC, we can turn a network node (e.g., Base Station) into a cloud node. This provides a new source of revenue for Telcos and improves latency of applications. However, it requires massive continual investments in Capex and Opex to keep up with the ever- increasing demands as new end-devices are added. The required additional investments further challenge the business case viability of 5G and network-MEC.

Available computing resources on end-devices will continue to grow thanks to Moore's laws. We have an opportunity to leverage these end-devices as cloud resources to address three main challenges of network-MEC highlighted below:

First challenge: lifetime of network-MEC appliances

Today, end-devices are confined to run only the frontend of apps. All other tasks are delegated to the access network or the data center. As a result, as new end-devices are added, the network infrastructure needs to be upgraded to keep up with the increased load.

With Hybrid-MEC, we increase the lifetime of network-MEC appliances by leveraging end-devices as server resources.

Second challenge: managing fluctuations in the device domain (including application session, physical device states)

With network-MEC, many of the end-device functions have to run on computing resources in the network. These end-devices have dynamic behaviour turning on or off or moving (detaching themselves from one network to reattach to another). Therefore, the management of these tasks is bound to inherent situational changes of an end-device.

The end-device turning on or off can be treated similarly to a blade disappearing or appearing in a data center. However, a moving end-device poses a different challenge. It puts a burden on the access network to closely follow the end-device as it moves across the network.

With hybrid-MEC, we can track and dynamically adjust to mobility issues right on the end-device.

Third challenge: Workflow Schizophrenia

Today, we deal with the business logic of complex workflows by applying sophisticated models in the cloud. If we offload some of the workflows from end-devices to the access network, we create an unnecessary intermediary. The access network is detached from the end-device and the user experience. Therefore, it should be an intermediary only if absolutely necessary.

With hybrid-MEC, we allow end-devices to manage as much of the workflow as possible to ensure optimal user experience. End-devices are the closest to the user experience and should therefore manage as much of the workflow as possible locally. In most scenarios, this will improve latency, performance, and user experience.

HYBRID-MEC, THE BEST OF BOTH WORLDS

One major objective of MEC /3GPP is to make applications more network-aware and the network more application-aware. This can be optimally achieved by the hybrid-MEC approach, where network-MEC can utilize end- device resources as cloud resources. The hybrid-MEC uses computational resources on end-devices for functions typically performed on the backend. It is also the most dynamic and ever-evolving node on the network.

The approach functions as a continuous fabric of cloud computing resources amongst end-devices:

- edge to/from other edge nodes,
- edge to/from a telco NaaS,
- edge to/from private/public cloud.

We can enable end-devices to act as servers when needed and runmicroservices on them. Moreover, end-devices can discover other end-devices and their microservices within clusters formed according to various scopes. Some examples of the scopes supported by the mimik platform are network, proximity, and account.

In effect, we can form instances of ad-hoc edge service mesh of microservices that communicate directly. We can use explicit addresses within and across particular clusters. With hybrid-MEC, the device becomes part of the cloud infrastructure as a full-fledged node of a distributed system.

From an architectural viewpoint, this leads to two major improvements and capabilities:

- Device as an extension of MEC and as an environment to run tasks (microservices),
- Device clusters forming ad-hoc instances of edge service mesh.

Device as an extension of MEC and as an environment to run tasks (microservices)

Process automation leveraging AI/ML will be embedded into almost all technology solutions as we transition to a hyper-connected world. This requires efficient communication between complex workflows amongst systems of heterogeneous nature. We need the lowest possible latency to ensure a highly personalized and real-time experience. Let's consider a few typical scenarios:

- Imagine the scenario where a neighborhood faces wildfire dangers. The local authority sends an alarm that gets picked up by the neighborhood sprinkle systems. The alarm then triggers all the sprinklers within each small residential neighborhood as needed to protect residential and commercial properties from fire.
- Or imagine when a robotic arm inside a factory detects its own abnormally, and as a result, it has to inform:
  - a dispatcher to schedule and dispatch operational maintenance personnel,
  - a risk management solution for liability and insurance,
  - dispatch operational management to mitigate any factory operational impact,
  - dispatch a person to perform the work manually while the robot arm is being fixed.

All the above examples illustrate various heterogeneous systems collaborating to manage workflows in a dynamic and fluid fashion.

Enabling end-devices that can run microservices as part of the MEC helps manage complex workflows and automation more efficiently. Also, given all these devices are powered up already, it can reduce the overall power consumption. We can save the power required to transmit the data back to the cloud and to power up the data center resources.

This approach drives sustainably starting from the endpoints (edge-in versus data-center-out). Finally, it accommodates the necessary dynamic response to the situation at the time.

In conclusion, providing edge context to solutions leads to a network-aware service (microservice) and service-aware network.

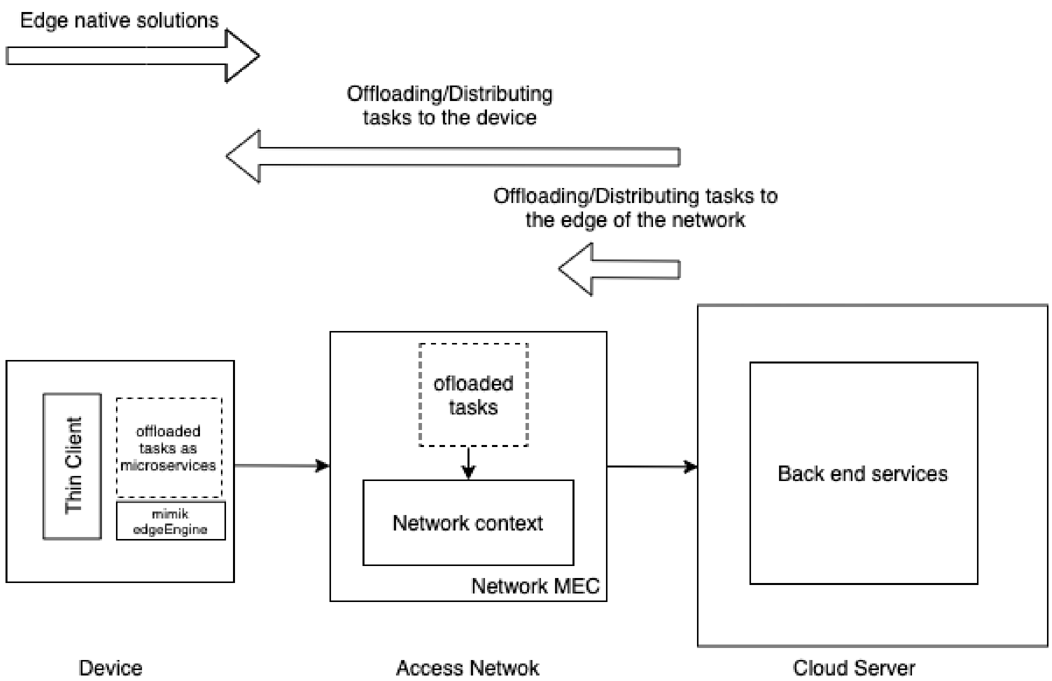

Figure 3: Hybrid approach of offloading and empowering edge native solutions

Device clusters forming ad-hoc instances of edge service mesh

Beyond the obvious benefits mentioned above, we can run tasks on end-devices as well as in the access networks. In effect, we can create instances of ad-hoc hybrid edge service mesh.

The combination of the ad-hoc edge service mesh and MEC has consequences on how solutions can be orchestrated:

- First, it is possible to bring global orchestration closer to targeted end-devices by running the orchestration in the desired network element. This will reduce the time for decision-making and will reduce bandwidth usage
- Second, we can implement smarts in microservices that run in end-devices where every microservice is aware of which service context it uns. Consequently, end-devices can choreograph themselves instead of being orchestrated. This alternative approach can reduce the time to make local decisions. Yet, it still can be complemented by global orchestration that runs in a network element close to end-devices.

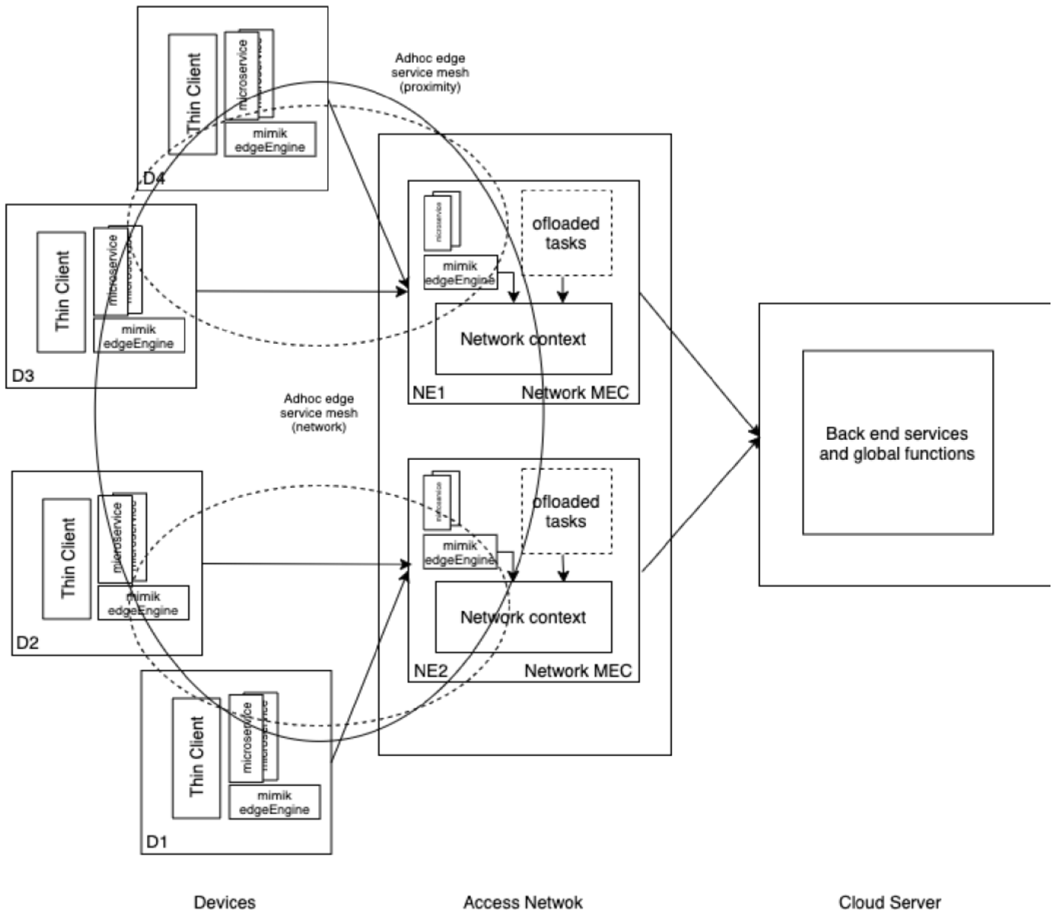

Figure 4: Hybrid approach for a service mesh

The benefits of the hybrid MEC

Running cloud tasks on end-devices as well as network nodes provides many benefits for both telcos and solution developers. Here are some of the potential benefits:

- Expand MEC capacity and reduce upgrade cycle:

  No longer limited to hyperscaler servers, tasks can run on all network elements and all connected devices. Consequently, the MEC capacity increases by the number of devices attached to it at the time. It also reduces the upgrade cycle of infrastructure, which reduces Capex and Opex costs of the MEC infrastructure for telcos. This also potentiallyreduces:

  - the cost of infrastructure as a service (IaaS) for the enterprises using MEC
  - the carbon footprint of enterprises aligned with their Environmental goals.

- Enables apps running on end-devices to have visibility to available resources on the network node and its capabilities.

  As a result, apps can trigger and instantiate tasks (microservices) to load and run on the best available resources. This also reduces the overhead on the network-MEC.

- Higher level of security

  All interactions within the hybrid-MEC (leveraging mimik edgeEngine) require OAuth2.0 authentication and authorization. Hence, security is built-in from the edge out. Additionally, all apps and microservices interacting with the edgeEngine must have proper authorization for access and must be authenticated.

  In other words, transactions are secured end-to-end starting from the device all the way to other nodes and back. Therefore, there is no need to build moats around network-MEC.

- Clear continuum for deploying tasks from the device to the cloud, including the network

  This allows developers to deploy services in the most appropriate place in a static or dynamic fashion.

- Flexibility to decide where along the cloud continuum from device to MEC to data centers:

  Processing needs, context, and need for global or local knowledge can define how far to the edge a task is deployed. The expanded choice from data centers to the network to end-devices helps optimize or eventually optimize the solution.

- Data generated by the device becomes actionable and can be consumed right at the source:

  When the applications on a device are only tasked to generate or render data, it takes a few processes to make the data actionable. This includes cleaning or normalizing but also adding more meaning and context (for instance, when the data is just a reference).

  Making the data actionable at the source or close to the source enhances the value of the data. Therefore, we can improve the ratio of the "value of the data over the cost of transport". We also eliminate the need to trigger small tasks running further in the cloud continuum. The data can become actionable at source and consumed in a meaningful way earlier, reducing latency and transport and hosting costs.

  The meaningful data can also trigger special routing to a specific cloud environment to either optimize further or to secure based on the data itself. Providing access to network context enables dynamic routing based fon the meaning of the data.

- Cascaded processing

  Some tasks may need to be split into many subtasks similar to a factory line concept. It is more efficient to be able to start the process on the end-device where most of the processes start.

- Shadowing reality and context

  End-devices are the closest to the real context. When an end-device is used as a server, the solution becomes more reality and context-aware. For example, when an end-device moves, it is important that the deployed service in an access network “follow” the device which may require tearing down/redeploying the service while the device is moving.

  If the service runs in the end-device, it will move with it. The end-device is already aware of the movement and doesn't have to use approximation or modeling.

- Call routing optimization using network context

  Hybrid-MEC clusters of devices can be discovered according to specific scopes. For instance, all devices that are on the same network, or within given proximity, or belonging to a given account. We can then create a service mesh of all the microservices that run on these edge devices.

  Indeed, all the end-devices are generally connected to the network. However, this doesn’t mean that they can communicate directly or that microservices at the solution level serving them can communicate directly.

  Access to the network context as per MEC and SON specs enables peer-to-peer device connectivity and peer-to-peer microservice communication. This approach reduces the need to go deep into the cloud continuum to facilitate the necessary communication between various microservices.

DETAILED INTEGRATION

There are four possible levels of integration between mimik and MEC/3GPP technology from light (no integration) to deep integration. Additionally, a combination of two or more levels could co-exist in the same solution.

Level 0: over-the-top (OTT) MEC

The mimik edgeEngine is agnostic to the network implementation and the cloud provider. Therefore, like many OTT-type solutions, the mimik platform can implement an edge solution similar to the microservice-based development model of backend solutions. The platform has a built-in API gateway sidecar concept (like with edge Istio) and uses container images for managing microservices. This provides an environment to develop microservice-based solutions at the edge, including ad-hoc instances of edge service mesh.

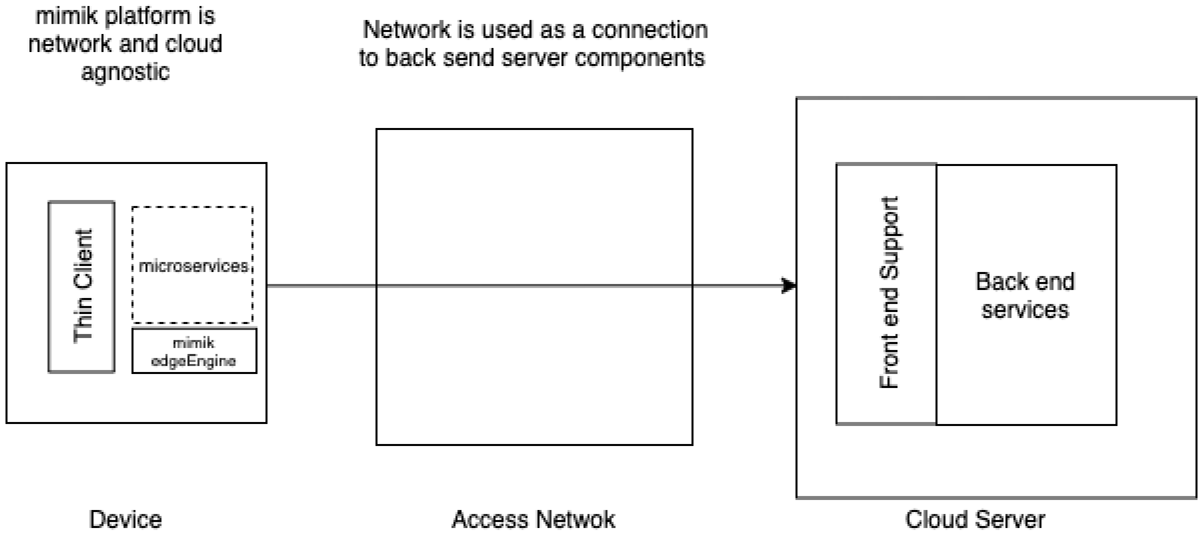

Figure 5: mimik platform as MEC OTT

In this model, there is zero integration required between mimik edgeEngine and network-MEC. The two can coexist and be utilized without even having any knowledge of one another. For example, a microservice running on edgeEngine can point to network-MEC instead of a resource in the data center when required.

Level 1: Using the network as a "closer" cloud

We can combine the capabilities of MEC and mimik edgeEngine by interfacing the edgeEngine with the Edge Enabler Client (EEC) as defined in the 3GPP Spec. In this model, mimik edgeEngine enables a microservice to run on an end-device. When an edge microservice makes an API call, the Edge Enable Client routes that call to the service running on network-MEC using EDGE-5 API.

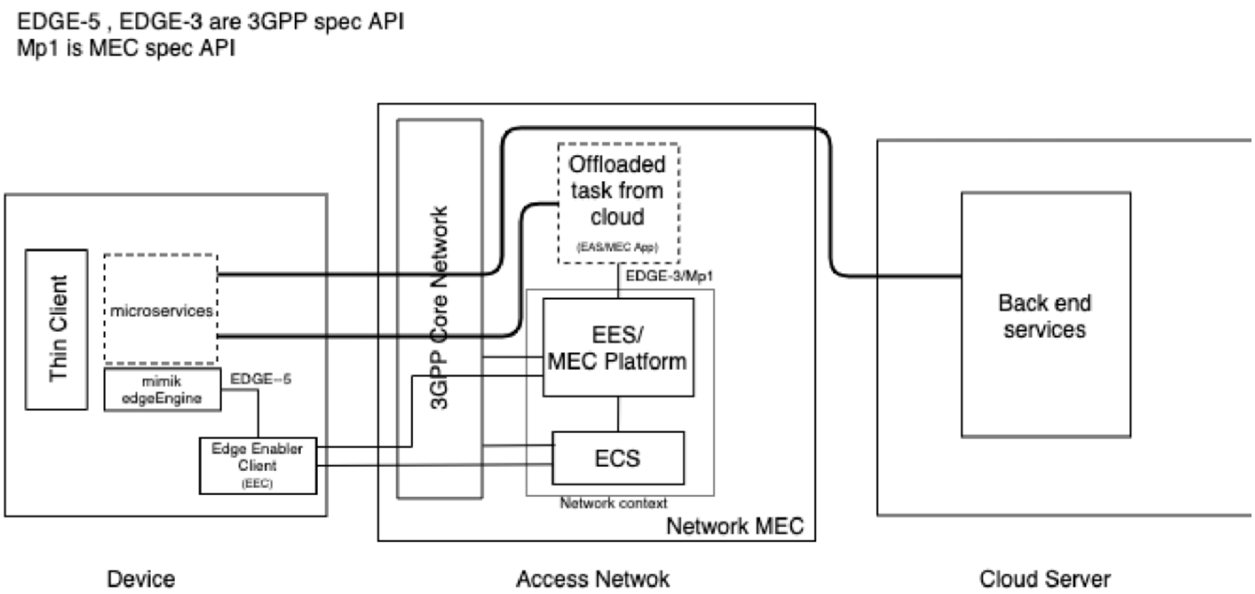

Figure 6: Network as a "closer" cloud

Level 2: Using the network as a network-aware virtual gateway

In many solutions, a designated gateway aggregates functions or does some orchestration of locally centralized decisions or actions. This generally requires special equipment, which can become an architectural bottleneck and a point of failure.

The mimik platform does not use a dedicated static gateway approach. Instead, it defines special node roles either by election and/or selection in a dynamic fashion. It is possible to use mimik edgeEngine to enable any network-MEC node to become a de-facto gateway. This eliminates the need for special equipment, making the solution more reliable.

The virtual gateway in an end-device gains access to the network context based on the application context. It can therefore optimize the network or be informed by the network of any relevant context changes.

In this model, one could even do without an Edge Enabler Client. Using edgeEngine on end-devices and on network elements enables seamless discovery and communication of microservices. It also provides the service mesh context necessary for the operation. We have already tested this approach by integrating the edgeEngine on eNodeBs with Lime Microsystems.

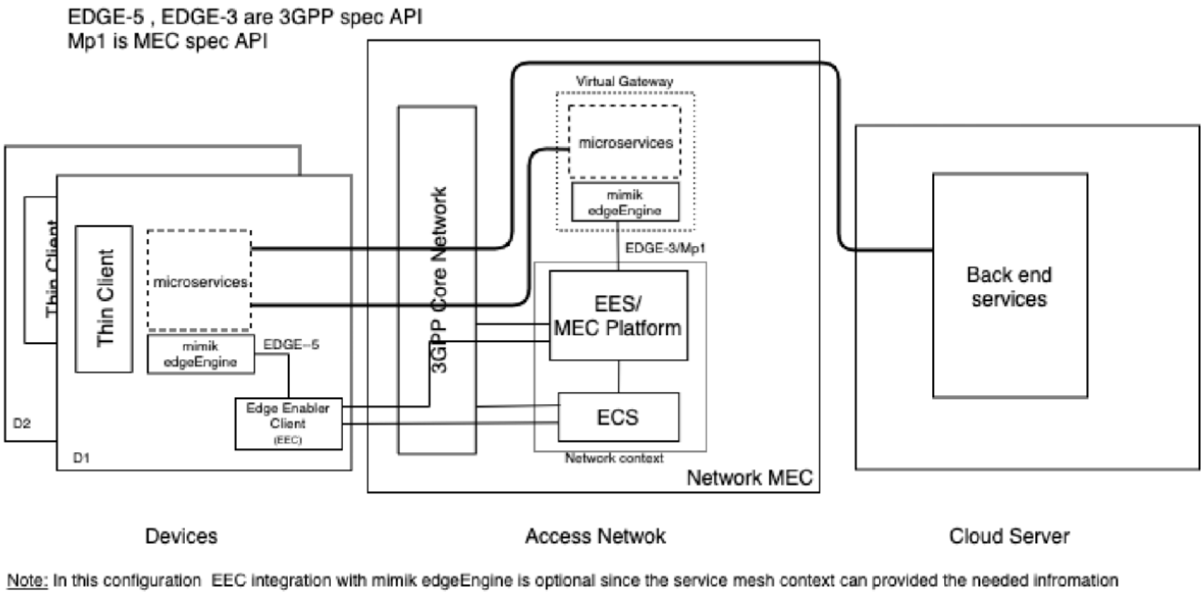

Figure 7: Network-aware virtual gateway

Level 3: Service-mesh-aware network

The mimik edgeEngine enables the formation of an edge service mesh consisting of a cluster of devices. The mimik platform enables microservices within the same service mesh to communicate directly via explicit addresses.

Each microservice is aware of its context and knows the addresses of all the microservices that are part of the context. This helps avoid a publish/subscribe model to communicate with other microservices.

While communication is guaranteed, the way the communication is achieved depends on the available connectivity. Without optimization (like the OTT approach shown in Figure 5), the communication can unnecessarily go all the way to data centers.

We can create a self-organized network to establish all necessary connections between the nodes throughout the network. We can either treat all nodes as network elements (see Figure 7) or provide all nodes access to the network context (see Figure 6). Moreover, this is feasible within one network element or amongst several network elements (see Figure 4 with direct communication between NE1 and NE2).

As a result, all network elements are aware of all instances of service mesh established at the solution level. This minimizes the need to go deep in the cloud continuum to enable direct microservice to microservice communications.

CONCLUSION

The hyperconnected world has introduced new challenges with the explosion of smart devices and apps with massive amounts of data that need to communicate. This has created complex integration requirements between siloed data-rich domains built on heterogeneous operating systems and networks.

To manage the complexity, it is essential to take a system-level approach that considers all the nodes and devices. At the system level, the network can expand its role and go beyond managing a connection fabric. In effect, we can use network appliances, end-devices, and backend systems together as one environment where services are deployed. To make this feasible, all the nodes need to discover and organize themselves in complex and dynamic instances of service mesh.

In these environments, end-devices play a key role as the boundary between physical and digital realities. To enable smart automation and efficient operations, digital systems need to follow reality more closely instead of modeling them like we commonly do today. For instance, smart cars should interact directly with the passengers' smartphones and the smart homes services, smart cities, and smart offices.

To make these possible, the network needs to adapt in real-time to the dynamic nature of the instances of the service mesh. It is always possible to take an OTT approach and go deep into the cloud continuum to enable communication within a service mesh. However, it is much more efficient for the network to become aware.

REFERENCES

[1] Alusha, D. (2020, April 6). Cloud-Native networking for a 5G Era. ABI Research: for visionaries. https://www.abiresearch.com/blogs/2020/04/06/cloud-native-networking-5g-era/.

[2] Baroudy, K., Kishore, S., Mahajan, N., Nair, S., Sigurdsson, H., & Sukumar, K. (2019, October 18). *Reinventing telco networks: Five elements of a successful transformation*. McKinsey & Company. https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/reinventing-telco-networks-five-elements-of-a-successful-transformation.

[3] Brown, G. (2016, July). Mobile Edge Computing Use Cases & Deployment Options . Juniper. https://www.juniper.net/content/dam/www/assets/white-papers/us/en/mobile-edge-computing-use-cases-and-deployment-options.pdf

[4] ETSI MEC Leadership Team. (2020). ETSI MEC: An Introduction (almost) everything you want to know about ETSI MEC. ETSI. https://www.etsi.org/images/files/technologies/ETSI-MEC-Public-Overview.pdf

[5] ETSI. (2020, April). Multi-access Edge Computing (MEC); V2X Information Service API. France. https://www.etsi.org/deliver/etsi_gs/MEC/001_099/030/02.01.01_60/gs_MEC030v020101p.pdf

[6] Kekki, S., Arora, R., Contreras, L. M., Fang, Y., Featherstone, W., Frydman, D., Jiangping, F., Kim, K., Kuure, P., Li, A., Odgers, A., Purkayastha, D., Ranjan, A., Scarpina, S., Verin, G., & Wen, K.-W. (2018, June). MEC in 5G networks. France; ETSI. https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp28_mec_in_5G_FINAL.pdf

[7] Liyanage, M., Porambage, P., Ding, A. Y., & Kalla, A. (2021, May 15). *Driving forces for MULTI-ACCESS edge COMPUTING (MEC) IoT integration in 5G*. ICT Express. https://www.sciencedirect.com/science/article/pii/S2405959521000631.

[8] Owen, G. (2020, June 25). *5G MEC - deployment options and challenges for mobile operators*. Counterpoint Research. https://www.counterpointresearch.com/5g-mec/.

[9] Shen, J., Eng, D., Gu, S., Huang, R., & Brower, J. (2020, September). Cloud Interfacing at the Telco 5G Edge. LF Edge. https://www.lfedge.org/wp-content/uploads/2020/09/Akraino_Whitepaper2.pdf

[10] Sprecher, N., Elloumi, O., Kuure, P., Neal, A., Soloway, A., Harmand, A., Chitturi, S., Reznik, A., Li, A., Costa, C., Sabella, D., Moses, D., Gunjal, D., Hall, E., Chen, G., Singh, H., Martin, J., Shen, J., Feng, J., … Yoon, M. (2020, July). Harmonizing standards for edge computing - A synergized architecture leveraging ETSI ISG MEC and 3GPP specifications. France; ETSI. https://www.etsi.org/images/files/ETSIWhitePapers/ETSI_wp36_Harmonizing-standards-for-edge-computing.pdf

[11] Verizon. (2019, December 3). 5G and edge computing White paper Unleashing the power of real time. https://www.verizon.com/business/resources/whitepapers/2019/5g-and-edge-computing.pdf?_ga=2.136114545.1718011598.1629758070-1691642166.1629758070

let's edgifi the world together