International Telecommunication Union

ITU-T Technical Specification

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(1 AUG 2019)

ITU-T Focus Group on Application of
Distributed Ledger Technology
(FG DLT)

Technical Specification FG DLT D3.3

Assessment criteria for distributed ledger technology platforms

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The procedures for establishment of focus groups are defined in Recommendation ITU-T A.7.

Deliverables of focus groups can take the form of technical reports, specifications, etc., and aim to provide material for consideration by the parent group in its standardization activities. Deliverables of focus groups are not ITU-T Recommendations.

The ITU Telecommunication Standardization Advisory Group established the ITU-T Focus Group on Application of Distributed Ledger Technology (FG DLT) in May 2017.

FG DLT concluded and adopted its Deliverables on 1 August 2019.

Type	**Number**	**Title**
Technical Specification	FG DLT D1.1	DLT terms and definitions
Technical Report	FG DLT D1.2	DLT overview, concepts, ecosystem
Technical Report	FG DLT D1.3	DLT standardization landscape
Technical Report	FG DLT D2.1	DLT use cases
Technical Specification	FG DLT D3.1	DLT reference architecture
Technical Specification	FG DLT D3.3	Assessment criteria for DLT platforms
Technical Report	FG DLT D4.1	DLT regulatory framework
Technical Report	FG DLT D5.1	Outlook on DLTs

The FG DLT Deliverables are available on the ITU webpage, at https://itu.int/en/ITU-T/focusgroups/dlt/.

For more information about FG DLT and its deliverables, please contact Martin Adolph (ITU) at tsbfgdlt@itu.int.

Technical Specification FG DLT D3.3

Assessment criteria for distributed ledger technology platforms

Summary

This technical specification is a deliverable of the ITU-T Focus Group on Application of Distributed Ledger Technology (FG DLT).

It defines an assessment framework for distributed ledger technology (DLT) platforms, and includes a set of criteria for function, performance and other aspects. The framework can be used as a guideline for DLT platform assessment, as well as for information disclosure of a certain DLT platform product.

Acknowledgements

This Technical Specification was researched and principally authored by Baixue Yang (CAICT), Wei Kai (CAICT), Ruifeng Hu (Huawei), and reviewed by Lisa Tan (Economic Design), Luca Boldrin (InfoCert), Prof. Javier Ibáñez (Comillas University).

Keywords

DLT; distributed ledger technology; ledger; blockchain; assessment; criteria; test

Disclaimers

(1) Sample projects and reference articles mentioned in this deliverable are only for the purpose of analysis of technical architecture. The Focus Group does not endorse any of these projects and/or reference articles, nor their technical aspects.
(2) Some contents of this deliverable may involve patents. The Focus Group does not take the responsibility of identifying patents.

Editors:

Yang Baixue
CAICT
China
Tel: +8613001251346
E-mail: yangbaixue@caict.ac.cn

Wei Kai
CAICT
China
Tel: +8601062300249
E-mail: weikai@caict.ac.cn

Ruifeng Hu
Huawei
China
Tel: +8602556621725
E-mail: huruifeng@huawei.com

CONTENTS

Page

Technical Specification FG DLT D3.3

Assessment criteria for DLT platforms

1 Scope

This technical specification defines an assessment framework for distributed ledger technology (DLT) platforms, which includes a set of criteria for function, performance and other aspects. The framework can be used as a guideline for DLT platform assessment as well as information disclosure of a certain DLT platform product.

2 Terms and definitions

2.1 Terms defined elsewhere

This document uses DLT related terms defined in ITU-T Technical Specification FG DLT D1.1 [b-DLT 1.1].

3 Abbreviations

API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
BFT	Byzantine Fault Tolerance
CFT	Crash Fault Tolerance
CPU	Central Processing Unit
DAO	Decentralized Autonomous Organization
DDoS	Distributed Denial of Service
DLT	Distributed Ledger Technology
KYC	Know Your Customer
SDK	Software Development Kit
SPV	Simplified Payment Verification
TPS	Transaction Per Second

4 Conventions

This clause is intentionally left blank.

5 Overview

The assessment criteria framework defined in this document consists of 25 assessment items. They can be classified into 3 domains including core functions, application support functions and operation functions of a DLT platform.

This framework also includes a performance domain and an ecosystem domain, see Figure 1.

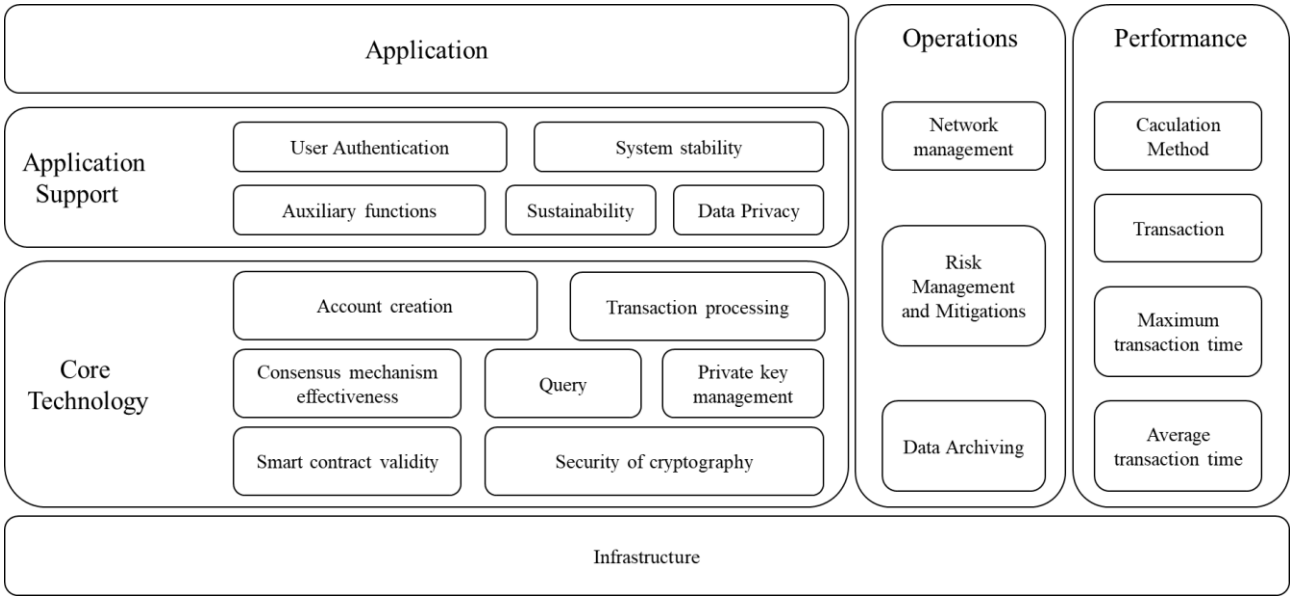

Figure 1: Overview of the DLT assessment framework

The framework corresponds to the reference architecture of a DLT platform [b-DLT 3.1], see Figure 2.

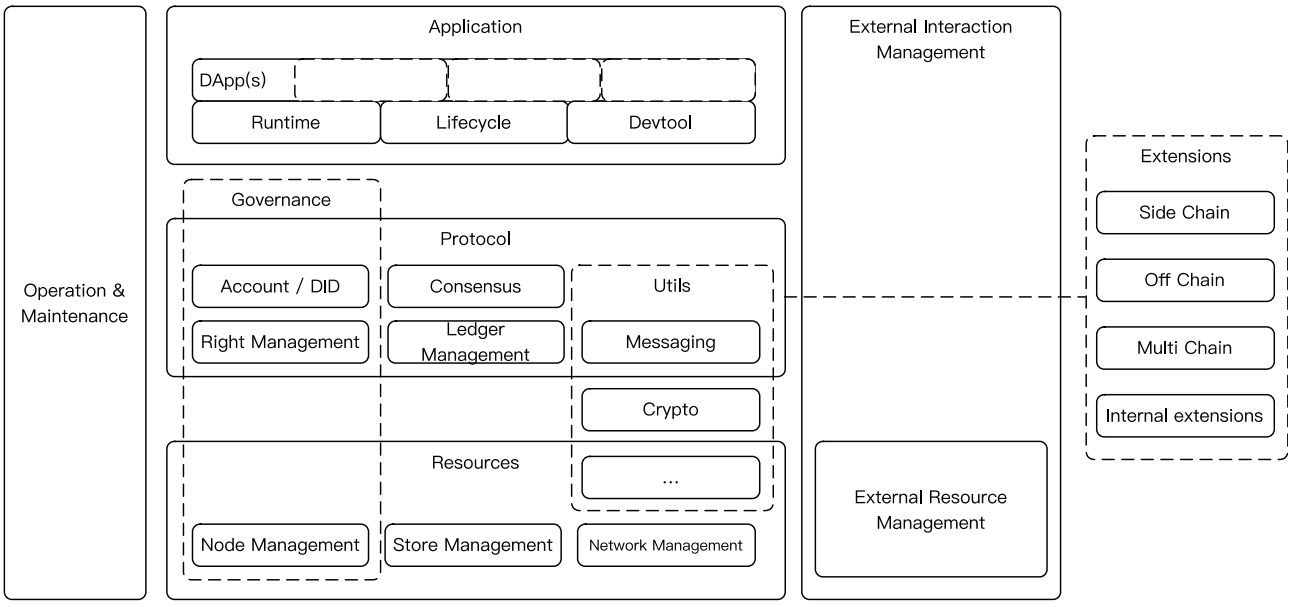

Figure 2: DLT reference architecture diagram [b-DLT 3.1]

The vendors are required to reveal details about these metrics, which can be verified through document review or functional testing.

6 Criteria for DLT core functions

6.1 Account creation

This refers to the ability to create user accounts. Accounts contain public and private key pairs. The create action can be launched by a client or by a smart contract. The private key shall be kept by the client only. If an account name can be customized, the account name must be unique in the system.

6.2 Transaction processing

This refers to the ability to process transaction(s). There are two types of transactions. It is not necessary for the DLT platform to support both types of transactions.

- **Asset transfer transaction** refers to the transfer of certain amount of an asset between accounts, ensuring the asset in the ledger is balanced.
- **Non-asset transfer transaction** (such as changing the configuration parameter of an account, modifying the status of a smart contract and other status modification operation within an account) refers to any transactions without any asset being transferred.

Users could check the result from any of the nodes in the DLT system after a transaction has been successfully committed.

6.3 Query

A user can get result(s) by information request(s).

6.3.1 Balance query

A user can acquire its account balance with a searching condition. DLT platform without asset transfer function does not need to support this function.

6.3.2 Conditional query

User can search its historical information in the DLT platform with a searching condition such as a time period specification or specified user account.

6.4 Consensus mechanism effectiveness

A consensus mechanism is a set of rules and procedures by which consensus is reached. It is a data consistency mechanism that is used in DLT platform to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems.

To ensure the effectiveness of the consensus mechanism, there should be enough nodes participating in the consensus process, aligned with the objectives of the blockchain platform. E.g., some blockchain platforms only requires a specific number of masternode participants, while others require all token holders.

6.4.1 Data consistency

The data synchronization module ensures that the distributed ledger is a consistent ledger. The synchronization module also validates the synchronized data to ensure its correctness and consistency no matter how immediate or eventual finality is achieved.

6.4.2 BFT (Byzantine Fault Tolerance) / CFT (Crash Fault Tolerance)

DLT systems should continue to function in spite of some nodes with malicious actions or systems failure. The threshold for tolerated malicious/crashed node is determined by the consensus mechanism selected, as well as the economics design of the DLT platform.

6.5 Private key management

Public-key cryptography is a cryptographic system that uses pairs of keys: public keys (which may be disseminated widely), and private keys (which are known only to the owner).

Private key management for use of DLT is an important function for user experience and security measures. It provides a reliable and safe way to keep users' private keys. Storage and usage of private keys may be separated. User should have full control of their private key usage. There are two common methods to store private keys: software wallet method and hardware wallet method.[1]

6.5.1 Software wallet

A software wallet can be a software application (i.e., client application, mobile application) or a service (i.e., digital asset exchanges) to store the private key and to track asset ownership.

In a deterministic wallet, a mnemonic sentence or a word seed is generated and that single root key is safe enough. In a non-deterministic wallet, each keys can be randomly generated on its own accord. Therefore, any backup of the wallet must store each and every single private key used as an address or account password.

6.5.2 Hardware wallet

A service can create a private key offline and provide the user with a device or physical medium, which stores the private key. It should have the ability to execute basic operations such as signing transactions.

6.6 Smart contract mechanism

DLT can support more complex transactions as technology evolves. Some complex transactions are stored in DLT systems in the form of source code/bytecode programs, and can be executed to deal with different business logics. These programs are called smart contracts.

Smart contract mechanism includes language definition, compilation and execution of the code. Smart contracts for different DLT systems can be implemented using simple interpreted scripts or programming languages.

6.6.1 Monitor ability of participants' status

This item refers to system's ability to monitor the status of nodes that are participating in the execution of a smart contract.

6.6.2 Lifecycle management of smart contract

This item refers to the availability of a smart contract state ID and the availability of functions for its lifecycle management, such as create, deploy, activate, suspend and destroy.

6.6.3 Security of smart contract

This item refers to the capability of writing high quality smart contracts (i.e., low bug rate) by using a programming language, which is more suitable to write smart contract or by providing contract templates.

6.6.4 Smart contract data access control

A DLT system should disclose how developers specify the authorization and confidentiality of their smart contract in a technical view.

[1] Private key storage and usage can be separated, which may lead to many other applications, such as custodian.

6.7 Security of cryptography

Security includes encryption, cryptography, crash tolerance, hack tolerance, etc. A DLT system should ensure the highest security for the system and disclose its security measures.

6.7.1 Encryption declaration

A DLT platform should specify where the encryption is derived; open-source encryption or regulatory compliance encryption.

6.7.2 Pluggable encryption algorithm

A DLT platform can use pluggable modular encryption and switch to a specified encryption algorithm online or offline as required.

6.7.3 Efficiency of encryption algorithm

A DLT platform should use and make available to use secure, strong enough encryption with acceptable efficiency, depending on the objectives of the system.

6.7.4 Strength of encryption

A DLT platform should declare the security level of the used cryptographic schemas. Category and cipher strength of the encryption could be taken as metrics. In addition, quantum-resistant encryption algorithms could be taken into account.

7 Criteria for DLT application functions

7.1 User authentication

A DLT platform should have user authentication modules and user access control management modules. Electronic signature is an effective way to authenticate user. The platform may allow the creation of smart contracts that can authenticate and control the access of users.

7.1.1 User account verification

This is the validation of information, such as keystore and password, or 2-step verification.

7.1.2 Login state management

The platform should update the user login-state after user login.

7.1.3 User classification and user management

This is to assign users into one of several types and manage their permissions.

7.1.4 Authorization

Users can grant authority to others to access or modify their private data.

7.1.5 Smart contract data access control

Smart contract data should be shared among participants of the smart contract, all relevant participants should have access to this information, for example, interfaces to represent smart contracts and query them.

7.2 System stability

The platform is required to satisfy at least, but not limited to, the following requirements.

7.2.1 Stability for manage nodes

The system should grant normal operations when some nodes join, leave or upgrade.

7.2.2 Stability for cross-chain operation

The system should grant normal operations when cooperating with other DLT system or cloud system.

7.2.3 Network latency

The system should remain stable after running 7x24 hours with latency of network. The tolerable extent of latency is based on design of system.

7.2.4 Memory utilization

The system should remain stable after running 7x24 hours without memory exceptions.

7.2.5 CPU utilization

The system should remain stable after running 7x24 hours without CPU exceptions.

7.2.6 Stability for concurrency

The system should remain stable with bursts of concurrent transactions.

7.3 Economic mechanism design

Economic mechanisms should be in place to incentivize user participation. This includes, but not limited to, consensus protocol, resolution mechanisms, voting protocol, allocation mechanism, bargaining protocol, monetary policy of the token, transaction fees.

Economics design focuses on both the economics of the blockchain platform, as well as the economics the tokens that is produced in the blockchain system (when applicable).

7.3.1 Incentive mechanisms

Incentive mechanisms includes both financial incentives and non-financial incentives. Non-financial incentives can include voting protocol, reputation mechanism, and allocation mechanisms. Financial incentives are a more direct form of rewards like monetary policy of the token, transaction fees, platform activities (i.e., block rewards when a block is mined). The latter is more applicable in the permissionless blockchains.

Incentives mechanisms are ways to coordinate activities amongst the decentralized participants, to achieve the objectives of the DLT system. The incentives (financial and non-financial) are in place to align the behaviors of decentralized participants with the DLT ecosystem as a whole.

E.g., Bitcoin's Proof-of-Work mechanism rewards users in financial ways (diminishing returns of block rewards and transaction fees) in return for the investment of electricity usage and special ASICs. This incentive mechanism should be detailed and be made known.

7.3.2 Token economics disclosure

The token is a digital representation of value. They can be fungible or non-fungible in nature. Fungible tokens can have different functions, namely security, utility and money. Depending on the token function, a tokenomics report detailing the token economics should be publicly released.

This should include, but not limited to, token policy (monetary policy of the token), valuation policy (for security tokens), platform activities (e.g., block rewards for miners), transaction fees, property rights and distribution of tokens.

7.3.3 Token transfer

A DLT platform should use a standard protocol for its tokens. The system should have functionalities to facilitate cross-chain or cross-DLT system operations, when applicable or required by regulations.

7.4 Information privacy

Information privacy is a key requirement for DLT platforms, useful in some industries like finance and healthcare. It refers to the generation, storage and transmission of data in a DLT system being confidential and user private information be safely stored.

7.4.1 Secure transmission

Confidential or proprietary information should be transferred over a secure channel, which is achievable by a specified secure transmission protocol.

7.4.2 Restricted data access

Any confidential or personal information that is protected by law or policy requires the appropriate level of differential access control and security protection whether in storage or in transit.

7.4.3 Privacy protection

A DLT platform may use privacy protection algorithm(s) such as zero knowledge proofs, ring signature, secure multi-party computation and homomorphic encryption to avoid privacy disclosure.

7.5 Application support functions

A DLT platform may implement application support functions to improve user-friendliness (i.e., user experience and user interface).

7.5.1 User interface for query

A DLT platform may provide a functionality (web page, app, browser plug-in, etc.) to perform a query, visualize the query result and show the status of ledger.

7.5.2 User interface for smart contract

A DLT platform may provide functionalities to visualize the deployment and invocation of smart contracts, and queries to smart contract data.

7.5.3 Multi-language software development kits (SDKs)

A DLT platform should provide at least one SDK and may translate it into other programming languages.

8 Criteria for DLT operation functions

8.1 Network management

The management and monitoring of nodes within a DLT system, including status, configuration, node type and behaviour.

8.1.1 Node status monitoring

A DLT system should have ability to monitor nodes statuses, such as number of nodes online/offline, synchronization status, client version and so on.

8.1.2 Multi type nodes

A DLT system should have ability to classify nodes. For instance, node can be classified into two categories, full node and lightweight node, according to whether a complete ledger copy is stored in the node. With lightweight nodes, SPV method should be applied to verify the correctness of the shared ledgers.

8.1.3 Node configuration modification

A DLT system should have the ability to support hot or cold modification of the node's configuration parameter, such as the block size, node type and so on.

8.2 Risk management and mitigation

A DLT system should have the ability to resist DDoS attack, Sybil attack or dishonest node(s). If failure exists under an attack, the DLT system should have ability to recover to its previous known state.

8.2.1 Recovery mechanisms

A DLT system should be able to recover from failure by downgrading recovery, security service, etc. Recovery solution should be flexible and problem-oriented.

8.2.2 Trouble shooting

A DLT system should have ability to execute rapid trouble shooting, automatically send failure notifications.

8.2.3 Avoid single point of failure

The DLT system should not depend on any centralized system, which might cause a single point of failure.

8.3 Data archiving

A DLT system is an append-only trusted ledger. However, mass data stored in DLT systems may degrade the query performance. It is essential to move data that is no longer actively used to a separate storage device for long-term retention, especially for a permissioned DLT system.

8.3.1 Data archiving

A DLT system should have ability to archive data that is no longer used or activity level under a threshold to independent storage and migrate the archived data from a node.

8.3.2 Data query

A DLT system should have ability to query archived data by providing APIs or tools.

8.3.3 Data recovery

A DLT system should have ability to recover archived data in some ways, and the ledger(s) should keep the same status as before it was archived.

9 Performance

The throughput and resource usage of processing standard transactions. Environment and deployment reasons may affect performance, such as network topology and test environment (CPU, memory, disk, network). Transactions per Second (TPS) is a standard performance indicator. When evaluating performance with TPS, topology deployment and test environment must be indicated.

9.1 Metric definitions

$$\text{TPS} = \frac{\text{number of processed transactions}}{\text{time used to process transactions}}$$

The data used to calculate TPS needs to be gathered when system is stable. Both system booting or shutting down and test environment instability may lead to inaccuracy.

$$\text{MTD (max transaction delay)} = \text{Max (transaction confirmed time} - \text{transaction committed time)}$$

$$\text{ATD (average transaction delay)} = \frac{\Sigma\,(\text{transaction confirmed time} - \text{transaction committed time})}{\text{number of transactions}}$$

9.2 Preconditions for performance evaluation

9.2.1 Test environment

When evaluating performance of a DLT system, the hardware such as CPU, memory, hard disk, network must be indicated.

9.2.2 Topology of the network

When evaluating performance of a DLT system, network topology must be indicated.

9.2.3 Test system deployment

When evaluating performance of a DLT system, the type of deployment must be indicated.

9.3 Transaction

When evaluating performance of a DLT system, numbers of transaction must be indicated.

9.4 Test tools

There are open source tools designed especially for DLT system to test performance, such as Hyperleder Caliper [b-caliper] or TrustedBench [b-trustedb]. Script can also be used to make performance test.

10 Criteria for DLT ecosystem

10.1 Platform maturity

Platform maturity of DLT includes many factors such as year of creation, launch of production version, deploying networks (mainnet/testnet), numbers of production networks, numbers of applications, and expertise of team. Whether permissioned network or permissionless network, DLT vendor should disclose this information to all consumers.

10.2 Open source

DLT platforms should be open sourced (to its users) and announce the license it is using.

10.3 Maintenance

DLT platforms should be well maintained by either individuals, companies or non-profit organizations. This can include regular updates on source repository, active discussion by the community regarding the platform and ease of updating the DLT platform with respect to the decentralized applications and DAOs that exists on it.

10.4 Availability of professionals

Different DLT platforms require compound technical backgrounds. This indirectly limits the number of professionals available in the market. Sufficient talent supply is an important factor to evaluate DLT platforms.

10.5 Running cost of DLT systems

It is an essential factor for DLT systems to evaluate the running cost. It includes transaction fees payable, transaction confirmation time and cost of writing, reading and executing smart contracts. In the future, with the importance of smart contract security, audit fee of smart contract and code review and many other intermediate fees should also be taken into consideration.

10.6 Avoid vendor lock-in

DLT system should have standardized APIs, such as, service access, data dictionary, communication protocol, encryption algorithm and system testing, so that there can be multiple vendors that provide similar services for the customers to avoid vendor lock-in.

Annex A: How to use the assessment criteria

Assessment criteria concerning the economic mechanism (clause 7.3) and ecosystem (clause 10) should be tested by disclose information. The tester conducts a comprehensive assessment of DLT system information and market information.

Criteria related to application (clause 7) and operation (clause 8) functions should be tested in DLT platforms in a production environment.

Performance criteria (clause 9) should be assessed in a laboratory environment, i.e., a "quality assurance environment", which should be stable and controllable [b-SQuaRE].

DLT core function criteria (clause 6) should be assessed in a test environment. These tests may affect normal operation of a DLT platform.

Bibliography

[b-caliper] Hyperledger Caliper, https://www.hyperledger.org/projects/caliper.

[b-DLT 1.1] ITU-T Technical Specification FG DLT D1.1 (2019), *DLT terms and definitions.*

[b-DLT 3.1] ITU-T Technical Specification FG DLT D3.1 (2019), *DLT reference architecture.*

[b-SQuaRE] ISO/IEC 25000:2014, *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE*.

[b-trustedb] TrustedBench, https://github.com/TrustedBlockchain/TrustedBench.