



Original software publication

GeoBroker: A pub/sub broker considering geo-context information

Jonathan Hasenburg*, David Bermbach

TU Berlin & Einstein Center Digital Future, Mobile Cloud Computing Research Group, Einsteinufer 17, 10587 Berlin, Germany



ARTICLE INFO

Keywords:
Geo-context
IoT
Data distribution

ABSTRACT

The majority of today's pub/sub brokers only focus on data content when deciding where to distribute data. In this paper, we introduce the GeoBroker pub/sub broker software that additionally considers the geo-context associated with data producers and consumers. As a result, data consumers only receive relevant messages which preserves bandwidth and compute power while data producers can use the geo-context of clients to control data distribution, e.g., for enhanced privacy or as an alternative to credential-based authentication.

Code metadata

Current code version	4.0.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2020-30
Permanent link to Reproducible Capsule	leave empty
Legal Code License	GPL-3.0
Code versioning system used	git
Software code languages, tools, and services used	Java, Kotlin, Maven
Compilation requirements, operating environments & dependencies	Linux, macOS, Microsoft Windows
If available Link to developer documentation/manual	https://github.com/MoeweX/geobroker/tree/single
Support email for questions	jh@mcc.tu-berlin.de

Software metadata

Current software version	4.0.0
Permanent link to executables of this version	https://github.com/SoftwareImpacts/SIMPAC-2020-30/tree/single/release
Permanent link to Reproducible Capsule	leave empty
Legal Software License	GPL-3.0
Computing platforms/Operating Systems	Linux, macOS, Microsoft Windows
Installation requirements & dependencies	Java 11
If available, link to user manual - if formally published include a reference to the publication in the reference list	https://github.com/MoeweX/geobroker/tree/single/release
Support email for questions	jh@mcc.tu-berlin.de

1. Introduction

Most data generated by IoT devices are only relevant to data consumers in a particular physical area [1]. Yet, the majority of pub/sub brokers available today only focus on data content when deciding where to distribute data. As a result, data consumers have to locally process and filter vast amounts of irrelevant messages.

In this paper, we introduce the pub/sub broker software GeoBroker that considers the geo-context associated with data producers and consumers. The GeoBroker software is based on our published work [2] and [3]. With GeoBroker, data consumers can specify that they only want to receive data from producers in a specific geographical area. A car, for example, might want to receive data only from other cars

* Corresponding author.

E-mail addresses: jh@mcc.tu-berlin.de (J. Hasenburg), db@mcc.tu-berlin.de (D. Bermbach).

<https://doi.org/10.1016/j.simpa.2020.100029>

Received 3 August 2020; Received in revised form 4 August 2020; Accepted 11 August 2020

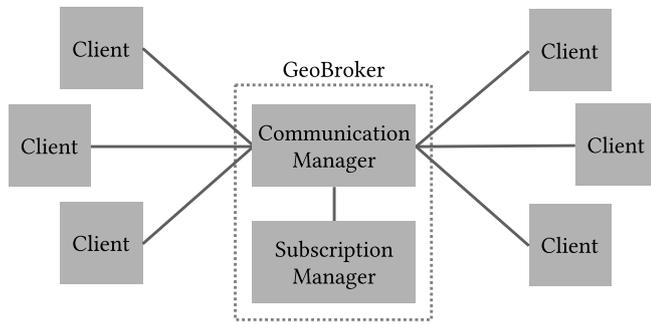


Fig. 1. The communication manager maintains connections to clients, the subscription manager is responsible for message filtering.

in its immediate vicinity. Data producers, on the other hand, can specify that their data should only be delivered to consumers within a specified geofence, e.g., to prevent data misuse or as an alternative to credential-based authentication.

2. GeoBroker software

The GeoBroker software offers a similar functionality and operational behavior as other software systems used to distribute IoT data. A general requirement for such a system is that it supports loosely coupled communication between clients that do not have to know each other [4]. Furthermore, the software should be lightweight, payload agnostic, and usable by devices that operate in a constrained environment. For these reasons, GeoBroker as well as other popular software solutions such as AWS IoT¹ or Google Cloud IoT² rely on the topic-based pub/sub messaging paradigm.

On a very high level, the GeoBroker architecture comprises two components (see Fig. 1): a communication manager and a subscription manager. The former is responsible for maintaining communication links with the clients that use GeoBroker for distributing and receiving messages (see also Section 2.1). The latter is responsible for message matching, i.e., it determines which clients should receive a given message (see also Section 2.2). All GeoBroker components, as well as the example clients provided along the GeoBroker software, are implemented in a mixture of Java 8 and Kotlin 1.3.

2.1. Interacting with GeoBroker

To ease integration into existing IoT applications, GeoBroker's operational behavior is very similar to other MQTT-v5.0-compliant [5] pub/sub brokers, i.e., which follow a widely used pub/sub protocol. Thus, GeoBroker's message types are very similar to the ones of vanilla MQTT, but we piggyback the information needed for the geo-context based data distribution on top of existing messages.³ Clients can connect and ping GeoBroker (on top of these messages, we piggyback client locations), subscribe/unsubscribe to MQTT compliant topics (on top of subscribe messages, we piggyback information on where data producers have to be located), and publish messages to MQTT compliant topics (on top of publish messages, we piggyback information on where data consumers have to be located). When disconnecting from GeoBroker, all active client subscriptions are removed.

¹ <https://aws.amazon.com/iot/>.

² <https://cloud.google.com/solutions/iot/>.

³ Note, that while the message types are similar, we use a more human readable message encoding scheme (JSON) than defined by the MQTT protocol.

2.2. Message matching process

The GeoBroker subscription manager component uses content and geo-context information to match incoming messages and active client subscription. This message matching process comprises three checks: Content Check, Consumer GeoCheck, and Producer GeoCheck (Fig. 2). When all three checks are successful for any of a client's subscriptions, GeoBroker delivers the incoming message to that client. The Content Check validates that the subscription topic matches the message topic as defined by the MQTT-v5.0 protocol [5]. The Consumer GeoCheck validates that the data producer, i.e., the client that published the incoming message, is inside an area defined by the data consumer, i.e., the client that created the respective subscription. The Producer GeoCheck validates that the data consumer is inside an area defined by the data producer.

3. Impact overview

GeoBroker is the first open source pub/sub broker that considers content and geo-context information of data producers and consumers when distributing messages. We have used the GeoBroker software and its underlying principles for two scientific publications [2,3]. By making it available as open source, however, GeoBroker can also become the basis for a new branch of pub/sub research that aims to make use of geo-contexts for additional purposes. For example, in [6], we describe a preliminary approach on how to use geo-context information for efficient rendezvous-based message routing between geo-distributed pub/sub broker instances. This approach builds upon the original GeoBroker Software and is still under active development. The GeoBroker software can also be used to quickly prototype applications for data-intensive scenarios, especially in the Internet of Things, as our approach reduces the amount of irrelevant messages that are distributed and received by clients [3]. Because GeoBroker considers more geo-context dimensions than related approaches, e.g., [7–9], it can be used for a wider-range of application scenarios which require fine-grained control of data distribution, i.e., it can be used to control data distribution based on client locations, areas defined by clients, or both.

Furthermore, GeoBroker comprises a novel indexing structure that allows us to quickly retrieve all areas that contain a given location (needed for the Consumer GeoCheck). Beyond its use inside a pub/sub system, this index data structure has been proven useful for other purposes: it is part of the data analysis pipeline of the SimRa project [10] where near miss incidents in bicycle traffic – recorded with the usual GPS accuracy – need to be mapped to the corresponding street segment or intersections.

4. Conclusion

In this paper, we introduced the pub/sub broker software GeoBroker. GeoBroker considers the geo-context associated with data producers and consumers when distributing messages. As a result, data consumers receive only relevant messages which preserves bandwidth and compute power. Data producers, on the other can, can use additional ways to control data distribution for enhanced privacy or as an alternative to credential-based authentication.

CRediT authorship contribution statement

Jonathan Hasenburg: Conceptualization, Software, Validation, Investigation, Visualization, Writing - original draft. **David Bermbach:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

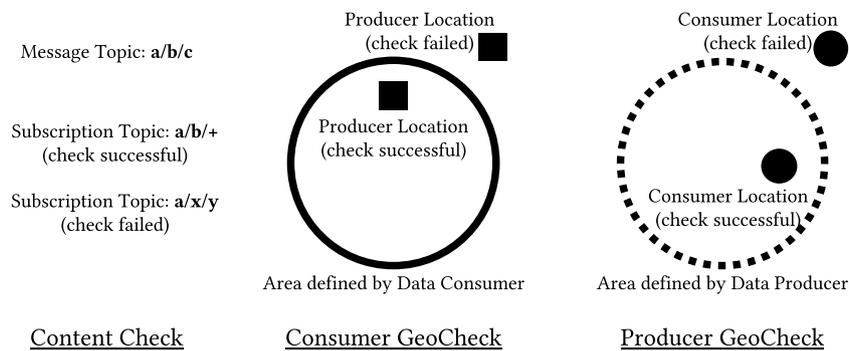


Fig. 2. Only if an incoming messages passes all three checks for a given subscription, GeoBroker delivers the message to the subscription’s client.

References

[1] P. Bellavista, A. Corradi, A. Reale, Quality of service in wide scale publish-subscribe systems, *IEEE Commun. Surv. Tutor.* 16 (3) (2014) 1591–1616, <http://dx.doi.org/10.1109/SURV.2014.031914.00192>.

[2] J. Hasenburg, D. Bermbach, Towards geo-context aware IoT data distribution, in: *Service-Oriented Computing – ICSSOC 2019 Workshops*, Springer, 2019, pp. 111–121, http://dx.doi.org/10.1007/978-3-030-45989-5_9.

[3] J. Hasenburg, D. Bermbach, GeoBroker: Leveraging geo-contexts for IoT data distribution, *Comput. Commun.* 151 (2020) 473–484, <http://dx.doi.org/10.1016/j.comcom.2020.01.015>.

[4] K. Paridel, E. Bainomugisha, Y. Vanrompay, Y. Berbers, W. De, Middleware for the internet of things, design goals and challenges, in: *EASST Context-Aware Adaptation Mechanisms for Pervasive and Ubiquitous Services*, vol. 28, 2010, <http://dx.doi.org/10.14279/tuj.eceasst.28.392>.

[5] A. Banks, E. Briggs, K. Borgendale, R. Gupta, MQTT Version 5.0, OASIS Standard. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>, (Accessed 29 July 2020), 2019.

[6] J. Hasenburg, D. Bermbach, Using geo-context information for efficient rendezvous-based routing in publish/subscribe systems, in: *KuVS-Fachgespräch Fog Computing 2020*, TU Wien, 2020, pp. 4–7, <http://dx.doi.org/10.34726/kuvs2020>.

[7] S. Herle, R. Becker, J. Blankenbach, Bridging GeoMQTT and REST, in: *Proceedings of the Geospatial Sensor Webs Conference*, 2016.

[8] B. Chapuis, B. Garbinato, L. Mourot, A horizontally scalable and reliable architecture for location-based publish-subscribe, in: *IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, IEEE, 2017, pp. 74–83, <http://dx.doi.org/10.1109/SRDS.2017.16>.

[9] R. Bryce, T. Shaw, G. Srivastava, MQTT-G: A publish/subscribe protocol with geolocation, in: *41st International Conference on Telecommunications and Signal Processing (TSP)*, IEEE, 2018, pp. 1–4, <http://dx.doi.org/10.1109/TSP.2018.8441479>.

[10] A.-S. Karakaya, J. Hasenburg, D. Bermbach, SimRa: Using crowdsourcing to identify near miss hotspots in bicycle traffic, *Pervasive Mob. Comput.* 67 (2020) <http://dx.doi.org/10.1016/j.pmcj.2020.101197>.