Working Paper presented at the

Peer-to-Peer Financial Systems 2020 Workshop

2020

A Contract Service Provider Model for Virtual Assets

Thomas Hardjono

MIT Connection Science

P2P Financial Systems

Powered by

Exponential Science

A Contract Service Provider Model for Virtual Assets

THOMAS HARDJONO, MIT Connection Science
ALEXANDER LIPTON, MIT Connection Science
ALEX PENTLAND, MIT Connection Science

With the recent rise in the cost of transactions on blockchain platforms there is a need to explore other service models that may provide a more predictable cost to customers and end-users. We discuss the *contract service provider* (CSP) model as a counterpart of the successful Internet ISP model. Similar to the ISP business model based on peered routing-networks, the CSP business model is based multiple CSP entities forming a *CSP community* or group offering a contract service for specific types of virtual assets. We discuss the *contract domain* construct which encapsulates well-defined smart contract primitives, policies and contract-ledger. We offer a number of design principles borrowed from the design principles of the Internet architecture.

Additional Key Words and Phrases: Blockchains, FaaS, smart contracts, virtual assets, contract service providers, contract domains

1 INTRODUCTION

Blockchains and distributed ledger technology (DLT) currently face a number of growing pains, one being the challenge of providing a suitable cost and pricing model for entities that participate in a blockchain ecosystem. The recent rise in the cost of transactions in Bitcoin and Ethereum illustrates the reality that the virtual asset industry is still in its nascency – even with Bitcoin being over a decade old. This rise in costs is exemplified not only by Bitcoin, but also by Ethereum with the recent hype in "decentralized finance" (DeFi) [1, 2]. This points to the need to consider alternative service pricing models that offer stability over time and affordability to ordinary end-users.

The current work proposes a new kind of service provider for blockchain-based asset related services, which we refer to as the *Contract Service Provider* (CSP). This contract-centric view of services follows from the concept of Function-as-a-Service (FaaS) [3, 4], and it represents a further logical narrowing of the notions of software-as-a-service (SaaS) and the variations of the hosted computing model (e.g. container-as-a-service, node-as-a-service, etc.).

The current work provides discussions around three (3) areas or themes of focus, namely that:

(1) Blockchain interoperability architectures that support the mobility of virtual assets across blockchain networks tempers or reduces fluctuations in transaction costs or fees.
(2) Decentralization infrastructures (i.e. blockchain nodes) must operate agnostically to the economic value of the virtual assets flowing through the nodes. This is one of the important corollaries of the end-to-end principle [5, 6] of the Internet architecture – which when implemented together with interoperable blockchain architectures permits true scaling of blockchain-based services.
(3) Simple and well-defined smart contracts permits the standardization of contract primitives for specific types of assets. This, in turn, permits smart contracts to be accessible as a Function-as-a-Service made available by contract service providers.

The overall goal of the CSP model is to simplify end-user access to services dealing with virtual assets, and supporting fixed fees to end-users for contract invocations (e.g. subscription model based on tiered pricing). The contract primitives are "light weight" in that it provides only a limited set of operations (in the sense of Bitcoin's limited set of op-codes), each of which consumes a deterministic number of CPU operations.

Authors' addresses: Thomas Hardjono, MIT Connection Science, Cambridge, MA 02139, hardjono@mit.edu; Alexander Lipton, MIT Connection Science, Cambridge, MA 02139, alexlip@mit.edu; Alex Pentland, MIT Connection Science, Cambridge, MA 02139, pentland@mit.edu.

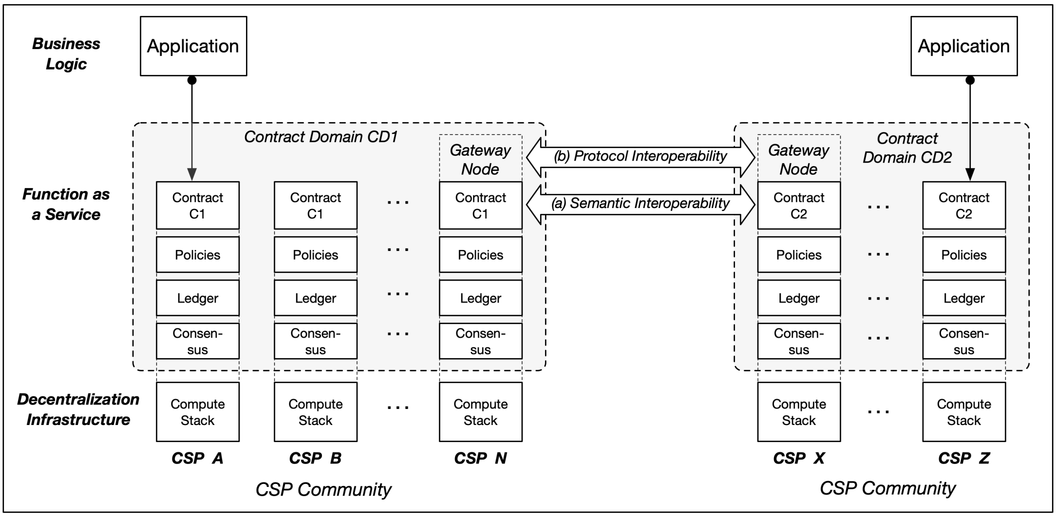

Fig. 1. Interoperability of contract semantics and gateway protocols

We define CSPs more precisely in Section 3, and discuss the notion of the contract-domain in Section 4. We discuss the use of well-defined contract primitives in Section 5 as the basis of services. In order to adhere to the end-to-end principle, we discuss the importance of externalizing the value of the virtual assets to the blockchain infrastructure in Section 6. We close the paper with some conclusions in Section 7.

We seek to make the current paper readable to a broad audience, and as such have sought to limit the usage of technical jargon. However, we assume the reader is at least familiar with Bitcoin, cryptocurrencies, and distributed ledgers generally.

2 INTEROPERABILITY, ASSET MOBILITY AND TRANSACTION COSTS

We believe the rise in transaction costs would be tempered if blockchain systems had the same degree of interoperability as IP routing autonomous systems. If virtual assets had ease of mobility or transferability across different blockchain systems, then end-users and asset holders would have the freedom to move their assets elsewhere should their current blockchain system become too costly to do business in. In general, blockchain interoperability is crucial not only for asset mobility, but also for the scalability and survivability of blockchain networks as a whole [7].

2.1 Interoperability at Peer Layers

In order for asset mobility to occur seamlessly across blockchain systems, interoperability needs to occur at several levels in the blockchain functional layers (see Figure 1):

- *Interoperability of contract semantics*: When virtual assets are to be transferred from one blockchain system to another, both blockchain systems must have the same semantic understanding of the meaning of the "transfer" transaction. This must be true regardless of the specific syntax of the contract language being used in either of the blockchain systems [8]. This is illustrated as item (a) in Figure 1.
  For example, when an Originator (Alice) seeks to transfer her virtual asset from its current blockchain system (e.g. B1) to an a Beneficiary (Bob) in a different blockchain system (e.g. B2), the smart contract implementing this unidirectional cross-chain transfer must have

compatible (or identical) state-machines – which must result in the states maintained by their respective ledgers to achieve consistency across systems (independent of whether the ledgers are permissioned or permissionless). Semantically, the asset in blockchain B1 must be extinguished and then introduced into in blockchain B2 in an atomic manner while preventing double-spending by Alice while the transfer is being carried-out.

- *Gateway protocol interoperability*: The nodes that perform asset mobility related functions – referred to as *gateway nodes* – must implement the same gateway protocols in order for both ledgers to arrive at the same consistent state [9].
  Using the previous Alice and Bob example of the cross-chain asset transfer, the gateway G1 in the origin blockchain B1 must execute the same technical transfer protocol as gateway G2 in the destination blockchain B2. This is illustrated as flow (b) in Figure 1. Since it is reasonable to assume that different technical transfer protocols may be developed and standardized in the future, gateways G1 and G2 must first negotiate the common protocol (type and version) supported by both gateways. Protocol-type and parameter negotiation is a common phase in many Internet protocols (e.g. TLS handshake negotiation).
- *Common reference to authoritative asset profile*: Virtual asset mobility requires an authoritative definition of the virtual asset in question. We refer to this metadata as the *asset profile*. It is essentially a prospectus file of a regulated asset that includes information and resources describing the virtual asset. The asset profile is independent from the specific instantiation of the asset (on a blockchain or otherwise) and independent from its instance-ownership information [9].
  When the Originator (Alice) in blockchain B1 seeks to transfer virtual assets to a Beneficiary (Bob) in blockchain B2, both Alice and Bob must have the means to refer to the same asset definition (namely the asset profile file). Consequently, the gateway nodes G1 and G2 that are performing the transfer on behalf of Alice and Bob must also have the means to refer to (point to, or hash of) the asset profile file. A copy of the signed asset profile file may be represented on-chain, or it may be elsewhere off-chain (e.g. at a well known asset depository [10]). We discuss this further in Section 6.

2.2 Entities in the Ecosystem

Figure 1 incorporates a number of entities and functions that we will discuss further in the ensuing sections. We provide a brief definitions of these here in order to facilitate discussion.

- *Contract*: The Smart contract service defined by the set of primitive operations implemented on-chain and the asset-types processed through the primitives. Figure 1 shows two contracts C1 and C2, which are implemented by the nodes participating in the Contract Domain CD1 and CD2 respectively.
- *Contract Service Provider* (CSP): A regulated service provider (e.g. registered business) that participates in making available contract services to its customers. Note that a CSP is in fact also Virtual Asset Service Provider (VASP) as defined under the FATF definition and the Travel Rule (see [11, 12]).
- *CSP Community*: A group of CSPs offering contract services for a given asset type on a blockchain network consisting of their nodes.
- *Contract Domain*: The various computing resources required to implement a contract service, including contract-specific functional components (i.e. primitives, contract-ledger, consensus algorithm and domain policies) and the other technological constructs that implement the domain.

- *Gateway node*: The node belonging to a CSP that performs cross-domain (cross-chain) asset transfers.

An overall summary of Figure 1 is as follows. A group of CSPs (labelled $A, B, \ldots, N$) form a CSP community that agrees to make available contract $C1$ on their nodes. The decentralization functions (contract, policies, ledger and consensus protocol) used to implement the contract form a contract domain CD1. Each member of the CSP community is free to use different compute stack technologies (e.g. bare metal, hosted node, container as a service, etc.) to stand-up their nodes that participates in the contract domain. Another CSP community is also shown in Figure 1, consisting of CSPs labelled $X, \ldots, Z$, which agree to make available contract $C2$ on their nodes forming domain CD2. As will be discussed later, within this contract-centric CSP model a given CSP as a business entity may participate in several different CSP communities (contract domains) simultaneously.

2.3 Principles from the Internet Architecture

The following summarizes a number of key principles from the Internet architecture that are relevant to blockchain interoperability:

- *Observance of the end-to-end principle*: Smart contracts and the blockchain infrastructure must relegate the notion of economic value of assets to the end-points of the transaction outside of the blockchain system (i.e. the originator and beneficiary). The principle in essence states that any semantic meaning associated with the data (bytes) flowing through a network must be external to the network itself [5, 6]. That is, it must be kept at the "edges" of the network at the the sender and the receiver. The network infrastructure must be agnostic as to whether a set of data packets (datagrams) belong to one user application (e.g. email) or another application (e.g. video streaming).
  When re-casted to a blockchain system consisting of multiple nodes, the end-to-end principle demands that for cross-chain interoperability the computing nodes of a blockchain network be agnostic to the economic value of the virtual assets (bytes) being processed collectively by the nodes. The blockchain ledger must be viewed simply as state-table with no attached economic significance, much as a routing-table in an ISP domain has no economic significance to the end-users.
- *Observance of the autonomous systems principle*: Contract service providers must be able to form their own communities by collectively pooling their resources (i.e. nodes), without being reliant on other platforms. That is, each blockchain system must operate as a true autonomous system [6] regardless of the number of nodes in the blockchain and the access model (permissionless or permissioned). Standard interfaces and cross-chain protocols must be defined to prevent end-users' assets from suffering from "platform capture". Blockchain systems must be designed so as not be dependent on third parties (e.g. crypto-exchange entities) in order to achieve asset mobility. Any such dependence reduces the autonomy of a blockchain system, and the overall technical value of blockchain/DLT technology.

We believe the end-to-end principle (i.e. lack of adherence to it today) contributes to the problems around fluctuating transaction costs. A blockchain service model that operates based on an inherent tight-coupling between the actual infrastructure operational costs (i.e. cost of CPU hardware, electricity, etc.) with the tokenization of user-access results in the unpredictability of transaction costs. Such a model essentially makes the infrastructure (i.e. nodes) no longer agnostic to the economic value of the "bytes" passing through the infrastructure.

Although this tight-coupling maybe motivated by the desire to attain an incentives equilibrium for honest nodes processing transactions (e.g. mining nodes in Bitcoin), this approach appears to be unsustainable in the long term. Node-operators (i.e. miners) and token-holders are instead

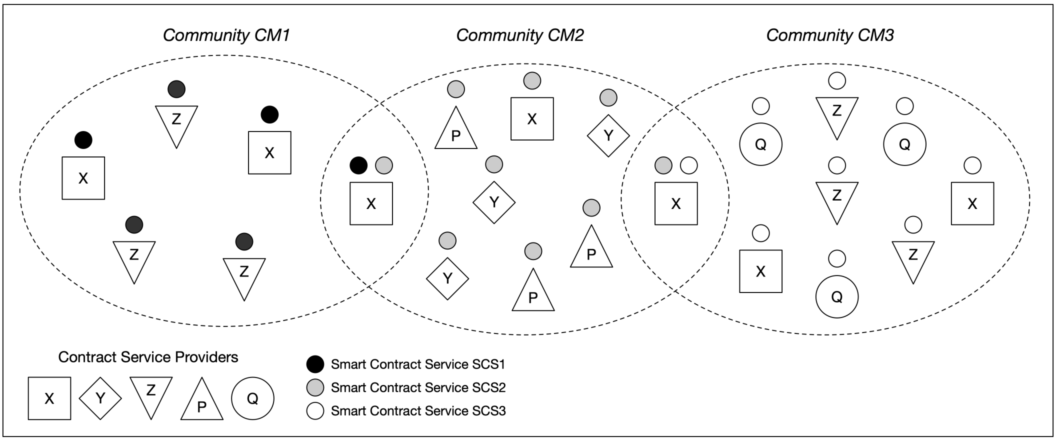
Community CM1
Community CM2
Community CM3
X
Y
Z
P
Q
Contract Service Providers
Smart Contract Service SCS1
Smart Contract Service SCS2
Smart Contract Service SCS3

Fig. 2. Simple example of three (3) CSP communities viewed from a contract-centric perspective

motivated to manipulate the price of tokens at the expense of other users (e.g. indirect inflation through fake transactions [13, 14]).

3 THE CSP AND CSP COMMUNITIES

A Contract Service Provider (CSP) is a regulated service provider who collaborates with other CSPs in making available on distributed nodes one or more *smart contract primitives* that consists of well-defined operations applicable to virtual assets. Collectively, the group of CSPs is said to offer *contract services* via well-defined APIs to one or more end-users (customers), which may include individuals and organizations. The contract primitives and the supported types of assets are chosen by the CSPs prior to deployment. All smart contracts employed by the CSPs are authored (programmed) by the CSPs, and the end-users are not permitted to to load/run their on smart contracts.

As mentioned previously, the goal of the CSP model is to simplify end-user access to services dealing with virtual assets. Contract invocations are based on fixed fees that are not tied to any dynamic incentives mechanisms (i.e. proof or work). The contract primitives must be "light weight" in that it provides only a limited set of operations, each of which consumes a deterministic number of CPU operations.

This allows a CSP to "containerize" node-stacks (i.e. virtual machine software images) that implement these contract primitives and to deploy these images in different platform configurations [15]. Thus, for a CSP the cost of operations are also deterministic and node-deployment automation can be further introduced to reduce operational costs. The CSP is free to employ different resource management strategies to implement the requirements of each contract domain (e.g. bare-metal nodes; virtualized nodes on its private cloud; multi-tenanted virtualized nodes on public cloud; SGX-secured nodes; etc.).

3.1 The CSP Community

We define a *CSP community* as a group of regulated contract service providers offering asset-related smart contract services on a blockchain network whose nodes are composed of the computing resources of the members of the CSP community. The notion of a community is contract-centric in that the CSPs in a community agree to allocate computing resources for a given smart contract service. A given CSP community may be constituted to provide contract services for a single smart contract or for several related smart contracts on the same blockchain network. A *contract service* is defined by the set of contract primitives implemented on-chain and the *asset-types* processed through those primitives. For a given contract domain (see below), a customer should be associated with (belong to) only one CSP in the CSP community. The CSP community must clearly define beforehand the contract primitives and the supported asset-types that constitute the contract-service, as well as the pricing structures in order to provide transparency to customers.

From a legal perspective, a CSP community must be founded on a legal contractual agreement that defines the obligations and liabilities of each of the member. The community must define a service level agreement (SLA) for the customers of its membership. This approach is akin to multi-lateral peering agreements used by Internet Service Providers (ISPs), which defines common data routing responsibilities across multiple networks.

A simple example of three (3) CSP communities is shown in Figure 2. Community CM1 are CSPs providing contract service SCS1, the Community CM2 providing contract service SCS2, while Community CM3 the contract service SCS3. The CSP-X (shown as the square box in Figure 2) is active in all three communities, where in each case it makes available computing resources (nodes) to each community.

3.2 CSP Community Membership Agreement

The computing resource to be allocated by each CSP in a CSP-community, the specifications of the technical mechanisms (e.g. protocols) to be used in the community, its core *operating rules* [16], as well as other operational aspects of the contract domain is expressed in the *CSP community membership agreement* document – which is a legally binding contractual agreement. The community membership agreement may also specify the number of CSP entities minimally (maximally) required to establish the contract domain, and the methods to add or subtract the CSP membership. The agreement may also place a time duration commitment on CSPs, meaning that once a contract service is made operational by a CSP community the CSPs are bound to be a member (i.e. allocate nodes and computing resources) until the end of the duration.

From a revenue perspective, the CSP community membership agreement must specify a fair and attractive revenue sharing structure for the CSP members in that community. For example, a revenue sharing model could be based on the number of transactions (local or cross-domain) and the size of the customer-base overall.

Although beyond the scope of the current work, the notion of a *node-stack diversity index* could be defined in the CSP community membership agreement as a measure of the diversity of the node-implementation technologies [15, 17]. The node-stack diversity index may provide customers with a tangible indicator of the resiliency of the blockchain network as a whole (e.g. degree of resistance against malwares/viruses targeted to specific node softwares).

4 THE CONTRACT DOMAIN

We use the notion of the contract domain in order to reason more accurately about the various technical and implementation aspects of a contract service, including access to the components of the contract service (e.g. contract calling APIs, ledger, etc). A *contract domain* is defined by a CSP community through a combination of the following: (i) the set of contract-primitives that constitute the contract service together with the contract ledger and consensus algorithm for the ledger; (ii) the policies regarding the asset types permitted to be transacted in the jurisdiction of operation of the CSP community; (iii) the nodes infrastructure that implements the contract-primitives and enforces policies. This is illustrated in Figure 3.

A contract domain coincides with the CSP community in that both represent the same participating business entities (namely CSPs) and the resources (i.e nodes) dedicated by the business entities

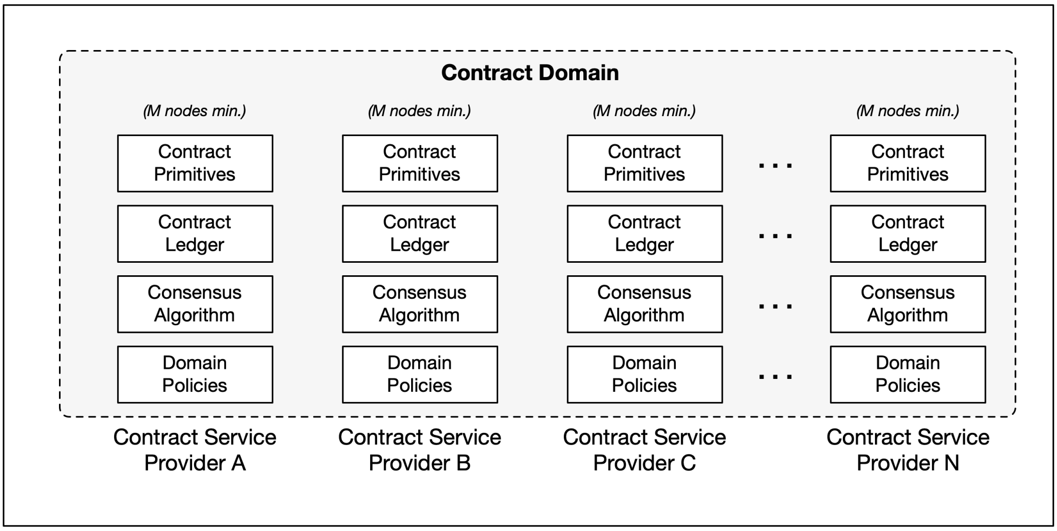
Contract Domain
(M nodes min.)
Contract Primitives
Contract Ledger
Consensus Algorithm
Domain Policies
Contract Service Provider A
Contract Service Provider B
Contract Service Provider C
Contract Service Provider N
. . .

Fig. 3. Overview of a contract domain

to establish the contract service. Thus, for business reasons a given CSP may simultaneously be a member of multiple CSP communities (each with a separate contract domain) at any given moment. In each case, the CSP must allocate the computing resources required for each contract domain (e.g. minimal M nodes) and observe the core operating rules of each community. In each community, the contract domain structure ensures that a separate ledger and consensus mechanism is used to record the asset transactions in that contract domain.

A given CSP may participate in multiple CSP communities (contract domains) simultaneously, dedicating the nodes required in each of the contract domains. Figure 4 illustrates this case. For example, CSP-A is shown to be participating in contract-domains CD1, CD2 and CD3, while CSP-C is participating in domains CD2 and CD3 only. This means that CSP-A, CSP-B and CSP-C share a common contract-ledger for CD3 and jointly participate in the consensus mechanism to maintain that contract-ledger. The CSP-A, CSP-C and CSP-D a common contract-ledger for domain CD2.

It is important to note that when a CSP participates in multiple CSP communities it must adhere by the membership agreement and operating rules of each community. The membership legal agreement must prohibit *domain intersections* from occurring, either inadvertently or intentionally by CSPs. Prohibiting domain intersection means that a CSP must not use the same node in two (or more) contract domains, as this may lead to the node acting as an unauthorized “bridge” between the two contract domains.

As will be discussed below, asset transfers between two contract domains must be performed in an explicit and transparent manner using an atomic cross-domain asset transfer protocol [9]. The gateway nodes in the two domains which perform the asset transfer protocol may happen to belong to the same CSP entity, but this coincidence (e.g. through the gateway selection algorithm in each domain) must be transparent to all other CSP members in the two respective domains.

There are several important aspects about a contract domain and the resources (nodes) implementing the contract domain:

- *Per-domain consensus algorithm and per-domain ledger*: The nodes that implement the contract service in a domain employ a separate consensus algorithm and contract-ledger specifically

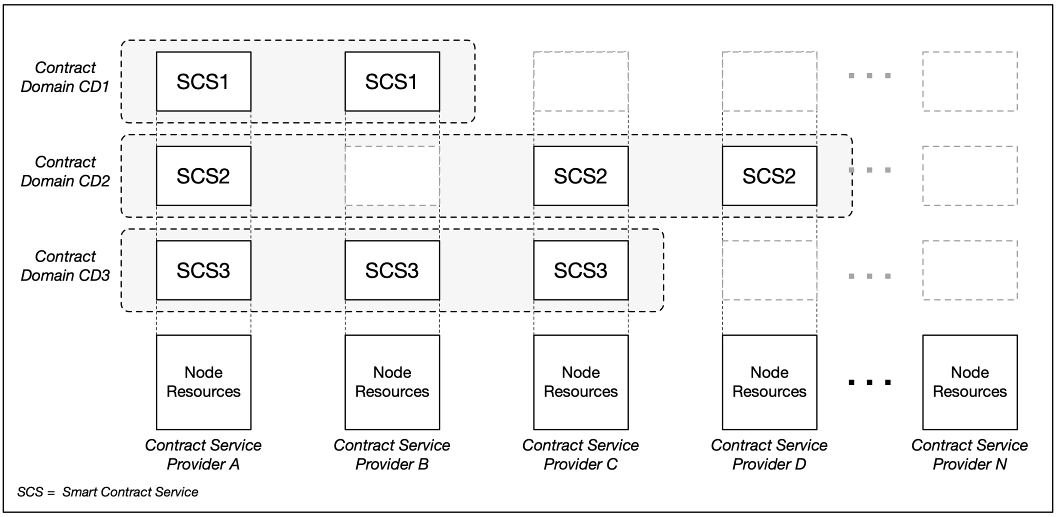
Contract Domain CD1
Contract Domain CD2
Contract Domain CD3
SCS1
SCS1
SCS2
SCS2
SCS2
SCS3
SCS3
SCS3
Node Resources
Node Resources
Node Resources
Node Resources
Node Resources
Contract Service Provider A
Contract Service Provider B
Contract Service Provider C
Contract Service Provider D
Contract Service Provider N
SCS = Smart Contract Service

Fig. 4. Illustration of CSP participation in several contract domains

for that contract service. The choice of consensus algorithm and the structure of the ledger-blocks are defined by the CSP community through their formative the legal agreements. Thus, in Figure 4 there is a separate ledger for contract service SCS1, SCS2 and SCS3. For example, CSP-A who participates in all three domains CD1, CD2 and CD3 must put forward distinct (non-intersecting) nodes (e.g. minimal M nodes per domain). For domain CD1, the nodes establish and maintain the contract-ledger CL1, for domain CD2 the ledger CL2 and for domain CD3 the ledger CL3.

- *Opaque ledgers and contracts*: When an end-user customer obtains access to a contract service from a CSP who is a member of a CSP community, the customer has visibility only into the relevant resources (e.g. contract ledger) for that contract domain. Using Figure 4, if a customer of CSP-A purchases access to contract-service SCS3 (domain CD3), the customer has visibility only to the ledger for SCS3 (i.e. no visibility or access to ledgers for SCS1 and SCS2).
- *No domain intersections*: A given CSP must not deploy the same node in two (or more) contract domains. Asset transfers between two contract domains must be performed in an explicit, transparent and non-repudiable manner.

In the remainder of this paper, we will use the term “CSP community” when discussing the business and legal aspects of contract services by a CSP, and we shall use “contract domains” when discussing the technical aspects of the contracts, nodes, ledgers and blockchain.

5 PRIMITIVES IN A CONTRACT SERVICE

A core requirement of the CSP model is the well-defined, simple and modular smart contract primitives. This ensures that efficient on-chain primitives have a limited number of operations. Furthermore, this ensures that complex business logic is located off-chain within the customer application.

Although there are several possible primitives that a contract-service may use, the following represents the basic primitives which must be implemented by a contract domain [9, 18]:

- *Asset transfer from one customer to another*: This operation moves the ownership of a virtual asset instance from one customer to a second customer, both of which must have been previously onboarded by a CSP member in the community. This is equivalent to Bitcoin's payment to public key (or to a hash of public key).
- *Asset escrow to another customer or to CSP*: This operation conditionally moves the ownership of a virtual asset instance from one customer to another, or from one customer to a CSP in the community. The escrows must be time-limited, meaning that if the condition fails to be satisfied within the specified time, the asset reverts back to the customer.
- *Asset ingress into blockchain*: This operation introduces a regulated virtual asset instance into the contract domain, making it available for exchanges between customers of the CSPs in that domain. This asset-introduction operation maybe at the request of a customer of a CSP. This operation is available only to a CSP because the CSP must validate the legal status of the asset prior to introducing it into the contract domain. A customer cannot introduce virtual assets on their own.
- *Asset egress from blockchain*: This operation removes (extinguishes) a regulated virtual asset instance from the contract domain, making it no longer available to customers of the CSPs in that domain. This operation is available only to a CSP. It marks the ledger to indicate that the asset has moved to another contract domain and therefore unavailable for further use.

Other possible contract primitives include those pertaining to key management and to different types of virtual assets. For example, the key management tasks include: introduction of a customer (new customer) public-key into the contract domain; key-rotation (or revocation) of customer's public-key; introduction of a new CSP public-key; key-rotation (or revocation) of CSP's public-key; and so on. Asset related tasks include: introduction of a new asset-type (e.g. new stablecoin, etc.), removal of an existing asset-type; and so on. Cross-domain (cross-chain) primitives are further discussed in [19].

6 PROFILES OF VIRTUAL ASSETS AND EXTERNALIZATION OF VALUE

As mentioned in Section 2, in order for interoperability to be achieved between two contract domains the nodes that participate in both domains respectively must have a common definition of the virtual asset being transacted. We refer to this authoritative definition (metadata) of a virtual asset as its *asset profile*. It is essentially a prospectus document of a regulated virtual asset that includes information and resources describing the virtual asset. This includes the asset name/code, issuing authority, denomination, date of issue, intended systems of circulation, jurisdictions, and the URLs and mechanisms to validate the information. The asset profile must be digitally signed by its *Profile Authority* (profile issuer), which may or may not be the same entity that creates the instantiations of the virtual asset defined by the profile document.

From the gateway protocol interoperability perspective (see Figure 1), there are a number of technical requirements to achieve interoperability across contract-domains:

- *Externalization of value*: The nodes and the gateway protocol between contract-domains must be agnostic (oblivious) to the economic or monetary value of the virtual asset being transferred [9].
- *Separation of asset profile from value sources and technological instantiation*: As a prospectus metadata, an asset profile document (e.g. signed JSON file) may define the permitted forms of the asset technological instantiation (e.g. blockchain-based, Chaumian eCash, etc). However, the profile document must be standalone and be independent from its specific asset-instantiations or evidence of instantiations. It must also be separated from the evidence

of legal ownership of the instantiations (e.g. asset bound to an owner's public-key on a given ledger).

The asset profile metadata is used when a regulated entity seeks to issue digital or virtual assets in a jurisdiction that recognizes the asset type. The asset profile may specify the technological implementations (e.g. blockchains, DLTs, hash-graphs, etc.) required to instantiate the virtual asset. The function of legally issuing virtual assets in a digital representation is assumed to start outside the contract domain, and is performed by an external entity referred to as the asset *Issuer* authority. A symmetrical role is assumed to be carried out by an asset *Acquirer* authority, which performs the reverse function. The asset Issuer and Acquirer can be the same external entity. This assumption is consistent also with a number of exploratory projects that have been reported (e.g. see [10, 20, 21]). The method to determine the value of an asset is outside of the current work, and several mechanisms have been proposed (e.g. see [22, 23] for a proposed taxonomy).

Figure 5 provides a high level illustration. Here the Asset Issuer (IA1) uses the asset profile definition (Step (a)) in order to digitally represent ("convert") an external source of value (Step (b)) – such as a basket of real-world assets – into its blockchain-based representation (Step 1). The Asset Issuer must ensure that the contract primitives in contract domain CD1 support this type of virtual asset, and that the CSPs in the domain operate within a regulatory jurisdiction (J1) that recognizes the asset type. Note that from a regulatory perspective the CSPs that form (participate in) a contract domain are considered Virtual Asset Service Providers (VASP) in the sense of the FATF definition [11, 12]. Similarly, the asset Issuers and Acquirers are also VASPs under the FATF definition.

When an owner (e.g. Alice) of an instance of the virtual asset in contract domain CD1 seeks to transfer the asset to another contract domain CD2 (e.g. to a beneficiary Bob), the owner invokes the relevant contract that performs the asset transfer across domains (Step (2)). However, the gateway nodes in domains CD1 and CD1 that handle the cross-domain transfer must validate that: (i) the blockchain system in domain CD2 mechanically supports the type of asset as defined in the asset-profile metadata, and that (ii) the jurisdiction (J2) – under which the CSPs in domain CD2 operate – permits incoming virtual assets of the type defined in the profile. Thus, both gateways at providers CSP-N (in domain CD1) and CSP-X (in domain CD2) must have access to the asset-profile metadata file (Step (c) and Step (d)). The gateways must employ an asset transfer protocol that includes (embeds) atomic commitment [24, 25].

Our use of the terms "issuer" and "acquirer" is borrowed from the classic *4-corners model* [26] in the card-payments industry (i.e. credit cards), which has been successfully deployed for over three decades now. In the card-payments ecosystem, the consumer (card-holder) obtains a credit card from the Issuer, which is often a bank or financial institution where the consumer has an account. When the consumer uses the card at a retail merchant to purchase goods (e.g. Point of Service (POS) terminal), the merchant forwards the transactions details to the Acquirer, which is typically the merchant's bank or financial institution. The Acquirer then obtains payment from the Issuer bank (e.g. debited from the consumer's bank account). Thus, in Figure 5 the loop in closed in Step (4) that represents the interaction between the Acquirer IA2 and the Issuer IA1.

The 4-corners paradigm is useful in the context of reasoning about the design of contract domains, for several reasons:

- *Separation of roles and responsibilities*: The paradigm permits the CSP role to be separated from the role of Asset Issuer/Acquirer, where the CSP takes-on legal obligations and in validating an asset prior to accepting it into the contract domain.

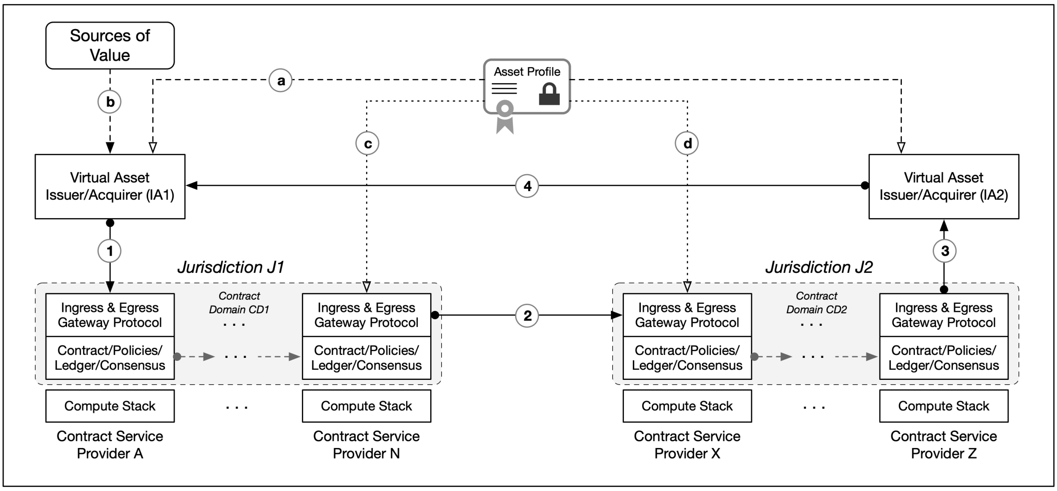
Sources of Value
Asset Profile
a
b
c
d
Virtual Asset Issuer/Acquirer (IA1)
Virtual Asset Issuer/Acquirer (IA2)
4
1
3
2
Jurisdiction J1
Jurisdiction J2
Contract Domain CD1
Contract Domain CD2
Ingress & Egress Gateway Protocol
Contract/Policies/ Ledger/Consensus
Compute Stack
Contract Service Provider A
Contract Service Provider N
Contract Service Provider X
Contract Service Provider Z

Fig. 5. Standard asset profiles and sources of value for virtual assets external to the contract-domain

- *Incorporation of jurisdictional regulations*: The paradigm permits the notion of global jurisdictions to come into the picture by recognizing that the issuers of virtual assets may reside in different legal jurisdictions (e.g. countries).
- *Value-conversions external to the contract domains*: The conversion of values between value-systems and/or monetary jurisdictions occurs at the Issuer-Acquirer level (outside the contract domain). This is consistent with the end-to-end principle [5, 6] discussed previously in Section 2.3.

Note that the 4-corners paradigm does not preclude an architecture in which the asset issuers and acquirers share a common blockchain system for the purposes of tracking conversions to/from sources of value into virtual assets and tracking which asset-profiles have been utilized in the process [27].

7 CONCLUSIONS

The success of blockchain networks hinges on their ability to make smart contract services affordable for everyone. This will not be achievable if transaction costs are unpredictable – something arising as a side effect from a system design that violates the end-to-end principle by tightly coupling the infrastructure operating costs with the tokenization of access to end-users.

Today there is a great interest in the possible use of digital currencies at the national level in the form of Central Bank Digital Currencies (wholesale and retail) and fiat-backed Stablecoins. Unless the problems of blockchain interoperability and the high costs of transactions are solved, it is unlikely that blockchains will be used for CBDCs and that other technical solutions will need to be developed (e.g. Chaum-based payments schemes).

In this paper we have proposed the notion of a contract service provider (CSP) as a counterpart to the well established ISP model for Internet access. The CSP model offers an opportunity to achieve interoperability at the smart contracts layer, and to standardize the contract primitives for specific types of virtual assets. Similar to groups of ISPs forming peering agreements for routing IP traffic, groups of CSPs can form communities that makes available well-defined smart contracts authored by CSPs in the community for specific types of virtual assets.

The CSP model may offer a path forward for the use of decentralized nodes as the basis for enabling CBDC distribution and usage at a low transaction cost to end-users. Similar to the existing interconnected autonomous systems that form the physical Internet, interconnected CSP Communities – as autonomous blockchain nodes with a high degree of cross-chain interoperability – should become the basis for making smart contracts affordable and independent from the incumbent blockchain platforms.

REFERENCES

[1] Z. Voell and W. Foxley, “Decentralized Finance Frenzy Drives Ethereum Transaction Fees to All-Time Highs,” *Coindesk*, August 2020. [Online]. Available: https://www.coindesk.com/decentralized-finance-frenzy-drives-ethereum-transaction-fees-to-all-time-highs

[2] D. Cawrey, B. Keoun, and O. Godbole, “First Mover: Ethereum Faces Inflation Problem as Gas Fees Soar,” *Coindesk*, August 2020. [Online]. Available: https://www.coindesk.com/first-mover-ethereum-faces-inflation-problem-as-gas-fees-soar

[3] F. Alder, N. Asokan, A. Kurnikov, A. Paverd, and M. Steiner, “S-FaaS: Trustworthy and Accountable Function-as-a-Service using Intel SGX,” October 2018. [Online]. Available: https://arxiv.org/pdf/1810.06080.pdf

[4] M. Fowler, “Serverless Architectures,” May 2018, available at https://martinfowler.com/articles/serverless.html, Accessed 7 May 2020.

[5] J. Saltzer, D. Reed, and D. Clark, “End-to-End Arguments in System Design,” *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, November 1984.

[6] D. Clark, “The Design Philosophy of the DARPA Internet Protocols,” *ACM Computer Communication Review – Proc SIGCOMM 88*, vol. 18, no. 4, pp. 106–114, August 1988.

[7] T. Hardjono, A. Lipton, and A. Pentland, “Towards an Interoperability Architecture Blockchain Autonomous Systems,” *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1298–1309, June 2019. [Online]. Available: doi:10.1109/TEM.2019.2920154

[8] J. Hazard and T. Hardjono, “CommonAccord: Towards a Foundation for Smart Contracts in Future Blockchains,” in *W3C Workshop on Distributed Ledgers on the Web.* Cambridge, MA, USA: W3C, June 2016, https://www.w3.org/2016/04/blockchain-workshop.

[9] T. Hardjono, M. Hargreaves, and N. Smith, “An Interoperability Architecture for Blockchain Gateways,” IETF, Internet-Draft draft-hardjono-blockchain-interop-arch-01, October 2020. [Online]. Available: https://www.ietf.org/archive/id/draft-hardjono-blockchain-interop-arch-01.txt

[10] DTCC, “Project Whitney: Case Study,” Depository Trust & Clearing Corporation, DTCC Report, May 2020. [Online]. Available: https://perspectives.dtcc.com/articles/project-whitney

[11] FATF, “International Standards on Combating Money Laundering and the Financing of Terrorism and Proliferation,” Financial Action Task Force (FATF), FATF Revision of Recommendation 15, October 2018, available at: http://www.fatf-gafi.org/publications/fatfrecommendations/documents/fatf-recommendations.html.

[12] ——, “Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers,” Financial Action Task Force (FATF), FATF Guidance, June 2019, available at: www.fatf-gafi.org/publications/fatfrecommendations/documents/Guidance-RBA-virtual-assets.html.

[13] P. Vigna, “Most Bitcoin Trading Faked by Unregulated Exchanges, Study Finds,” *Wall Street Journal*, March 2019, https://www.wsj.com/articles/most-bitcoin-trading-faked-by-unregulated-exchanges-study-finds-11553259600?shareToken=ste90e80f8d18f47cb896491914f06cea3.

[14] J. M. Griffin and A. Shams, “Is Bitcoin Really Un-Tethered?” *SSRN*, November 2019. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3195066

[15] T. Hardjono and N. Smith, “Towards an Attestation Architecture for Blockchain Networks (to appear),” *World Wide Web Journal – Special Issue on Emerging Blockchain Applications and Technology*, 2021. [Online]. Available: https://arxiv.org/abs/2005.04293

[16] Visa, “Visa Core Rules and Visa Product and Service Rules,” Visa, Specification, October 2017.

[17] D. Yaga, P. Mell, N. Roby, and K. Scarfone, “Blockchain Technology Overview,” National Institute of Standards and Technology Internal Report 8202, October 2018, https://doi.org/10.6028/NIST.IR.8202.

[18] M. Hargreaves and T. Hardjono, “Open Digital Asset Protocol,” IETF, Internet-Draft draft-hargreaves-odap-01, November 2020. [Online]. Available: https://www.ietf.org/archive/id/draft-hargreaves-odap-01.txt

[19] T. Hardjono, A. Lipton, and A. Pentland, “Towards a Contract Service Provider Model for Virtual Assets and VASPs,” September 2020. [Online]. Available: https://arxiv.org/abs/2009.07413

[20] MAS, “Project Ubin Phase 5: Enabling Broad Ecosystem Opportunities,” Monetary Authority of Singapore, MAS Report, July 2020. [Online]. Available: https://www.mas.gov.sg/-/media/MAS/ProjectUbin/Project-Ubin-Phase-5-

Enabling-Broad-Ecosystem-Opportunities.pdf

[21] DTCC, “Project Ion: Case Study,” Depository Trust & Clearing Corporation, DTCC Report, May 2020. [Online]. Available: https://perspectives.dtcc.com/articles/project-ion

[22] P. Tasca and C. J. Tessone, “Taxonomy of Blockchain Technologies: Principles of Identification and Classification,” *Ledger Journal*, vol. 4, February 2019. [Online]. Available: 10.5195/ledger.2019.140

[23] T. Ankenbrand, D. Bieri, R. Cortivo, J. Hoehener, and T. Hardjono, “Proposal for a Comprehensive (Crypto) Asset Taxonomy,” in *Proceedings of the 2020 Crypto Valley Conference on Blockchain Technology (CVCBT)*, June 2020. [Online]. Available: https://arxiv.org/abs/2007.11877

[24] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, “Adding Concurrency to Smart Contracts,” in *Proceedings of the ACM Symposium on Principles of Distributed Computing PODC’17*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 303–312. [Online]. Available: https://doi.org/10.1145/3087801.3087835

[25] M. Herlihy, “Blockchains From a Distributed Computing Perspective,” *Communications of the ACM*, vol. 62, no. 2, pp. 78–85, February 2019. [Online]. Available: https://doi.org/10.1145/3209623

[26] IBM, “Four-party credit/debit payment protocol (EU Patent EP1017030A2),” December 1999. [Online]. Available: https://patentimages.storage.googleapis.com/68/7d/68/51b6337757ed7a/EP1017030A2.pdf

[27] A. Lipton, T. Hardjono, and A. Pentland, “Digital Trade Coin (DTC): Towards a more stable digital currency,” *Journal of the Royal Society Open Science (RSOS)*, August 2018, available at https://doi.org/10.1098/rsos.180155.