The POINT Token System: Gamification And Achievements For The Blockchain

August 2017

Version 1.0

There are huge upsides to bringing the benefits of the blockchain to rewards and loyalty programs. Just as the blockchain has transformed the exchange of money, blockchain technologies have the potential to transform how businesses incent and reward customers. The core tenets of decentralization and transparency that underlie blockchain can be equally applied to loyalty programs and gamification initiatives.

Imagine a landscape where consumers earn the same points whether they are buying a beverage at a coffeeshop, filling up their gas tank or purchasing a plane ticket. Likewise, imagine consumers can spend points across multiple properties and even exchange those points for other currencies. Consider a future where businesses that implement loyalty programs outsource the entirety of unlocking achievements and earning points to the blockchain, relying on its existing infrastructure for managing a user's point balance and awards.

Such a system has benefits for businesses and consumers alike.

For consumers, such a system would bring transparency to today's closed systems of rewards and loyalty programs. The mysteries of what a point was "worth" and how it was "earned" would no longer be hidden behind a vendor, but instead would be stored on the public blockchain. Once participating in this system, a consumer could earn points on one property and spend them on another. A user's points would be transferable to any vendor that participated in the system. How many times have consumers been asked to join a loyalty program and declined because they didn't want to manage yet another account with another card and another system? A unified--yet decentralized--loyalty system could motivate users to participate, because their points had meaning beyond just one vendor.

For businesses, putting rewards and points on the blockchain has many advantages. Firstly, the entirety of managing and tracking points would be handled by the blockchain itself, reducing system management and customer support costs for awarding and storing users' awards and point balances. Using an on-chain system has security benefits as well, reducing a company's attack surface area. The transparency of using a public cryptocurrency also has benefits for dealing with regulators and tax compliance. Lastly, participating in a system that users *want to use* means more customers motivated by rewards, more customers earning and spending points, ultimately enabling businesses to reap the well-documented benefits of gamification.

The POINT Token System

The POINT Token System is a solution designed to meet these requirements. It runs on the Ethereum public blockchain and provides an end-to-end, multi-tenant system for managing rewards and loyalty programs on the blockchain. It has three major components:

1. An ERC20 compliant token that can be issued to users by businesses
2. A methodology for storing tenants of the system, the rewards issued by those tenants and the users who have earned those rewards
3. A governance model for managing the point economy

All three of these components are managed by an Ethereum smart contract. This contract is already written and can be found here: https://github.com/pointtoken/Contracts.

This system leverages the existing Ethereum ecosystem on several fronts. Because it uses an ERC20 compliant token, it benefits from existing wallets, exchanges, marketplaces and token systems. Because it stores achievements and point balances on-chain, it benefits from the Ethereum global computer for storage and a distributed ledger.

How It Works

The logic of the smart contract itself is the best way to understand the rules of the POINT Token system. We will define some terms and then explain how the contract works:

OWNER: The owner of the entire POINT token smart contract and system.

AWARDER: A tenant within the POINT token system.

USER: An end user of the POINT token system.

ACHIEVEMENT: An event that can be awarded to a USER by an AWARDER.

POINT: The ERC20 token governed by the POINT token smart contract.

The OWNER has the power to mint POINTS and transfer them to AWARDERS. The OWNER can either transfer the POINTS gratis to the AWARDER or the OWNER could sell the POINTS for Ethereum or potentially some other cryptocurrency. The thinking here is that the cost for an AWARDER to buy POINTS would always be at a discount compared to their current market value because the reason to have POINTS is to give them away.

The OWNER is represented by an Ethereum account. This account could be a “human” account or a smart contract.

Note that POINTS have zero decimal places a currency, keeping things simple and congruent with the idea of points.

The OWNER has the power to add and remove AWARDERS to the system. This list of AWARDERS is stored on the Ethereum blockchain in storage. An AWARDER is represented by an Ethereum address and is stored as a hash table mapping address to a boolean, which represents if the AWARDER is in good standing. This system allows the OWNER to suspend AWARDERS and reinstate AWARDERS depending on business factors. The code for storing, adding, removing and checking status of AWARDERS is as follows:

```
mapping(address => bool) awarders;

function isAwarder(address _addr) constant returns (bool) {
    return awarders[_addr];
}

function addAwarder(address _addr) onlyOwner {
    awarders[_addr] = true;
    AwarderAdded(_addr);
}

function deleteAwarder(address _addr) onlyOwner {
    awarders[_addr] = false;
    AwarderRemoved(_addr);
}
```

An AWARDER can create AWARDS, issue ACHIEVEMENTS and give POINTS. An ACHIEVEMENT is an integer that is owned by the AWARDER. ACHIEVEMENTS are stored in storage on the Ethereum blockchain. There is a one-to-one mapping between AWARDERS and ACHIEVEMENTS. In other words, if AWARDER **XYZ** claims ACHIEVEMENT **1001**, no one else can claim that ACHIEVEMENT and it is immutably stored on the blockchain and cannot be deleted. Because the ACHIEVEMENT is an unsigned integer there is plenty of space to create ACHIEVEMENTS. Here is the code for managing ACHIEVEMENTS:

```
mapping(uint => address) achievements;


function addAward(uint index) returns (bool) {
    if (!isAwarder(msg.sender)) throw;

    if (achievements[index] == 0x0) {
        achievements[index] = msg.sender;
        AwardAdded(msg.sender, index);
        return true;
    }
    return false;

}
```

Note how only AWARDERS can create ACHIEVEMENTS. Also note how, if the ACHIEVEMENT is already claimed, it can't be used.

The meaning and rules that govern an ACHIEVEMENT are not the concern of the POINT token system. What an ACHIEVEMENT is called and how it is earned is outside the scope of the system. It is the responsibility of the AWARDER to create a mechanism for storing what an ACHIEVEMENT means and the rules governing it.

AWARDERS can award ACHIEVEMENTS. To reduce costs and bloat for AWARDERS, ACHIEVEMENTS are stored as events/logs on the Ethereum blockchain, much in the way transactions are stored. When an ACHIEVEMENT is awarded, AWARDERS can optionally award POINTS at the same time. They are not required to do so and can optionally pass zero when awarding an ACHIEVEMENT, allowing for a tenant to use the system to award ACHIEVEMENTS but not participate in the POINTS economy. There are no rules governing if how many times an ACHIEVEMENT can be earned; this is entirely governed by the system built by the AWARDER. Here is the code for awarding an ACHIEVEMENT:

```
    function awardAchievement(address user, uint aType, uint amount)  {

        if (!isAwarder(msg.sender)) throw;
        if (achievements[aType] != msg.sender) throw;
        if (super.balanceOf(msg.sender) < amount) throw;
        super.transfer(user, amount);
        Award(msg.sender, user, aType, amount, now);

    }
```

Note that only AWARDERS are able to award ACHIEVEMENTS and they can only award ACHIEVEMENTS that they own. If the AWARDER does not have a sufficient balance of POINTS, the method will fail. If they do have a sufficient balance to give the POINTS, an award event is fired to the logs, passing the AWARDER, the USER, the ACHIEVEMENT, the number of POINTS awarded and a timestamp. Here is the signature of the event itself:

```
    event Award(address indexed _from, address indexed _to,  uint indexed _type,
uint _amount, uint _date);
```

It indexes on AWARDER (_from), USER (_to) and AWARD (_type) so that fast queries can be issued to the blockchain to get a list of all the awards issued to a USER.

A Real-World Example: eFolio

Let's walk through a real-world scenario of a company that has already implemented the POINT token system. eFolio (https://www.efolio.info) is a website that allows users to track and analyze their cryptocurrency portfolios. eFolio was onboarded as a tenant to the to the POINT token system and provisioned 1000 points to do with as they wished. Once added as an AWARDER, the eFolio engineers used the POINT token command line interface (CLI) to create 3 achievements to incent users on the website. (The Point token CLI is available today here: https://github.com/pointtoken/CLI.) On the POINT token system, these achievements were represented by the numbers 1001, 1002 and 1003. For the eFolio engineers, they mapped these numbers to achievements:

Number	Name	Rule
1001	Create a portfolio	Awarded if a user adds 5 positions to their portfolio. Can only be earned once. Worth 1 point.
1002	Convert currencies	Awarded if a user uses the conversion feature of the website to convert between cryptocurrencies. Can be earned 10 times. Worth 1 point.
1003	Give feedback	Awarded if a user finds a bug on the site or provides valuable feedback. Worth 10 points.

Note how the different achievements have different rules. This business logic and validation is entirely owned by eFolio and is outside the scope of the POINT token system.

The eFolio engineers created a backend service that listened for the events representing these achievements.

Let's walk through how a user would unlock an award on eFolio. Consider the case of a user named Jane. Imagine she is on the eFolio website and adds five positions to her portfolio, unlocking award **1001**. When this event occurs on the website, Jane is notified in the application, letting her know that she unlocked an achievement and that she needs to claim her reward. To claim her reward, she would have to do two things. First, she would have had to provide her Ethereum wallet address. Second, she would have to complete a Recaptcha. Efolio implemented Recaptcha to prevent someone from writing a bot to earn achievements.

After validating the Recaptcha, the backend service would then call the POINT token system smart contract, sending the achievement ID, the number of points to give, the award and Jane's Ethereum address. The backend service would sign the transaction with eFolio's Ethereum address – the address they used when onboarding as a tenant onto the system.

The smart contract would log the fact that Jane got the award and would transfer points from the eFolio token balance to Jane's token balance. The website would then provide the transaction ID to Jane, so she could see the transaction get validated on the blockchain. Once processed on the Ethereum network, Jane would see her POINT token balance had increased in whatever wallet she used to keep tokens. Additionally, she could go back to the eFolio website and see her new point balance as well as what achievements she had unlocked. Behind the scenes, the website would render this data by querying the blockchain, passing the Jane's Ethereum address as an identifier.

Road Map

If you are interested in learning more about the POINT Token system and potentially partnering with us, please contact info@pointtoken.io.