Publish-Pay-Subscribe Protocol for Payment-driven Edge Computing

Gowri Sankar Ramachandran, Sharon L.G Contreras, and Bhaskar Krishnamachari
University of Southern California

Abstract

IoT applications are starting to rely heavily on edge computing due to the advent of low-power and high data-rate wireless communication technologies such as 5G and the processing capability of GPU-driven edge platforms. However, the computation and the data communication model for the edge computing applications are quite diverse, which limits their interoperability. An interoperable edge computing architecture with a versatile communication model would lead to the development of innovative and incentive-driven edge computing applications by combining various data sources from a wide array of devices. In this paper, we present an edge computing architecture by extending the publish-subscribe protocol with support for incentives. Our novel publish-pay-subscribe protocol enables the data producers (publishers) to sell their data with data consumers and service providers (subscribers). The proposed architecture not only allows the device owners to gain incentive but also enable the service providers to sell the processed data with one or more data consumers. Our proof-of-concept implementation using AEDES publish-subscribe broker and Ethereum cryptocurrency shows the feasibility of publish-pay-subscribe broker and its support for data-driven and incentive-based edge computing applications.

1 Introduction

Applications in the area of connected and autonomous vehicles [7] and smart cities [14] are considering edge computing to offload heavy computations and to aggregate information from multiple data sources. Such a model not only reduce the response time but also reduce the bandwidth usage by limiting the data sent to the cloud. For example, the connected and autonomous vehicles can share their LIDAR (Light Detection and Ranging) data to neighboring edge devices for creating a real-time traffic map for the cars in the geographical neighborhood.

A wide-array of edge computing architectures and data communication models have been presented in the literature [5, 12, 13, 15]. The state-of-the-art approaches combine publish-subscribe communication model with microservices architecture to build services at the edge [15], but such architectures either assume that the data owners share their data with the services or the data and service owners belong to the same organization. Such models limit the effectiveness of the data and lack interoperability since the organization that owns the infrastructure may define a custom data format to build edge services.

Besides, the edge devices including embedded sensors, smartphones, cars, and traffic signals gather valuable data from operational environments. Aggregation of data at the edge server increases the utility of the data. However, device owners have no incentive to share their data. Emerging data marketplaces [8–10] discuss the benefits of data marketplaces and explains how such marketplaces enable the application developers and service providers to build novel and innovative applications by buying data streams from devices. Marketplace for machine learning tasks is also presented in the literature [2] which explains how the data can be used to build robust machine learning models. Existing literature motivates the need to integrate incentives for building new classes of applications and edge services by acquiring data from multiple sellers at the edge.

In this paper, we propose an edge computing architecture with built-in support for incentives. Our novel publish-pay-subscribe payment protocol allows the device owners to gain incentive for publishing their data to a data broker. The subscriber has to pay the fee set by the publisher before the data consumption. We introduce an edge computing architecture using the publish-pay-subscribe (pub-pay-sub) broker to create a value chain, wherein the device owners are incentivized for sharing their data with service providers, and the service providers may sell the processed data feedback to one or more device owners. Application developers can deploy computation services or containers on the edge platform and create a subscription channel to a data broker for buying valuable application data. The integration of a publish-subscribe communication pattern and the introduction of data payments allow the edge service providers and consumers to build data-intensive edge computing applications. Note that the proposed pub-pay-sub protocol is also applicable to services and applications operate in a cloud environment, but we focus on the edge computing case in this version of the paper. Our proof-of-concept implementation using AEDES publish-subscribe broker and Ethereum cryptocurrency shows the feasibility of publish-pay-subscribe broker.

2 Motivation and Related Work

2.1 Publish-Subscribe Communication Pattern for Edge Computing

The publish-subscribe communication pattern follows a centralized architecture, wherein the devices or data producers *publish* their data to a data broker using a human-readable *topic* string. Applications subscribe to the data by registering their interest to a topic. The lightweight nature of publish-subscribe clients coupled with the broker's ability to isolate the publishers and subscribers in space, time, and synchronization [6] make this communication pattern more popular in distributed applications.

The application of publish-subscribe communication pattern to edge computing allows the application developers to build scalable and interoperable edge computing applications. Data producers including embedded devices, sensors, and connected vehicles publish the data to a broker. One or more applications or services can consume the data from the broker by merely subscribing to the relevant topic. The publish-subscribe communication pattern is widely discussed in the context of edge computing [5].

Building edge computing application using a publish-subscribe communication enables the application developers to create portable and lightweight services at the edge for machine learning, data aggregation, and real-time control by merely subscribing to the data sources from a broker. Contemporary edge application architectures focusing on customized network stack require a dedicated client-side software for sending data to the nearest edge service. Such architectures result in a collection of edge service providers each following their architecture and a messaging model for delivering services at the edge. We argue that the integration of payment to the publish-subscribe communication model for edge computing provides the application developers and the edge service providers to create a scalable and interoperable system for edge computing.

2.2 Support for Data Economy

Payments, incentives, and pricing models have been discussed for IoT and edge computing in the literature [2, 8–10]. With the introduction of low-cost sensors and the innovations in the connected vehicles sector, a large number of devices are expected to collect data from their operational environment. Sharing these data with other application developers to create services at the edge would lead to the development of innovative data-driven applications powered by machine learning and artificial intelligence algorithms. Note that the data infrastructure is critical to train and create accurate models [2]. Protocols such as SDPP [11] allow the device owners to sell their data using a client-server architecture. The publish-subscribe communication model is widely considered in edge computing scenario, but the existing literature does not integrate payment to the publish-subscribe model, which we believe is essential to support the emerging data economy.

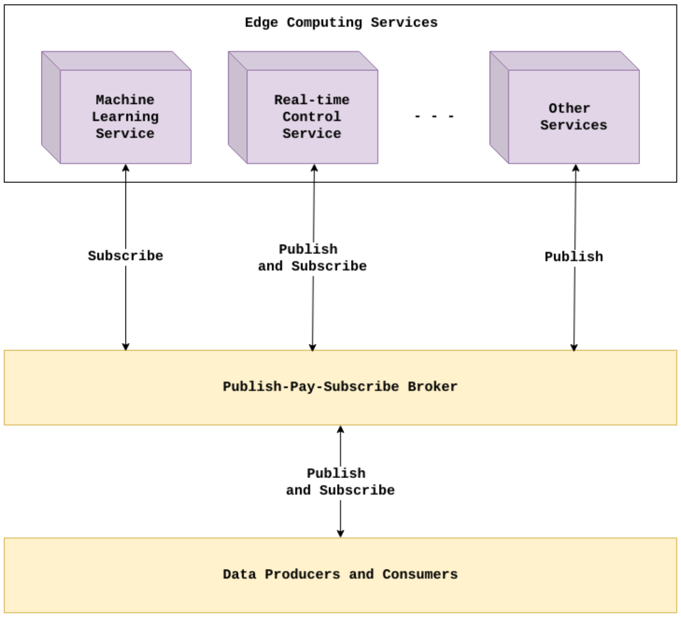

Figure 1: Publish-Pay-Subscribe based Edge Computing.

3 Edge Computing with Publish-Subscribe Payment Protocol

We propose an architecture for edge services using the publish-subscribe payment protocol. Figure 1 shows our proposed architecture, which consists of three layers:

Publishers and Subscribers: At the south-side of the broker, we have a collection of publishers and subscribers. Publishers are edge devices with the capability to publish data to brokers, and they can charge for their data by setting a price. A subscriber at the south-side may buy data from a broker. Service providers (from the north side) publish their processed data feed for other subscribers at the south-side.

Broker: A broker is a centralized software entity that handles the publishers and subscribers. Contemporary brokers

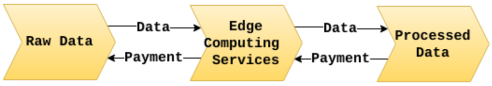

Figure 2: Value Chain Created by Publish-Pay-Subscribe Protocol based Edge Computing.

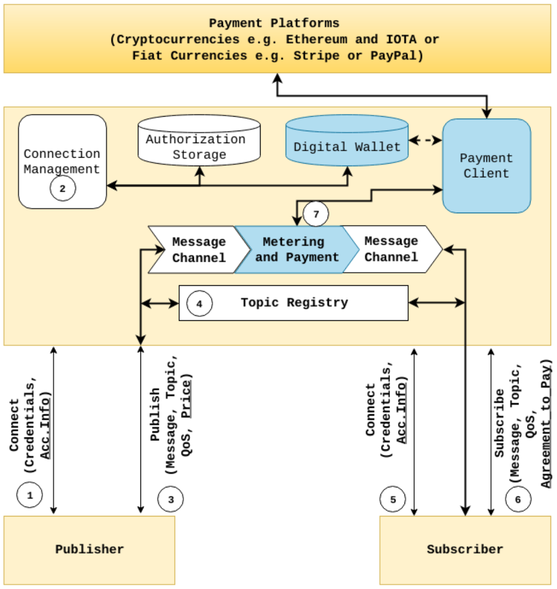
Payment Platforms
(Cryptocurrencies e.g. Ethereum and IOTA or
Fiat Currencies e.g. Stripe or PayPal)
Connection Management
2
Authorization Storage
Digital Wallet
Payment Client
7
Message Channel
Metering and Payment
Message Channel
4
Topic Registry
Connect (Credentials, Acc.Info)
1
Publish (Message, Topic, QoS, Price)
3
Connect (Credentials, Acc.Info)
5
Subscribe (Message, Topic, QoS, Agreement to Pay)
6
Publisher
Subscriber

Figure 3: Publish-Pay-Subscribe Communication Model.

directly forward the published data to the subscribers, and there is no support for the publishers to monetize the data. In this work, we extend the traditional publish-subscribe communication model with payment support, wherein the publishers have the opportunity to charge for their data and subscribers are required to pay the charge set by the publishers to consume the data. We assume that the broker is hosted on an edge server. Services can be built around the broker, but we do not define where the services should be hosted. It is up to the application developers to host their services on edge, cloud, or even on the edge devices at the south side.

Service Providers: At the north end of the broker, we have a collection of service providers for edge computing. An example could be a collection of connected cars publishing their LIDAR data to the nearest edge service provider to create a real-time traffic map of a given region. The service provider can sell the processed information in the form of a real-time map to the cars on the road. Combining the data from multiple vehicles allow the vehicles on the road to get a live traffic update on freeways. Such an application can be developed using any architecture, but the loose coupling provided by the publish-subscribe communication model enables the vehicles to share the same LIDAR information with machine learning and AI algorithms to create models for autonomous cars. Figure 2 shows the value that the pub-pay-sub broker adds to the edge computing applications.

The architecture of publish-pay-subscribe broker in Figure 1 shows that a) the traditional publish-subscribe communication model requires extensions to handle payments and b) a reference service model is needed to build novel edge services to buy and sell data streams from a broker. The rest of the section discusses these building blocks.

3.1 Publish-Pay-Subscribe Protocol

Figure 3 shows the architecture of our novel publish-pay-subscribe broker. We will explain our architecture through the interaction sequence below:

1. **Publisher connects to Broker:** Contemporary publishers connect with a broker using a credential provided by the broker. The broker manager provides the credentials including the URI, socket port no, username, and password. We extend the connection operation by introducing the payment information block, which includes the account name to which the payment should be made and the currencies (see Section 5).

2. **Broker accepts the connection from Publisher:** The broker accepts the connection by checking the credentials and store the payment information to a database.

3. **Publisher starts to Publish:** When a publisher starts to publish the data to the broker, it will include a topic, message, and the price per message, if applicable. Note that the publisher can decide to charge for the data if he/she wishes to. Note the publishers may have to choose a reasonable price to encourage the subscribers to buy the data (see Section 5).

4. **Broker registers the Topic:** When a publisher sends a message including topic and payment information, the broker checks whether there are any subscribers interested in consuming the data. If there are interested subscribers, the broker charges each subscriber based on the amount set by the publisher. The announcement of topics and price has to be implemented using a novel announce/query function (see Section 5).

5. **Subscriber connects to Broker:** A subscriber connects to the using a credential and the account information that he/she wishes to use for making payments. The broker stores the information on its storage to handle payments in the future. The subscriber is assumed to know the topic he/she wants to subscribe to, but we plan to implement a query/announce function to inform the subscribers of the new topics and their price(see Section 5).

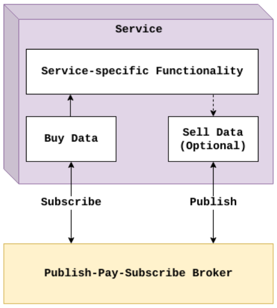

Figure 4: Service architecture for Publish-Pay-Subscribe Protocol.

6. **Subscriber subscribes to a Topic:** A subscriber subscribes to the topic, and he/she agrees to pay the fee charged by the publisher for the data.

7. **Broker charges the Subscriber for Messages:** Whenever the broker forwards the priced data to a subscriber, it charges the subscriber by transferring the payment from the subscriber's account to the publisher's account. Figure 3 shows the introduction of metering and payment to the message channel, which is responsible for charging the subscriber as the data flows from the publisher to the subscriber.

The above steps explain the novel publish-pay-subscribe protocol. Note that the broker-owner or manager recuperates the operational cost by charging a transaction fee from the publisher and the subscriber. We explain how to integrate these protocol with edge services in the next subsection.

3.2 Building Services around Publish-Pay-Subscribe Protocol

Figure 4 shows the service architecture for the publish-pay-subscribe protocol. Our publish-pay-subscribe protocol charges the data consumers for their subscriptions. Unlike traditional pub-sub based service architectures, our approach requires the integration of payment to the subscription channel. As described in Step 5 of Section 3.1, the subscriber has to share the account information and the payment currency at the connection phase, and it should agree to pay for the data when buying a priced data. Note that the architecture we present does not require subscribers to share any metadata with other subscribers, including what data they are paying for and how much. Service providers can develop services at the edge using microservices architecture or other SaaS models as long as they use our modified subscriber client to pay for the data. Besides, the service providers can sell the processed data to the devices by publishing the data back to the broker with the price attached it. This model allows the device owners to get an incentive for selling their data and the service providers to sell the processed data by including the processing cost to the price. The applications, whether it is machine learning, AI, or other services, have to define channels to publish and subscribe based on their underlying application.

4 Preliminary Implementation and Evaluation

We implemented a proof-of-concept publish-pay-subscribe broker using AEDES [1], which is a barebone publish-subscribe broker based on MQTT specifications [4]. Our proof-of-concept implementation is available as an open-source software here: https://github.com/ANRGUSC/pspp. The current version provides support for publishing data with pricing information to the broker, subscribe to paid data, and the support for payment at the broker. For payment, we used Ethereum (ETH) as a cryptocurrency on Ropsten Ethereum testnet. We evaluated our proof-of-concept implementation on a laptop, which influences the end-to-end delay due to lack of network routers and hops on the route.

The introduction of payment to the publish-subscribe channel adds a delay since the broker has to process the published data stream and charge the subscriber as the data flow between the publishers and subscribers. Our implementation handles the payment for each data stream in the background while the data is relayed to the subscriber, which means the subscriber may receive the data before the payment is debited from his/her account. The metering module is responsible for tracking the data flow and charging the subscribers. Figure 5 compares the end-to-end delay of the publish-subscribe broker with the publish-pay-subscribe broker. The invocation of the payment handler increases the end-to-end delay. As the number of messages per second increases, the end-to-end delay also increases due to the payment handling process and the buffering delay.

4.1 Payment per Each Data Point vs. Aggregated Payment

Triggering the payment handler for each data point is expensive for two reasons. First, the broker has to invoke the payment API to charge the subscriber, which adds a small computation and networking overhead to the broker. Second, the payment platforms charge a transaction fee for each transaction. To minimize the transaction and resource overhead, we introduce the k-parameter, which charges the subscriber for every k^{th} message. By reducing the number of payment triggers, the broker can handle the messages much faster with a small delay in the order of milliseconds as shown in Figure 5. We believe that the trend will remain the same on evaluations involving brokers, publishers, and subscribers running on a

Protocol	Transaction fee per K messages
Pub-Pay-Subscribe with K=1	1$ = (10 × 0.1$)
Pub-Pay-Subscribe with K=10	0.1$

Table 1: Transaction Fees for Pub-Pay-Sub Broker with K=1 and K=10.

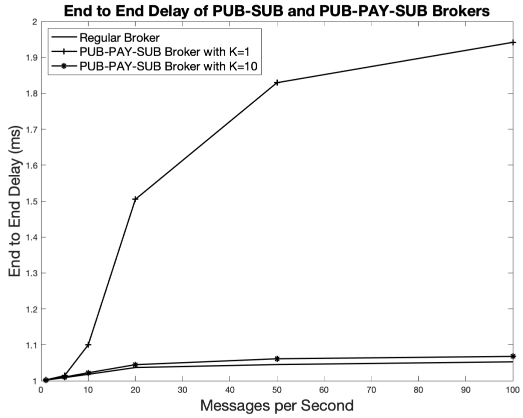

Figure 5: Service architecture for Publish-Pay-Subscribe Protocol.

large network.

4.2 Message Size Limitations for Services

Our evaluation using AEDES broker supported a message size of up to 256 MB, which means even data from cameras and other large data rate sensors can publish data to the publish-pay-subscribe broker. Service providers can build a large collection of applications using the data collected using publish-pay-subscribe broker. Note that the 256 MB limit is defined in the MQTT standard [4].

5 Open Issues

5.1 Payment Currencies

We provide support for both fiat and cryptocurrencies at the broker. Therefore, the publisher can choose their currency as long as the broker has the library to handle the payment transactions for the selected currency. Cryptocurrencies such as Ethereum and IOTA are suitable for microtransactions, but such platforms have a higher transaction cost and take tens of seconds to process the transactions.

5.2 Processing and Pricing of Data

In our current version, we restrict the broker from processing the data to minimize the processing overhead. The service providers at the north-side of the broker can buy the data from a broker and process it before reselling as shown in Figure 4. Allowing the publishers to decide on the price for the data may result in the data not being purchased by subscribers if the publisher charges a very high price. At the other side, the data owners may not be willing to give away their ownership to the broker to let the broker manager select the price for data/topics. We are investigating different models for pricing to help the data owners to preserve their ownership while choosing a satisfactory price for the data. We are exploring a concept called “discount code" to allow a specific set of subscribers to get instant access to the data. A monthly subscription model with an unlimited number of subscriptions is also a possibility. Also, we intend to explore the ideas of dynamic, differentiated pricing and auctions in our future work.

5.3 Rating

A rating system is essential to keep track of the behavior of the publishers and subscribers and to weed out dishonest buyers or sellers [3]. We plan to explore rating framework to the pub-pay-sub protocol in our future work.

6 Conclusion

In this paper, we have presented an edge computing architecture based on a novel publish-pay-subscribe protocol. The proposed edge computing architecture built using pub-pay-sub protocol allows service providers to buy data from a large collection of data owners by paying incentives. We have presented the architecture involving publishers, subscribers, service providers, and broker and explained how the publish-pay-subscribe protocol could be used to create value-added services at the edge. Our preliminary implementation using AEDES publish-subscribe broker and Ethereum cryptocurrency showed the feasibility of publish-pay-subscribe protocol. We believe that our publish-pay-subscribe protocol has the potential to impact future edge computing applications as we step into the era where "data is the new oil".

Acknowledgments

This material is based in part upon work supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001117C0053. Any views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] AEDES- A Barebone MQTT Broker, 2019.

[2] Anish Agarwal, Munther A. Dahleh, and Tuhin Sarkar. A marketplace for data: An algorithmic solution. *CoRR*, abs/1805.08125, 2018.

[3] Aditya Asgaonkar and Bhaskar Krishnamachari. Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator. *CoRR*, abs/1806.08379, 2018.

[4] Andrew Banks and Rahul Gupta. Mqtt version 3.1. 1. *OASIS standard*, 29, 2014.

[5] S. Chun, S. Shin, S. Seo, S. Eom, J. Jung, and K. Lee. A pub/sub-based fog computing architecture for internet-of-vehicles. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 90–93, Dec 2016.

[6] Patrick Th. Eugster, Pascal A. Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, June 2003.

[7] Dennis Grewe, Marco Wagner, Mayutan Arumaithurai, Ioannis Psaras, and Dirk Kutscher. Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions. In *Proceedings of the Workshop on Mobile Edge Communications*, pages 7–12. ACM, 2017.

[8] Bhaskar Krishnamachari, Jerry Power, Seon Ho Kim, and Cyrus Shahabi. I3: An iot marketplace for smart communities. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 498–499. ACM, 2018.

[9] Krešimir Mišura and Mario Žagar. Data marketplace for internet of things. In *2016 International Conference on Smart Systems and Technologies (SST)*, pages 255–260. IEEE, 2016.

[10] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing*, 3(4):585–598, 2015.

[11] Rahul Radhakrishnan and Bhaskar Krishnamachari. Streaming data payment protocol (sdpp) for the internet of things. In *Workshop on Blockchain for the IoT (BIoT)*, 2018.

[12] T. Rausch, S. Nastic, and S. Dustdar. Emma: Distributed qos-aware mqtt middleware for edge computing applications. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 191–197, April 2018.

[13] H. Shi, N. Chen, and R. Deters. Combining mobile and fog computing: Using coap to link mobile device clouds with fog computing. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 564–571, Dec 2015.

[14] T Taleb, S Dutta, A Ksentini, M Iqbal, and H Flinck. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3), March 2017.

[15] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing*, 3(6):76–83, Nov 2016.