SUPRA, a distributed publish/subscribe protocol with blockchain as a conflict resolver

Jean-Philippe Abegg, Quentin Bramas, Timothée Brugière, Thomas Noël

SUPRA, a distributed publish/subscribe protocol with blockchain as a conflict resolver

Jean-Philippe ABEGG *Transchain, ICUBE, University of Strasbourg, France*

Quentin BRAMAS *ICUBE, University of Strasbourg, France*

Timothée BRUGIÈRE *Transchain, Strasbourg, France*

Thomas NOEL *ICUBE, University of Strasbourg, France*

December 22, 2021

Abstract

Publish/subscribe is a communication paradigm used in distributed applications to easily exchange messages. This paradigm usually has a centralized architecture where a *broker* is responsible for transferring all the messages, hence can be a source of trust issues. In a threat model where the broker can be malicious, two honest entities cannot be sure of the origin of a message. There exist propositions for distributed publish/subscribe protocols replacing the broker by a blockchain. Those propositions all have in common an extensive usage of the blockchain, which makes them expensive over time, due to blockchain fees, and not scalable.

In this paper, we introduce SUPRA, a distributed publish/subscribe protocol. This protocol has the same security guarantees than other solutions relying on blockchains, but where the vast majority of messages are off-chain. The message exchanges are done mostly directly between publishers and subscribers and the blockchain is only used in case of network issues, if a message is lost, or an entity is suspected to be malicious.

blockchain, publish/subscribe protocol, missing messages detection, MQTT

1 Introduction

Publish/subscribe model is a paradigm for communication protocols. This communication model is more scalable and resource-efficient than the request-reply model [2]. There are different kinds of publish/subscribe protocols and we will focus on topic-based publish/subscribe protocols [2]. In such protocols, subscribers declare their interest in data, associated with a given topic, from a publisher, by sending a subscription. Then, when the publisher generates new data with this topic, it sends it to its subscribers. This paradigm allows a unidirectional message flow from the publisher to its subscribers.

These lightweight protocols are well suited for Internet of Things (IoT) applications but have a major drawback: the central entity in the system, the broker. In most existing protocols, such as MQTT, the publisher never sends directly the messages to its subscribers, but to the broker, which forwards the data to the subscribers. The subscribers also send their requests to the broker and not directly to the publishers. This third party between the publisher and the subscribers can create trust issues. This lack of trust makes most of the existing publish/subscribe protocols unable to be used for sensitive data.

In this context, cryptography can help to create verifiable communications and blockchain can be used as a trusted third-party between unknown entities to resolve conflicts.

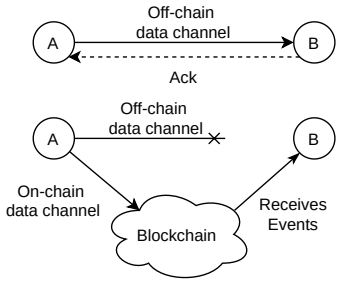

Figure 1: The two modes of communication of our unidirectional on/off chain channel protocol

Related works

Trinity [5, 6] is to our knowledge the first proposition for a publish/subscribe protocol that relies on a blockchain to work. In this solution, brokers use the blockchain to replicate data and store them in an immutable way. When a client publishes data by sending it to its broker, the broker forwards it to a blockchain node. The data eventually appears in a block so that other brokers retrieve this information by reading the blockchain and forward the published data to their local subscribers.

This interaction between a publish/subscribe protocol and a blockchain creates a secured system where we can trust the brokers because each received data can be linked to the entity that published it. However, this approach does not scale for two reasons. First, writing on the blockchain is not free. Each new transaction/data has a cost (transaction fees). With this price, the network can stay alive by paying blockchain nodes for their work. Over time, if a lot of data are published, this kind of solution could become expensive for the brokers or the clients.

Second, data in the blockchain is immutable, so each newly published data increases the size of the blockchain. The more data is sent on the blockchain, the more storage is needed for nodes. It is based on these two reasons that we design a distributed publish/subscribe protocol that uses as few blockchain transactions as possible.

After Trinity, other variants were presented [3], but keeping this extensive use of blockchain for each newly published data.

2 General concepts of SUPRA

2.1 Managers and workers

The publish/subscribe model is composed of three entities: the broker, the publisher, and the subscriber. They create a star-topology where the broker is in the center. As long as the broker is owned by a third party, there is a trust issue with the broker.

In the manager/worker model, there are two kinds of entities: the workers and the managers. The workers generate data, spontaneously, like a sensor, or with an input from another worker, like a server with a request. This input dependency can be translated into topics used by the publish/subscribe protocol. Each worker is connected to a manager and can only send messages to its manager. The manager is in charge of the good behavior of a set of workers, and it handles the security policies in place between it and its workers. The manager is also connected to other managers. The model does not impose properties on the link between the worker and the manager, nor specific protocols on this link. We assume that each worker has total trust in its manager. This assumption is easily achievable if the worker and the manager are owned by the same entity.

Managers can be publishers and subscribers at the same time. The managers do not trust each other and to share data, they will use the hybrid channel presented in the next section.

2.2 Unidirectional Channel with On-Off Chain Proof of Delivery

We present now a new communication protocol, used by SUPRA, but can be of independent interest. The purpose of this protocol is to allow one manager to send messages to another one, with delivery guarantees, and such that each manager obtains proofs for each of those guarantees. The communication protocol is a unidirectional channel, but the goal is similar to the bidirectional channels defined in the Lightning network [4]. We suppose that the managers are reliably connected to the same blockchain.

The publisher manager sends its messages to the subscriber manager using two methods, as illustrated by Figure 1. With the first method, the messages are sent like any other messages on the internet, directly to the subscriber manager, using its IP address, such messages are said to be sent *off-chain* because the blockchain is not used by this method. The receiver has to acknowledge this message. With the second method, the messages are sent on the blockchain, included in a block, and the subscriber manager indirectly receives the message by reading the blockchain. In this case, the messages are said to be sent *on-chain*. With these two methods, the sender is sure that the messages are delivered before a timeout T, because the message is either explicitly acknowledge, or acknowledge by assumption because the managers are connected to the blockchain.

Also, just like the blocks in a blockchain, messages are chained together with signatures. It allows the receiver to detect if messages are missing.

3 SUPRA

SUPRA is a publish/subscribe protocol between managers using the unidirectional channel with on-off chain proof of delivery. A more complete presentation of the protocol can be found in our technical report [1].

Users use the blockchain to declare public keys, and these keys are used to sign the messages. The hybrid channel is set up with a triple handshake between the publisher and the subscriber, where they share the topic name. Once it is set up, the publisher can send data to the subscriber. The publisher is sure that the subscriber will receive the data, because of the proofs of delivery generated with the channel. Since messages are chained together and signed, the subscriber is sure that it has not missed a message, and that the messages are from the publisher.

4 Comparison with existing solutions

SUPRA and Trinity are two publish/subscribe protocols using the blockchain. They resolve the issue with the central broker. Firstly because the message is signed by the sender. Secondly, because missing messages can be detected. Trinity does it by sending all the messages on-chain, and SUPRA by chaining the message together. The major difference between these two protocols is on the blockchain usage. With SUPRA, published data go through the blockchain only if there is a connectivity issue with the off-chain link. It reduces the amount of messages on-chain, and so the cost of the communications because there are less transaction fees. It also impact the delivery delay. Indeed, the off-chain link is faster than the on-chain link because we do not have to wait for the message to be added in a block. The channel makes in general SUPRA deliver data faster than Trinity. In the worst-case scenario, when the off-chain link is lost, SUPRA has the same delay as Trinity, because just like Trinity, every message will go through the blockchain.

5 Conclusion

The blockchain technology can be used to increase the security of the publish/subscribe paradigm. Existing solutions overuse the blockchain, which makes than expensive overtimes. We propose a new technique to use the blockchain to reduce drastically the number of messages on-chain, while keeping the same security properties.

References

[1] Jean-Philippe Abegg, Quentin Bramas, Timothée Brugière, and Thomas Noël. Supra, a distributed publish/subscribe protocol with blockchain as a conflict resolver, 2021.

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

[3] Pin Lv, Licheng Wang, Huijun Zhu, Wenbo Deng, and Lize Gu. An IOT-oriented privacy-preserving publish/subscribe model over blockchains. *IEEE Access*, 7:41309–41314, 2019.

[4] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[5] Gowri Sankar Ramachandran, Kwame-Lante Wright, and Bhaskar Krishnamachari. Trinity: A distributed publish/subscribe broker with blockchain-based immutability. *arXiv preprint arXiv:1807.03110*, 2018.

[6] Gowri Sankar Ramachandran, Kwame Lante Wright, Licheng Zheng, Pavas Navaney, Muhammad Naveed, Bhaskar Krishnamachari, and Jagjit Dhaliwal. Trinity: A byzantine fault-tolerant distributed publish-subscribe system with immutable blockchain-based persistence. *ICBC 2019 - IEEE International Conference on Blockchain and Cryptocurrency*, pages 227–235, 2019.