

БРОКЕР СООБЩЕНИЙ НА БЛОКЧЕЙНЕ: БЕЗОПАСНАЯ ПЕРЕДАЧА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ДВУХУРОВНЕВОЙ ПЛАТФОРМЫ HYPERLEDGER FABRIC

Г. В. Слепенков^{1,}, В. В. Корхов^{1,**}*

¹ Санкт-Петербургский государственный университет, Санкт-Петербург, Россия

Передача данных между организациями по ненадежным публичным сетям сопряжена с рисками потери данных, их искажения и несанкционированного доступа. Для решения этой проблемы представлен прототип брокера сообщений, основанный на Hyperledger Fabric и использующий двухуровневую архитектуру: партиции на PBFT для передачи данных и топика на Raft для управления системой. В блокчейне хранятся только метаданные, а отправляемые сообщения размещаются во внешних хранилищах. Такой подход обеспечивает высокие отказоустойчивость и масштабируемость, превосходящие как традиционные, так и основанные на блокчейне решения.

Data transfer between organizations over untrusted public networks is associated with risks of data loss, corruption, and unauthorized access. To address this problem, we present a message broker prototype based on Hyperledger Fabric using a two-layer architecture: PBFT partitions for data transfer and Raft topics for system management. Only metadata is stored in the blockchain, while messages are placed in external storages. This approach provides higher fault tolerance and scalability compared to existing traditional and blockchain-based solutions.

PACS: 89.20.Ff

ВВЕДЕНИЕ

Процесс безопасной передачи данных (Secure Data Transer, SDT) между различными организациями требует наличия проверок целостности передаваемых данных и их подлинности, поскольку осуществляется через ненадежные публичные сети. Имеющиеся решения, позволяющие осуществить безопасную передачу данных, либо опираются на централизованные системы, что приводит к появлению единой точки отказа и предоставлению полного контроля за системой одной из сторон передачи данных, либо используют блокчейн, но сталкиваются с ограничениями по масштабируемости и размеру передаваемых данных.

* E-mail: gslepenkov@gmail.com

** E-mail: v.korkhov@spbu.ru

В нашей предыдущей работе [1] была предложена архитектура брокера сообщений на базе блокчейна с двухуровневой структурой. В данной работе представлен прототип подобной системы, реализованный на платформе Hyperledger Fabric [2]. В разд. 1 описана архитектура прототипа, в разд. 2 представлены детали реализации и используемые технологии, а в заключении подведены итоги и обозначены направления дальнейших исследований.

1. АРХИТЕКТУРА ПРОТОТИПА

Как показано в работе [1], архитектура прототипа основана на двухуровневом блокчейне. В случае Hyperledger Fabric для построения такой архитектуры использован механизм каналов. Партиции реализованы как небольшие (до 7 узлов) каналы с алгоритмом консенсуса PBFT [3]. Все узлы партиционных каналов затем добавляются в более крупный канал с алгоритмом Raft [4], называемый *топиком*.

Поверх описанной двухуровневой сети работают два смарт-контракта. Смарт-контракт SDT функционирует на партиционных каналах и отвечает непосредственно за передачу данных. Он предоставляет API для публикации и потребления сообщений и содержит механизмы проверки сообщений и подтверждения доставки. Второй смарт-контракт работает на канале-топике и в настоящее время применяется для задач управления системой.

Для взаимодействия с блокчейном используется набор микросервисов на Java, называемых *коннекторами*. Коннектор выступает посредником между Hyperledger Fabric, приватными хранилищами и конечными пользователями SDT. Коннекторы предоставляют традиционный REST API для конечных пользователей, инкапсулируя для них все взаимодействия с блокчейном и приватными хранилищами данных.

На рис. 1 показана общая структура прототипа, включающая основные компоненты и их взаимосвязи.

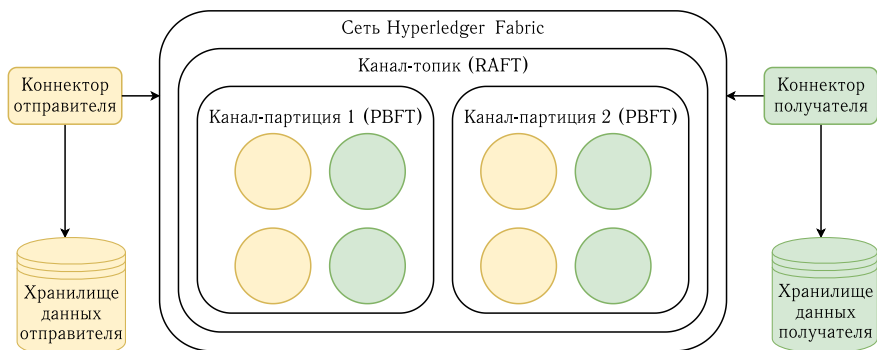


Рис. 1. Пример структуры прототипа с одним топиком и двумя партициями

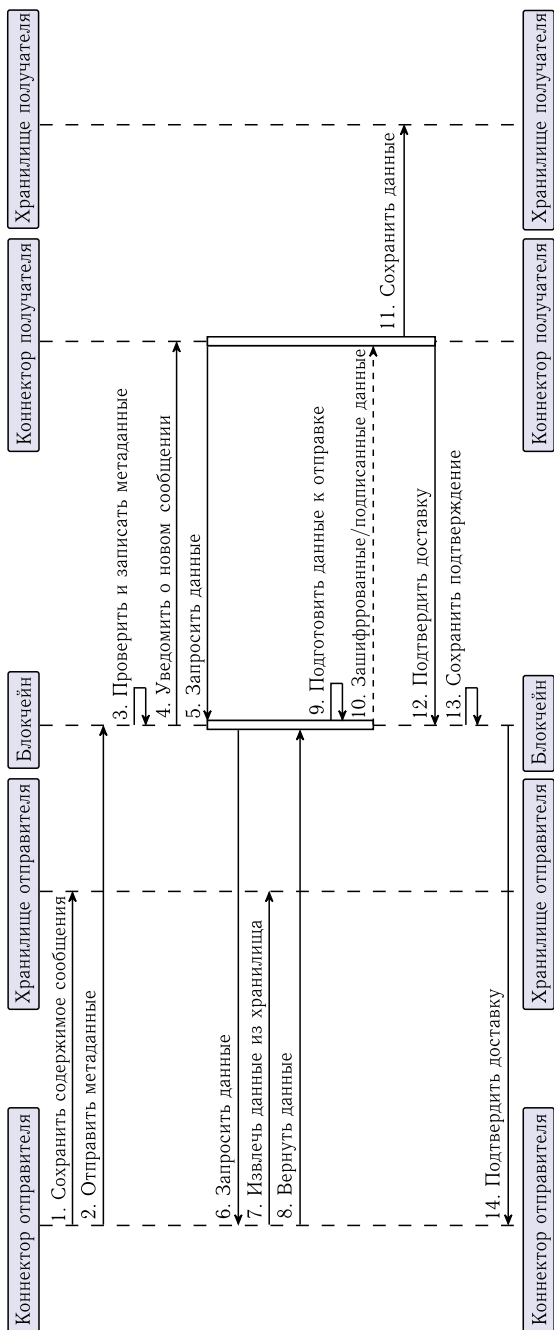


Рис. 2. Схема процесса безопасной передачи данных

Последовательность обмена данными при типичном безопасном переносе показана на рис. 2. Диаграмма описывает следующую последовательность взаимодействий между отправителем, получателем, блокчейном и компонентами хранилища:

- отправитель инициирует процесс, отправляя сообщение через приложение-коннектор, принадлежащее организации отправителя;
- коннектор сохраняет содержимое сообщения в приватном хранилище и отправляет метаданные в блокчейн;
- блокчейн проверяет метаданные и в случае успеха записывает их в реестр и порождает событие о новом сообщении для получателя данных;
- коннектор получателя принимает событие и инициирует процесс получения данных через специальный метод смарт-контракта, который запрашивает данные у коннектора организации-отправителя;
- когда блокчейн получает данные от отправителя, он проверяет их с помощью ранее сохраненных метаданных и шифрует, подписывает данные для их передачи получателю;
- после приема данных коннектор получателя валидирует их и сохраняет в хранилище данных организации-получателя;
- заключительным шагом является подтверждение завершения передачи данных.

2. ДЕТАЛИ РЕАЛИЗАЦИИ

Все компоненты прототипа реализованы на языке Java с использованием официальных SDK (Software Development Kit) Hyperledger Fabric. Каждый смарт-контракт имеет отдельный protobuf-базируемый API и ядро, полностью независимое от Fabric SDK, что позволяет отделить логику передачи данных от деталей реализации блокчейна и при необходимости перенести реализацию на другую платформу. Кроме того, у коннекторов имеется отдельный protobuf-базируемый API, используемый ядром смарт-контракта, что повышает гибкость системы и дает возможность реализовать коннекторы на других технологиях при наличии совместимого API смарт-контракта.

Хранилища в прототипе реализованы с использованием PostgreSQL и MinIO, чтобы продемонстрировать гибкость архитектуры.

Исходный код прототипа доступен в репозитории GitHub [5].

ЗАКЛЮЧЕНИЕ

В работе представлен прототип брокера сообщений на основе блокчейна, реализованный на платформе Hyperledger Fabric. Прототип использует двухуровневую архитектуру с малыми каналами для передачи данных и большими каналами для управления системой, что повышает масштабируемость, отказоустойчивость и характеристики производительности. Все компоненты прототипа отделены от деталей реализации

блокчейна, что обеспечивает возможность переноса на другие платформы в будущем.

Параметры прототипа, такие как пропускная способность, задержки и другие показатели производительности, в настоящее время исследуются и будут представлены в будущей работе. Важным направлением дальнейших исследований является завершение разработки дополнительной функциональности, описанной в работе [1], такой как асинхронная репликация данных и возможности, основанные на методах ИИ.

СПИСОК ЛИТЕРАТУРЫ

1. *Slepenkov G., Korkhov V.* Towards Secure Cross-Organizational Data Transfer: A Blockchain-Enabled Message Broker Approach // Proc. Conf. "Computational Science and Its Applications" (ICCSA 2025). Cham: Springer Nature Switzerland, 2026. P. 115–130; https://doi.org/10.1007/978-3-031-97589-9_9.
2. *Androulaki E. et al.* Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains // Proc. Thirteenth EuroSys Conf. "EuroSys '18". New York: ACM, 2018. 15 p.; <https://doi.org/10.1145/3190508.3190538>.
3. *Barger A. et al.* A Byzantine Fault-Tolerant Consensus Library for Hyperledger Fabric // 2021 IEEE Intern. Conf. on Blockchain and Cryptocurrency (ICBC). IEEE. 2021. <https://doi.org/10.1109/ICBC48261.2021.9459547>.
4. *Ongaro D., Ousterhout J.* In Search of an Understandable Consensus Algorithm // 2014 USENIX Ann. Tech. Conf. (USENIX ATC 14). 2014. P. 305–319; <https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro>.
5. *Slepenkov G.* Hyperledger Fabric-Based Implementation of Message Broker for Secure Data Transfer. <https://github.com/SkyWaet/secure-data-transfer-fabric>. 2025. GitHub repository.