A Semi-Fungible Token for Regulatory Compliance

Chris McConnell and Tracy Auerbach
March 15, 2018

Abstract

To help address regulatory concerns revolving around new issues of cryptographic coins and tokens (e.g. initial coin offerings) and associated secondary markets, we describe herein a specialized token S.A.I.F. and its smart contract to facilitate investments while complying with securities law. By design, S.A.I.F. only can be held by and used by *accredited investors*, a classification that underpins many U.S. and several international investment regulations. Because S.A.I.F. only can be held by accredited investors, sellers who receive S.A.I.F. can be confident that they are receiving payment from a person or entity who has affirmed their status as an accredited investor. We describe in detail how S.A.I.F. is restricted to accredited investors.

In addition, S.A.I.F. is quantitatively bound with Ether. Stronger than pegging, this hard linkage derives from autonomy: the smart contract, not the owner, receives all tokens and controls all transactions. Initially, each new user must be “seeded” with S.A.I.F. Only then can the user purchase more S.A.I.F. and invest with it. The system is frictionless. Users incur no fees whatsoever for purchasing or transferring S.A.I.F. To encourage participation, a modest redemption fee (2% maximum) accrues as a pro rata reward to all existing S.A.I.F. holders. There are no administration fees.

Background

While Bitcoin has demonstrated the value and integrity of blockchain technology, Ethereum, due to its open and accessible programmability, has unleashed a torrent of new and specialized tokens. The number of initial coin offerings (ICOs) rose dramatically in 2016 and especially in 2017. In many ways, this phenomenon still may be in its infancy. Relative to traditional vehicles like stocks and bonds, which amount to many trillions of dollars, cryptocurrency remains a tiny fraction of global financial assets.

Because participants in coin offerings often desire appreciation in their investment, many cryptographic coins and tokens seem like securities. Yet ICO issuers often go to great lengths to avoid SEC scrutiny. Surely one reason is to avoid government oversight, largely based on a culture which embraces decentralization and autonomy. Another important reason may be democratization, allowing anyone—not just the wealthy—to participate in exciting new technologies.

Ironically, traditional fundraising under the auspices of U.S. securities law has powerful advantages, perhaps none more important than this: no taxes on funds raised. Whenever an enterprise receives money, an accountant will need to know what to credit (offsetting the cash debit). If construed as equity or some financial obligation (e.g., debt), then no taxes are due; however, if construed as revenue or sales, then immediately a tax liability is created, unless the enterprise is a nonprofit.

Furthermore, ICO issuers often spend substantial sums on legal counsel, either to defend the position that their token is not a security, and/or to create an off-shore or non-profit haven. In other cases, ICO issuers go in the opposite direction. They engage a strict accreditation process whereby every investor must *verify* that they meet the accredited investor qualification by submitting documentation such as a

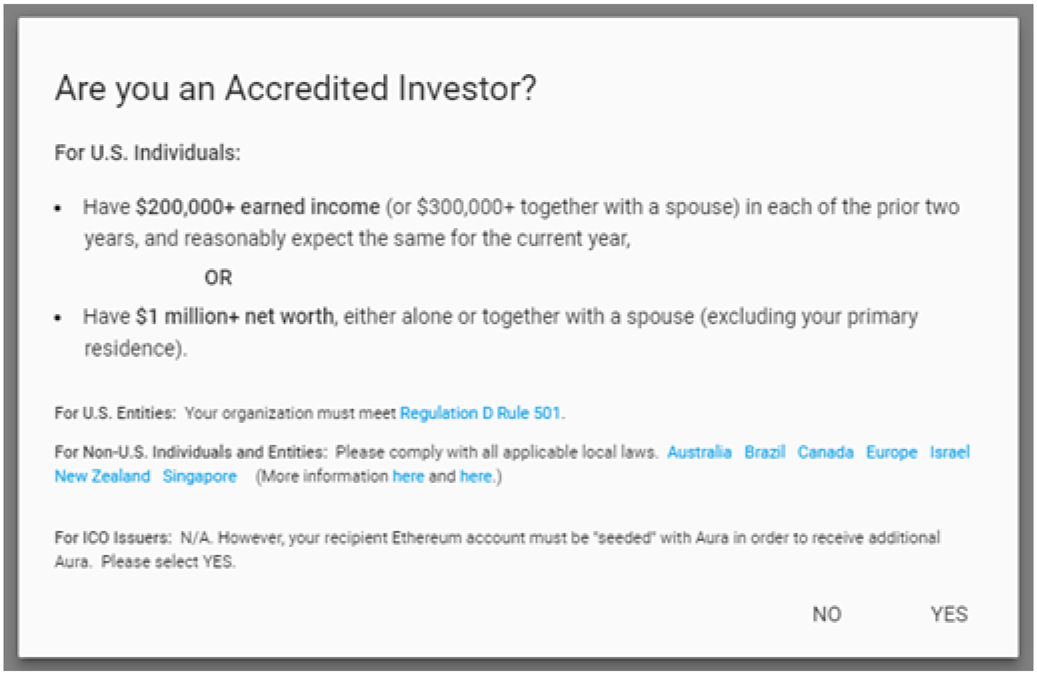
Form W-2, bank/brokerage statement, or other financial affidavit. This proof bar likely is sought in conjunction with Reg D Rule 506(c), which permits advertising, and is substantially higher than the "pre-existing substantive relationship" bar generally associated with Reg D Rule 506(b), which forbids advertising. Traditionally, most conventional U.S. private placement offerings rely on Reg D Rule 506(b), i.e., the lower bar.

What is needed is a simple and efficient mechanism that complies with U.S. regulations *and* facilitates investment in coin/token offerings, as well as associated purchases in their secondary markets. A particularly effective solution would reduce anxiety among issuers and investors, allow entrepreneurs to focus on innovation (not legal concerns), reduce costs and burdens on all fronts, and allow the cryptocurrency revolution to proceed unimpeded.

S.A.I.F. – Security for Accredited Investors

We envision a semi-fungible cryptographic token that only can be held by and used by individuals and entities that self-affirm their status as accredited investors. This self-affirmation relies on trust and prior relationships, conforming to the widely adopted Reg D Rule 506(b) exemption of the U.S. Securities Act, a.k.a the 1933 Act (https://www.sec.gov/fast-answers/answers-rule506htm.html). In addition, due to the broad use of the accredited investor terminology throughout various regulations, both in the U.S. and internationally, this self-affirmation can serve other purposes as well, especially to help protect sellers and issuers, where the compliance burden generally lies.

To transfer S.A.I.F., senders and receivers alike must possess S.A.I.F. already. Designed purposely as a "catch-22," all participants need a bit of S.A.I.F. to get more S.A.I.F. How does this happen? First, using a web application, a new user responds to a single-question, like the one shown below.



Are you an Accredited Investor?

For U.S. Individuals:

- Have **$200,000+ earned income** (or $300,000+ together with a spouse) in each of the prior two years, and reasonably expect the same for the current year,

  OR

- Have **$1 million+ net worth**, either alone or together with a spouse (excluding your primary residence).

For U.S. Entities: Your organization must meet Regulation D Rule 501.

For Non-U.S. Individuals and Entities: Please comply with all applicable local laws. Australia Brazil Canada Europe Israel New Zealand Singapore (More information here and here.)

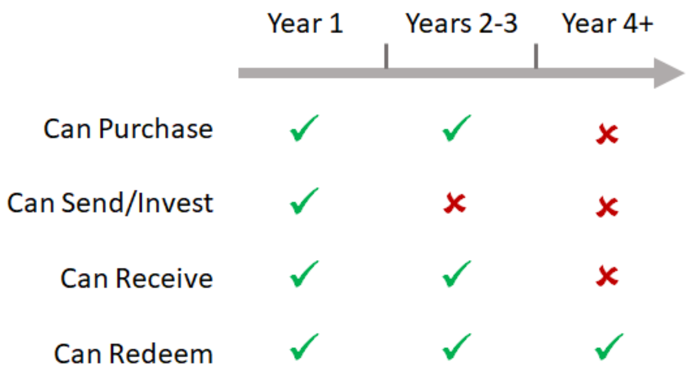
For ICO Issuers: N/A. However, your recipient Ethereum account must be "seeded" with Aura in order to receive additional Aura. Please select YES.

If the user clicks “NO,” then a message appears saying that, unfortunately, S.A.I.F. is restricted to accredited investors only, and therefore the user cannot participate. If the user clicks “YES,” then the following process occurs between the web application and the S.A.I.F. smart contract:

- The web application invokes `affirmStatusAsAccreditedInvestor()`.
- Acting independently and autonomously, and relying on trust, the S.A.I.F. smart contract accepts this response automatically.
- The smart contract then “seeds” the new user’s account with a small amount of S.A.I.F. * *Importantly, there is no external oversight.*

Once seeded with S.A.I.F., the user is free to purchase more S.A.I.F., hold S.A.I.F., redeem (sell) S.A.I.F., and invest with (send or transfer) S.A.I.F. Purchases and redemptions always are done between the user and the S.A.I.F. smart contract, in Ether swaps. S.A.I.F. transfers between external parties require that the recipient also is seeded with S.A.I.F. Thus, S.A.I.F. becomes an immutable and semi-fungible token that can be exchanged between accredited investors and ICO issuers who also have affirmed their status.

Because accredited investor status may change over time, if the user does not refresh his or her status, the S.A.I.F. smart contract will begin to limit that user’s capabilities. After one year, the user no longer will be able to send or transfer S.A.I.F. (make investments). This is because ICO issuers, and potentially other sellers, often must be certain that they are transacting only with accredited investors who have affirmed their status relatively recently. After three years, the S.A.I.F. smart contract only will allow the S.A.I.F. holder to redeem. S.A.I.F. holders always can redeem (sell) their S.A.I.F. back to the contract. Other than a periodic or per transaction volume limit (described later), *there are no restrictions on redemption.*



	Year 1	Years 2-3	Year 4+
Can Purchase	✓	✓	✗
Can Send/Invest	✓	✗	✗
Can Receive	✓	✓	✗
Can Redeem	✓	✓	✓

At any time, an accredited investor can refresh his or her status and re-enable all functions by again invoking `affirmStatusAsAccreditedInvestor()`. The user thereby reaffirms and updates his or her status, and then can carry on with any desired S.A.I.F. activity. Self-affirmation as an accredited investor is unrestricted and essentially instantaneous. In this manner, the S.A.I.F. smart contract complies with

U.S. securities law while simultaneously allowing users to self-administer. The system is completely decentralized, with no external oversight.

To implement the logic described above, the S.A.I.F. smart contract includes three modifiers, `isSeeded()`, `isFresh()`, and `isNotStale()`. The first one indicates that the user has affirmed his or her accredited investor status and has received some S.A.I.F. The second two relate to the notion that a user's status may change over time. These modifiers then constrain S.A.I.F. purchases and transfers as follows:

```
permitPurchase() = isSeeded(buyer) && isNotStale(buyer)                      (1)

permitTransfer() = isSeeded(_to) && isNotStale(_to) && isFresh(sender)       (2)
```

The one-year and three-year timeframes used by the S.A.I.F. smart contract to define "fresh" and "stale" are adjustable within limits. This flexibility anticipates the possibility that regulators may start requiring timeframes different from the current S.A.I.F. contract settings. As of today, no U.S. regulations address these timeframes. (S.A.I.F. contract settings are described in more detail later.)

To help issuers and sellers comply with Reg D Rule 506(c), the S.A.I.F. smart contract includes a more advanced function, `verifyStatusAsAccreditedInvestor(account, verifyMetaData)`. This function is restricted for use only by *trusted verifiers* whose accounts have `isTrustedVerifier` set to true. Beyond seeding and updating the target account, this verify function will write metadata (text, such as hash values and/or urls) for the account of any S.A.I.F. holder who has provided "reasonable" supporting evidence of his or her status as an accredited investor. So long as the verification time stamp associated with the account is fresh, `isVerified(account)` will return true. The metadata can be read using `getVerifyMetaData(account)`. These capabilities provide straightforward mechanisms to record and query Reg D Rule 506(c) verification information for any accredited investor.

Binding to Ether

Because its purpose is to provide assurance, S.A.I.F. is designed from the ground up to avoid risk. This includes avoiding exposure to value fluctuations separate from Ether. While establishing a pegged exchange rate between S.A.I.F. and Ether might be the expected approach, history shows that the pegging method can break down, especially when imbalances occur. As a stronger yet more flexible method, S.A.I.F. uses what we term *quantitative binding*.

The S.A.I.F. smart contract overcomes any possibility of external breakdowns by maintaining total ownership of all tokens, and complete control over all transactions. The S.A.I.F. contract uses a variable mint, creating newly purchased coins (increasing `_totalSupply`) and burning redeemed coins (decreasing `_totalSupply`). No S.A.I.F. is provided at any time to anyone at any discount, not even to the contract owner. In this regard, the contract owner is no different than any other S.A.I.F. user. The owner starts with no S.A.I.F.

Exchange Rate and Formulaic Pricing

Liquidity is one of the most important needs for a successful financial instrument. Without liquidity investors may become locked into an illiquid asset that they cannot monetize. Ideally, after making an investment, investors always should feel comfortable that they will be able to sell it. Liquidity is especially important with a semi-fungible token, where transfer and ownership are restricted to persons meeting specific qualifications (e.g., accredited investor). Such restrictions narrow the available market for liquidity.

One approach for ensuring adequate liquidity is to design the system so that the issuer (i.e., the smart contract) stands ready to repurchase the asset any time at a reasonable price. In other words, the issuer provides the liquidity. To do so, the issuer must maintain sufficient capital reserves and must adopt a trustworthy pricing and control model.

We have tested extensively a base case with the price (exchange rate) locked into one unchanging value, regardless of supply and demand. Using an Ethereum smart contract on the Ropsten Testnet, we began with a large fixed token mint held entirely by the smart contract. Then, we allowed buyers to purchase tokens at a 1:1 exchange rate for Ether. Every transaction was an even swap. The smart contract also held all swapped Ether internally, so that the Ether reserve always equaled exactly the tokens outstanding (i.e., not held by the contract). Lastly, users could redeem their tokens anytime, again at a 1:1 exchange rate with Ether. As expected, the system always maintained perfect balance, down to 18 decimal places.

Two external functions, **`purchase`**`()` and **`redeem`**`()`, handled all swap transactions. We stress-tested the system by incorporating redemption fees, and fee-sharing among token holders (discussed later). Even with these advanced features, the system still maintained perfect balance. Importantly, being an autonomous smart contract, the system demonstrated high security, requiring no private key management. Like similar Ethereum smart contracts (e.g., https://github.com/etherdelta), this smart contract performed its duties as an immutable and trustless token exchange.

A major shortcoming of a system with a fixed price is lack of incentive to purchase or hold the tokens. While fixed-price tokens may convey valuable information regarding the holder (e.g., status as an accredited investor), there is no financial upside to purchasing them, and therefore no investment opportunity. An intriguing challenge is how to design a useful financial instrument with two critical properties: 1) reasonable expectation for price appreciation, and 2) reasonable assurance of liquidity. For a semi-fungible token entirely controlled by a smart contract, we propose a novel approach using formulaic pricing based on the overall demand for the token, as follows, where both are expressed as a fraction or a percentage:

$$\Delta \text{ Price} = \Delta \text{ TokensOutstanding} \qquad (3)$$

This formula states that the price or exchange rate for the token will increase (or decrease) as the number of outstanding tokens increases (or decreases). For example, if the outstanding balance of tokens rises by, say 3%, then the price or exchange rate for that token also would rise by 3%.

We further suggest that this formula can be calculated periodically, perhaps daily or hourly. Alternatively, it can be calculated on a transaction by transaction basis. Limits then can be set for maximum and minimum percent price change, even if there are large swings in tokens outstanding. For example, we suggest limiting price increase to 10% and price decrease to 5%. These rules help stabilize the price and reduce opportunities for manipulation.

The above formula and recommended limits can be written as follows for S.A.I.F. tokens:

$$\text{Price}_{(new)} = \text{Price}_{(old)} * \text{S.A.I.F.outstanding}_{(new)} \;/\; \text{S.A.I.F.outstanding}_{(old)} \qquad (4)$$

$$\text{where Price}_{(new)} <= 1.1 * \text{Price}_{(old)} \text{ and Price}_{(new)} >= 0.95 * \text{Price}_{(old)} \qquad (5)$$

To maintain system integrity, the mathematics and associated rules must be controlled entirely by the smart contract, with no external influence or intervention. The system must be completely autonomous and immutable. In addition, at any time and to the extent possible, the smart contract must fill all purchase and redeem requests using consistent, transparent, formula-based pricing.

Clearly, formulaic pricing is not perfect. For starters, it is not symmetric. If the tokens grow in popularity, early purchasers/holders will be rewarded with the opportunity to redeem at a higher price. Likewise, later purchasers might be penalized (usually modestly) on redemption if the total number of tokens outstanding declines. Certainly, we are not suggesting that this model emulates a free market. What we are proposing is that such a formula-based pricing methodology can provide both purchase incentive (investment opportunity), and simultaneously, reasonable assurance of liquidity for redemption.

Two fundamental constraints are inherent in this system. First, with a fixed mint, tokens outstanding cannot exceed the supply. This can be overcome with a variable mint. Second, and more noteworthy, the Ether balance in the smart contract cannot decrease below zero. If the formula-based price for S.A.I.F. has increased, then it may be possible to drain all Ether in the contract by having a complete "run on the bank." In that scenario, further redemption cannot occur unless or until more Ether enters the contract via additional S.A.I.F. purchases.

We have carried out extensive Monte Carlo simulations to explore various control methods, associated parameters, and operational scenarios. From this work, we have devised a simple and effective “guardrail” that eliminates the possibility of depleting Ether held by the smart contract. By limiting the global amount of daily, hourly, or transaction-by-transaction redemption to 5% of the outstanding tokens, the same limit as imposed on price decrease, the system prevents depletion. Given a sufficiently large pool of outstanding S.A.I.F. tokens, most participants will not encounter any inconvenience. In conjunction with formulaic pricing controls, we believe that such a compromise provides necessary and prudent price protection.

Ironically, the redemption limit creates another possibility—instead of becoming exhausted, the smart contract can end up with excess Ether. Fortunately, the system easily can overcome this condition by stipulating that the S.A.I.F. price (the exchange rate with Ether) never can go below the ratio of Ether in the contract to S.A.I.F. outstanding, in effect the average price for all outstanding S.A.I.F. At that point, the redemption gate is no longer necessary and can be lifted. Regardless of further redemption volume, the S.A.I.F. price will hold precisely at this ETH-to-S.A.I.F. ratio. Even the last bits of S.A.I.F. and Ether will diminish in perfect concert (unchanging price), creating a mathematical analogy to a three-point landing.

The periodic redemption limit and associated minimum price rule may be expressed as follows:

$$\text{S.A.I.F.outstanding}_{\text{(new)}} >= 0.95 * \text{S.A.I.F.outstanding}_{\text{(old)}} \qquad (6)$$

$$\text{while Price} > \text{ETHinContract} / \text{S.A.I.F.outstanding} \qquad (7)$$

As a final measure, to protect from abuse, the system uses forward or anticipatory pricing (i.e., "post-trade" rather than "pre-trade"). Much like a bid-ask spread, forward pricing has minimal economic impact, especially for small trades relative to S.A.I.F. outstanding. Based on the relative size of the transaction (rather than any arbitrary spread), forward pricing captures the impact of each purchase or redemption, however small, in accordance with Equation 3. Importantly, forward pricing tends to penalize potential price manipulators; and, due to quantitative binding, reward all other participants.

The following table summarizes the advantages and disadvantages of several different pricing models: fixed pricing, formula-based pricing, and two free-market pricing scenarios, one with an actively-traded asset ("hot"), and one with an illiquid one ("cold"). As shown, a carefully-designed formulaic pricing system, implemented in conjunction with prudent limits governed by an autonomous smart contract, can lead to an optimal blend of price stability, liquidity, and protection from manipulation. Furthermore, formulaic pricing can offer transparency and a genuine opportunity for price appreciation.

Table 1 – Comparison of Pricing Models

	Fixed Pricing	Formulaic Pricing	Market Pricing (hot)	Market Pricing (cold)
Trading Liquidity	**Perfect**	**High**	**High**	**Low**
Price Stability (relative to Ether)	**Perfect**	**Good**	**Low**	**Medium**
Prospects for Appreciation	**None**	**Good**	**Good**	**Fair**
Chances of Manipulation	**None**	**Low**	**Low**	**Medium**

Redemption Fee Sharing

As mentioned before, we have explored redemption fees and fee-sharing in conjunction with the fixed-price smart contract discussed previously. Originally conceived as a possible mechanism to cover administrative costs, we foresee fees serving a much better purpose as an incentive for S.A.I.F. participants to hold S.A.I.F., and thereby earn fees. With this benefit in mind, the S.A.I.F. smart contract implements a modest redemption fee (set at 2% maximum) entirely for the benefit of S.A.I.F. participants.

The S.A.I.F. smart contract handles redemption fees by burning 2% of the redeemed S.A.I.F. This results in accretion to all remaining S.A.I.F. holders. While the largest S.A.I.F. holders will receive the most gain, even the smallest holders will receive some benefit. Calculations are extremely precise due to Ethereum's native 256-bit architecture.

In the case of deploying a fixed mint rather than a variable mint, we recommend using a point-based accrual system for fee distribution, and disbursing fees on-demand whenever a user initiates a transaction. This *amortization of work* method avoids potentially large computational overhead on the blockchain. Nick Johnson describes this method in detail here: https://medium.com/@weka/dividend-bearing-tokens-on-ethereum-42d01c710657.

Contract Functions, Fields, and Settings

The S.A.I.F. smart contract uses the *ERC20 Interface*, including these primary functions: **`totalSupply`**`()`, **`balanceOf`**`()`, **`allowance`**`()`, **`transfer`**`()`, **`approve`**`()`, and **`transferFrom`**`()`. It also incorporates SafeMath (https://theethereum.wiki/w/index.php/ERC20_Token_Standard). A suite of ten event definitions emit relevant information whenever primary or secondary functions are invoked.

Public fields defined by the S.A.I.F. smart contract are like those of other ERC20 tokens (symbol, name, decimals, _totalSupply, and owner account). S.A.I.F. uses uint256 with 18 decimals throughout. The database structure for S.A.I.F. holders includes six fields: **`balance`** (S.A.I.F.), **`affirmDate`**, **`verifyDate`**, **`ETHpaymentDue`**, **`verifyMetaData`**, and **`isTrustedVerifier`**. In the spirit of decentralization, both the contract owner and other trusted verifiers are permitted to modify the last field by invoking **`toggleTrustedVerifier`**`(`*`account`*`)`.

Beyond the ERC20 Interface, the S.A.I.F. smart contract exposes four additional primary functions: **`affirmStatusAsAccreditedInvestor`**`()` and **`verifyStatusAsAccreditedInvestor`**`()`, as well as **`purchase`**`()` and **`redeem`**`()`. Conforming to Solidity best practices, the **`depositPayment`**`()` *payable* fallback function supports purchase() by performing the task of depositing Ether into the contract. Similarly, the **`withdrawPayment`**`()` function supports redeem() by adhering to the recommended *Check-Effects-Interactions* withdrawal pattern (http://solidity.readthedocs.io/en/develop/common-patterns.html). Prior to withdrawal, **`ETHpaymentDue`**`()` will return the quantity of ETH awaiting withdrawal for any given account.

Aggregate S.A.I.F. and ETH balances are returned by two functions, **`S.A.I.F.outstanding`**`()` and **`ETHinContract`**`()`, defined as follows where **`totalETHpaymentsDue`** is the sum of all ETHpaymentDue:

`S.A.I.F.outstanding() = _totalSupply`
(8)

`ETHinContract() = this.balance – totalETHpaymentsDue` (9)

Five secondary functions exposed by the S.A.I.F. smart contract include: **`getAffirmDate`**`()`, which retrieves a user’s self-affirmed accredited investor date stamp, plus **`getVerifyDate`**`()`, **`getVerifyMetaData`**`()`, **`isVerified`**`()`, and **`isTrustedVerifier`**`()` for information relating to the “reasonable steps to verify” associated with Reg D Rule 506(c).

An internal function calculates the forward exchange rate (S.A.I.F. price relative to ETH) on a transaction-by-transaction basis. The S.A.I.F. contract imposes no time constraint. As the S.A.I.F. price rises, `limitRedemptions` is set to true and the `redemptionLimit` is set as 5% of outstanding S.A.I.F. This gate is lifted (limitRedemptions = false) when the ETH:SAIF price falls to its stipulated minimum defined as the ratio of all internal ETH to all outstanding S.A.I.F. (the average price for outstanding S.A.I.F.).

Already we have discussed the contract settings that only the owner can control. These are summarized below. We subscribe unconditionally to principles of autonomy and decentralization; however, it is prudent to maintain access to a few select controls in anticipation of unforeseeable events (such as changes in legislation) that could impact the efficacy of the S.A.I.F. smart contract. These `ownerOnly` settings are constrained to prevent any adverse effects on users.

Setting	Contract Value	Minimum	Maximum
SeedAmount	0.0001	0.00000001	0.01
DaysFresh	370	14	-
DaysStale	1110	30	-
DaysVerify	370	14	-
RedemptionFee	2%	0%	2%

Summary and Conclusions

We have described a semi-fungible token, S.A.I.F., and its associated smart contract. S.A.I.F. begins by deploying a "catch-22" scheme which assures that all users have affirmed their status as accredited investors. We have described rules that anticipate investor status potentially changing over time, and how the S.A.I.F. smart contract responds accordingly. These rules are accomplished immutably, in a fully decentralized fashion, where all users can self-administer their own status. In addition, the S.A.I.F. smart contract supports advanced features supporting "reasonable steps" taken by third parties to verify status.

Recognizing that semi-fungibility presents an inherent challenge in terms of liquidity, we have described a series of experiments and Monte Carlo simulations with different formulaic pricing models. In all cases, the S.A.I.F. smart contract owned all tokens, controlled all transactions, and best assured that owners could purchase and redeem from the smart contract at any time. While the base case of fixed, unchanging price certainly met the liquidity and security needs, it failed to offer any financial incentive for purchasing and holding tokens.

To address this need, we have proposed a simple formulaic pricing model whereby the price of S.A.I.F. rises and falls in proportion with the demand for S.A.I.F., but within certain limits. These limits help stabilize the price, even if there are large swings in the number of tokens outstanding. Furthermore, we have described detailed pricing policies that thwart potential manipulation, and reward all other holders. Fundamental to this system is the notion of quantitative binding: where all tokens and transactions are owned and controlled by the smart contract. The system is frictionless, and there are no administration fees. The system is quantitatively bound, meaning that there is no token leakage whatsoever.

Appendix A – Summary and Weblinks to Select U.S. Securities Regulations

I. Public Offerings and International Offerings

   a. Reg A and A+
      https://www.sec.gov/oiea/investor-alerts-bulletins/ib_regulationa.html

   b. Form S-1
      https://www.sec.gov/about/forms/forms-1.pdf

   c. Reg S
      https://www.sec.gov/rules/final/33-7505.htm

II. Private Offerings - Reg D
   https://www.ecfr.gov/cgi-bin/text-idx?SID=465dc4251925603a672a767b7916fc49&node=sg17.3.230_1498.sg11&rgn=div7

   a. Rule 501 - Definition of accredited investor.

   b. Rule 502 - Information requirements and other conditions.

   c. Rule 503 - Filing notices and amendments.

   d. Rule 504 - For offerings up to $5M in a 12-month period.
      https://www.sec.gov/fast-answers/answers-rule504.html

   e. Rule 506 - For offerings without regard to dollars.
      https://www.sec.gov/fast-answers/answers-rule506htm.html

      i. Rule 506(b)
         1. No general solicitation or advertising.
         2. Unlimited number of accredited investors.
         3. Up to 35 other sophisticated investors.

      ii. Rule 506(c)
         1. Can broadly solicit and generally advertise.
         2. Unlimited number of accredited investors.
         3. Reasonable steps to verify status, e.g. Form W-2.

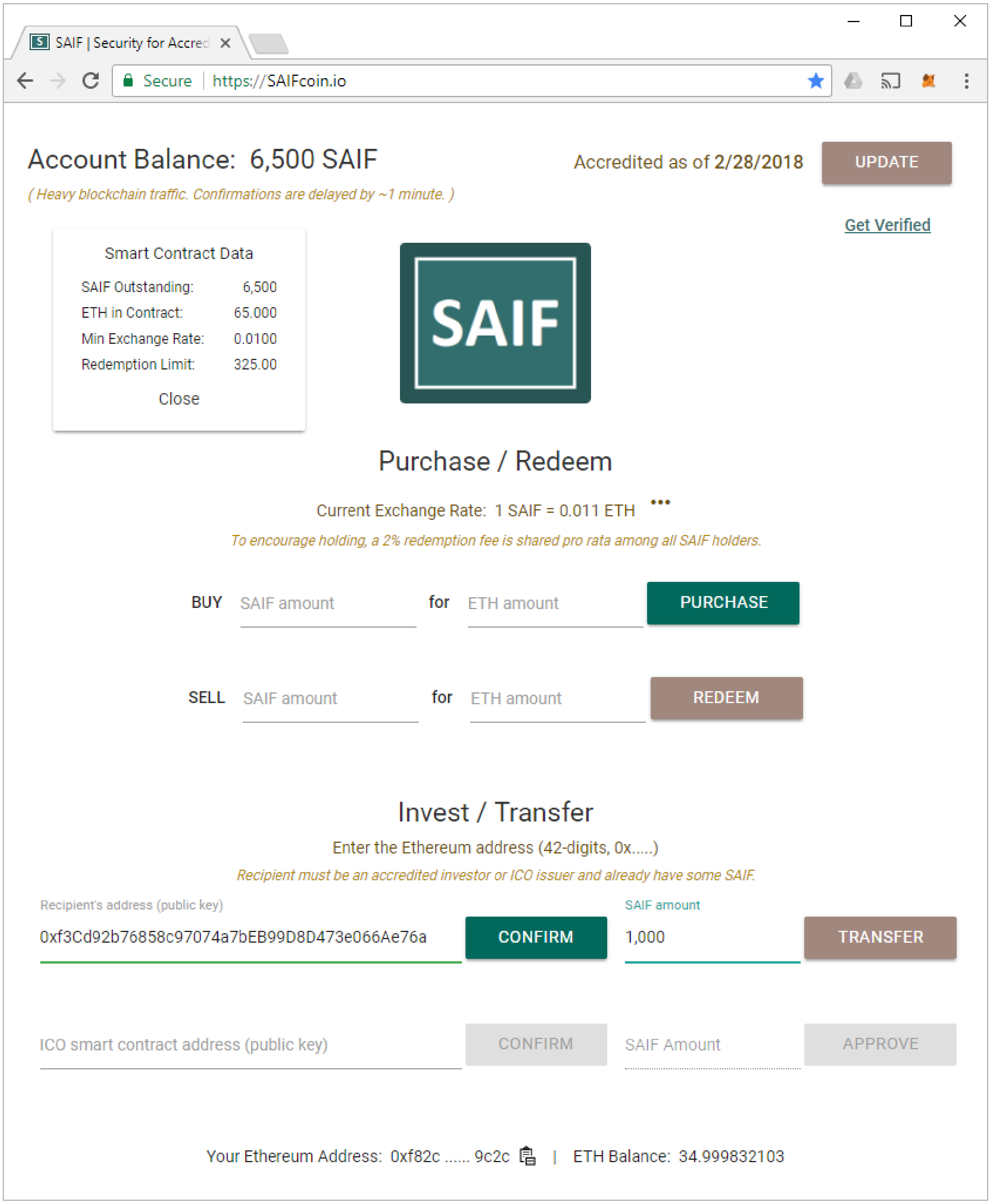
III. Resale of Restricted Securities

   a. Rule 144
      https://www.sec.gov/reportspubs/investor-publications/investorpubsrule144htm.html

   b. Section 4(a)(7) and 4(a)(1½)

      i. Each purchaser must be an accredited investor.
      ii. No general solicitation or advertising.
      iii. Neither sellers nor buyers can be "bad actors."

Appendix B – S.A.I.F. DApp (https://S.A.I.F.coin.io)



Account Balance: 6,500 SAIF

(Heavy blockchain traffic. Confirmations are delayed by ~1 minute.)

Accredited as of **2/28/2018**

Smart Contract Data

SAIF Outstanding:	6,500
ETH in Contract:	65.000
Min Exchange Rate:	0.0100
Redemption Limit:	325.00

Purchase / Redeem

Current Exchange Rate: 1 SAIF = 0.011 ETH

To encourage holding, a 2% redemption fee is shared pro rata among all SAIF holders.

BUY SAIF amount for ETH amount

SELL SAIF amount for ETH amount

Invest / Transfer

Enter the Ethereum address (42-digits, 0x.....)

Recipient must be an accredited investor or ICO issuer and already have some SAIF.

Recipient's address (public key): 0xf3Cd92b76858c97074a7bEB99D8D473e066Ae76a

SAIF amount: 1,000

ICO smart contract address (public key)

SAIF Amount

Your Ethereum Address: 0xf82c ...... 9c2c | ETH Balance: 34.999832103

Appendix C – Provisional Patent Application

The authors intend to contribute S.A.I.F., its smart contract, and all associated inventions to the public. As a means of establishing prior art, and to avoid potential conflicts with others who may claim rights to implementing semi-fungible tokens, on September 18, 2017, we filed the following provisional patent application (U.S. Application Serial No. 62/560,018).

Systems and Methods for Specialized Cryptocurrency Transactions

ABSTRACT

A method for specialized cryptocurrency transactions may comprise determining that an entity comprises a member of a class. An account associated with the cryptocurrency or a blockchain may be generated for the entity. At least a portion of a unit of the cryptocurrency may be transferred to the entity. A transaction may be generated to transfer units of the cryptocurrency to a recipient. It may be determined that the recipient is a member of the class associated with the cryptocurrency. If the recipient is a member, the units may be transferred to the recipient. The transaction may be added to the blockchain. If the recipient is not a member, the units may be exchanged for units of another cryptocurrency. The units of the other cryptocurrency may be transferred to the recipient.

At the time of this writing, we have no plans to pursue further intellectual property protection on this matter.