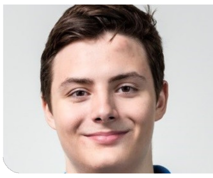
Network insights in OpenTelemetry

Graduate

Michael Brändli



Leandro Ceriani

Initial Situation: Distributed applications are increasingly becoming state-of-the-art in software development in the age of containers and various cloud providers. This naturally offers excellent flexibility for application developers. However, distributing applications has some downsides. The more systems are involved in a distributed application, the more challenging it becomes to troubleshoot in case of an error or slow application behavior. OpenTelemetry, a vendor-neutral observability framework, excels at capturing application-level insights. In OpenTelemetry, an essential component is still missing. Specifically, there is currently no information about the network devices between the different tiers of an application. Telemetry of Network components is especially useful when the application is hosted across different data centers or clouds.

Approach: The aim was to evaluate different variants and test how telemetry data can be collected from network devices. Four different variants were considered and evaluated. The next step was to collect this data in a central location and integrate it into a trace in OpenTelemetry.

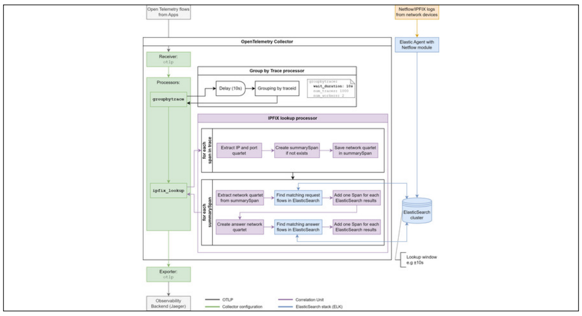
OpenTelemetry is an open-source observability framework. It helps developers gain insights into the performance and behavior of their systems. It is not designed out of the box to integrate with telemetry data from network devices. In order to integrate network telemetry data, the code base of an OpenTelemetry collector must be extended.

Result: Network components like routers and firewalls sent Netflow / IPFIX data to a central ElasticSearch cluster. A newly developed processor in the OpenTelemetry collector was created. This ipfix_lookup processor searches the ElasticSearch cluster for network telemetry data as soon as the application telemetry data arrives. This processor then integrates the network data into existing application traces.

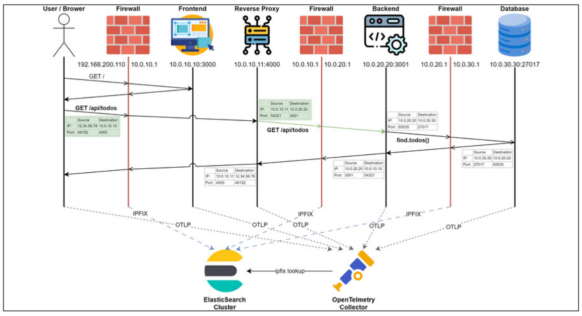
The developed solution successfully provides a comprehensive view of latency and performance across the distributed system by integrating network component data into the existing traces. Latency at the network layer could be analyzed and correlated with application-level requests, enabling more effective troubleshooting and optimization. This advantage minimizes the effort required for troubleshooting, which indirectly leads to a reduction in resources and costs. The enhanced visibility provided by the ipfix_lookup processor will invariably contribute to an extended mean time between failures.

While the developed solution achieved the desired objectives, there are opportunities for further improvement. Future work could focus on enhancing the analysis capabilities by including information about the reason for a delay. This could be archived by incorporating additional data from the network components, such as the device status and error information. Additionally, the solution could be extended to support other observability backends and protocols for collecting network component data.

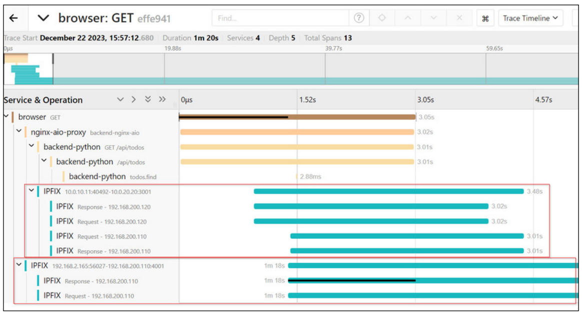
Custom collector / correlation unit
Own presentment



3-tier-app traffic flow example
Own presentment



Network telemetry data span - final implementation
Own presentment



Advisors
Urs Baumann, Yannick Zwicker

Co-Examiner
Philip Schmid, Isovalent GmbH, Zürich, ZH

Subject Area
Networks, Security & Cloud Infrastructure