Splyt Blockchain Technical Paper

This paper details the technical aspects Splyt has developed, and its strategies for future research and development to build a robust product capable of mainstream use and adoption.

Version 2.0

Overview

Splyt aims to decentralize affiliate marketing brokers and providers in retail e-commerce so that anyone can have a fair opportunity to create an online business without having to rely on a single entity. Specifically, we use blockchain, NFTs (non fungible tokens), and smart contract technology to reduce the barriers of entry for small entrants to reliably and efficiently provide the same quality of service to Consumers[1] as the current expensive and inefficient middlemen Providers[2] that currently dominate the industry.

Leaders in the blockchain space have indicated that economies can be formed via blockchain by applying the technology to databasing and search engine functionality, trust-facilitating environments such as reputation/dispute resolution systems.[3] Additionally, those thought leaders recognize that mainstream adoption of a blockchain-based e-commerce system in the short run will likely require stable coin capability and payment processing[4] to ease the current players' transition as seamlessly as possible from their current systems. Splyt uses blockchain, NFTs, and smart contract technologies to create an efficient marketplace of decentralized providers to perform those functions in the same ways that Bitcoin and Ethereum did with computer processing power.

Splyt has initially focused on building and perfecting databasing and search engine functionality necessary for marketing retail e-commerce goods via tokenizing asset

[1] This document uses the capitalized "Consumers" to refer to those who purchase retail goods online.

[2] This document uses the capitalized "Providers" to refer to legacy intermediaries who facilitate transactions between e-commerce sellers and e-commerce purchasers or those who facilitate transactions between e-commerce sellers and affiliate marketing partners.

[3] https://twitter.com/VitalikButerin/status/1023923633175945222

[4] https://twitter.com/vitalikbuterin/status/711696782556733440

listings through smart contract escrows. These tokenized asset listings, also called eNFT's (ecommerce non-fungible tokens) carry the ability for Sellers[5] to offer bounties for Affiliates[6] and other parties who help facilitate a sale or transaction. Our team is now researching the relevant remaining issues, including refining and developing efforts by others in the ecosystem so they can be implemented in an all-encompassing retail-ecommerce ecosystem that provides affiliate marketing intermediary services through blockchain and eNFT organizational structures.

1. How Real Assets are Represented and Transacted on the Blockchain

When Sellers list an item for sale on Splyt as a eNFT, they must connect through a node running the Splyt SDK that organizes input data into a format that can be circulated on the blockchain within each eNFT. This allows multiple websites or interfaces (that are not controlled by the same entities) to display and sell items tethered to eNFTs using the same inventory tracking system using Splyt. The Splyt SDK allows for both single-purchaser and to be listed. Sellers may also opt to set a predefined "Reward" to each eNFT to incentivize an Affiliate or another party who helps generate a sale to a Consumer. Should a Consumer purchase an item off an Affiliate's website[7] or interface, their purchase amount will be placed into an escrow contract until the transaction successfully completes. Upon a transaction's

[5] This document uses the capitalized "Sellers" to refer to those who warehouse inventory available for retail sale online.

[6] This document uses the capitalized "Affiliates" to refer to those who help Sellers gain visibility and marketing presence for Sellers, including influencers, referral-link traffic aggregators, or even other Sellers. Unlike Sellers, Affiliates do not warehouse the inventory of the goods they promote.

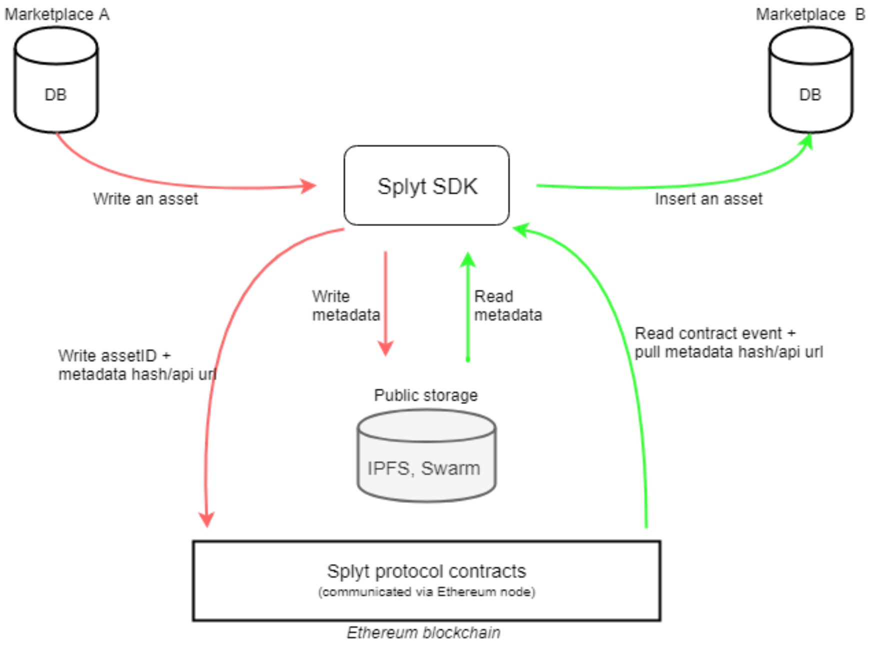
[7] "Websites and services based on Web 2.0 concepts—blogging and interactive online communities, for example—have impacted the affiliate marketing world as well. These platforms allow improved communication between merchants and affiliates." https://en.wikipedia.org/wiki/Affiliate_marketing

successful completion, the purchase amount is automatically distributed to the Seller's and the appropriate Affiliate's wallets. This section describes how the Splyt SDK structures asset metadata, how the blockchain interacts with a data store, and how Affiliate Rewards are successfully executed on-chain.

a. eNFT retail e-commerce listing databasing and indexing functionalities

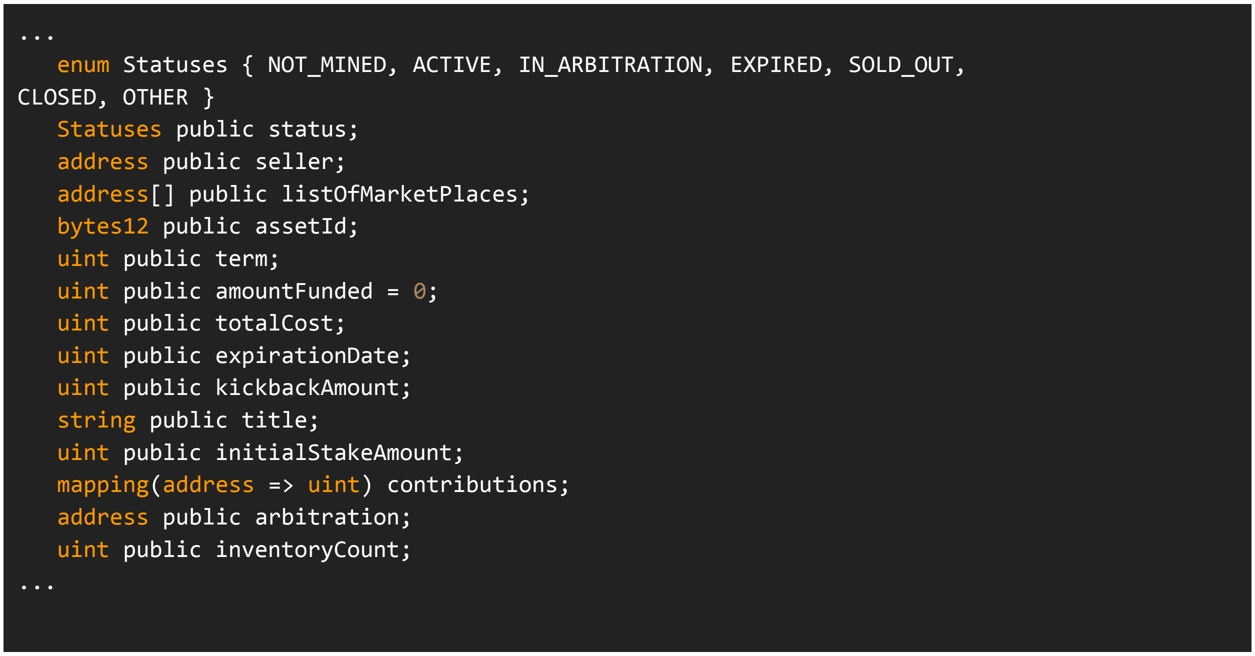
A thorough and effective search through existing catalogs and inventories is necessary for Consumers to locate items they wish to purchase from Sellers. The following illustrates how Splyt approaches this issue:

Spl.yt data storage and event mechanism



The Splyt SDK supplies one method for developers to join the Splyt ecosystem within this framework. A developer managing an e-commerce website can run the Splyt SDK on their back-end in order to sync their existing inventories with the Splyt ecosystem. When a Seller pushes a (or creates a new) listing to the Splyt ecosystem, a contract will be deployed to represent that item. As the above flow-chart indicates, Splyt is data-storage agnostic, meaning the service can easily draw from information stored on any data store that can be resolved to and read by the Ethereum blockchain. Splyt currently houses metadata on its own servers, though it intends to migrate to the InterPlanetary File System or a similar Web3 data storage service as a default in the near-term. Nonetheless, others building on the Splyt platform can pull from any data store that is appropriate to their purposes at any time, including but not limited to rest API calls.

Splyt's current iteration includes the following code:



```
...
   enum Statuses { NOT_MINED, ACTIVE, IN_ARBITRATION, EXPIRED, SOLD_OUT,
CLOSED, OTHER }
   Statuses public status;
   address public seller;
   address[] public listOfMarketPlaces;
   bytes12 public assetId;
   uint public term;
   uint public amountFunded = 0;
   uint public totalCost;
   uint public expirationDate;
   uint public kickbackAmount;
   string public title;
   uint public initialStakeAmount;
   mapping(address => uint) contributions;
   address public arbitration;
   uint public inventoryCount;
...
```

This includes the key on-chain identifiers necessary to facilitate crypto-payments and escrowing, such as the total cost of the item, owner of the asset listing, listing expiration etc. While the e-commerce industry has not standardized metadata requirements, information such as descriptive identifiers (color, make, model, year, condition, or product category), specific descriptive identifiers (item description, shipping size, warranty), item pictures, and item ID numbers (SKU, IBSN, UPC, or another serial number) are held off-chain.

E-commerce also does not have a standard method for inventory managers to derive unique keys for their catalog databases. This leads to a potential problem where two marketplaces may create clashing unique keys while tracking their various inventories, resulting in a clash in Splyt's global catalog. Splyt's SDK will resolve this by hashing the combination of an inventory manager's local unique key for a listing with their assigned ethereum wallet address into one single hexadecimal uniqueId, ensuring that no unique key will clash in the global Splyt system. This method is favorable to using an item's token address as a unique Id because it gives sellers the flexibility of manipulating their listing information before committing it to the blockchain. In other words, a seller can operate locally as they are always accustomed to, and only commit resources to joining the Splyt ecosystem (and, consequently, the Ethereum network) once they are ready to push such information.

This method is sub-optimal given that the majority of goods listed online are generally not available for fractional purchase, making the redundancy unnecessary in most cases. Splyt plans to convert its listings to the ERC-721 standard (which gained popularity after we completed this portion of our technology), with the option to assign an ERC-721 token (that represents an item for sale) to a fractionally ownable DAO that is itself the fractionally purchasable item.[8]

Local copies of the current state of Splyt's global catalog is held locally (including real-time changes to inventory), so Affiliates can display these items (including any relevant metadata) on their own websites. Search indexing and querying then can occur locally, allowing Affiliates to only display certain items or subsets as they please. When an Affiliate sells an item from their website or interface, the SDK will then automatically update all other ecosystem copies to show the item had already sold. As each item is represented by a single contract, this will prevent accidental double-sales of any item in the same way that blockchains prevent double-spends.

b. Polkadot Substrate

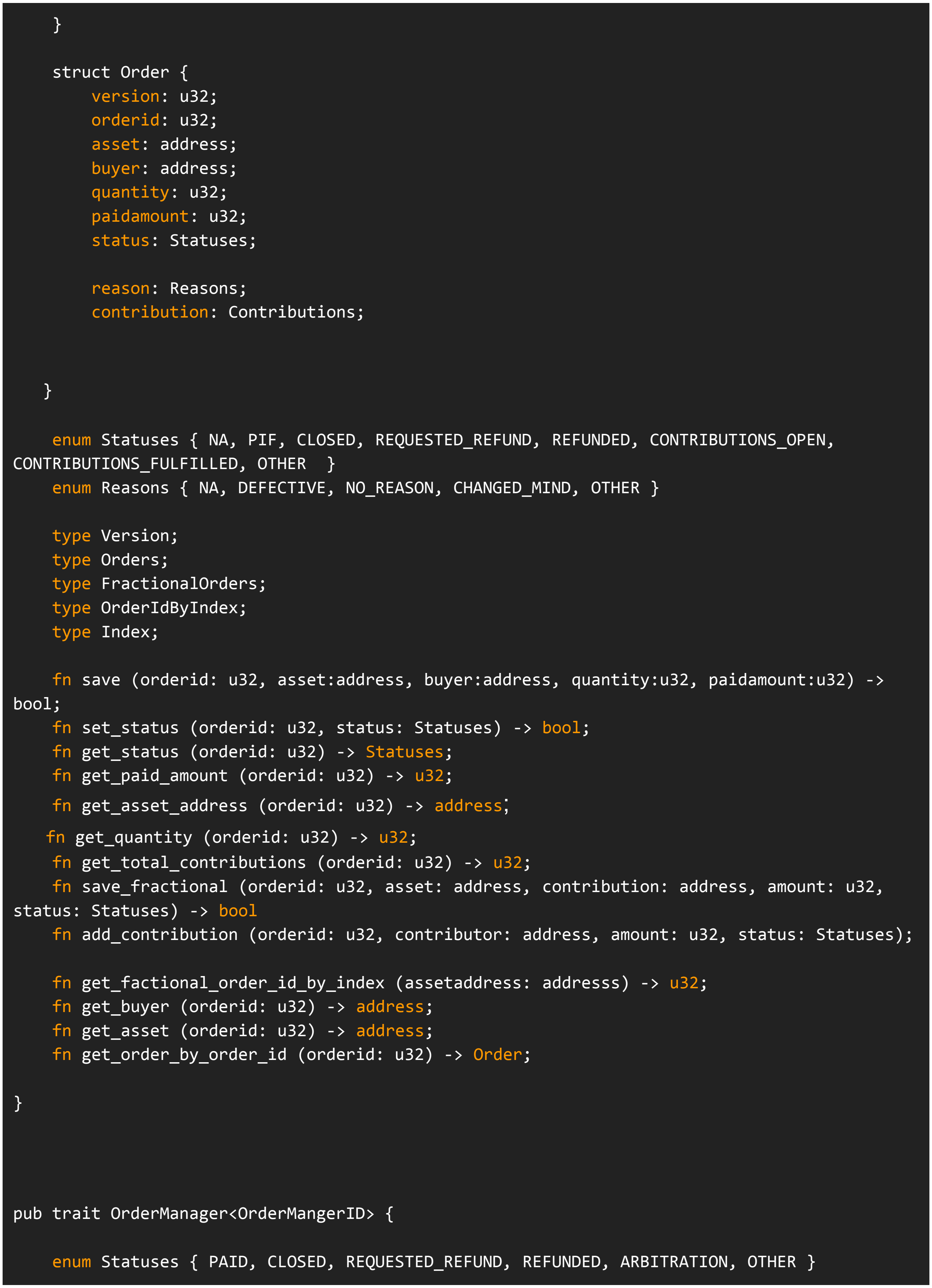
Polkadot, in contrast to Ethereum, requires fewer validators, which results in shortening the transaction finality time (12 - 60 seconds). Polkadot's parachains (parallel blockchains) are meant to process more transactions than a single blockchain, spread across multiple computers. Their governance decision can be made on-chain, which enables upgrades without hard forks. With this, Splyt can easily adapt and upgrade as technologies improve, ensuring we are continuously at the forefront of industry.

Polkadot is built on Substrate which uses "pallets," modular blocks of code, and ships with FRAME for runtime development. Substrate developers can make their own

[8] This also creates a potential for a collectibles aftermarket of closed listings. For example, should a celebrity purchase a unique item online through the Splyt system, they may then be able to sell the ERC-721 that represented that transaction as collectible merchandise after it is closed. While it is not clear that any demand exists for this feature, the use of ERC-721 in the Splyt system will enable exploring this possibility.

pallets, reuse ones that have already been made, or they can use up to 50 pre-made pallets that come with Substrate. Below is a pallet written in Rust, the language Parity used to create Substrate:

```
//! # Splyt eNFT Protocol

pub trait Asset<AssetID> {


    enum AssetTypes { NORMAL, FRACTIONAL }
    enum Statuses { NOT_MINED, ACTIVE, IN_ARBITRATION, EXPIRED, SOLD_OUT, CLOSED, OTHER }

    type SellerAddress;
    type ListOfMarketPlacesAddress;
    type AssetID;
    type Term;
    type AmountFunded;
    type TotalCost;
    type ExpirationDate;
    type KickBackAmount;
    type Title;
    type InitialStakeAmount;
    type Contributions;
    type Arbitration;
    type InventoryCount;

    fn set_status(Statuses);
    fn get_marketplace_by_index(index: u32);
    fn get_marketplace_length()
    fn add_marketplace(marketplace: address)
    fn add_inventory(quantity: u32)
    fn subtract_inventory(quantity: u32)
   fn set_inventory(qty: u32)
    fn is_only_affiliate() -> bool



}


pub trait Order<OrderId> {

   struct Contribution {
        contributor: address;
        amount: u32;
        date: u32;
```



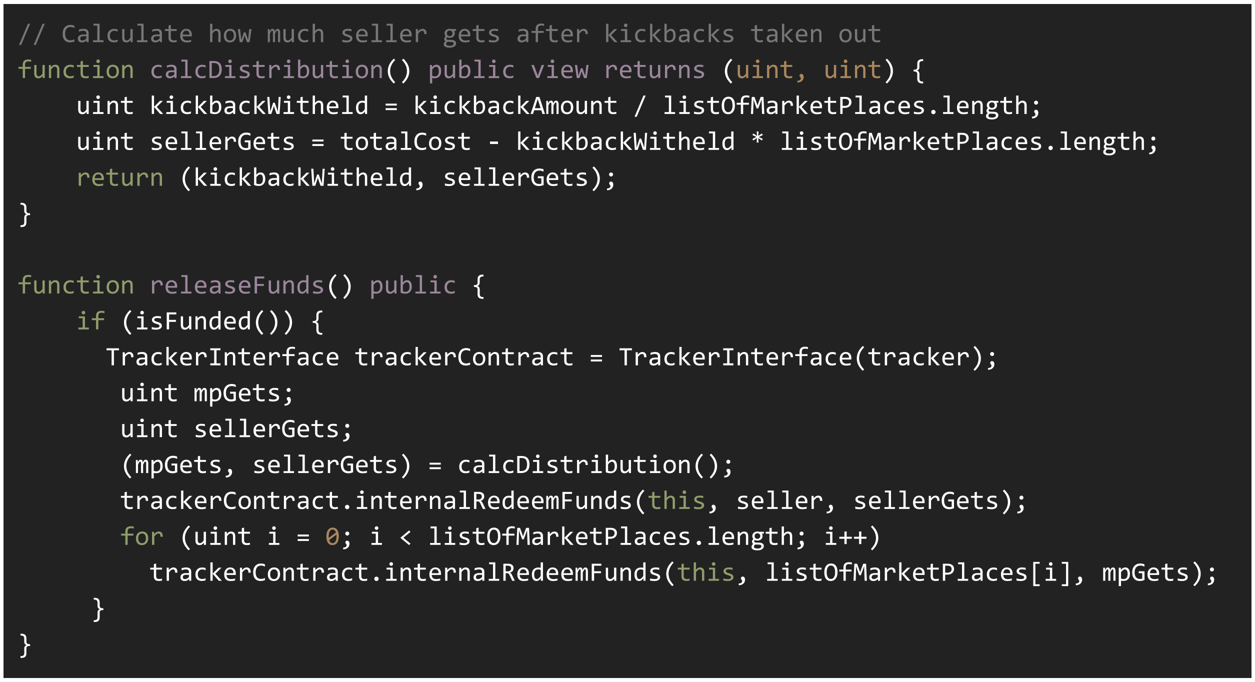
```
    }

    struct Order {
        version: u32;
        orderid: u32;
        asset: address;
        buyer: address;
        quantity: u32;
        paidamount: u32;
        status: Statuses;

        reason: Reasons;
        contribution: Contributions;


   }

    enum Statuses { NA, PIF, CLOSED, REQUESTED_REFUND, REFUNDED, CONTRIBUTIONS_OPEN,
CONTRIBUTIONS_FULFILLED, OTHER  }
    enum Reasons { NA, DEFECTIVE, NO_REASON, CHANGED_MIND, OTHER }

    type Version;
    type Orders;
    type FractionalOrders;
    type OrderIdByIndex;
    type Index;

    fn save (orderid: u32, asset:address, buyer:address, quantity:u32, paidamount:u32) ->
bool;
    fn set_status (orderid: u32, status: Statuses) -> bool;
    fn get_status (orderid: u32) -> Statuses;
    fn get_paid_amount (orderid: u32) -> u32;
    fn get_asset_address (orderid: u32) -> address;
   fn get_quantity (orderid: u32) -> u32;
    fn get_total_contributions (orderid: u32) -> u32;
    fn save_fractional (orderid: u32, asset: address, contribution: address, amount: u32,
status: Statuses) -> bool
    fn add_contribution (orderid: u32, contributor: address, amount: u32, status: Statuses);

    fn get_factional_order_id_by_index (assetaddress: addresss) -> u32;
    fn get_buyer (orderid: u32) -> address;
    fn get_asset (orderid: u32) -> address;
    fn get_order_by_order_id (orderid: u32) -> Order;

}



pub trait OrderManager<OrderMangerID> {

    enum Statuses { PAID, CLOSED, REQUESTED_REFUND, REFUNDED, ARBITRATION, OTHER }
    enum Reasons { DEFECTIVE, NO_REASON, CHANGED_MIND, OTHER }

    type OrderData;
    type SplytManager;



    fn purchase (orderid: u32, assetaddress: address, quantity: u32, tokenamount: u32,
marketplace: address) -> bool;
    fn create_order (orderid: u32, asset: Asset, quantity: u32, tokenamount: u32);
    fn is_fractional_order_exists (asset: address) -> bool;
    fn contribute_order (orderid: u32, asset: Asset, tokenamount: u32);
    fn set_status (orderid: u32, status: Statuses) -> bool;
    fn get_status (orderid: u32) -> Statuses;
    fn calc_distribution (totalcost: u32, length: u32, kickbackamount: u32) -> u32;
    fn approve_refund (orderid: u32) -> Statuses;
    fn request_refund (orderid: u32) -> Statuses;
    fn get_total_contributions (orderid: u32) -> u32;
    fn set_splytmanager (address: address);
    fn get_data_contract_address () -> address;
    fn get_data_contract (orderdata: address) -> OrderData;
    fn get_fraction_order_id_by_asset_address (asset: address) -> u32;
    fn get_order_info_by_order_id (orderid: u32) -> OrderData;
    fn get_order_info_by_index (index: u32) -> OrderData;
    fn get_contribution_length (orderid: u32) -> u32;
    fn get_contribution_by_order_id_and_index (orderid: u32, index: u32) -> u32;



}
```

c. Affiliate Bounty Incentive Contract Structure

Sellers can use Splyt to incentivize Affiliates to attract Buyers for their products. Sellers do this by pre-defining an amount of an item’s sale price that will be automatically transferred to an Affiliate’s wallet should a Buyer purchase an item from the Affiliate’s website or storefront. This acts as a colloquially affiliate reward or commission, with invoicing and settlements processed automatically and verifiably by the blockchain.

For example, a seller can post an asset for 100 SHOPX tokens with an 10 SHOPX reward for any Affiliate who sells the item. An Affiliate then lists the item on their website or interface. Should a Buyer purchase the item from the Affiliate (and the transaction is successful), the Seller will receive 90 SHOPXs while the Affiliate receives 10 SHOPXs.

We have coded this feature in Solidity as follows:



```
// Calculate how much seller gets after kickbacks taken out
function calcDistribution() public view returns (uint, uint) {
    uint kickbackWitheld = kickbackAmount / listOfMarketPlaces.length;
    uint sellerGets = totalCost - kickbackWitheld * listOfMarketPlaces.length;
    return (kickbackWitheld, sellerGets);
}

function releaseFunds() public {
    if (isFunded()) {
      TrackerInterface trackerContract = TrackerInterface(tracker);
       uint mpGets;
       uint sellerGets;
       (mpGets, sellerGets) = calcDistribution();
       trackerContract.internalRedeemFunds(this, seller, sellerGets);
       for (uint i = 0; i < listOfMarketPlaces.length; i++)
         trackerContract.internalRedeemFunds(this, listOfMarketPlaces[i], mpGets);
     }
}
```

Reward mechanism from asset contract (for fractionally/single purchased items)

This structuring also allows for multiple Affiliates to facilitate fractional purchases of an asset. For example, a Seller lists an item for fractional purchase for 100 SHOPXs with a 10 SHOPX Affiliate Reward. Two independent Affiliates can list the item for purchase on their website. Each Affiliate secures a Buyer who each contributes 50 SHOPXs for the purchase of the item, allowing the transaction to eventually successfully complete. By splitting the Affiliate Reward among Affiliates listed in the "listOfMarketPlaces.length," each Affiliate will receive their portion of the Reward. This can help Sellers market fractionally purchasable items, especially for high value assets.

2. Establishing trust through automated and decentralized mechanisms

C2C e-commerce requires established trust between Consumers and Sellers. Without any other factors, an unknown Seller finds it difficult to convince an unknown Consumer to send them money online in exchange for the promise of a good to be delivered to their home at a future date.

“Spam” -- offers to sell an item that will never be fulfilled because they are erroneous, fake, or redundant -- represents a large disincentive for Consumers to shop on an online platform if they think the effort (and money) they spend to order an item will not result in receiving that item. Centralized e-commerce providers curate their catalogs through various means to minimize spam on their platform and help Consumers. Some of the most popular methods that these providers engage in includes leveraging human resources or automated tools to detect spam postings. Centralized entities may even block or take other disciplinary action against Sellers to encourage their trustworthy behavior. Providers who successfully establish an environment that Consumers can consistently trust to transact with unknown Sellers online typically charge a high premium for their services, with reported fees as high as 30-50% of an items’ listed prices. Further, their position as a trusted intermediary affords Providers the opportunity to gather and exploit market data on transaction counter-parties. This allows Providers to maximize their value extraction efforts against them, sometimes without counter-parties' awareness.

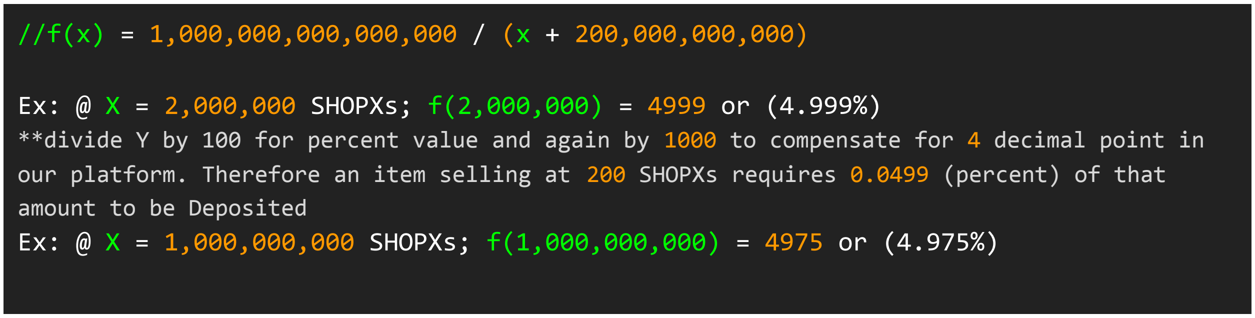
Splyt aims to provide the same or a higher level of integrity of items listed in its catalog to avoid dissuading Consumers from transacting with ecosystem Sellers without a trusted third-party facilitator. This section explains our technical strategy to holistically minimize the chance for disputes to occur using established trust methods in a decentralized fashion.

a. Calculating Seller Listing Deposit Amounts On-Chain

Splyt requires Sellers to "Deposit" an amount of SHOPXs to successfully list an item. This Deposit is returned to Sellers unless they are found to be engaging in bad behavior (such as by submitting spam listings or not transacting as agreed with a Consumer). By holding a Deposit on-chain that may be forfeited if bad behavior is found, Sellers are encouraged to perform to the highest degree of professionalism expected by Consumers. If they do, then the Deposit model does not heavily burden their participation. Conversely, if a Seller, or other bad actors, do not act in good faith, then they risk losing their Deposit.

Splyt does not use fixed or linearly-calculated (based on the price of an asset) Deposits because those methods would have inconsistent effectiveness across various circumstances. For example, a fixed Deposit amount may discourage low-value item Sellers to use the Splyt system altogether while Sellers listing higher-value assets might not be adequately disincentivized from engaging in bad behavior given their potential payoff. Linearly-calculated Deposits are also imperfect because legitimate high-value item Sellers may find it uneconomical or unaffordable to place a large capital deposit in escrow without any guarantee of return.

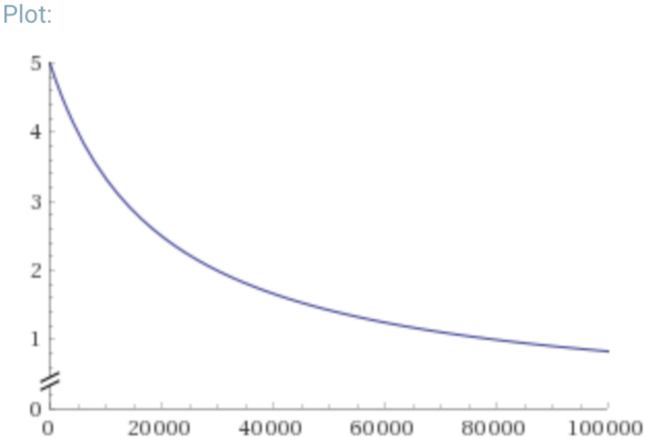
Instead, Splyt calculates required Deposit amounts with the following formula for first-time Sellers:



```
//f(x) = 1,000,000,000,000,000 / (x + 200,000,000,000)

Ex: @ X = 2,000,000 SHOPXs; f(2,000,000) = 4999 or (4.999%)
**divide Y by 100 for percent value and again by 1000 to compensate for 4 decimal point in
our platform. Therefore an item selling at 200 SHOPXs requires 0.0499 (percent) of that
amount to be Deposited
Ex: @ X = 1,000,000,000 SHOPXs; f(1,000,000,000) = 4975 or (4.975%)
```

Where Y is the required Deposit percent of item price X.

This formula optimizes the accessibility for Sellers to use the Splyt system across all listing prices, as demonstrated by the following curve:

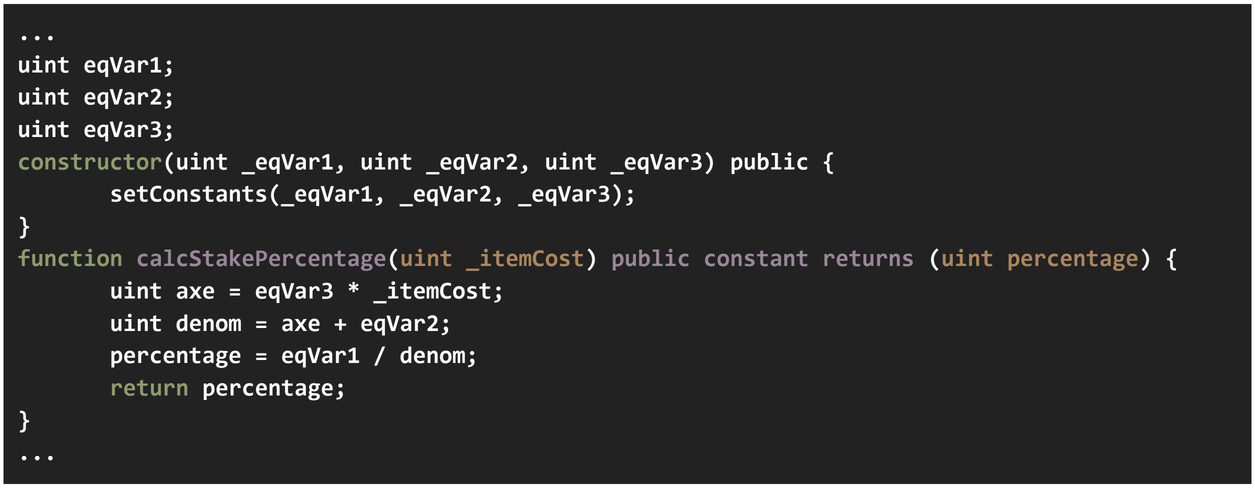


[9]

Where Y-axis is Deposit % and X-axis is price of goods

[9] https://www.wolframalpha.com/input/?i=graph+y+%3D+(10,000,000)+%2F+(100*x+%2B+2,000,000)+from+0+to+100000

Ethereum's Solidity does not support decimal places, so round-off error could occur when dividing two numbers. We exaggerate the formula to return an integer instead of a decimal, for up to 5 decimal places. As accuracy of Deposit functionality is not critical to initial platform trials, our requirements to minimize round off errors are not strict. Thus minimizing round off errors for this case was assumed instead of eliminating round off errors. With these understandings in place, the following is Splyt's Solidity implementation of the Deposit functionality:



```
...
uint eqVar1;
uint eqVar2;
uint eqVar3;
constructor(uint _eqVar1, uint _eqVar2, uint _eqVar3) public {
       setConstants(_eqVar1, _eqVar2, _eqVar3);
}
function calcStakePercentage(uint _itemCost) public constant returns (uint percentage) {
       uint axe = eqVar3 * _itemCost;
       uint denom = axe + eqVar2;
       percentage = eqVar1 / denom;
       return percentage;
}
...
```

Splyt's Deposit formula implementation in an Ethereum contract

b. Poker Blind Dispute Resolution Incentives

Dispute resolution is an important aspect of any form of commerce. Disputes arise around contracts or agreements when one (or multiple) parties to a transaction believe other parties did not perform according to the prearranged agreement. Courts are supposed to provide dispute resolution to any agreement, but in practice are reserved for high-stakes issues engaged by those with resources to engage in litigation. C2C e-commerce and affiliate marketing transactions frequently involve goods or services of such low value that relying on courts for resolution is impractical. Nonetheless, disputes must still be reliably resolved in order to maintain Consumer trust in any given platform.

Legacy Providers typically provide this service by maneuvering to resolve disputes should one user of their services claim another violated an agreement in some way. Our research has identified two common problems reported by users in these legacy e-commerce dispute resolution mechanisms: 1) high volumes of fake or erroneous complaints are submitted since there is usually no disincentive for filing a complaint, driving up operational costs and diluting attention given to legitimate complaints; and 2) dispute resolution systems are rarely impartial and are frequently biased by business or other interests. The following describes how Splyt is approaching these problems

To list an asset on Splyt, a Seller must deposit an amount of SHOPXs to ensure the transaction occurs properly (see Section 2 above). A Consumer then purchases the item for the full value of the item. The purchase amount (and the deposit) are held in escrow for a pre-set amount of time for the transaction to occur successfully (the Consumer receives the item to their satisfaction). If the transaction occurs successfully, then the deposit is returned to the Seller.

If the transaction does not occur to the Consumer's satisfaction, then the Consumer may contest the transaction by placing an amount equal[10] to the Seller's Deposit ("Poker Blind") into a challenge escrow. The Poker Blind encourages the Consumer to

[10] Early iterations of the Splyt system will involve equal amounts between the Poker Blind and the Seller's deposit. However, it is possible that users discover average payouts from the full resolution system are higher than what would be considered equitable, creating an incentive to submit Poker Blinds or Double Blinds even when the relevant party is not as confident that they will succeed in resolution as the blinds suggest. Splyt will monitor this possibility and address it in a future update should such behavior become significant.

ensure they are confident that the transaction did not occur as agreed. The Seller then has the options of a) admitting fault, releasing the full purchase amount plus their Deposit back to the Consumer, b) contacting the Consumer to reach a resolution between them, or c) signaling their confidence that the transaction did in fact occur as agreed by placing an equal amount to the Consumer's Poker Blind ("Double Blind"). The Consumer will then have the option to a) forfeit their Poker Blind, deeming the transaction successful allowing the full escrowed amounts to be distributed accordingly, or b) reaffirm their confidence that the transaction was not correctly carried out by placing an amount matching the Double Blind into an escrow. If the Consumer submits a matching Double Blind, only then will a dispute be sent to a resolution process. We will be split testing to optimize the amount of time allocated to each party's response time.

While this system does not eliminate the need for off-chain resolution, it nonetheless carries multiple benefits. First, it minimizes operational costs incurred by discouraging erroneous complaints while encouraging Consumers with legitimate complaints to participate through the potential upside of receiving more than the purchase amount if a dispute is resolved in their favor. Second, similar to settlements in traditional litigation, it encourages the Consumer and Seller to reach resolution without the need for third-party intervention when possible. Third, the Deposit, Poker Blind, and Double Blinds can be used to compensate a third-party arbitrator should one still be desired by both parties.

Splyt does not seek to solve problems associated with on-chain dispute resolution arbitration. However, this system is compatible with leading emerging projects, including Aragon, Kleros, Mattereum, Sagewise, and other notable efforts which can use our tokens to become arbitrators. As those efforts mature, we will look into integrating their services into the Splyt ecosystem.

c. Curation List of Goods for Sale And Its Governance

Requiring Buyers to endure dispute resolution damages user experience and should be avoided to promote their participation. Adding methods for bad actors' deposits to be forfeited before initiating the Double Poker Blind mechanism can heighten disincentive for bad actors while transferring costs of dispute from Buyers to Splyt stakeholders. We are exploring adapting a form of Mike Goldin's Token Curated Registry ("TCR") game theoretical incentive structure[11] to incentivize stakeholders to curate all credible inventory and self-police questionable listings before a Buyer has a chance to interact with them. Goldin's TCR framework can be described as follows:

- Published lists face conflicting interests of those who wish to be included on the list ("Listees"), those who read a list ("Readers"), and those who moderate a list ("Validators").
    - A list only has value if Readers find it valuable. For example, a list of colleges ranked by average amount of debt graduates possess ten years after graduation is only useful if it is accurate.
    - Listees benefit from inclusion on a list, and are therefore incentivized to try to "game" the system to obtain inclusion.
    - Validators obtain value through monetization or other incentives of providing the list. Validators may be able to sufficiently monetize the list by attracting traffic of Readers (e.g. via advertising to Readers), but are

[11] https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7

    vulnerable to corruption in the form of seeking other incentives that may compromise the integrity of a list (e.g. by accepting bribes to include a Listee on the list despite the Listee’s lack of qualification for inclusion). Therefore, a list’s integrity is balanced between a Validator’s ability to provide value to Readers and a Validator’s willingness to extract incentives in ways counter to Readers’ desire for value.

- Goldin hypothesizes that centralized Validators are prone to corruption through their position as a single “gatekeeper” for Listees’ inclusion onto a list. He suggests that this vulnerability may be mitigated by decentralizing list-curation-power among a larger set of Validators, increasing the number of parties that must act in a faulty or corrupt manner in order to negatively reduce value to Readers. For example, one single gatekeeper/Validator might be willing to corrupt a list if a Listee offers a $1,000 USD bribe to do so; but a system that requires bribing at least ten gatekeepers/Validators would likely cost more to the briber than $100/Validator, thereby increasing the cost of corruption by spreading authority across multiple actors and minimizing its practicality.
- Goldin’s TCR model provides a framework to accomplish this using tokens on a blockchain.
    - Validators receive some tokens. Holders of these tokens are entitled to a portion of the returns generated by the list by attracting Readers.
    - Listees must apply to join the list by submitting their qualifications plus a deposit.
    - Validators construct a mechanism through which they collectively assess Listee applications. Validators are united only through the common interest of increasing returns by attracting Readers.

    - A Listee who successfully joins the list receives their deposit (in whole or in part) back. Conversely, a Listee who Validators collectively agree is not worthy for list-inclusion forfeits their deposit, which is burned or distributed among Validators.
- This mechanism by itself sorts potential Listees in three categories:
    - Confident Listees, who are sure they will be included and therefore will not shy away from placing a deposit
    - Bad-Actor Listees, who recognize they will probably not be included and therefore will not risk forfeiting a deposit against the high likelihood of rejection.
    - Unconfident Listees, who are likely to receive heavier scrutiny from Validators, and cannot be overwhelmingly confident one direction or the other about their inclusion.
- If Goldin's assumptions are true, this creates a method for a list less vulnerable to corruption than those curated through centralized entities, while creating a plausible value to be harvested by stakeholders participating in the system.

We have incorporated a form of Goldin's TCR model to promote the legitimacy of the curation of minting eNFTs as inventory to the Splyt global catalogue. In the Splyt ecosystem, Sellers are analogous to Listees and Buyers are analogous to Readers in the Goldin system. We have refined the Validators function to take place after a listing has been curated to reduce friction of doing business in real life. We expect someToken holders to act as Validators. We have explored the viability in requiring Validators to stake tokens to have skin in the game.

```
function createArbitration(bytes12 _arbitrationId, address _assetAddress,
Arbitration.Reasons _reason) public {
  Asset asset = Asset(_assetAddress);
  uint stakeAmount = asset.initialStakeAmount();
  address reporter = msg.sender;

  Arbitration arbitration = new Arbitration(_arbitrationId, _assetAddress, _reason,
reporter, stakeAmount);
  arbitrationData.save(_arbitrationId, address(arbitration));
  splytManager.setAssetStatus(_assetAddress, Asset.Statuses.IN_ARBITRATION);
  splytManager.internalContribute(reporter, asset, stakeAmount);
  emit Success(3, address(arbitration));
}
```

Above is our initial code to enable any user to act as a Validator and submit a listing as suspicious/ invalid. This gives the user freedom to mark bad actors/listings to be taken out of inventory of the Splyt global catalogue.

Our TCR model should reduce the amount of arbitrations because we have implemented a deposit of tokens which prevents falsely marking a listing as spam or invalid. The Validator needs to deposit the same amount as Seller when they initially posted the listing (as defined in Section 2a). If the Seller thinks their listing is valid and poses no threat then they are allowed to stake one more time, shown below. After Seller's stake is 2x and Validator's stake is 1x, the Validator gets a chance to stake one more time making a 2x and 2x stake for the Seller and Validator. There will be a default period of time to resolve a deadlock stake amount before being sent to the arbitration process.

```
function set2xStakeBySeller(bytes12 _arbitrationId) public
onlySeller(_arbitrationId) onlyStatus(_arbitrationId,
Arbitration.Statuses.REPORTED){
  address arbitrationAddress =
arbitrationData.addressByArbitrationId(_arbitrationId);
  Arbitration arbitration = Arbitration(arbitrationAddress);
  arbitration.set2xStakeBySeller();

  splytManager.internalContribute(arbitration.getSeller(), arbitration.asset(),
arbitration.baseStake());
  emit Success(4, address(arbitration));
}

function set2xStakeByReporter(bytes12 _arbitrationId) public
onlyReporter(_arbitrationId) onlyStatus(_arbitrationId,
Arbitration.Statuses.SELLER_STAKED_2X){
  address arbitrationAddress =
arbitrationData.addressByArbitrationId(_arbitrationId);
  Arbitration arbitration = Arbitration(arbitrationAddress);
  arbitration.set2xStakeByReporter();
  splytManager.internalContribute(arbitration.reporter(), arbitration.asset(),
arbitration.baseStake());
}
```

In order to be chosen as an Arbiter, you must stake tokens to toggle your participation to “on”. The more tokens you stake, the higher “Stakes” of arbitration you will be eligible for. If you are chosen to be an Arbiter and you do not respond to the process, you lose your stake. This will incentivize participation in a timely manner. Arbiters can unstake their tokens at any time to be taken out of the queue. Arbiters will be designed to be chosen at random however as we begin our implementation, the arbitration process will be done by our team until proper testing and game theory research has been conducted.

The Splyt TCR enabled Validators to discover listings that are clearly faulty before that item is sold. This may be done manually, or by automated clients that search for statistically prominent patterns common in faulty listings may also be developed to reduce labor costs required for this process.

We are still exploring the game theoretical leverage nuances to promote the best outcomes of this system. For example, as the game was suggested by Goldin, Validators may be incentivized to collude to over-reject Listees because they receive monetary benefit to do so; however, we cannot simply eliminate those distributions as it provides the short-term incentive for participating in the process. We may be able to resolve or mitigate this conflict of interest (should it manifest) by reducing the portion of deposits distributed to Validators (thereby reducing their incentive to collude), and either burning the remaining deposit amount or transferring them to another recipient (such as a pool to reward Dispute Resolution providers). This would leave the soft-incentive of enhancing the value of the overall Splyt ecosystem (and therefore their token stake) and the remaining portion of deposit distributed to override the costs associated with moderating the list.

We also may look to ways to encourage the Splyt community to reach consensus around a set of standards by which Validators will judge Listers against for more consistent enforcement results. This would be beneficial because it gives clear and transparent standards through which Validators behave.

d. E-commerce oriented reputational identities persistently maintained across platforms

Reputational ratings play an important role in establishing trust in e-commerce because they provide a means of knowing an individual's track record that otherwise is unavailable to Consumers or Sellers who have never transacted together before. For example, a Consumer is more likely to purchase from a Seller who has a near-perfect rating over thousands of transactions than one who has only completed a few sales or is completely new. Sellers frequently leverage established reputations to price items higher than those who have not established a reputation, providing an economic incentive to engage in good behavior over a long period of time. Without ratings or another reliable means of demonstrating reputation, many Consumers and Sellers would likely be discouraged from transacting online because they cannot differentiate their counterparties from potential bad actors.

Providers maintain systems for supplying this endorsement system via platform ratings. Operational costs associated with managing accurate ratings systems drive platform fees incurred by e-commerce counterparties using a Provider's services. However, Providers' position as a trusted intermediary creates room for rent-seeking behavior and causes redundant efforts throughout the ecosystem. For example, a Seller who has paid eBay's platform fees while establishing a strong reputation over a long period of time fails to fully capture the benefit of those costs should they open a channel on Amazon Marketplace.

The issue of establishing trust is more pronounced in a system like Splyt -- which seeks to eliminate intermediaries to benefit the whole industry -- because Consumers may be purchasing items from a party who has never interacted with the Seller providing a listed good. Counterparties cannot rely on a centralized player to create and enforce a rule set. Splyt solves this by employing a system similar to that suggested by Evan Miller[12] that uses Wilson score binomial proportion intervals to promote beneficial behavior by actors across multiple interfaces[13].

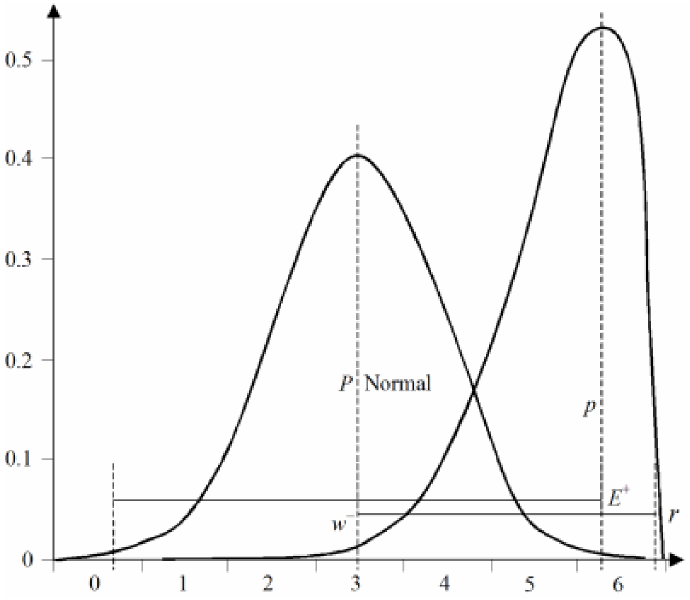
[12] https://www.evanmiller.org/how-not-to-sort-by-average-rating.html

[13] For now, identities are defined as public addresses, presumably operated by a secret private key. Nonetheless, the Foundation plans to integrate its reputational systems with other projects focusing on decentralized identity (such as Civic or uPort) as they mature to remove problems related to key-paired identities.

Following this logic, Splyt ranks Sellers relative to each other and shows their “Reputation Confidence” as a percentile in the system. A seller who ranks in the top of the “Raw Reputation Score” will be given a 99% Reputation Confidence. This standardized comparison system functions similarly to how students’ raw scores are ranked in the Scholastic Aptitude Test. In our initial testing, we are using Miller’s formulation of binary positive/negative ratings (not 5-star scale):

$$\left(\hat{p} + \frac{z^2_{\alpha/2}}{2n} \pm z_{\alpha/2}\sqrt{[\hat{p}(1-\hat{p}) + z^2_{\alpha/2}/4n]/n}\right)/(1 + z^2_{\alpha/2}/n).$$

The lower bound can be found by using the subtraction operation where the +/- sign is indicated. “p̂” is the observed number of positive ratings divided by the total number of ratings, [zα/2] is the [1-α/2] quantile of the standard normal distribution, and “n” is the total number of ratings.



We use this method against using scores based on average ratings or subtraction of negative ratings from positive ratings because it uses transactions to compute a raw score and then incorporates statistical confidence to compare against other parties. For example, under an average rating method, a Seller that has 1 positive and 0 negative ratings will have the same score as a Seller that has 1000 positive and 0 negative ratings. Our Wilson score method accounts for this by incorporating confidence levels derived from total ratings. Similarly, subtraction-based methods (where ratings are derived by subtracting negative ratings from positive ratings) would result in a Seller who has 10 positive and 0 negative ratings will be rated the same as a Seller that has 20 positive and 10 negative ratings. Our Wilson score method resolves this through its relative ranking system that also reflects binomial confidence.

We are also exploring the possibility of dynamic Deposit[14] requirements that adjust based on these Wilson score intervals. For example, a Seller in the 50th percentile indicates they are near the mean of Sellers in the system and are a reliable seller. But a Seller in the 90th percentile has proven a very strong track record, and therefore requiring as steep of a Deposit amount to ensure they are not a bad actor is not necessarily. If this proves useful, a highly ranked Seller may be rewarded with a xx% discount from the stake amount required to add your assets to the Splyt Global Inventory. Conversely, a Seller in the 20th percentile may be penalized, requiring a larger disincentive to continue using the system (or, more likely, forced to create a new account without any track record).

The system will also account for great sellers across the board. With a large sample

[14] As described in Section 2a above.

size of seller, if the top seller has 99 positive and 1 negative reviews, and the lowest rated seller has 95 positive and 5 negative reviews, we will not penalize the latter seller for being in the lowest percentile. The system will account for this with a Simple Score that will override the Reputation Confidence score if certain conditions are met. For example, if we take the same scenario stated earlier, we would take a rolling 12 month average of positive reviews divided by total reviews and if they met a certain threshold, the seller will be categorized in the top tier of the Confidence Interval

.

3. Side-Chains and Staking

Splyt will be using side-chains to speed up, stake and incentivize platform users to adopt the technology. E-commerce platforms need high transaction blockchains. Using ethereum's 12-second average settlement time, a low frequency e-commerce platform may be sufficient. However, high frequency e-commerce will be required to provide more instantaneous speeds.

By utilizing one sidechain per vertical, ethereum's main network will be bridged which will provide the speed that is required to do multiple settlements. These settlements include purchasing, staking, escrowing, and arbitrating. This, in turn, will conclude the asset's life-cycle. All of Splyt's functionality will be done within the Splyt sidechain. When users are ready to 'cash out', Splyt will automatically convert sidechain tokens to main-network tokens.

The Staking feature will be operated within the sidechain. Users will send main-network tokens to the Splyt contract. That will hold mainnet tokens and distribute sidechain tokens to be used in the future to stake for sellers or arbitrators. Once they receive sidechain tokens, they can use it in the Splyt system to stake.

4.Research and Development Dual Tokens

In order to implement Splyt into existing e-commerce systems as seamlessly as possible, we aim to minimize expectations of changing Consumer behavior to allow adopting this system. We hope that one day Consumers will be well versed and willing to transact in cryptocurrencies when shopping online, but do not expect that to be the case in the near future. Therefore, we must create methods for operating fiat values in the Splyt ecosystem.

Splyt has already opened discussions with some partners who can help us with fiat payment gateways for credit card, ACH, and certain types of gift cards. Nonetheless, this will require a token with similar values as the fiat currencies that Consumers use to move throughout the Splyt system, and should be close to identical (or improved) to existing payment gateway interfaces.

This section proposes a novel method for a dual-token that accomplishes this goal while creating a crystalized cash flow mechanism that aligns SHOPX-holder interests with growth of the Splyt's network effect. It then analyzes other stable coin efforts and why we believe they are not appropriate for the Splyt system.

Overview

We propose a “soft-pegged” dual-token to facilitate adoption of the Splyt ecosystem while creating a unique tie between Splyt token holders and growth of network effects. Rather than seeking fully eliminated volatility, our approach acknowledges that the average or typical Consumer can psychologically transact with a token under a Schelling belief that it is nominally worth a certain value even if its real value may fluctuate within a short period. We can therefore achieve this soft-peg without the need for (over-)collateralization, making participation in this scheme more economically favorable to stakeholders in terms of risk and reward than other approaches.

A Dual-Token Application of the Quantity Theory of Money

The dual-token approach involves two types of tokens: a “Governance Token” with a fixed supply that cannot be minted or burned via smart contract, and a “Payment Token” with a supply that can be minted or burned and is intended to achieve the soft-peg.[15] Payment Tokens will be minted or burned to maintain the desired peg, with minted tokens distributed among Governance Token holders. The supply of Payment Tokens will be regulated by a smart contract implementation of an updated version of Irving Fisher’s equation of exchange formulating the quantity theory of money (the “Fisher Contract”):

[15] Current forms of Splyt use SHOPXs as both a governance and payment medium. However, this is undesirable as (like most, if not all, “utility tokens”) the mixture of purposes instills the token with volatility, possibly creating confusion among users as to its value. This solution seeks to separate those purposes among multiple tokens, with the SHOPX tokens identified in Splyt’s White Paper serving as the Governance Token.

$$M * V_n = P_n * Q$$

Where:

- *M* is defined as the target supply of Payment Tokens (“Monetary Base”),
- *V* is the average number of times a Payment Token changes hands (“Velocity”),
- *P* is the price of the Payment Token divided by the asset on which a peg is desired (“Exchange Price”),
- *Q* is the real value of goods transacted through the Splyt ecosystem (“Real Quantity”), and
- *N* is the time period over which an analysis is performed.

This approach intends to maintain the Exchange Price at or near parity to the desired peg asset (for example, stabilizing one Payment Token to be worth one American dollar). Most economic scholars who have commented on the quantity theory of money agree that the Fisher equation holds in the long-term, meaning *P* will stabilize in the long-run if, and only if, other variables stabilize as well.

Balancing the Money Base

Q can be measured as the number of Payment Tokens that transfer due to a successful transaction, assuming a Schelling point has been established for users at the desired pegged price. In our example, if Consumers believe that each Payment Token is worth one American dollar (which should be established on an interface displaying the price of an item) and one million Payment Tokens flow through the Splyt ecosystem during a given period *n*, then *Q* is equal to one million American dollars. But since this concept does not involve a tangible backing or collateral to ensure that one Payment Token is equal to one American dollar, *P* can be used to determine the nominal value of each Payment Token (in terms of Payment Token/USD). An information feed from a source external to the Splyt smart contract infrastructure is required to inform the Fisher Contract of *P*.[16] As such, *PQ* can be described as the nominal value of goods transacted through the Splyt ecosystem.

The quantity theory of money states optimal velocity balances the demand of forward-facing parties who hold a currency (perhaps as a store of value or simply due to convenience) against present-facing parties who spend currency as a medium of exchange. Since we aim to peg Payment Tokens against another asset, we assume it will act as well or worse than that asset directly in terms of its utility as a store of value. Therefore, we assume Velocity will be predominantly affected by those using it as a medium of exchange inside the Splyt ecosystem.[17] If this assumption holds, then any change to *Q* should approximately proportionally affect *V*.

Following this assumption, we recognize that if *V* and *Q* are roughly identical, then *M* and *P* ought to be roughly identical modifiers to *V* and *Q*. Therefore, if *P* is not equal to one, then *M* is not equal to one by roughly the same amount. By increasing or decreasing *M* until it equals one (assuming *V* is equal to *Q*) should ultimately result in *P* stabilizing at one as well.

[16] We are investigating the potential to use a protocol like Uniswap or Bancor, which are on-chain exchanges that facilitate arbitrage to establish a token's price, to conduct this price discovery. But we can also establish *P* from an oracle measuring the price set via off-chain exchanges if on-chain methods are inappropriate.

[17] This assumption will be used in our initial trials of the Splyt system. We are exploring other ways to conceptualize Velocity (such as via the Cambridge equation of exchange) in case this initial assumption is determined to be flawed.

Assuming Payment Tokens are already flowing through the Splyt system (indicating non-zero *V* and *Q* values), the Fisher contract can mint or burn Payment Tokens to pressure *P* into its desired value. Minted Payment Tokens will be distributed as cash flow to Governance Token holders as an incentive to responsibly govern and grow the Splyt system at "Minting Amount" *C*. The Splyt system can accumulate Payment Tokens for the Fisher Contract to burn by collecting a "Transaction Fee" on each transaction at "Burn Amount" *B*. These values can be measured over a period of time (*n*), allowing comparable figures over time of a "Mint Rate" and a "Burn Rate," respectively. Thus, the Monetary Base at the end of a given time period is defined by the following equation:

$$M_n = M_{n-1} + C_n - B_n$$

If the system is in equilibrium, meaning there is no change to *V* or *PQ* over period *n* (and therefore no change is needed to value *M*), then the Burn Amount and the Minting Amount would be equal over time *n*. If the average Exchange Price over period *n* decreases, then it follows that $C_{n+1} > B_{n+1}$ until *M* is reaches the appropriate value where *P* is expected to equal one; conversely, if there is a decrease in the average Exchange Price over period n, then it follows that $C_n < B_n$.until the same result.

Increasing or decreasing the rate of the Transaction Fee can have different effects on the system. A high Transaction Fee allows for a faster Burn Rate, meaning that *M* may be decreased faster; however, a high Transaction Fee also causes the system to become more expensive for users, potentially causing them to seek other, cheaper methods to obtain the same utility. A low Transaction Fee allows the system to be more competitive against alternate methods, but also limits the Burn Rate (and, increasingly, the amount of time for *P* to reach the desired parity). Thus, the system must balance users' need for low fees against the amount of time Payment Tokens can stay out of parity against the desired pegged asset without undermining users' psychological confidence that one Payment Token is equal to one USD.

Areas for Further Research

Theoretical consensus has not formed around the nature of velocity. The classical majority view tends to advocate that velocity is the average number of times a unit of currency is spent on a newly produced consumable good or service. However, an alternative sound perspective has also been advocated that velocity captures the average number of times a unit of currency changes owner for any reason. Further, the Cambridge school places more emphasis on the demand side of currency, focusing on the tendency to hoard cash during negative market sentiment. We still must research how this disagreement affects the Splyt system, but begin by assuming the majority view.

The dual-token model assumes that Payment Tokens minted and distributed to Governance Token holders will be quickly transferred to users of the Splyt ecosystem. However, this may not be the case. We have identified possible solutions based on user interface, including the UniSwap protocol, the placement of a "rent" that effects after tokens remain idle in an account for a certain amount of time, or incorporating more Cambridge thinking into our formula. We will continue researching these and other avenues to promote liquidity in the Splyt ecosystem necessary to maintain the dual-token concept.

We are uncertain about how long a Schelling point can be maintained with users while volatility occurs in the market for Payment Tokens, and how this affects the transaction fee. This issue will be addressed after we gather data during our initial trials.

This concept assumes that most Payment Token spending occurs within the Splyt ecosystem. This may be true naturally if the system is constructed so that transaction costs are significantly lower for exchanging Payment Goods for items sold through Splyt than for any other purpose. However, if an unforeseen event indicates this is not the case, we may explore other options to account for transactions external of Splyt like adjusting our definition of Velocity or white/black listing certain addresses for participation.

LEGAL DISCLAIMER

This Splyt Technical Paper is intended to present relevant information to potential purchasers (the "**Purchasers**" and each a "**Purchaser**") in connection with the proposed offering (the "**Token Launch**") by Splyt Core, an exempted company incorporated with limited liability in the Seychelles ("**Splyt Core**"), of cryptographic ERC20 standard tokens with the symbol 'SHOPX' (the "**Tokens**"), including information about the smart contract connected to the Tokens (the "**Splyt Smart Contract**") and the decentralized ledger technology platform and marketplace operated by Splyt Core Foundation a foundation company incorporated in the Seychelles ("**Splyt Core Foundation**" or "**Foundation**" and together with Splyt Core "**Splyt**", "**we**" or "**us**") in connection with the Tokens (the "**Splyt Platform**").

This Splyt Technical Paper should be read in conjunction with the Confidential Private Placement Memorandum and the whitepaper.

Nothing in this Splyt Technical Paper shall be deemed to constitute a prospectus of any sort, a solicitation for investment or investment advice nor does it in any way pertain to an offering or a solicitation of an offer to buy any securities in any jurisdiction. This Splyt Technical Paper is not composed in accordance with, and is not subject to, laws or regulations of any jurisdiction which are designed to protect investors. To the maximum amount permitted by applicable law, each of Splyt Core, Splyt Core Foundation and their respective directors and officers (collectively, the "**Associated Parties**" and each an "**Associated Party**") expressly disclaim and shall not be liable for any and all responsibility for any direct or any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits, and loss of use or data), arising out of or in connection with (i) the Purchaser's acceptance of or reliance on any information contained in this Splyt Technical Paper, (ii) any error, omission or inaccuracy in any such information or (iii) any action resulting therefrom.

All statements, estimates and financial information contained in this Splyt Technical Paper, made in any press releases or in any place accessible by the public and oral statements that may be made by Splyt or any Associated Party that are not statements of historical fact, constitute "forward-looking statements". Nothing contained in this Splyt Technical Paper is or may be relied upon as a promise, representation or undertaking as to the future performance or policies of Splyt.

Further, Splyt disclaims any responsibility to update any of those forward-looking statements or publicly announce any revisions to those forward-looking statements to reflect future developments, events or circumstances, even if new information becomes available or other events occur in the future.

This Splyt Technical Paper, and any sale of Tokens referred to in this Splyt Technical Paper, will be subject to and governed by any terms and conditions of purchase of such Tokens. Such terms and conditions of purchase will be delivered by Splyt to the prospective Purchaser after its receipt of a completed Indication of Interest or such other documents or materials as Splyt may require from such Purchaser in connection with the satisfaction of its AML and KYC procedures (as defined in the *Important Notices* section below).

Where there is any inconsistency between such terms and conditions of purchase and this Splyt Technical Paper, the terms and conditions of purchase shall prevail and govern.