



Linus Basig



Fabrizio Lazzaretti

Graduate Candidates	Linus Basig, Fabrizio Lazzaretti
Examiner	Prof. Dr. Olaf Zimmermann
Co-Advisor	Dr. Gerald Reif, Innovation Process Technology, Zug, ZG
Subject Area	Internet Technologies and Applications
Project Partner	CARU AG, Zürich, ZH

Reliable Messaging Using the CloudEvents Router



The CARU Device placed in the home of a senior citizen
CARU AG

Problem: CARU is an AgeTech startup offering a voice-controlled emergency-call device designed to help the elderly live an independent life for longer. The device will react to a cry for help and automatically activate an emergency call to up to 5 family members. CARU's software architecture is heavily event-driven, envisioning a unified event plane letting events flow between all connected systems. To support this vision, CARU's events are structured according to the CloudEvents specification of the Cloud Native Computing Foundation (CNCF). In our previous student research project, we designed, implemented, and open-sourced the CloudEvents Router (github.com/ce-rust/cerk) – a first step towards this vision.

To entrust mission-critical events to the CloudEvents Router, it has to guarantee that these messages arrive. The goal of this bachelor thesis is to extend the router to provide such a delivery guarantee. This guarantee should also apply when an event is routed across different protocols. We focused on the protocols that the CloudEvents specification defines a protocol binding for: AMQP, Kafka, MQTT, NATS, and WebHooks.

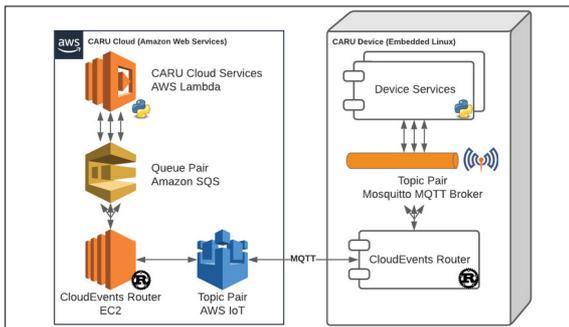
Approach: As a starting point, we conducted extensive literature research to identify and compare existing reliable messaging definitions and patterns.

Our research showed that the most feasible approach to provide reliable routing is to implement an end-to-end delivery guarantee of "At Least Once". This guarantee applies from the event source, via the router, to the event destination. We then analyzed which delivery guarantees are provided by the mentioned messaging protocols and how interoperable the supporting concepts are. Having adapted the architecture design of the existing CloudEvents Router to fulfill the new requirements, we implemented a proof-of-concept that routes messages between AMQP and MQTT with an "At Least Once" delivery guarantee.

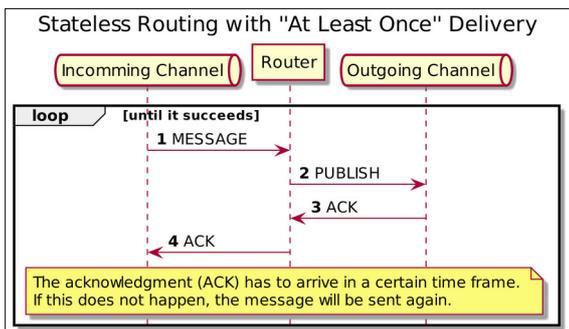
We validated the correctness of our implementation in two end-to-end test-scenarios. Both test cases demonstrated that the shortcomings of the previous implementation could be overcome.

Result: We released our enhanced open-source CloudEvents Router implementation, now supporting the desired "At Least Once" delivery guarantee in addition to the already existing "Best Effort" policy. The new version also supports the AMQP protocol complementing the already existing support for MQTT.

One challenge was that the MQTT specification does not require the implementation to include the message processing in the application in its "At Least Once" delivery guarantee. This behavior can lead to message loss during the routing process. This imprecision in the specification materialized in the consequence that none of the eligible libraries allows extending the delivery guarantee to the message processing. To solve this issue, we proposed a change to optionally include the message's processing in the "At Least Once" delivery guarantee for the Mosquitto MQTT library. With this new guarantee, CARU has started evaluating the CloudEvents Router for productive use on their device.



CARU's envisioned system context
Own presentation



Stateless routing with the "At Least Once" delivery guarantee
Own presentation