Connect - Blockchain and Self-Sovereign Identity Empowered Contact Tracing Platform

Eranga Bandara[1], Xueping Liang[2], Peter Foytik[1], Sachin Shetty[1], Crissie Hall[4], Daniel Bowden[4], Nalin Ranasinghe[3], Kasun De Zoysa[3], and Wee Keong Ng[5]

[1] Old Dominion University, Virginia, USA
{cmedawer, pfoytik, sshetty}@odu.edu
[2] University of North Carolina at Greensboro, North Carolina, USA
{x_liang}@uncg.edu
[3] University of Colombo School of Computing, Sri Lanka
{dnr, kasun}@ucsc.cmb.ac.lk
[4] Sentara Healthcare, Norfolk, VA, USA
{cehallre, dsbowden}@sentara.com
[5] School of Computer Science and Engineering
Nanyang Technological University, Singapore
{awkng}@ntu.edu.sg

Abstract. The COVID-19 pandemic in 2020 has resulted in increased fatality rates across the world and has stretched the resources in healthcare facilities. There have been several proposed efforts to contain the spread of the virus among humans. Some of these efforts involve appropriate social distancing in public places, monitoring and tracking temperature at the point of access, etc. In order for us to get back to the "new normal", there is a need for automated and efficient human contact tracing that would be non-intrusive and effective in containing the spread of the virus. In this paper, we have developed "Connect", a Blockchain and Self-Sovereign Identity (SSI) based digital contact tracing platform. "Connect" will provide an automated mechanism to notify people in their immediate proximity of an occurrence of a positive case and would reduce the rate at which the infection could spread. The platform's self-sovereign identity capability will ensure no attribution to a user and the user will be empowered to share information. The ability to notify in a privacy-preserving fashion would provide businesses to put in place dynamic and localized data-driven mitigation response. "Connect's" SSI based identity wallet platform encodes user's digital identities and activity trace data on a permissioned blockchain platform and verified using SSI proofs. The user activities will provide information, such as places travelled, travel and dispatch updates from the airport etc. The activity trace records can be leveraged to identify suspected patients and notify the local community in real-time. Simulation results demonstrate transaction scalability and demonstrate the effectiveness of "Connect" in realizing data immutability and traceability.

Keywords: Blockchain ; COVID-19 ; Contract Tracing ; E-Health ; Big Data

1 Introduction

The COVID-19 pandemic has challenged countries to invest in resources to control the spread of the virus. The number of positive COVID-19 cases have been rising all over the world, with the majority of the confirmed cases found in the U.S. The reasons for the rapid spread through humans have been attributed to symptomatic, pre-symptomatic and asymptomatic cases [30]. The current approaches to limit the spread of the virus includes various methods to enforce safe social distancing and limiting air travel. Though these approaches have value, there is a need for a platform that can alert the presence of a positive case to a regional community in a timely and privacy-preserving fashion.

Organizations are working on “safe back to work” policies to realize a “new normal working environment”, that would provide a data-driven mitigation response to limit the spread of the virus. These “back to work” policies are not just limited to COVID-19 and will also be effective against any infectious disease. It has been acknowledged that an effective means to limit the spread of the virus is to continuously track user activities at various points of visit and access [15] and notify potential cases or exposure to local and regional communities. The resultant data-driven insights would help organizations to operate safely and people to access public spaces.

However, on the flip side, current approaches that provide capabilities to address the aforementioned need are plagued with data centralization, privacy concerns and location tracking concerns. The centralization of user data in a cloud environment can be vulnerable to adversarial attacks. The privacy concern for both potential patient and the people they could come in contact should be preserved [2].

In this paper, we propose “Connect”, a “back to work safely” system based on a blockchain and self-sovereign identity (SSI) empowered digital contract tracing platform. The platform keeps employees’ digital identities and events related to testing/symptoms on a blockchain platform using SSI [27] proofs. The employer can use a mobile app to self-report the requested information that uses SSI to record results anonymously and without location tracking. The employer can use the back end analytics to monitor workplace conditions and use a data-driven approach to inform workplace safety policies and guidelines. The main contributions of “Connect” are as follows.

1. Blockchain and SSI empowered digital contact tracing platform to realize decentralized and privacy-preserving digital contact tracing
2. SSI based identity wallet to capture/verify the user identity proofs and activity trace record proofs.
3. Store user identity data and activity trace record data on blockchain platforms by using self-sovereign identity proofs.
4. Self-sovereign identity proof-based identity and activity trace storage address the common issues in cloud-based data storages(e.g lack of data privacy, lack of data immutability, lack of traceability, lack of data provenance [25, 35]).

The rest of the paper is organized as follows. Section 2 discusses the architecture of the Connect platform. Section 3 implementation details of the Connect platform. Section 4 performance evaluation, Section 5 surveys related work. Section 6 concludes the Connect platform with suggestions for future work.

2 Connect Platform

2.1 Overview

Connect is a blockchain, self-sovereign identity-based user identity, and activity tracking platform. It can be used to track the activity of COVID-19 suspected patients during a quarantine process. The Connect platform is built using a layered architecture shown in Figure 1 containing four main layers.

1. Distributed ledger - Where all user cryptographic artifacts for identity (DIDs) and proofs of activity are stored.
2. DID communication layer - Where peer to peer data exchange between user identity wallets happens within the DID communication layer.
3. Credential layer - Where different entities in the platform(users, admins) create and exchange credentials for verification via credential layer.
4. Activity trace layer - Where user activity trace recording and verification happens.

Distributed ledger is the blockchain-based peer to peer storage system used in the Connect platform. The blockchain can be deployed among multiple organizations such as government organizations, hospitals, airport/port customer offices, banks, identity authorities etc. Each organization in the network can run its own blockchain node connected as a ring cluster, Figure 8. It stores all user digital identity proofs (which are identified as DID or decentralized identity proof [6]) and user activity trace record proofs on Connect platform.

The DID communication layer is used to exchange the actual credential information(such as user image, id numbers, etc) between the credential approvers/versifiers(admins) mobile wallet and the credential owners(users) mobile wallets. Peer to peer data exchange between user identity wallets happens in this layer. When a user's identity needs to be verified/approved, the admin requests proof of identity from the holder, the holder consents and shares data along with cryptographic proof stored on the blockchain. The Connect mobile app fetches the identity information stored in local storage to send to the admins Trace mobile wallet. The admin can do further verification/approvals based on this information.

There are two main types of entities(users) in the connect platform, credential owners, credential verifiers(admins). Connect provides a self-sovereign identity based mobile wallet application for each type of user. Credential owners use "Connect mobile wallet" and admins use "Trace mobile wallet". Credential owners register their DID proofs on blockchain and enroll in the Connect platform with the Connect mobile application. Admins verify credentials(DID

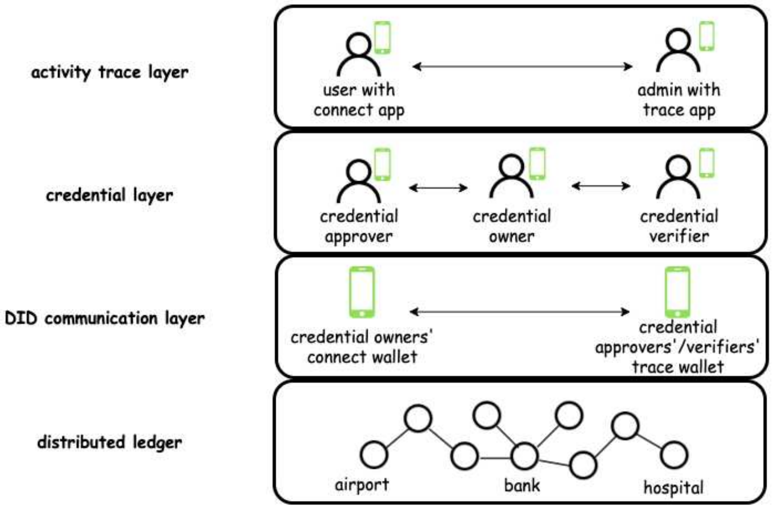


Fig. 1: Connect platform layered architecture. Distributed ledger used to store DIDs. Peer to peer data exchange between user identity wallets happens in the DID communication layer. Credential create, verification happens in Credential layer. User activity trace recording and verification happens in the activity trace layer.

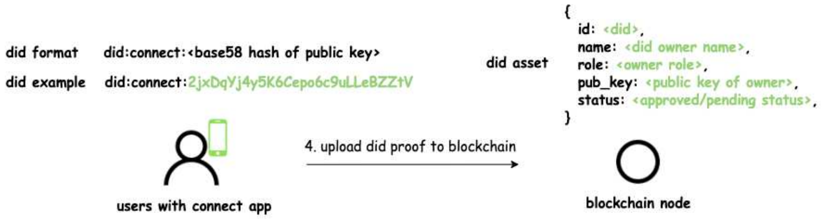
proofs) via Trace mobile wallet. The credential exchange process happens in the Credential layer, where credential owners and admins exchange the credentials for verification.

All user activity trace data in the Connect platform are stored in the blockchain ledger based on an SSI approach. When a user goes to a specific place(e.g airport, bank, hospital, office) the admin officers there can verify the identity of the user and create an activity trace record for the user on the blockchain. This identity verification and activity trace data creation process is done via Trace mobile wallet application given to the admin officers. Admins also can fetch user activity trace records which are stored in the blockchain when consent is given, verify them, and view through the Trace mobile application. Trace mobile app comes with a QR code scan-based identity and activity trace data verification process. All activity trace data is handled with functions(activity trace data creation, activity trace data verification) implemented in the Activity trace layer.

2.2 Functionality

Consider a scenario where a blockchain network is deployed at the Airport, Hospital network, Government Bank and Identity office. The admin officers at each organization installed the Trace mobile app. A user who comes from overseas installed the Connect mobile wallet and registered on it before entering the airport.

As shown in Figure 3, when registering it first captures basic user information with Id no/Passport no. After that, it asks users to capture their photo and put a signature on top of the photo. This information can be used as additional proof which administrators can use to approve/verify the user identity. The captured information will be saved in secure storage in a mobile application and the proof of this information will be uploaded to the blockchain as self-sovereign identity proof(DID proof). When uploading credentials, the app will generate a public/private key pair which corresponds with the user/mobile wallet. The private key will be saved on the Keystore on the mobile application. The public key and base58 [16,28] hash of the public key will be uploaded to the blockchain along with other DID proof information. The base58 hash of the public key will be used as the digital identity(DID) of the user on the Connect platform. Figure 2 shows the format of the DID proof on the connect platform. This DID will be embedded to QR code in the mobile app, which the user can show to admin officers (e.g admin at hospitals, custom officers at the airport, banks officers) for verification, Figure 3.

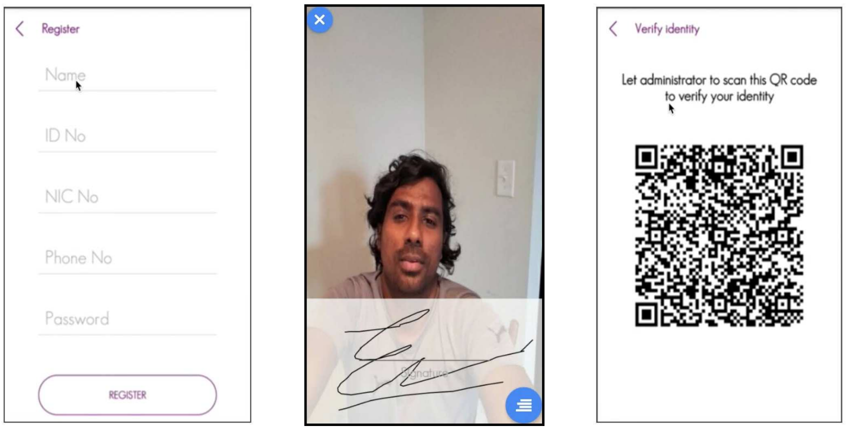


```
did format     did:connect:<base58 hash of public key>
did example    did:connect:2jxDqYj4y5K6Cepo6c9uLLeBZZtV
```

```
did asset  {
             id: <did>,
             name: <did owner name>,
             role: <owner role>,
             pub_key: <public key of owner>,
             status: <approved/pending status>,
           }
```

Fig. 2: DID format which generated in Connect app and DID asset format which stored in blockchain.

Assume a user comes from an overseas country and installed the Connect wallet on his/her mobile phone. When the user comes to the airport he/she needs to show the QR code identity which is embedded in the Connect mobile wallet to the admin officer(e.g customer officer) at the airport in order to have their digital identity issued, Figure 3. The officer will scan the QR code via Trace application and fetch the user identity proofs which are saved in the blockchain. After that, it requests for consent to specific data and connects to users through the "Connect mobile wallet" application. This process is achieved via push notification(DID communication layer) in order to fetch the actual user identity information(e.g Id numbers, photo, signature) to the Trace mobile app, Figure 4. Then the admin could check the information against the passport/id card of the user. If the data is correct according to the passport/id card, the admin approves the identity of the user. When approving, it updates the status of users' digital identity in the blockchain. This is the first-time vetting process which needs to be done in order to approve the user identity saved in the blockchain is authentic and verified by a trusted source. Once identity is approved by an authorized administration user



Let administrator to scan this QR code to verify your identity

(a) Add identity information

(b) Capture photo and signature

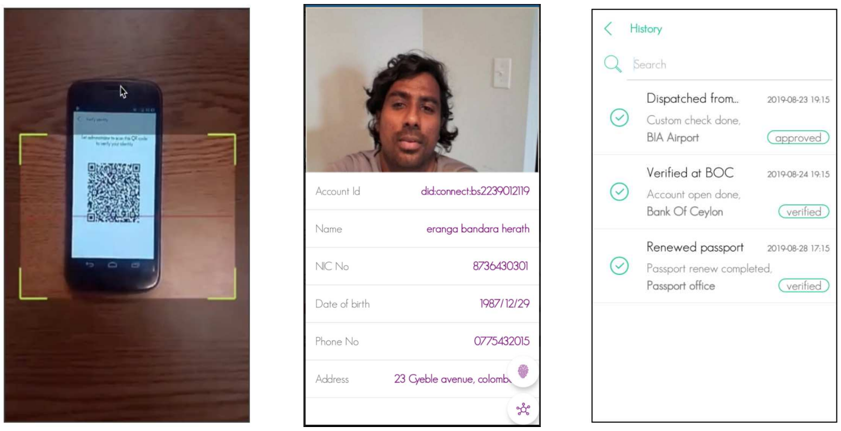
(c) Connect identity in QR code

Fig. 3: Connect mobile wallet application. It will embed users' digital identity on QR code.

can use his/her identity wallet in any other place to prove his/her identity (ex in a bank, hospital etc). When approving the identity, it will use the Identity smart contact. After identity approved blockchain will create an activity trace record (along with user digital identity/DID, date/time and location) by using Trace smart contract. This activity trace record specifies the user is dispatched from the airport. Once the activity trace record is created in the blockchain node at the airport, it will be available to other blockchain nodes at hospitals and banks.

For example, assume the user goes to a bank a few days after he/she enters the country. User needs to show his/her identity wallet QR code in order to prove identity at the bank. Then the admin at the bank scans the QR code, fetches the identity proof of the blockchain and verifies the user. At the end of this process, blockchain will save another activity trace record which mentions that the user came to the bank with date/time and location, Figure 5. In this way, the connect platform traces all the user activities as self-sovereign identity proofs (or proof of location). Now assume the user goes to the hospital for some various treatment. The user shows the identity wallet with QR code, then an officer at the hospital scans it and fetches the user identity proof with all user activity record proofs from the blockchain. The activity trace contains an activity that mentions the user came from a foreign country and dispatched from the airport on a specific date, Figure 4. With this information, it can be easily identified if a person could be suspected of Covid-19. Further precautions can be taken before spreading the virus to more people.

By recording an activity trace of users, Connect platform can support in identifying spread from three main transmission methods of Covid-19 virus, Symptomatic transmission(direct transmission from an asymptomatic individid-



Account Id did:connect:bs2239012119

Name eranga bandara herath

NIC No 8736430301

Date of birth 1987/12/29

Phone No 0775432015

Address 23 Gyeble avenue, colomb

History

Search

Dispatched from... 2019-08-23 19:15
Custom check done,
BIA Airport approved

Verified at BOC 2019-08-24 19:15
Account open done,
Bank Of Ceylon verified

Renewed passport 2019-08-28 17:15
Passport renew completed,
Passport office verified

(a) Scan user identity (b) View user identity (c) View activity trace

Fig. 4: Trace mobile application. It can view users' identity information and activity trace.

ual), Pre-symptomatic transmission(direct transmission from an individual that occurs before the source individual experiences noticeable symptoms), Asymptomatic transmission(direct transmission from individuals who never experience noticeable symptoms).

2.3 Contact Tracing

The users in the Connect platform can be notified via the peer to peer notification system. These notifications can be used to notify the users who are at risk of getting infected with Covid-19 virus. For example, assume a user who has registered in the Connect platform is diagnosed as a Covid-19 infected person. The medical officer at the hospital can report the patient to the Connect platform via Trace mobile application. Additionally, the Connect mobile wallet provides a feature to self-report the diagnosis of the users. This diagnosis information will be uploaded and stored in the blockchain. Once Covid-19 case is reported, the Connect platform can identify all places where the patient has visited(during the last 14 days) by using the activity trace data in the blockchain. The activity trace data contains information about user identity(DID), times and location(latitude, longitude) the diagnosed person has visited. Based on these activity trace information, it can identify the other users who have been in these places at the same time with Covid-19 infected person without revealing personal information. Then Connect platform can send the notifications to these users mentioning there is a risk of contact with Covid-19 since they have been in a place where Covid-19 infected person visited, Figure 6. With the notification function, the user can be aware of the risks in time and take some actions accordingly.

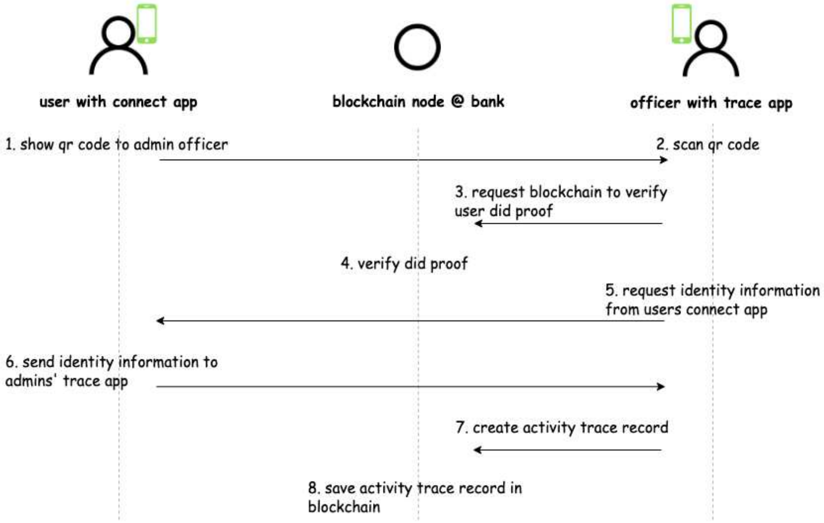

user with connect app
blockchain node @ bank
officer with trace app
1. show qr code to admin officer
2. scan qr code
3. request blockchain to verify user did proof
4. verify did proof
5. request identity information from users connect app
6. send identity information to admins' trace app
7. create activity trace record
8. save activity trace record in blockchain

Fig. 5: Admin officer scans QR code identity of the user and creates activity trace record in the blockchain. The trace record contains the information about user identity(DID), time and location(latitude, longitude)

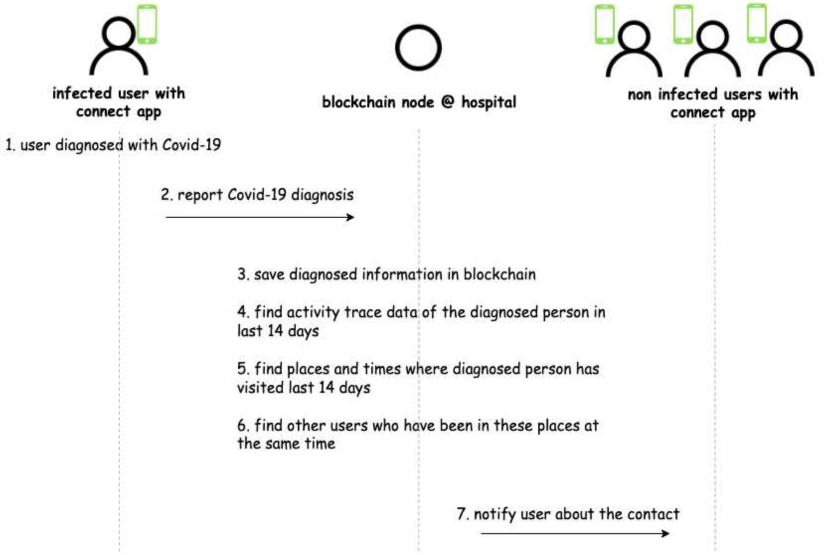
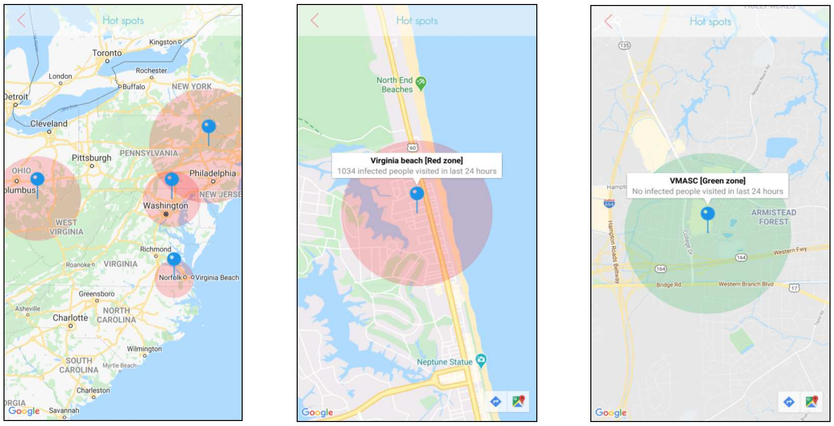

infected user with connect app
blockchain node @ hospital
non infected users with connect app
1. user diagnosed with Covid-19
2. report Covid-19 diagnosis
3. save diagnosed information in blockchain
4. find activity trace data of the diagnosed person in last 14 days
5. find places and times where diagnosed person has visited last 14 days
6. find other users who have been in these places at the same time
7. notify user about the contact

Fig. 6: Covid-19 contact tracing. Once a user is identified as Covid-19 infected person, Connect platform finds the other users who have contacted that person and notify them.

Based on the activity trace data of the Covid-19 diagnosed people, Connect platform can identify Covid-19 hot spots. These hot spots information will be shown in a Map view(Figure 7(a)) on Connect mobile wallet application. The users can search for a specific location in the map and see the critical level of that place, red zone(Figure 7(b)) or green zone(Figure 7(c)). Hot spot critical level is decided based on the infected people count visited that place recently. These location data traced with the activity trace records of the Covid-19 diagnosed people in the Connect platform.



Virginia beach [Red zone]
1034 infected people visited in last 24 hours

VMASC [Green zone]
No infected people visited in last 24 hours

(a) Hot spots (b) Red zone (c) Green zone

Fig. 7: View hot spot information. Hot spot critical level decided based on the infected people count visited to the place. These location data are traced with the activity trace records of the Covid-19 diagnosed people in Connect platform.

3 Connect Implementation

We have built the production version of the Connect platform with the collaboration of Sentara hospital chain USA [32]. The Connect platform has been built using microservices architecture [33] to support high scalability and high transaction load. All the services in the Connect platform are implemented as small services(micro-services) with the single responsibility principle. These services are dockerized [26] and deployed using Kubernetes [9] container orchestration system. To cope with high transaction load and back-pressure [14] operations we have adopted reactive streams based approach with using Akka streams [12]. All the microservice communications are handled via Apache Kafka [20,23] message broker. We run 3 Kafka broker nodes with 3 Zookeeper nodes in Connect. The platform is running as a permissioned blockchain system in a private cloud. Figure 8 shows the architecture of the Connect platform.

Rahasak blockchain has been used to implement the functionalities of the Connect platform. Rahasak blockchain [7] comes with concurrency enabled Aplos smart contacts [8] which are written with Scala [1, 29] and Akka actor-based [4] concurrency handling [18, 19]. All the functionalities of blockchain implemented with Aplos smart contracts. There are four main smart contracts a) identity contract, b) asset contract c) notification contract d) verification contract. "Connect" and 'Trace" mobile wallets are the client applications on the Connect platform. The functions which are implemented in the blockchain smart contracts will be invoked by Mobile clients. The requests generated from Mobile apps will be directed to blockchain smart contracts via Connect gateway service which is HTTPS REST API [3] built with Golang [31]. There is a peer to peer communication channel between "Connect" and "Trace" mobile wallets(to exchange the credential data). Firebase push notification service [22] has been used to implement the peer to peer communication between mobile wallets. Client authentication/authorization will be handled by JWT-based [21] auth service in the Connect platform. Client credential information will be stored in auth-storage(database) in the auth service.

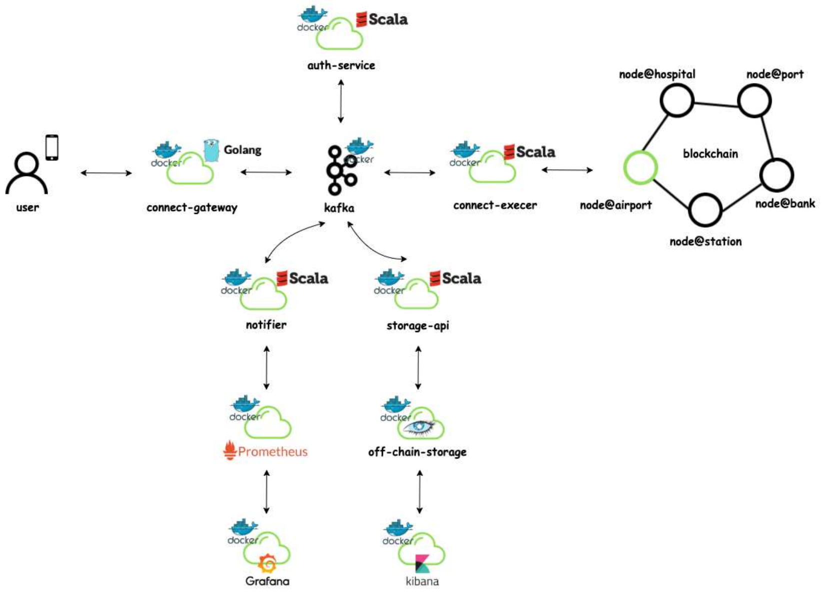


Fig. 8: Connect platform's microservices based architecture. All the services are dockerized and available to deploy with kubernetes.

4 Performance Evaluation

Performance evaluation of Connect was completed and is discussed. To obtain the results, we deployed the Connect platform with multi peer Rahasak blockchain cluster in AWS 2xlarge instances(16GB RAM and 8 CPUs). Rahasak blockchain runs with 4 Kafka nodes, 3 Zookeeper nodes and Apache Cassandra [24] as the state database. The smart contracts on the Rahasak blockchain implemented with Scala functional programming and Akka actor based Aplos [8] smart contract platform. The evaluation results are obtained for the following, with a varying number of blockchain peers (1 to 5 peers) used in different evaluations.

1. Transaction throughput
2. Transaction execution and validate time
3. Transaction scalability
4. Transaction execution rate
5. Block generate time

4.1 Transaction Throughput

For this evaluation, we recorded the number of DID proof create transactions and DID proof query transactions that can be executed in each peer in the Connect platform. When creating a DID, an invoke transaction will be executed in the underlying blockchain. Invoke transaction creates a record in the ledger and updates the status of the assets in the blockchain. Query transaction searches the status of the underlying blockchain ledger. They neither create transactions in the ledger nor update the asset status. We flooded concurrent transactions for each peer and recorded the number of completed results. As shown in Figure 9 we have obtained consistent throughput in each peer on the Connect platform. Since queries are not updating the ledger status, it has high throughput(2 times) compared to invoke transactions.

4.2 Transaction Execution and Validation Time

In this evaluation, we evaluated the transaction execution and transaction validation time. We recorded time to execute and validate different sets of transactions(100, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 transactions). Transaction validation time includes the double-spend checking time. Transaction execution time includes the double-spend checking time, ledger update time, data replication time. Figure 10 shows how transaction execution time and validation time varies in different transaction sets.

4.3 Transaction Scalability

For this evaluation, we recorded the number of transactions that can be executed (per second) over the number of peers in the network. We flooded concurrent

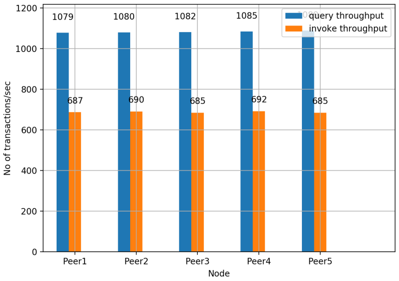


Fig. 9: Invoke transaction throughput and query transaction throughput of Connect blockchain.

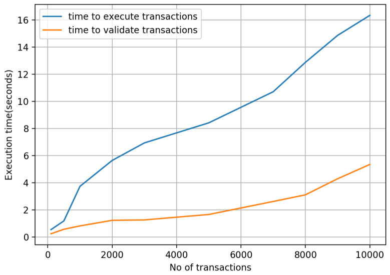


Fig. 10: Time to execute transactions and validate transactions in the Connect platform.

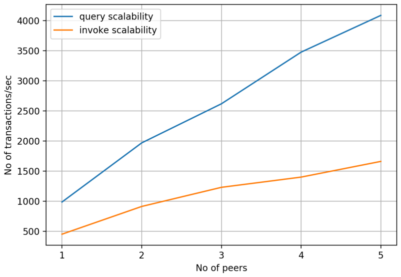


Fig. 11: Transaction scalability of Connect blockchain.

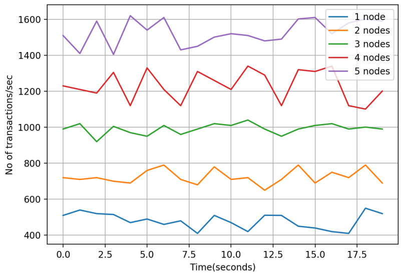


Fig. 12: Transaction execution rate with no of peers in the Connect blockchain.

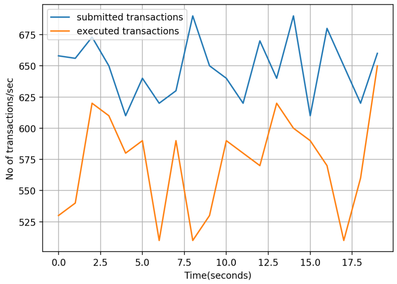


Fig. 13: Transaction execution rate and transaction submission rate in a single blockchain peer.

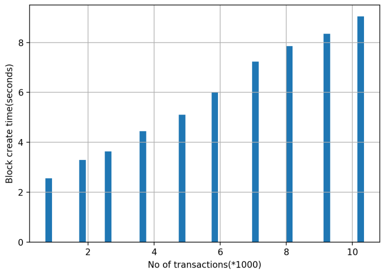


Fig. 14: Block creation time against the no of transactions in the block.

transactions in each peer and recorded the number of executed transactions. Figure 11 shows transaction scalability results. When adding a node to the cluster, it nearly linearly increases the transaction throughput. Query transactions have high scalability when comparing to invoke transactions. The main reason is question transactions are not updating the ledger status like invoke transactions.

4.4 Transaction Execution Rate

Next, we evaluate the transaction execution rate in the Connect platform. We tested the number of submitted transactions and executed transactions in different blockchain peers recording the time. Figure 12 shows how transaction execution rate varies when having a different number of blockchain peers in the Connect platform. When the number of peers increases, the rate of executed transactions is increased relatively. Figure 13 shows the number of executed transactions and submitted transactions in a single blockchain peer. There is a back pressure operation [14] between the rates of submitted transactions and executed transactions. We have used a reactive streaming-based approach with Apache Kafka to handle these backpressure operations in the Connect platform.

4.5 Block Generate Time

Finally, we have evaluated the time taken to create blocks in the underlying blockchain storage of the Connect platform. The statistics recorded against the no of transactions in a block. Block generate time depends on a). data replication time b). Merkel proof/block hash generate time c). transaction validation time. When the transaction count increases in the block, these factors will be increased. Due to this reason, when the transaction count increases, block generation time also increases correspondingly. As shown in Figure 14 to create a block when having a 10k transaction, it takes 8 seconds.

5 Related Work

There are some research works which have been conducted to find contract tracing technologies to control Covid-19 outbreak [5, 11, 34]. In this section, we outline the main features and architecture of these research works.

TraceTogether [34] is a mobile application-based platform to detect potential Covid-19 virus carriers in Singapore. It works by exchanging short distance Bluetooth signals with other users of the app, giving officials a database to track potential Covid-19 carriers. If a user is diagnosed with Covid-19, the respiratory illness caused by the coronavirus, they could allow Singapore's health ministry to access their app data to identify people who had close contact with the infected individual. Then the app alerts those who come in contact with someone who has tested positive or is at high risk for carrying the coronavirus.

Google/Apple Contact Trace [17] Google and Apple recently announced a joint initiative to build a contact tracing application to help contain the Covid-19 spread. They will be launching a comprehensive solution that includes application programming interfaces (APIs) and operating system-level technology to assist in enabling contact tracing. Their system uses Bluetooth, a standard way for most mobile devices to communicate with each other. Apple and Google stressed that their system preserves users' privacy. Consent is required and location data is not collected. The technology also won't notify users who they came into contact with, or where that happened.

WeTrace [13] is a fully privacy-preserving approach and application, which built on top of BTE(Bluetooth Low Energy). This solution meets major GDPR (General Data Protection Regulation) requirements, which are in force in certain European countries. WeTrace here fulfils exactly this key requirement on privacy-preserving for arbitrary mobile devices, being able to communicate via BTE and being used by their owners in a once-used, once-associated manner. The application of low-range BTE communications determines a highly suitable coincidence between the COVID-19 "social distancing" requirements and the communications technology.

COVID Credentials Initiative(CCI) [10] is a collaboration of more than 60 organizations working to deploy self-sovereign identity(SSI) based verifiable credential solutions to help stop the spread of COVID-19. The goal of CCI is to build an "immunity passport", which is a digital certificate that lets individuals prove (and request proof from others) that they have recovered after testing negative, have tested positive for antibodies, or have received a vaccination once one is available. These digital certificates would be issued by health care institutions but controlled by the user and shared in a peer-to-peer manner. The CCI group includes individuals who are part of Evernym, ID2020, uPort, Dutch research organization TNO, Microsoft, ConsenSys Health and consultants Luxoft.

The comparison summary of these platforms and the Connect platform is presented in Table 1. It compares Architecture(Centralized/Decentralized), Running blockchain, Supported credential types(e.g biometric), SSI support, Activity trace support, Privacy level details.

Table 1: Self-sovereign identity and activity trace tracking platform comparison

Platform	Architecture	Running Blockchain	Credential Type	SSI Support	Activity Trace Support	Privacy Level
Connect	Decentralized	Rahasak	Any	Yes	Yes	High
TraceTogether	Centralized	N/A	Any	No	Yes	Low
Google and Apple Contact Trace	Centralized	N/A	N/A	No	Yes	Mid
WeTrace	Centralized	N/A	N/A	No	Yes	High
CCI	Decentralized	Sovrin	Medical	Yes	No	High

6 Conclusions and Future Work

In this paper, we have presented "Connect", a Blockchain and SSI empowered digital contract tracing platform that can leverage the information on positive cases and let people in the immediate proximity be notified, which would thereby reduce the rate at which the infection could spread. This would particularly be effective if sufficient people use the platform and benefit from the targeted recommendations. The recommendations would be made in a privacy-preserving fashion and contain the spread of the virus without the need for an extended period of lockdown. We have developed a prototype for the proposed platform and conducted simulations to evaluate scalability and transaction throughput.

Acknowledgements

This work was funded by the Department of Energy (DOE) Office of Fossil Energy (FE) (Federal Grant #DE-FE0031744).

References

1. The scala programming language, `https://www.scala-lang.org/`
2. Abeler, J., Bäcker, M., Buermeyer, U., Zillessen, H.: Covid-19 contact tracing and data protection can go together. JMIR mHealth and uHealth **8**(4), e19359 (2020)
3. Adamczyk, P., Smith, P.H., Johnson, R.E., Hafiz, M.: Rest and web services: In theory and in practice. In: REST: from research to practice, pp. 35–57. Springer (2011)
4. Akka: Akka documentation, `https://doc.akka.io/docs/akka/2.5/actors.html`
5. Allam, Z., Jones, D.S.: On the coronavirus (covid-19) outbreak and the smart city network: Universal data sharing standards coupled with artificial intelligence (ai) to benefit urban health monitoring and management. In: Healthcare. vol. 8, p. 46. Multidisciplinary Digital Publishing Institute (2020)
6. Baars, D.: Towards self-sovereign identity using blockchain technology. Master's thesis, University of Twente (2016)
7. Bandara, E., NG, W.K., DE Zoysa, K., Fernando, N., Tharaka, S., Maurakirinathan, P., Jayasuriya, N.: Mystiko—blockchain meets big data. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 3024–3032. IEEE (2018)
8. Bandara, E., NG, W.K., De Zoysa, K., Ranasinghe, N.: Aplos: Smart contracts made smart. BlockSys'2019 (2019)
9. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., Wilkes, J.: Borg, omega, and kubernetes. Queue **14**(1), 70–93 (2016)
10. CCI: Cci, `https://www.covidcreds.com/`
11. Cho, H., Ippolito, D., Yu, Y.W.: Contact tracing mobile apps for covid-19: Privacy considerations and related trade-offs. arXiv preprint arXiv:2003.11511 (2020)
12. Davis, A.L.: Akka streams. In: Reactive Streams in Java, pp. 57–70. Springer (2019)
13. De Carli, A., Franco, M., Gassmann, A., Killer, C., Rodrigues, B., Scheid, E., Schoenbaechler, D., Stiller, B.: Wetrace–a privacy-preserving mobile covid-19 tracing approach and application. arXiv preprint arXiv:2004.08812 (2020)
14. Destounis, A., Paschos, G.S., Koutsopoulos, I.: Streaming big data meets backpressure in distributed network computation. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. pp. 1–9. IEEE (2016)
15. Ferretti, L., Wymant, C., Kendall, M., Zhao, L., Nurtay, A., Abeler-Dörner, L., Parker, M., Bonsall, D., Fraser, C.: Quantifying sars-cov-2 transmission suggests epidemic control with digital contact tracing. Science (2020)

16. Fisher, J., Sanchez, M.H.: Authentication and verification of digital data utilizing blockchain technology (Sep 29 2016), uS Patent App. 15/083,238
17. Google, Apple: Privacy-Preserving Contact Tracing. https://www.apple.com/covid19/contacttracing (2020), [Online]
18. Hewitt, C.: Actor model of computation: scalable robust information systems. arXiv preprint arXiv:1008.1459 (2010)
19. Hoare, C.A.R.: Communicating sequential processes. Communications of the ACM **21**(8), 666–677 (1978)
20. Hunt, P., Konar, M., Junqueira, F.P., Reed, B.: Zookeeper: Wait-free coordination for internet-scale systems. In: USENIX annual technical conference. vol. 8. Boston, MA, USA (2010)
21. Jones, M.B.: The emerging json-based identity protocol suite. In: W3C workshop on identity in the browser. pp. 1–3 (2011)
22. Khawas, C., Shah, P.: Application of firebase in android app development-a study. International Journal of Computer Applications **179**(46), 49–53 (2018)
23. Kreps, J., Narkhede, N., Rao, J., et al.: Kafka: A distributed messaging system for log processing. In: Proceedings of the NetDB. pp. 1–7 (2011)
24. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review **44**(2), 35–40 (2010)
25. Liang, X., Shetty, S., Zhao, J., Bowden, D., Li, D., Liu, J.: Towards decentralized accountability and self-sovereignty in healthcare systems. In: International Conference on Information and Communications Security. pp. 387–398. Springer (2017)
26. Merkel, D.: Docker: lightweight linux containers for consistent development and deployment. Linux journal **2014**(239), 2 (2014)
27. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. Computer Science Review **30**, 80–86 (2018)
28. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
29. Odersky, M., Altherr, P., Cremet, V., Emir, B., Maneth, S., Micheloud, S., Mihaylov, N., Schinz, M., Stenman, E., Zenger, M.: An overview of the scala programming language. Tech. rep. (2004)
30. Organization, W.H.: Coronavirus disease 2019 (covid-19) situation report, https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200402-sitrep-73-covid-19.pdf?sfvrsn=5ae25bc7_2
31. Schmager, F., Cameron, N., Noble, J.: Gohotdraw: Evaluating the go programming language with design patterns. In: Evaluation and Usability of Programming Languages and Tools. p. 10. ACM (2010)
32. sentara: sentara, https://www.sentara.com/hampton-roads-virginia
33. Thönes, J.: Microservices. IEEE software **32**(1), 116–116 (2015)
34. TraceTogether: Tracetogether, https://www.tracetogether.gov.sg/
35. Yu, Y., Au, M.H., Ateniese, G., Huang, X., Susilo, W., Dai, Y., Min, G.: Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Transactions on Information Forensics and Security **12**(4), 767–778 (2016)