

# Smart Vows Contract

Marco Montes Neri

**Community based insurance is possible leveraging a specific form of smart contracts called smart vows. We propose the creation of applications calling the Smart Vows Contract in the Ethereum Network to facilitate the offering and acceptance of policies (vows) among community members to hedge against unforeseen events.**

## Vision

Some companies make vows to their clients and default. Insurers have done this in the past and consequently a regulatory framework was once needed and eventually created. According to common knowledge, the purpose of regulatory insurance law is to protect consumers by (mainly): “Monitoring and preserving the financial solvency of insurance companies” and “Controlling market conduct and preventing unfair trade practice”. In simple terms, the rules of the game the insurers must meet are: don’t be evil and don’t make promises you can’t keep.

Leveraging technology that exists today, it is possible to enforce these agreed rules, encouraging the entry of new market players to make trustworthy *vows* to consumers without the need of traditional controls. We call these reliable promises *Smart Vows* and believe that, in the near future, a new set of centralized and decentralized organizations will start designing and offering them transforming part of the insurance industry.

## Context

Smart Vows fit as a subset of Smart Contracts bounded to conditional expressions involving future events that would affect people's economies in the case of occurring. An example of one Smart Vow look something like this: *Organization O* vows that IF *Authority A* states your home *H* burns, THEN will pay you *X* amount of USD. For this example to work as a Smart Vow, it would require few things to be in place:

1. A trusted source of information (Authority) responsible for heralding new events (This can be a prediction market or a reputable institution).
2. A transparent Proof Of Funds (POF) mechanism, which is essential for the establishment of trust among policyholders (an Ethereum contract can hold an amount of assets equal or greater that to the sum of all coverage limits)
3. Availability of information for scrutiny and performance analysis
4. An identity system that facilitates the association of a specific policy with the property ownership.

## Security

Cryptography will enable the encryption and source authentication of new events notifications. Honesty of Authorities will be motivated by establishing coverage limits. This means that the reputation of *Authority A* has to be economically more valuable than the aggregated amount of coverage limits associated with a single event. Competition and reputation systems among Authorities is desirable. The validation of policy conditions (vow

logic) can be programmed into an open source Ethereum contract that will, besides validating conditions, automate the claim payment process.

## The Contract

Within the Ethereum contract two roles exist: *Acceptant* and *Vower*; *Vowers* take risk motivated by the potential of making a profit and *Acceptants* cover themselves from risks. *Vowers* are members of a community who have the permission to create *Vows*. Permissions are given through a tokens called RRR described in more detail in the next section.

The data structure corresponding to these Vows expressed in Solidity language looks like this:

```
// Vow data structure
struct Vow {
    address vower;
    address oracle;
    uint funds;
    uint payoff;
    uint numAcceptants;
    uint premiums;
    uint expiration;
    uint taar; // total amount at risk
    mapping (uint => Acceptant) acceptants;
}
```

This data structure shows two Ethereum personas (addresses) related with each Vow: One persona is the vower we mentioned above, and the second one, denominated oracle is an “offchain” trusted third party responsible of notifying the contract that when real world events occur. The functions of the Ethereum contract are four: createVow, acceptVow, settleVow and terminateVow, shown in the following Diagram 1.

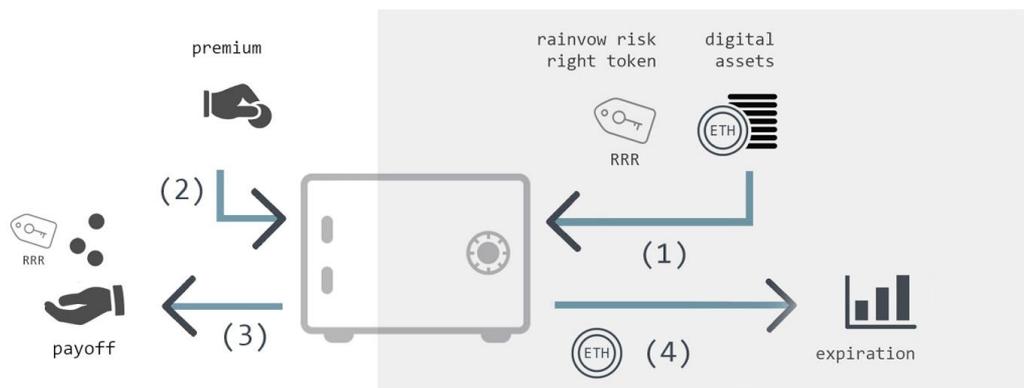


Diagram 1

The step 1 is the creation of the Vow which means publishing offers that other members can accept. Step 2 is accepting the offer through an Ethereum transaction where the amount of such transaction is equivalent to the premium. The step 3 is a settlement function. Acceptants would receive ETH if contract conditions were met, or RRR as a reward if not. Step 4 is the termination of the Vow, premiums minus claims are collected providing feedback to the actuarial model for the improvement of efficiency in the underwriting of future vows.

## The Rainvow Risk Right Token.

Permissions in protocol are given via a token denominated *RRR* and the issuance of such tokens occurs when a *Vow* expires without claims. *RRR* are then sent to the *Acceptant* as a form of reward and as an incentive to keep participating in the system. This token is a retribution for the participation in the construction of the actuarial model and, after aggregation of enough data, represents a possibility to make a profit by providing risk capacity (funds) to the *Vows*.

## The Rainvow club

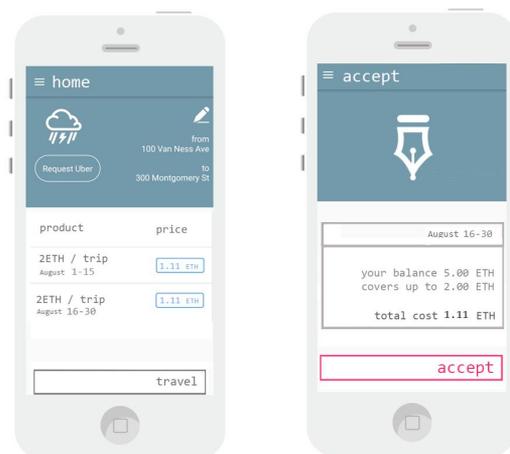
Rainvow is an implementation of the Smart Vow Protocol that acts as an authoritative system for the validation of coverage policy conditions associated with transportation extra costs in rainy days. It is designed to serve the majority of community members (club). Rainvow is designed for urban communities with access to connectivity and smart phones whose members typically walk or bike but have to take uber rides when there is rain or snow. This situation typically generates more demand and a surge in price affecting economically everybody. *Vows* are policies that start at least 10 days in the future when there is enough weather uncertainty to make the model work.

In the Rainvow the oracle is an off chain system responsible of notifying the contract that the trip took place while it was raining. In this case it depends on two services: The Weather data API and Uber API.

## Rainvow Organization

Rainvow is the entity responsible for implementing the Smart Vows Contract and publishing it in the Ethereum network. It also builds the actuarial model, acts as Vower and maintains an application that serves as a reputable oracle by integrating with The Weather API and the Uber API.

The application allows users input their usual paths they are interested to be covered for. According to this data users can also browse and accept the vow, if available in the specified city.



## Roadmap

Private beta app is available for Android users and be open in October 2016 for both Android and iOS. The app targets the Ethereum community and people interested in becoming *RRR* holders. Early 2017 the initial data will be published as well as the Ethereum contract will be made open source for audits.