EPI Framework: Approach for traffic redirection through containerised network functions

Jamila Alsayed Kassem[1]

[1]UvA, MNS group

On the road towards personalised medicine, secure data-sharing is an essential prerequisite to enable healthcare use-cases (*e.g.* training and sharing machine learning models, wearables data-streaming, etc.). On the other hand, working in silos is still dominating today's health data usage. A significant challenge to address here, is to set up a collaborative data-sharing environment that will support the requested application while also ensuring uncompromised security across communicating nodes. We need a framework that can adapt the underlying infrastructure taking into account norms and policy agreements, requested application workflow, and network and security policies. The framework should process and map those requirements into setup actions. On a low packet level, the framework should be able to enforce the setup route via intercepting and redirecting traffic.

In our approach, we utilise programmable infrastructures to set up, on the fly, the chain of functions mapping to the requirements communicated by other EPI components; Bridging Function Chain (BFC). As a result, the EPI (Enabling Personalised Interventions) [1] Framework (EPIF), dynamically provisions and instantiates virtual network services to support an application archetype and accommodate to different infrastructural capabilities across communicating nodes.

The current generation of network and ICT infrastructures already heavily rely on virtualisation, given the successful evolution of virtualisation over the past decades [7]. The ETSI (European Telecommunications Standards Institute) standardised the NFV architecture, which can be extended to define the next generation of network infrastructures [1]. The architecture addresses the management and coordination of network resources for cloud-based applications and the network services lifecycle. Moreover, the NFV paradigm allows on-demand implementing and instantiating of NF's such as firewalls, segmentation, Deep Packet Inspection (DPI), etc. This is fundamental when dealing with heterogeneous collaborative domains.

These technologies can be utilised to build a dynamic network infrastructure to adapt to different requirements/ healthcare application request. We use container-based Virtual Network Function (VNF's) to accomplish fast deployment, high reusability, and low-performance overhead. In this case, network functions and all their dependencies are encapsulated in lightweight Docker containers to offer platform independence, fast instantiation time, low resource utilisation, and high processing rate. After setting up the NF's, redirection tools should be employed through a specified route.

In the EPIF, the BFC's are configured and deployed (as in Figure 1), and we enforce the instantiated specified route by deploying different redirection methods: SOCKS [5], and NGINX [6]. Then, we evaluate each method according to overhead and throughput performance. But the questions remain, *"what BFC we need to set up knowing that a certain application workflow was requested?"*, and *"where do we host these functions?"*.

To answer these questions, different EPIF components are put in collaboration, as in Figure 1. Once a collaborate request is submitted to the **application orchestrator** [8], it builds the workflow needed and communicate to the infrastructure orchestrator the end nodes required to run this. The infrastructure orchestrator queries requirements for such an archetype from the **policy management system**[4] and the **logic area generator** [2]. The infrastructure orchestrator then translate the requirements into configuration actions and fires-up the needed BFC. The **proxy** component is the framework's actor, and it is used to host the pool of implement BF's and control traffic redirection.

In [3], we evaluated and benchmarked the two redirection approaches in terms of time overhead, and the rate of processed transaction per second. We determined that the overhead of the EPIF proxies differs in the various implementations, and it relates directly to the positioning of the proxy within a network topology.

In the ICT.Open presentation we will be focusing on introducing the framework in Figure 1, and discussing our benchmark results. In the future, we plan on evaluating the adaptiveness and resilience of the EPIF

[1] https://delaat.net/epi/

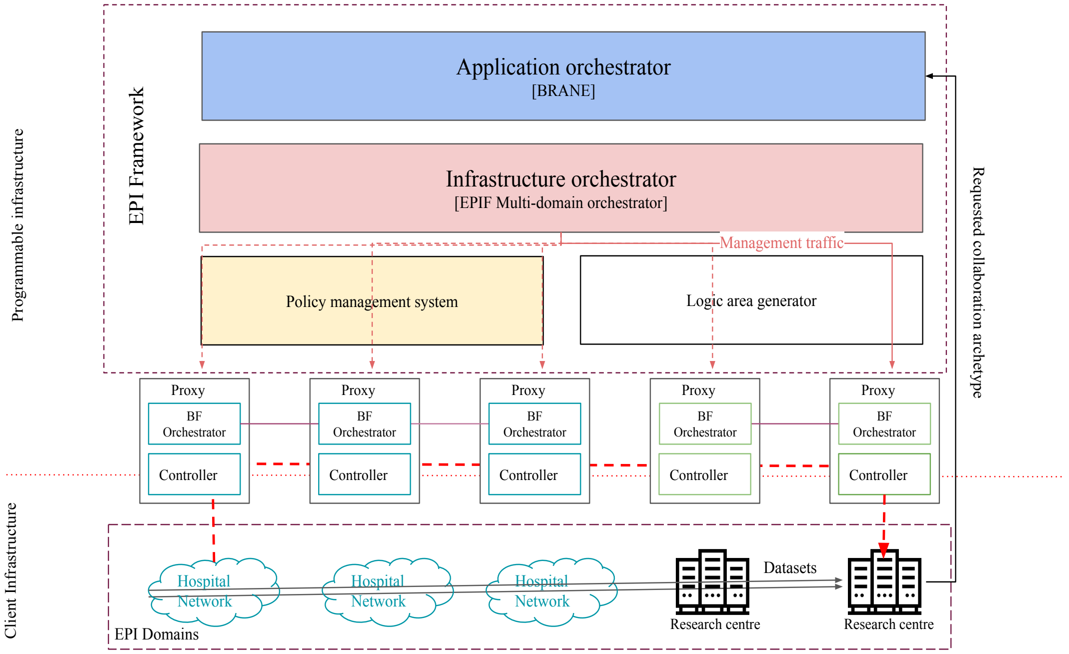

Figure 1: The EPI framework and its components to set up a collaborative environment between different EPI domains.

setup by defining a scaling strategy per BFC topology. The key limitation we expect to face is the framework's acceptance by health providers and the lack of bridgeability of several security functions, and hence the ability to fulfil some security requirement. We aim to work collaboratively with hospital IT professionals to address these problems.

References

[1] URL: https://www.etsi.org/technologies/nfv.

[2] J. A. Kassem et al. "The EPI Framework: A Dynamic Data Sharing Framework for Healthcare Use Cases". In: *IEEE Access* 8 (2020), pp. 179909–179920. DOI: 10.1109/ACCESS.2020.3028051.

[3] Jamila Alsayed Kassem et al. "EPI Framework: Approach for Traffic Redirection Through Containerised Network Functions". In: *2021 IEEE 17th International Conference on eScience (eScience)*. IEEE. 2021, pp. 80–89.

[4] Milen G Kebede, Giovanni Sileno, and Tom Van Engers. "Automated regulatory constraints and data governance for healthcare". In: ().

[5] Marcus D. Leech. *SOCKS Protocol Version 5*. RFC 1928. Mar. 1996. DOI: 10.17487/RFC1928. URL: https://www.rfc-editor.org/info/rfc1928.

[6] *NGINX Reverse Proxy*. Tech. rep. URL: https://docs.nginx.com/nginx/admin-guide/web-server/reverse-proxy/?_ga=2.225849692.737245468.1617007670-911961388.1613593493.

[7] N. Omnes et al. "A programmable and virtualized network IT infrastructure for the internet of things: How can NFV SDN help for facing the upcoming challenges". In: *2015 18th International Conference on Intelligence in Next Generation Networks*. 2015, pp. 64–69. DOI: 10.1109/ICIN.2015.7073808.

[8] Onno Valkering, Reginald Cushing, and Adam Belloum. "Brane: A Framework for Programmable Orchestration of Multi-Site Applications". In: *2021 IEEE 17th International Conference on eScience (eScience)*. 2021, pp. 277–282. DOI: 10.1109/eScience51609.2021.00056.