Travel Rule Information Sharing Architecture for Virtual Asset Service Providers (TRISA)

Version 7

June 23, 2020

Additions since Version 6:

- Provision to support PayID Protocol integration

By: Dave Jevans, Thomas Hardjono, Jelle Vink, Frank Steegmans, John Jefferies, and Aanchal Malhotra
TRISA Overview ................................................................................................................................. 4
FATF Funds Travel Rule .......................................................................................................................... 4
BSA Travel Rule – 31 CFR 103.33(g) .................................................................................................... 5
The Solution: Modify Blockchains or Add an Overlay Layer? ............................................................... 5
What the FATF Travel Rule Requires VASPs to Retain and Share .................................................... 6
Applying the Certificate Authority Model to Reliably Identify and Verify VASPs ...................... 6
Establish Secure Communications Between VASPs ........................................................................... 8
Verifying VASP Identities ....................................................................................................................... 9
Required VASP Identity Fields ............................................................................................................. 9
VASP Verification Questionnaire: TRIXO ............................................................................................ 10
A Standard Process for Verifying VASPs and a Global Directory ..................................................... 11
VASP Identity Certificates .................................................................................................................... 11
Extended Validation Certificates .......................................................................................................... 12
Identity Certificates for VASPs with Extended Validation .................................................................... 12
Transactions Signing Certificates for VASPs ....................................................................................... 13
Certificate Hierarchy for TRISA ............................................................................................................ 13
TRISA Business Information for VASP EV-Certificates .................................................................... 15
Reliable Communications Between VASPs ......................................................................................... 16
Protocol ................................................................................................................................................ 16
Connection Optimization ...................................................................................................................... 17
Mitigating the Risk of Sending Private Information to the Wrong Entity ............................................. 17
Determining by Receiver if Sender Is a VASP ...................................................................................... 19
Automatically Determining a VASP from a Blockchain Address ...................................................... 20
Optimization of the Network ................................................................................................................. 21
Security Concerns ................................................................................................................................ 21
Revocation and Blacklisting ................................................................................................................. 22
Encryption of Transmitted Data .......................................................................................................... 22
Messaging Formats ............................................................................................................................... 23
Encrypted Transaction Envelope .......................................................................................................... 23
Versioning Paradigm ............................................................................................................................ 27
Provision for Batch Processing ................................................................. 28
Provision to Support Account and VASP Identifiers .................................... 29
Verifiable PayID Protocol Integration .......................................................... 30
   Integration of PayID with TRISA for VASPs Flow ................................... 32
   PaymentSetupDetails ............................................................................. 33
   Identity ................................................................................................. 33
   Response .............................................................................................. 34
   PaymentInformation ............................................................................. 35
   addresses .............................................................................................. 36
   addressDetails ...................................................................................... 36
   ProofOfControlSignature ..................................................................... 36
Collaboration on Message Formats ............................................................... 38
Open Source Project ..................................................................................... 39
TRISA Overview

The Travel Rule Information Sharing Architecture (TRISA) was initiated in July 2019 as a response to the FATF and FinCEN emerging regulations around data transfer for cryptocurrency transactions between Virtual Asset Service Providers (VASPs). The goal of TRISA is to enable compliance with the FATF and FinCEN Travel Rules for cryptocurrency transaction identity information without modifying the core blockchain protocols, and without incurring increased transaction costs or modifying virtual currency peer-to-peer transaction flows. TRISA aims to do this on a global level while:

- Protecting user privacy
- Ensuring fast and inexpensive transactions
- Remaining open source and decentralized
- Having an open governance body
- Maintaining interoperability with other approaches

In June 2019 The Financial Action Task Force (FATF) proposed global rules regarding the sharing of beneficiary and originator information between Virtual Asset Service Providers (VASPs), inspired by regulation from the Financial Crimes Enforcement Network (FinCEN) in the US. Entities subject to these regulations are cryptocurrency exchanges, custodial wallets, DEX operators, and others depending on the interpretation of regulations in each jurisdiction.

This whitepaper proposes a peer-to-peer mechanism for complying with these regulations, with minimal cost impact to participants, and with consideration for preserving high performance transaction processing at cryptocurrency virtual asset service providers.

FATF Funds Travel Rule

In June 2019, the Financial Action Task Force (FATF)—an international money-laundering watchdog organization based in Paris—released clarification to its guidance to member nations regarding the regulation of VASPs and other crypto entities. In response to the increasing use of virtual assets for money laundering and terrorist financing, the updated guidance includes a “Travel Rule.” It requires VASPs to share
sender (originator) and receiver (beneficiary) information in cryptocurrency transactions. This is similar to so-called Travel Rules that have for years required financial institutions to share this information when executing bank wire transfers and SWIFT electronic funds transfers.

At the close of a summit on June 29, finance ministers and central bankers of the G20 economic bloc formally announced their support for FATF’s updated cryptocurrency guidelines, which include the Travel Rule. The G20 member countries have now rapidly begun the process of translating the Travel Rule into their respective local laws.

BSA Travel Rule – 31 CFR 103.33(g)

The Bank Secrecy Act (BSA) established a Funds Travel Rule for fiat currency transfers in the US in 1996 and amended it to include electronic funds transfers 2012. FinCEN is charged with enforcing BSA rules, and in May 2019 it released guidance that the US Department of Treasury would classify many cryptocurrency exchanges as money service businesses (MSBs), meaning exchanges operating within the United States must now comply with the BSA Travel Rule. According to the rule, any time a transfer of funds is greater than or equal to $3,000, financial institutions must include the following in the transmittal order: the name, account details, and financial institution of the recipient and the transmitter. The regulation’s text does not dictate exactly how financial institutions must collect, verify or transfer this information.

The Solution: Modify Blockchains or Add an Overlay Layer?

The goal of the Travel Rule Information Sharing Architecture (TRISA) is to enable compliance with the FATF and FinCEN Travel Rules for transaction identity information without modifying the core blockchain and cryptocurrency protocols. Trying to modify the protocols is bound to fail, as there are many different protocols, and forcing hard forks is simply not feasible. A better option involves creating a separate out-of-band mechanism to augment existing blockchains and cryptocurrencies for compliance purposes.

This whitepaper describes a peer-to-peer mechanism for VASPs to comply with the respective Funds Travel Rule for transaction identification exchange between originators and beneficiaries.
What the FATF Travel Rule Requires VASPs to Retain and Share

The FATF guidelines require both sending and receiving VASPs to exchange and store originator and beneficiary identification information in addition to the cryptocurrency addresses and transaction ID for each transaction. Regulators require the latter since cryptocurrency addresses can be used by multiple beneficiary customers. For example, some exchanges use a single address to send all transactions. Also, cryptocurrency addresses can be recycled and consequently may be used by multiple customers at a VASP.

Applying the Certificate Authority Model to Reliably Identify and Verify VASPs

The FATF rule creates a technical challenge for VASPs—how to comply with the requirement to exchange information while still protecting user privacy. The solution requires the equivalent of a certificate authority (CA) that verifies the identity of VASPs and serves as a dictionary for their public key certificates so that they can be identified and establish secure communications between VASPs.

![Diagram of Enhanced Validation Know Your VASP (EV KYV)](image)

*Figure 1: Implementing Enhanced Validation Know Your VASP (EV KYV)*
Receiving VASPs should return receipts, ideally digitally signed, to sending VASPs to confirm that the transaction identity information has been received. It is desirable to be able to reject a transaction in a receipt, for example if the sender’s identity or purported beneficiary’s identity data fails sanctions or other blocking tests by the receiving VASP.

In such cases, the sending VASP should not proceed with the blockchain transaction and should notify the originator of a failed transaction. This enhanced validation standard is known as Extended Validation Know Your VASP (EV KYV). See Figure 1.

In a CA model, one or more third parties verify the identity of a VASP through a number of steps such as email identification, domain name ownership identification, phone call verification, and business paperwork verification. The CA can then issue a digital certificate signed by the CA and the VASP to serve as identification and a way to establish secure encrypted communications with the verified VASP (Figure 2). These certificates should have an expiration date. They should also be subject to revocation by the CA through an OCSP mechanism or revocation list.

Figure 2: A validated certificate X.509 from a certificate authority (CA) protects communications between two VASPs by encrypting the connection between them. In this model, a VASP applies for certification through a registered VASP CA. The CA would then verify that all legal requirements have been met before the VASP can send a certificate signing request (CSR) and the TVASP CA (Trusted VASP Certificate Authority) can produce the signed certificate.
Another approach is to use a mechanism similar to Domain Keys Identified Mail (DKIM) whereby public keys and configuration information regarding where to connect to the transaction identity services are published in the DNS records of VASPs. A problem with this approach, however, is that many VASPs operate multiple domain names for different services, sometimes in different jurisdictions. The CA model provides more oversight and simplification; however, it does require one or more trusted third parties to operate the verification, issuance, and revocation of certificates.

In web connections, the CA is the cornerstone trust for public key infrastructure (PKI), by issuing trusted digital certificates and managing, distributing, and revoking these certificates. It works by using two different cryptographic keys: a public key and a private key. In an TLS interaction, for example, the public key is available to any user that connects with the website. The private key—a unique key generated when a connection is made—remains secret. When communicating, the client app or browser uses the public key to encrypt and decrypt data, while the server uses the private key. This match-up of keys enables the sending and receiving machines to establish a secure connection, which protects users’ information from theft or tampering. In addition, since the CA issues digital certificates that associate an entity with a given public key, this approach ensures users interact with the intended party, not an imposter.

The CA model is one that has been developed and used by the Internet for over 20 years. The model has been successfully used to establish trust across untrusted networks and facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking, and government communications. It provides proven and well-established trust models, audit procedures, issuance, and revocation mechanisms.

**Establish Secure Communications Between VASPs**

When a VASP wishes to send transaction originator and beneficiary information to another VASP in support of the Travel Rule requirements, they must establish secure communication with the other VASP. One way to do this is to identify the VASP, obtain its trusted certificate, communications address and port, and then establish a SSL/TLS
secure connection directly to the receiving VASP. This is similarly to the way that browsers connect securely to web servers.

**Verifying VASP Identities**

Extended Verification SSL (EV SSL) certificates provide an additional layer of trust for web communications beyond that provided by standard SSL certificates. Similar to the way in which an EV SSL Certificate authenticates a website and the entity controlling the website, an EV KYV Certificate contains the following required fields, which are validated by a trusted third party:

**Required VASP Identity Fields**

1. Subject Organization Name – Must contain the full legal name of the entity
2. Registration Number – The unique Registration Number assigned by the Incorporating Agency in the Jurisdiction of Incorporation
3. Address of Place of Business – Must contain the address of the physical location for the Subject. City, state, and country information are required
4. Business Category – Must contain one of the following strings: “Private Organization,” “Government Entity,” “Business Entity,” or “Non-Commercial Entity”
5. Subject Jurisdiction of Incorporation or Registration – The Jurisdiction of Incorporation or Registration
6. Domain name – Must contain one or more host domain name(s) owned or controlled by the Subject for association with the Subject’s publicly accessible server

Mutual authentication can be facilitated over the connection protocol by the originating VASP providing their identification certificate to the receiving VASP, which is then verified during the secure communication session establishment and checked for revocation status. In this way, both VASPs are certain the counterparty on the other end of the connection trusted.
VASP Verification Questionnaire: TRIXO

The biggest challenges in developing and maintaining a global VASP network are the ability to reference and validate VASPs and identify which protocols each VASP supports. The initial validation of a VASP must include verification of business identity, regulatory jurisdiction, control over URLs and email addresses, and risk ratings. VASP validation must also be available on an on-going, real-time basis to notify if a VASP has lost its certification as a regulated entity, as well as theft and fraud incidents.

The TRISA working group has worked with the GDF (Global Digital Finance) working group and VASPs to develop a VASP verification questionnaire that can be used across various protocols and technical implementations.

The TRIXO VASP questionnaire comprises of:

<table>
<thead>
<tr>
<th>TRIXO- VASP Questionnaire</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entity Details</strong></td>
<td></td>
</tr>
<tr>
<td>Full Legal Name</td>
<td></td>
</tr>
<tr>
<td>DBA</td>
<td></td>
</tr>
<tr>
<td>Full Legal (Registered) Address</td>
<td></td>
</tr>
<tr>
<td>Full Primary Business Address (if different from Registered Address above)</td>
<td></td>
</tr>
<tr>
<td>Date of Entity Incorporation / Establishment</td>
<td></td>
</tr>
<tr>
<td>Incorporation Number</td>
<td></td>
</tr>
<tr>
<td>EIN or Legal Entity Identifier (LEI), if available</td>
<td></td>
</tr>
<tr>
<td>Does the Entity need to send Travel Rule Information?</td>
<td></td>
</tr>
<tr>
<td>Name of the Entity’s primary financial regulator / supervisory authority</td>
<td></td>
</tr>
<tr>
<td>Please provide a list of national jurisdictions, other than the primary national jurisdiction, where the Entity has been granted licenses or other approvals to operate and the name of the regulator / supervisory authority</td>
<td></td>
</tr>
<tr>
<td>Entity’s License / Registration Number(s) for each jurisdiction it operates in</td>
<td></td>
</tr>
<tr>
<td>Is the Entity required by local regulations to safeguard customer PII?</td>
<td></td>
</tr>
<tr>
<td>Are you compliant with GDPR / CCPA?</td>
<td></td>
</tr>
<tr>
<td>Is the Entity permitted to send and/or receive transfers of virtual assets in the jurisdictions in which it operates?</td>
<td></td>
</tr>
<tr>
<td>Does the Entity have a customer identification program?</td>
<td></td>
</tr>
<tr>
<td>At what threshold does the Entity conduct customer identification before permitting them to any and all send / receive virtual asset transfers?</td>
<td></td>
</tr>
<tr>
<td>What is the minimum threshold above which the entity is required to collect / send Travel Rule information?</td>
<td></td>
</tr>
<tr>
<td>What is the Endpoint or URL for sending and receiving Travel Rule information? If more than one, list them</td>
<td></td>
</tr>
<tr>
<td>Does the Entity have a program that sets minimum AML, CFT and Sanctions standards per the requirements of your jurisdiction’s regulatory regime regarding the following components?</td>
<td></td>
</tr>
<tr>
<td>If Yes, please specify the applicable regulation</td>
<td></td>
</tr>
<tr>
<td>Name, email and phone number of travel rule contact</td>
<td></td>
</tr>
</tbody>
</table>
A Standard Process for Verifying VASPs and a Global Directory

VASP Identity Certificates

One of the key requirements for the establishment of secure messaging between two VASP entities is the correct identification of the endpoints of the corresponding VASPs. Although a VASP may possess a private-public key pair used to establish a secure channel (e.g. SSL/TLS session) with another VASP, the possession of the private-key itself is insufficient to prove that the VASP is in fact connecting to the entity with whom it seeks to transact. A secure channel is needed when transferring customer information as mandated by the Travel Rule.

Figure 3: Use of Digital Certificate to Identity VASPs during Secure Channel establishment.

The purpose of Identity Certificates is to provide unambiguous binding between a public-key and the entity who (legally) owns the private-public key pair. The most common format for digital certificates is the X.509 Standard (ISO9594 [1]). A digital certificate that carries a copy of the public-key of the subject (i.e. the VASP or entity
being issued the certificate), as well as other standard fields. The certificate itself is issued (signed) by a certification authority (CA) [2,3].

**Extended Validation Certificates**

For entities requiring additional business-related information not included by the base X.509 digital certificate, extended validation (EV) certificates can be used to obtain, validate, and represent additional relevant information. These certificates are typically issued to subjects (entities) who are legal entities (versus individual persons) and are subject to verification by the certificate issuer.

The purpose of extended validation (EV) of subjects (entities) are:

- **Identification of the entity (subject) who controls a given domain** (i.e. website): Provide assurance to the end-user that entity controlling a domain website (e.g. merchant) is a true legal entity. This involves incorporating various legal information into the EV certificate (e.g. business name & address, LEI number, etc.).

- **Assurance of the endpoint for SSL communications**: Provide assurance that an endpoint is the correct endpoint in an SSL secure channel establishment – which involves performing a standard key-exchange handshake using the X.509 certificate bound to that endpoint.

The TRISA community will define a certificate profile for EV-certificates for VASPs.

**Identity Certificates for VASPs with Extended Validation**

For Virtual Asset Service Providers (VASPs) there are at least two basic needs for EV certificates:

(a) **VASP-to-VASP messaging protection**: When an originator-VASP transmits messages (e.g. off-chain) to a beneficiary-VASP, these messages must be signed (for source authenticity) and must be delivered over a secure channel. To ensure that each VASP obtains assurance that the destination VASP is the correct legal entity, EV certificates can be employed by both VASPs. Here the intended
consumer of the business information (contained in the EV certificate) are VASPs themselves.

(b) **VASP identification for customers using browser and wallet connections:** The second requirement for EV certificates for VASPs follows the classic usage of EV certificates—namely to enable customers using a browser or a wallet to know that he or she is connecting to the desired VASP.

This family of certificates is referred to as **VASP Identity Certificates**, which are issued by the TRISA organization.

**Transactions Signing Certificates for VASPs**

In various circumstances, VASPs may transmit transactions to the blockchain which are signed using the VASP’s own key-pair.

To maintain strong security practices, it is beneficial to use different private-public keys for the identification of entities (e.g. an SSL/TLS channel) from that of transactional digital signatures from the entity (e.g. signing documents, contracts, etc.).

For VASPs, such private-public keys are referred to as transaction signing-keys, and the corresponding certificates are transaction **signing-key certificates**. The sole purpose of this certificate is to certify the ownership of the private-public keys owned by the VASP.

A given VASP may own multiple transactions signing-keys and multiple signing-key certificates.

**Certificate Hierarchy for TRISA**

Acting as a common industry organization, TRISA streamlines the validation of identity certificates between VASPs. Similar to the Browsers/SSL community and the Cable Modem industry, the TRISA organization acts as the certification authority (CA) for the certificates issued in the VAPS messaging network. This is shown in

**Figure 4**, where TRISA becomes the Root-CA for the certificates issued to all VASPs in the TRISA organization.
There are a number of benefits from this configuration based on a common TRISA certification authority. Some of the benefits include:

- **Certificate issuance based on a common legal framework**: As a certification authority, the TRISA organization will publish Certification Practices Statement (CPS) document, which is a form of a legal service-level agreement to which all TRISA members must abide by. This allows VASPs to develop their respective business model based on a common set of certificate processing procedures (the “plumbing”), thereby allowing VASPs to focus on the higher-layer aspects of their business where revenue can be found.

- **Efficient VASP certificate-status check**: On the operational side, by sharing a Root CA a VASP can very quickly and efficiently validate the status of the certificate of another VASP. This allows a secure channel between VASPs on the messaging network to be established, allowing VASPs to quickly exchange customer information as required by the Travel Rule.

*Scalability through bridging with other VASP jurisdictions*: When different communities of VASPs in different jurisdictions observe the Travel Rule, there are often additional constraints that implemented by the local regulations in their
respective jurisdictions. When each community employs a certificate-hierarchy based in an organizational Root CA, the communities can establish a shared *bridge* across the two Root CAs, allowing VASPs in the respective communities and jurisdictions to quickly and efficiently cross-validate the status of VASPs.


In order to ensure a high degree interoperability among a community of users (i.e. VASPs), typically a *Certificate Profile* for the community is defined. Among others, the profile defines the base aspects of the certificate that the community agrees to employ (e.g. cipher type, hash function, format of timestamp, etc.).

For Certificate Authorities (CA) who wish to legally issue certificates conforming to one or more certificates profiles, the CA must publish a *Certificate Practices Statement* (CPS) document. A given CPS is formal statement that explains the methods and practices which a CA employs in the issuance, suspension, revocation, and the renewal of certificates and providing access to them, in accordance with specific requirements (i.e., requirements specified in this Certificate Policy, or requirements specified in a contract for services) [5].

We anticipate that Certificate Authorities within TRISA will publish a CPS document, including for X.509 EV certificates and signing-key certificates used within the TRISA network.

**TRISA Business Information for VASP EV-Certificates**

For a VASP, an Extended Validation (EV) SSL Certificate authenticates: the VASP’s website (by domain name), the VASP connection endpoints, and the VASP entity controlling the website.

Under TRISA, the EV Certificate for VASPs (referred to as the Subject) includes the following required fields [6]:

- **VASP Organization name**: The Organization field must contain the full legal name of the entity controlling the VASP website as listed in the official records in the VASP’s Jurisdiction or as otherwise verified by the CA according to the TRISA EV Guidelines.
Reliable Communications Between VASPs

In the case where peer-to-peer communication of transaction identity information is used to satisfy Travel Rule requirements, it is essential that communication between VASPs is reliable. To that extent, sending VASPs must ensure that their systems can reliably re-try and resend information if the receiving party’s servers are unreachable, or they do not receive a transmission receipt notification.

The protocol to send transaction identity information must also include a receipt that is provided by the receiver to the sender to prove that the data has been received. The receiver should timestamp and digitally sign this receipt and include the hash of the identity information that was sent, so that the sender can store this for non-repudiation purposes in case of a future audit.

Similarly, receivers should not deposit received funds into the account of a beneficiary until the transaction identity information is received from the sending VASP. This brings up the challenge of the receiving VASP determining if an inbound transaction is from a VASP or from an individual private wallet, which is not required to provide transaction identification information under the FATF Travel Rule.

Protocol

The protocol for sending transaction identity information should be:

1. Establish a secure, mutually authenticated SSL/TLS connection between VASPs by the originator to assure privacy of data in transit
2. Originator posts a transaction identification message
3. Receiver posts a signed receipt
4. Originator posts transaction to a blockchain and receives a transaction ID
5. Originator posts the transaction ID to the receiver
**Connection Optimization**

Because establishing a new SSL/TLS-authenticated session between VASPs for every single transaction could prove to be overly costly in computation for key exchange and session establishment, it is acceptable to keep a connection open and exchange data for multiple transactions over a single connection. This is similar to how a browser keeps HTTPS web connections open for accessing multiple web pages in a single connection with a web server.

**Mitigating the Risk of Sending Private Information to the Wrong Entity**

The simple way to do this is to ask the sending user if they are sending to a VASP, and if so, which one. VASPs such as Coinbase have used this mechanism for several years. It does rely on trusting the user, but it is the simplest way to implement compliance by senders. To fully comply with the Travel Rule requirements, the user would also have to enter the beneficiary’s information before the transaction can proceed, so that this can be stored by the originating VASP and also communicated to the receiving VASP.

This poses the danger that the user can claim the wrong VASP as the intended destination. For example, a user claims that they are sending to VASP CoinAA when they are really sending to VASP CoinBB. Assuming both VASPs are legitimate and are registered in the system, then the sending VASP would establish a connection to CoinAA and would send the user and beneficiary’s information to CoinAA. This discloses this private data to the wrong VASP, and puts the actual receiving VASP, CoinBB, in a compliance violation.

Mitigating this risk is done by verifying that the receiving address is actually controlled by the declared beneficiary VASP. This requires a high-speed lookup whereby the sending VASP can query the beneficiary VASP about the address and confirm that the receiving address actually belongs to that VASP (VASP Address Confirmation protocol). Coinbase and several others call this “Proof of Control,” which requires the queried entity to response with a *challenge*, using the private key of the queried entity to sign a randomly generated number with the private key corresponding to the cryptocurrency address. This allows the querying VASP to ensure that the receiving VASP does in fact control the address. One challenge here is that this must be implemented across multiple blockchains, each of which may have different public-private key systems.
Potential risk and considerations:

1. Transactions should not be posted to the blockchain until the receiving VASP confirms the receiving address. This can delay transaction delivery if the lookup mechanism goes down for any reason.
2. Smaller VASPs would have to ensure 7x24 high availability of this mechanism, unlike today where VASPs can operate in batch mode.
3. It could allow mining of addresses to map them to VASPs.
4. It will create errors and delays for users if there is not a match, or the matching is delayed for several minutes. In such cases, the originating VASP would notify the user that there is no match. This could cause much customer support overhead if in fact it was due to the receiving VASP having a maintenance delay or other problem with the mechanism, when in fact it really is a match.

**Figure 5:** Communication between exchanges begins when the originating exchange establishes a secure connection through CA.

The diagram in Figure 5 demonstrates how mutual authentication can be facilitated over the connection protocol as the Originator VASPs (here called “Alice’s VASP” as a real-world example) also provides their identification certificate to the destination exchange (here Bob’s VASP). Once a secure connection is established, the Originator VASP can
then initiate the virtual asset transfer to the Destination VASP along with the information required under the Travel Rule. To prove it has received the data, the Destination VASP sends a timestamped and digitally signed receipt that includes the hash of the identity information that was sent. The Originator VASP must keep this information to meet its record-keeping obligations under FATF recommendations.

**Determining by Receiver if Sender Is a VASP**

A somewhat more complicated problem is how a receiving VASP, who gets an inbound transaction to one of their addresses, can determine if the inbound transaction is from a VASP or not. For full compliance if the inbound transaction is from a regulated VASP the receiving VASP should not make funds available to the beneficiary until the Travel Rule transaction identity information is received and recorded.

One approach would require the receiving VASP to wait for a period of some minutes, and then if no transaction identity information is received to assume that it is from a private individual and not another VASP. The VASP would record it as such, and only then make the funds available to the beneficiary. This delay could be due to problems with the transaction identity system by either the sender or receiver’s systems, which operate separately from the underlying blockchain.

Lack of clarity surrounding transaction identities, VASP or private individual, could result in slowed transaction speed for as VASPs attempt to comply with Travel Rule, and ultimately a violation of the guidance.

Sending VASPs should always ensure they receive a signed receipt of transaction identity information from a receiving VASP before transactions are placed onto a blockchain. This can complicate the processing systems at sending VASPs because multiple transfers of value are typically batched up into a single transaction in order to reduce blockchain processing fees. That workflow would have to be reworked to only queue transfers to happen once the Travel Rule processing has been confirmed.
Automatically Determining a VASP from a Blockchain Address

We propose that originators with cryptocurrency stored at a VASP declare that they are sending funds to another VASP rather than to an individually managed address. This mechanism has been used by VASPs such as Coinbase for several years. However, some parties have requested the ability to automatically reference a receiving address to verify if it is hosted by a VASP and not require users to self-declare this. This capability can be provided in two ways:

1. By using the VASP Address Confirmation peer-to-peer protocol to test every VASP to see if a new beneficiary address belongs to that VASP. An implementation of this “data mining” approach is highly undesirable, as it tests every address against every VASP until a hit is found. The process can be optimized by storing address-to-VASP correlations for addresses that are reused by beneficiaries. The following section discusses optimization of the protocol and detection and revocation of data miners in more detail.

2. Encourage VASPs to publish address-to-VASP mappings to a high-speed blockchain when those addresses are created. VASPs can post hashes of addresses rather than actual addresses, providing a modicum of privacy, however it does not protect against data mining of address-to-VASP relationships. This approach requires a very high speed blockchain with minimal confirmation times (seconds) in order to avoid delaying the sending of transactions. If performance of that system is slow, address-to-VASP mapping would become unreliable. Another challenge is that this blockchain would contain every beneficiary address and its associated VASP for all blockchains. It seems like a very heavyweight approach with the aforementioned reliability issues. A more efficient mechanism involves using a centralized service to provide this functionality. Such a service could expire address-to-VASP mappings over time and throttle data mining attempts but would constitute a central point of failure.

The peer-to-peer discovery mechanism is the preferred approach for the reasons of privacy, decentralization, performance and reduction of impact on existing transaction workflows.
Optimization of the Network

If sending VASPs do not want to trust their customers to declare that funds are being sent to another VASP or a personal wallet, then they need to query other VASPs to confirm if they control the receiving address and associated account. If done randomly, then we can expect each sent transaction to require an API call to 50% of the VASPs. With over 300 active VASPs, this would be inefficient.

Transaction analysis of hundreds of millions of transactions shows that there are trading clusters where exchanges tend to send and receive transactions between a small group of other exchanges. In fact, as much as 60% of transactions from one exchange can be with two other exchanges.

Sending VASPs can employ caching to optimize automated discovery of the receiving VASP without the input of the end user customer. This cache is simply the ordered list of the most frequent VASPs that receive funds. More elaborate caching can be performed on a per-user basis as well. In such a case, queries required prior to sending a transaction can be reduced by well over 90% for a typical exchange or hosted wallet provider.

Certificate authorities can further optimize the network by delivering VASPs a prioritized list of exchanges to query, based on transaction flows.

Security Concerns

This type of system would create several new security concerns for the VASP industry.

Since the transaction identity sending and receiving services must be online and highly available (7x24), these services are particularly vulnerable to security breaches and attack. A distributed denial of service (DDoS) attack could take a VASP’s entire transaction capability offline, and a large-scale attack on the transaction identity services of major exchanges could take the entire industry offline. Today, it is only possible to take a service’s user interface offline with a DDoS attack because back-end
transaction processing that interfaces with blockchains is typically separated from the visible interface.

Once a VASP implements Travel Rule compliant data exchange and storage the VASP has massive amounts more personal data to protect from data breaches. Today, VASPs only store the personal data of their customers, and only in one location. That data can be stored offline and encrypted in extreme cases. However, with travel rule data requirements, every VASP will have originator and beneficiary data for every transaction. This means that VASPs will find themselves storing the personal information not only of their customers but also of everyone who ever sends them funds.

**Revocation and Blacklisting**

Certificate Authorities must provide a revocation service to remove VASPs from the trusted counterparty list. The reasons for revoking a VASP can include bankruptcy, fraud, criminal activities, or sanctions. Both Certificate Revocation Lists (CRLs) and real-time Online Certificate Status Protocol (OCSP) lookups should be supported. OCSP lookups can be performed on a transaction level.

OCSP lookups can also provide blacklisting of malicious VASPs that are trying to data mine counterparty address information. For example, if a VASP decides to try and find which exchanges hosts every address on a blockchain, this lookup data can be reported to the CAs, and then used to create revocation data and update a blacklist.

**Encryption of Transmitted Data**

In a solution where data is encrypted during transmission between VASPs, using SSL/TLS connections eliminates the further data encryption of originator or beneficiary information during transit. The onus for checking transaction identity data for sanctioned or suspicious persons or entities falls on both sending and receiving VASPs. Thus, they must access this information in plaintext. Naturally, once checked, VASPs should store the information in an encrypted database, but the data must remain accessible at any time for filing CTR, SAR, for audits, and financial investigations.
Messaging Formats

TRISA defines a trust architecture and peer-to-peer model for compliance with the Travel Rule. In Version 4, TRISA did not define message formats for sending the actual compliance data (account number, name, geographic address, beneficiary name and account number). TRISA V5 defined message formats as follows. These are subject to review and modification based on collaboration with GDF and Digital Chamber collaboration.

The peer-to-peer exchange protocol is defined in two different layers:
1. The wire protocol itself, which defines the API to exchange the Transaction Data envelop
2. The Transaction Data itself is composed of two distinct parts:
   a. Identity Information
   b. Blockchain Data

All APIs follow a clear versioning paradigm to allow for future expansion and integrations. When integrating with TRISA, maintaining additional libraries, or SDK’s, you can verify on the TRISA documentation website which versions are current or deprecated.

TRISA V6 uses IVMS101 as the message contents protocol. IVMS is an industry standard for exchanging Travel Rule information. See the appendix for more information. IVMS101 has been integrated with the TRISA source code as of June 1st, 2020.

Encrypted Transaction Envelope

![Figure 6]
The peer-to-peer TRISA protocol is defined under `proto/trisa/protocol`. It defines the RPC exchange endpoint and the Transaction Data envelope. Next to the Identity and Blockchain Data, the envelope applies additional encryption of the message and an HMAC digital signature.

Although all messages are exchanged over a secure channel, by adding this additional layer VASP’s can securely store the full Transaction Envelope at rest in a backend of their choice while maintaining full repudiation of the exchange.

https://github.com/trisacrypto/trisa/blob/master/proto/trisa/protocol/v1alpha1/trisa.proto

```protobuf
// Peer-to-peer Exchange Service for the Transaction Envelope
service TrisaPeer2Peer {
  rpc TransactionStream(stream Transaction) returns (stream Transaction) {} 
}

message Transaction {
  // The transaction identifier generated by the sender. Any response
  // to a transaction request needs to carry the same identifier.
  string id = 1;

  // Encrypted TransactionData
  bytes transaction = 2;

  // Encryption key used to encrypt the transaction blob. This key itself
  // is encrypted using the public key of the receiver.
  bytes encryption_key = 3;

  // The encryption algorithm used to encrypt the transaction blob.
  string encryption_algorithm = 4;

  // HMAC signature calculated from encrypted transaction blob.
  bytes hmac = 5;

  // The HMAC secret used to calculate the HMAC signature. This secret
  // itself is encrypted using the public key of the receiver.
```
bytes hmac_secret = 6;
    // The algorithm used to calculate the HMAC signature.
    string hmac_algorithm = 7;
}

// Transaction Data composed of both Identity and blockchain Data
message TransactionData {
    // Identity contains any valid identity structure.
    google.protobuf.Any identity = 1;

    // Data contains the network specific data.
    google.protobuf.Any data = 2;
}

The Transaction Data is split into two section to be able to maintain and version them separately:

1. The different fields for the Identity section depend on the country/region of the subject being described in it. TRISA suggests using ISO 3166-1 alpha-2 country codes to maintain the different formats. If any subdivision is needed, this can be easily accomplished using dedicated region suffixes (i.e. “us-ca”).

2. The Blockchain data allows for specific properties defining the transaction based on the specific blockchain used. Although there might be some common ground among different blockchains, from a protocol perspective TRISA suggest maintaining clear and dedicated formats per blockchain type.

Note that next to the versioning of the wire protocol itself, both Identity and Blockchain Data sections maintain their own versioning. Once a message has been vetted by the community it will become stable and should be accepted by any production client.

At its current state, TRISA defines Identity examples for “us” (United States of America) and “be” (Belgium), see https://github.com/trisacrypto/trisa/tree/master/proto/trisa/identity.

    proto/trisa/identity/us/v1alpha1

message Identity {
    string first_name = 1;
string last_name = 2;
string ssn = 3;
string state = 4;
string driver_license = 5;
}

proto/trisa/identity/be/v1alpha1

message Identity {
  string first_name = 1;
  string last_name = 2;
  string national_number = 3;
  string city_of_birth = 4;
}

For the Blockchain specific data there are two examples for “bitcoin” and “ethereum”, see https://github.com/trisacrypto/trisa/tree/master/proto/trisa/data.

proto/trisa/data/bitcoin/v1alpha1

message Data {
  string source = 1;
  string destination = 2;
  string amount = 3;
}

proto/trisa/data/ethereum/v1alpha1

message Data {
  string source = 1;
  string destination = 2;
  string amount = 3;
}

Based on community feedback, TRISA would like to expand on all different Blockchains and different country Identity definitions. Note that the above Identity and Blockchain data are examples and need to be formalized and vetted by the TRISA community.
**Versioning Paradigm**

Each message starts at \textit{v1alpha1}. When the TRISA community vetted an API for prime time, it gets bumped to \textit{v1}. If needed any intermediate version like \textit{v1alph2}, \textit{v1alpha3} or even \textit{v1beta1}, … can be applied. TRISA is using gRPC which allows for compatible schema evolution as not all clients will get updated at once.

By decoupling the wire protocol from the messages TRISA can evolve much easier, depending on the introduced changes.

Message formats are:

- **Does Cryptocurrency Address belong to this VASP?**
  - Cryptocurrency type (chain)
  - Cryptocurrency address

- **Sending Transaction Identity Information**
  - Originator Name
  - Originator Address
  - Originator State
  - Originator Account Number
  - Beneficiary Name
  - Beneficiary Account Number

- **Receipt**
  - Receipt ID tied to Transaction Identity Information Send

- **Transaction Confirmation**
  - Receipt ID
  - Blockchain Transaction Number
Here are the initial message formats for exchanging identity information.

```plaintext
message Identity {
    string first_name = 1;
    string last_name = 2;
    string ssn = 3;
    string state = 4;
    string driver_license = 5;
}
```

```plaintext
message Identity {
    string first_name = 1;
    string last_name = 2;
    string national_number = 3;
    string city_of_birth = 4;
}
```

The TRISA transaction ID which is in the Transaction message ("id" field) needs to be present in each message exchange. This is how both parties can correlate messages:

1. Originator --> sends transaction identification - the "TRISA transaction id" is generated by the originator
2. Receiver --> validates the wallet, sends back receipt - this message uses the same "id" as received from #1
3. Originator --> correlates using this TRISA transaction "id" to which the response belongs
4. After putting the TX on the blockchain, originator send the confirmation message containing both TXID + TRISA transaction id
5. Receiver again uses the TRISA transaction id to know to which message exchange this confirmation belongs

---

**Provision for Batch Processing**

Many VASPs—primarily exchanges who have large daily transaction volumes—are looking to retrofit Travel Rule compliance onto existing cryptocurrency transaction flows. The idea is to support batch transaction processing at the end of a day of trading
and transactions. This can be done without impacting the Straight Through Processing (STP) of existing VASP data processing and transaction processing pipelines.

For example, some VASPs have optimized blockchain transactions to group 50-200 payments in a single blockchain transaction, which dramatically reduces transaction costs. VASPs can use TRISA to batch the originator and recipient data and not interrupt their existing optimized payment flows. However, there are two issues with this approach that are not solved in this version (V6) of the TRISA whitepaper:

- **Funds availability:** According to FinCEN, batch processing is acceptable, however, funds received cannot be delivered to the recipient until the corresponding originator information has been provided and scanned for sanctions and risk compliance. This can delay the availability of inbound cryptocurrency payments to customers for many hours.

- **Transaction finality:** FATF, FinCEN and other guidance and regulation allows VASPs to process private transactions without Travel Rule compliance. VASP-to-VASP transactions, however, must send originator and beneficiary information. This raises the problem of how a VASP is to know that an outbound or inbound blockchain transaction is from a private wallet, or from another VASP. In a world of batch processing, there is no current solution for a VASP to determine if a payment inbound or outbound is to or from a private wallet in a deterministic fashion. This argues for real-time address detection. The TRISA community is still researching this issue. Perhaps timeouts (12 hours for example) will suffice. Otherwise a real-time verification and processing model is preferred, as this preserves one of the most valued capacities of cryptocurrencies—near real-time transfer of funds globally with provable receipt of said funds.

### Provision to Support Account and VASP Identifiers

TRISA supports multiple protocols which can be TRISA, OpenVASP, Shyft, BIP75, PayID or others. The Travel Rule message standard can be one of these message formats.
TRISA supports the inclusion of multiple account and VASP identifiers. As of the date of this whitepaper, the TRISA team has worked to incorporate the OpenVASP VANN number as an identifier of the VASP and the account holder. OpenVASP VANN numbers are similar to IBAN numbers and require a customer to input the number when sending.

The TRISA community is working with other industry efforts that are defining payment IDs, that can be tied to internet domain control, and engage their own payment protocols once the identity of the counter-party is verified.

Some VASPs want to only provide an interface that customers understand today, which is send to a Bitcoin or Ethereum address, versus having a separate address for VASP-to-VASP payments. This requires the discovery algorithm that TRISA proposes. Some VASPs are fine to request customers to enter VASP-specific account numbers (such as OpenVASP VANN or other forthcoming proposals which we will reveal in v7 of this whitepaper).

TRISA has been working with OpenVASP and other industry efforts to integrate these notions of a universal payment identifier that is not tied to cryptocurrency addresses.

**Verifiable PayID Protocol Integration**

Verifiable PayID protocol allows for secure and private out-of-band mechanism to retrieve payment addresses corresponding to PayID. Integrating PayID protocol into the TRISA flow enhances the protocol in several aspects:

1. **Determining by Sender if Receiver is a VASP: The Risk of Sending Private Information to the Wrong Entity**

   In the TRISA flow, the sending user provides the payment address to the Sender VASP. It is not possible for the Sender VASP to know from the payment address the Receiver VASP without blockchain analytics. TRISA suggests two solutions:
   - Asking the sending user if they are sending to a VASP.
   - VASP address confirmation protocol.
Both the above approaches currently have constraints as described in the respective sections.

Some of the potential ways that PayID can solve these constraints are:

Adding “PayID domain” in the EV KYV identity certificate. This:
   a. Allows the Receiver VASP to send the signed payment address to the Sender VASP. This proves that the identity who signed this address (i.e. the Receiver VASP) provided this payment address.
   b. Allows the Receiver VASP to send “proof of control signature” to the Sending VASP to prove the ownership of the private key corresponding to the payment address.

The above two proofs together tie the ownership of a private key for an on-ledger payment address to the identity of the Receiver VASP.

2. Determining by Receiver if Sender is a VASP:

This is a challenging problem when sending to a ledger address since the address is not actively provided by the Receiver VASP but rather is sent by the Sender to the Sending Institution. From the perspective of the Receiver VASP, determining if the Sending Institution is a VASP and any corresponding compliance implications is cumbersome at best.

When a payment is instead sent to a PayID, the on-ledger payment address to make the transaction is sent by the Receiver VASP and signed with their private key (that identifies the VASP) after determining if the sending side is a VASP or not.

A corollary benefit of PayID is that it precludes both false positives and false negatives, regardless of what combination of VASPs and non-VASPs are involved in a transaction. That is, a payment to a PayID will definitively determine the counterparty without ambiguity. This is impossible in a payment to an on-ledger address unless every VASP participates in the same compliance system.
3. PayID enhances the compliance screening and privacy of TRISA because the blockchain address to make the payment is only sent by the Receiver VASP to the Sending VASP after:
   a. Receiver VASP and the Sending VASP have verified each other’s identity and have decided to proceed with the transaction.
   b. Each side has received the required Travel Rule information about the Sender and the Receiver.

Integration of PayID with TRISA for VASPs Flow

The participating entities i.e. the Sending and Receiving institutions MUST acquire the following three certificates:
- EV KYV Identity Certificates
- Transactions Signing Certificates for VASPs
- Web PKI domain certificate (Non-VASP certs)

PayID – TRISA integration

Figure 7
The integrated protocol flow begins at the Sending VASP as a PayID client. The prerequisite is Sender issues a Payment Request that contains the Receiver’s PayID and the transaction amount along with the other meta-data to the Sending VASP. The Sending VASP resolves the PayID URI to VASP’s URL as described in the PayID discovery protocol.

1. The Sending VASP establishes a secure, mutually authenticated TLS 1.3 connection (non-VASP web PKI certificate) with the receiving endpoint.

2. If the TLS session is successfully established, Sending VASP (PayID client) generates the signed PaymentSetupDetails message. Among other optional fields, the body of the message MUST contain
   a. isVASP field set to true to indicate to the receiving endpoint that the sending endpoint is a VASP.
   b. transactionAmount field set to the amount of intended payment

### PaymentSetupDetails

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>identity</td>
<td>optional</td>
<td>Identity</td>
<td>TBD</td>
</tr>
<tr>
<td>isVASP</td>
<td>required</td>
<td>boolean</td>
<td>Indicates if the Endpoint is a VASP. Defaults to false</td>
</tr>
<tr>
<td>transactionAmount</td>
<td>optional</td>
<td>integer</td>
<td>Amount of intended payment</td>
</tr>
<tr>
<td>currency</td>
<td>optional</td>
<td>string</td>
<td>ISO currency code</td>
</tr>
<tr>
<td>scale</td>
<td>optional</td>
<td>integer</td>
<td>Orders of magnitude necessary to express one regular unit of the currency e.g. a scale of 3 requires an amount of 100 to equal 1 US dollar</td>
</tr>
<tr>
<td>memo</td>
<td>optional</td>
<td>string</td>
<td>Specifies additional metadata</td>
</tr>
</tbody>
</table>

### Identity

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fullLegalName</td>
<td>required</td>
<td>string</td>
<td>Full legal name of the Sending Endpoint</td>
</tr>
<tr>
<td>postalAddress</td>
<td>required</td>
<td>string</td>
<td>Principal place of Business Address of the Sending Endpoint</td>
</tr>
</tbody>
</table>
Then Sending VASP sends an HTTP POST request with path parameter /payment-setup-details to the receiving endpoint (PayID server) with the appropriate request headers as described in the HTTP request and response headers.

3. Upon receiving this PaymentSetupDetails message, the receiving endpoint (PayID server) parses the message body for isVASP field to check if the sending endpoint is a VASP. The receiving endpoint (PayID server/beneficiary VASP) generates a signed Response message that includes
   a. “Travel Rule” as a compliance requirement in the list of complianceRequirements field.
   b. isVASP field set to true to indicate to the Sending VASP that the Receiving endpoint is a VASP.
   c. redirectURI field set to redirect URI to redirect PayID client (Sending VASP) to TRISA server to initiate the TRISA flow.
   d. An empty paymentInformation field. The Beneficiary VASP MUST NOT send the payment address information yet.

**Response**
This message is sent by the Receiving Endpoint in response to the PaymentSetupDetails message sent by the Sending Endpoint.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/ Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>required</td>
<td>string</td>
<td>The value of this field is the UUID as described in RFC 4122</td>
</tr>
<tr>
<td>identity</td>
<td>required</td>
<td>Identity</td>
<td>TBD</td>
</tr>
<tr>
<td>isVASP</td>
<td>required</td>
<td>boolean</td>
<td>Indicates if the Endpoint is a VASP. Defaults to false.</td>
</tr>
<tr>
<td>redirectURI</td>
<td>optional</td>
<td>string</td>
<td>A redirect URI for PayID client</td>
</tr>
<tr>
<td>transactionAmount</td>
<td>optional</td>
<td>integer</td>
<td>Amount of intended payment</td>
</tr>
<tr>
<td>currency</td>
<td>optional</td>
<td>string</td>
<td>ISO currency code</td>
</tr>
<tr>
<td>scale</td>
<td>optional</td>
<td>integer</td>
<td>Orders of magnitude necessary to express one regular unit of the currency e.g. a scale of 3 requires an amount of 100 to equal 1 US dollar</td>
</tr>
<tr>
<td>expirationTime</td>
<td>required</td>
<td>integer (milliseconds from epoch)</td>
<td>This message is considered void and payments MUST NOT be made on the</td>
</tr>
<tr>
<td>Field name</td>
<td>Required/Optional</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------</td>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>paymentInformation</td>
<td>required</td>
<td>PaymentInformation</td>
<td>Contains details as to how a payment can be made to the Beneficiary. Defaults to empty.</td>
</tr>
<tr>
<td>complianceRequirements</td>
<td>required</td>
<td>string[]</td>
<td>List of the regulatory requirements that the Beneficiary must satisfy during the proposed transaction. Allows the client to send relevant compliance data corresponding to the data in this field. e.g. Travel Rule data in case of Travel rule compliance requirement. Defaults to empty list</td>
</tr>
<tr>
<td>previousMessage</td>
<td>required</td>
<td>string</td>
<td>This is the previous PaymentSetupDetails message received from the Sending Endpoint.</td>
</tr>
<tr>
<td>memo</td>
<td>optional</td>
<td>string</td>
<td>Specifies additional metadata to a payment</td>
</tr>
</tbody>
</table>

**PaymentInformation**

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addresses</td>
<td>required</td>
<td>Address[]</td>
<td>The value of this field is an array of one or more JSON objects of type addresses</td>
</tr>
<tr>
<td>proofOfControlSignature</td>
<td>optional</td>
<td>ProofOfControlSignature</td>
<td>The value of this field is a JSON object as described in ProofOfControlSignature. This is the digital signature proving ownership of the on-ledger address</td>
</tr>
<tr>
<td>identity</td>
<td>optional</td>
<td>string</td>
<td>This field may specify any additional identity information about the PayID owner or PayID server. See here.</td>
</tr>
<tr>
<td>payId</td>
<td>optional</td>
<td>string</td>
<td>The value of this field is the PayID URI in the client request that identifies the payment address information</td>
</tr>
<tr>
<td>memo</td>
<td>optional</td>
<td>string</td>
<td>Specifies additional metadata corresponding to a payment</td>
</tr>
</tbody>
</table>
addresses
This is a required field in the PaymentInformation message

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>paymentNetwork</td>
<td>required</td>
<td>string</td>
<td>The value of this field is the payment-network as specified in the client request's &quot;Accept&quot; header (e.g. XRPL)</td>
</tr>
<tr>
<td>environment</td>
<td>optional</td>
<td>string</td>
<td>The value of environment as specified in the client request's &quot;Accept&quot; header (e.g. TESTNET)</td>
</tr>
<tr>
<td>addressDetailsType</td>
<td>required</td>
<td>string</td>
<td>The value of this field is the string “CryptoAddressDetails” or “ACHAddressDetails”</td>
</tr>
<tr>
<td>addressDetails</td>
<td>required</td>
<td>CryptoAddressDetails</td>
<td>ACHAddressDetails</td>
</tr>
</tbody>
</table>

addressDetails
This is a field in the PaymentInformation message. addressDetails for each specific ledger MUST be registered at payid.org.

<table>
<thead>
<tr>
<th>Address Type</th>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CryptoAddressDetails</td>
<td>address</td>
<td>required</td>
<td>string</td>
<td>On-ledger address</td>
</tr>
<tr>
<td></td>
<td>tag</td>
<td>optional</td>
<td>string</td>
<td>Tagging mechanism used by some cryptocurrencies to distinguish accounts contained within a singular address. E.g XRP</td>
</tr>
<tr>
<td>ACHAddressDetails</td>
<td>accountNumber</td>
<td>required</td>
<td>string</td>
<td>ACH account number</td>
</tr>
<tr>
<td></td>
<td>routingNumber</td>
<td>required</td>
<td>string</td>
<td>ACH routing number</td>
</tr>
</tbody>
</table>

ProofOfControlSignature
This is an optional field in the PaymentInformation message.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Required/Optional</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>publicKey</td>
<td>required</td>
<td>string</td>
<td>on-ledger public key of the PayID server</td>
</tr>
<tr>
<td>payID</td>
<td>required</td>
<td>string</td>
<td>PayID of the receiver.</td>
</tr>
<tr>
<td>hashAlgorithm</td>
<td>required</td>
<td>string</td>
<td>The value of this field is the hash algorithm used to hash the entire contents of the “ProofOfControlSignature” message. E.g. “SHA512”</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>signature</td>
<td>required</td>
<td>string</td>
<td>The value of this field is the digital signature over the hash of the entire contents of the “ProofOfControlSignature” message using the private key corresponding to the public key in “publicKey”. This is a proof that the owner of the private key corresponding to the public key in the “publicKey” used to sign this message is the owner of the on-ledger public key in “publicKey”.</td>
</tr>
</tbody>
</table>

4. Upon receiving this Response message, the Sending endpoint (PayID client) parses the message body for isVASP field to check if the Receiving endpoint is a VASP. If it is, the PayID client (Sending VASP) forwards the request to the TRISA client. TRISA client parses the redirectURI field to initiate a secure, mutually authenticated TLS connection between VASPs by the Sending VASP to assure privacy of data in transit using EV KYV identity certificate.

5. Upon receiving the Sending VASP’s EV KYV identity certificate, the Receiving VASP verifies the certificate and decides if they want to proceed with the transaction with the Originating VASP. If they do, they send their EV KYV identity certificate to the Sending VASP.

6. Upon receiving the Receiver VASP’s TRISA identity certificate, the Sending VASP verifies the certificate and decides if they want to proceed with the transaction. If they do, the Sending VASP sends a “transaction identification message”. The transaction identification message MUST contain the amount and Travel Rule information as described in the TRISA paper.

*Note here that there are two changes in the TRISA transaction identification message here:*

a. Sending VASP sends an additional field that “PayID” that includes  
   a. receiver’s PayID,  
   b. payment-network, e.g. BTC, XRPL, ACH, etc.  
   c. environment, e.g. testnet, devnet, mainnet, etc.

b. Sending VASP does not send the “Blockchain” field as described in the TRISA flow. This is because this transaction identification message is a query for the Receiving VASP for payment address corresponding to the
queried “PayID” field.

1. Upon receiving the “transaction identification message”, Receiving VASP perform the following steps:
   1. Performs basic PayID protocol with the PayID server to retrieve PaymentInformation corresponding to the queried “PayID”
   2. Sends a signed receipt to the Sending VASP. The receipt MUST contain the *Beneficiary’s information* and signed *PaymentInformation*.

2. Sending VASP extracts the payment address from *PaymentInformation* and posts the transaction and receives a transaction ID.

3. Sending VASP posts the signed transaction ID to the Receiving VASP.

For more details on message formats for PayID flow, refer to the PayID Protocol whitepaper.

**Collaboration on Message Formats**

The TRISA project is collaborating with other efforts to standardize the data interoperability through message formats.

InterVASP IVMS101 is the recently ratified standard across the industry for message data formats for exchanging Travel Rule information. This standard has had the involvement of 100 companies and industry standard experts. It is a multi-lingual standard for sharing Travel Rule information between VASPS. The TRISA open source project has integrated IVMS101 in May 2020, to be the first Travel Rule regulations project to implement IVMS101.

TRISA participants are working with GDF and Digital Chamber on message format standardization.
The TRISA group is working with other working groups including the Coinbase-initiated “Group of 18”, initiatives driven by other cryptocurrency providers who have not announced yet (as of 6/10/2020).

Global Digital Finance (GDF) AML Working Group  
https://www.gdf.io/mem_wgroup/kyc-aml-ctf/

Digital Chamber of Commerce  
https://digitalchamber.org

TRISA participants are working with Shyft on interoperability of VASP identity certificates.  
https://www.shyft.network

TRISA participants are working with NetKi on how BIP 75 might be adapted to support Travel Rule requirements, and interoperability with the TRISA framework.  
https://github.com/bitcoin/bips/blob/master/bip-0075.mediawiki

Open Source Project

The TRISA project is open source. The latest code can be found here:  
https://github.com/trisacrypto/trisa

For more information, get involved, or to submit comments

Website: trisa.io  
Code: https://github.com/trisacrypto/trisa  
Email: support@trisa.io  
Slack: Trisa.slack.com
References


