

Quantified Prestige Documentation

V0.03

Michael Hrenka

2015-09-04

General notes:

- Feel free to contact me (radivis@radivis.com) if you notice a mistake or have a suggestion how to improve any part of this documentation.
- Terms which refer to specific parts of the Quantified Prestige system are written capitalized. This serves to distinguish them from the more general everyday meaning of the respective non-capitalized words.
- There is a Discourse forum that I've set up in 2015 which can be used to discuss the details of Quantified Prestige. Please check out the "Digital Abundance Network" category of the Fractal Future Forum.
- This documentation is distributed under a CC BY 4.0 license.

Contents

I. What is Quantified Prestige?	4
II. What is Quantified Prestige good for?	4
III. How does a Qualified Prestige Network work?	5
1. A few comparisons	5
1.1. Comparison between QP and budgets	6
1.2. Comparison between QP and karma systems	6
1.3. Comparison between QP and Klout	6

IV. Quantified Prestige Light	7
2. Esteem	7
2.1. Automatic EP Reallocation	9
2.1.1. Simultaneous Automatic EP Reallocation	10
3. Prestige	12
3.1. Trust	12
3.2. Prestige Computation	12
V. Quantified Prestige - Full Version	14
4. Esteem	14
4.1. Default System	14
4.2. Advanced System	15
4.2.1. Subtypes of allocated EPs	15
4.2.2. User Circles	16
4.2.3. Splitters	16
4.2.4. The general concept of an Allocator	17
4.3. User defined Esteem Points	17
4.4. The Esteem Matrix	17
5. Trust	18
5.1. Trust Scores, Trust Levels and Trust Factors	18
5.2. The Native Trust System	19
5.2.1. Top Trust	19
5.2.2. Esteem Trust	20
5.3. Using External Trust Systems	22
6. Prestige	22
6.1. Received Esteem Points	22
6.2. Spread Factor	22
6.3. How the Prestige Score is calculated	23
6.4. Why your Prestige doesn't depend on the Prestige Scores of your sup- porters directly.	24
6.5. Prestige dependent Prestige computation	25
VI. Prestige Guided Applications (PGAs)	26
7. Electronic Currency Distribution Funds (ECDFs)	26
7.1. How does an ECDF work?	27
7.2. Who pays for the Fund and what is it good for?	28

8. Fluido Currencies (FCAs)	29
8.1. The discrete system	29
8.1.1. Demurrage	30
8.1.2. Evaporation	31
8.1.3. Adjusting the timescale	32
8.1.4. Problems of the discrete system	32
8.2. Continuous system	33
8.2.1. Getting to the basic temporary Fluido formula	33
8.2.2. The basic global Fluido formula	35
8.2.3. Fluido computation with evaporation	37
8.2.4. Continuous Fluido transfer: Inflows and Outflows	37
8.2.5. Event times	38
8.2.6. Final Fluido formulas	40
9. Prestige Polls	40
VII. User Interface	41
VIII. Appendix	42
A. Reasoning behind reputation economies	42
B. Possible issues of Quantified Prestige	44
B.1. Wealthy participants could simply buy Prestige	44
B.2. Celebrities and widely popular persons are favored in this system	44
B.3. General popularity doesn't mean that someone or something is actually good	45
B.4. This system could favor already wealthy persons	45
B.5. Minorities could be discriminated against	45
B.6. Multiple user accounts per person could distort Prestige Scores	45
B.7. Fluido is just another "fiat currency" that's backed by nothing	45
C. Where do these ideas come from?	46

Part I.

What is Quantified Prestige?

Quantified Prestige is basically a rewarding system. Users of a **Quantified Prestige Network (QPN)** can reward others for appreciated actions or behavior by giving them **Esteem Points (EPs)**. Those Esteem Points are then used by the network to compute a positive reputation index, the **Prestige Score (\mathbb{P})**, for each user. By default, the Prestige Scores of all users can be seen publicly.

Those publicly visible Prestige Scores provide the basis for a reputation based economy. So called **Prestige Guided Applications (PGAs)** can expand the system, for example by coupling Prestige Scores with digital currencies like Bitcoin, or the native **Fluido (\mathbb{F})** currency which is generated through Prestige. So, Quantified Prestige is actually a hybrid reputation/money system. It can act as building block for an economic system which is much more flexible and universal than the current monetary system.

Part II.

What is Quantified Prestige good for?

A Quantified Prestige Network has the purpose to allow its users to reward other users. There are some different types of rewards:

1. *Appreciation*: It's nice to get Esteem Points from others, because that shows that you are appreciated in some way. Prestige is more valuable than points in karma systems like those on Reddit, because Prestige is harder to get, due to the fact that each user can only allocate a limited amount of Esteem Points to others, and also due to the (optional) trust system.
2. *Status*: Prestige Scores are publicly visible information, so you can quickly see which users have high status in a network by viewing a ranking of the user with the highest Prestige Scores. It's also very nice to appear on top (or close to the top) of such a ranking.
3. *Trust*: When using the native Trust System you can quickly see which users are probably legitimate users, namely those with high Trust Levels. Only trusted users are able to profit from being a member of a Quantified Prestige Network. The native Trust System has the purpose to prevent illegitimate use of the QPN.

4. *Basic income*: By using Prestige Guided Applications which couple Prestige Scores with payments in an electronic currency, it is very easy to implement a basic income in that currency for all (trusted) users. In the Fluido currency the basic income is a native (although optional) element of the system. For basic incomes in other electronic currencies a Distribution Fund is required from which (trusted) users are paid in regular intervals. Such Distribution Funds can be financed with a part of an income stream of an organization, or by donations.
5. *Bonus payments*: In addition to an (optional) basic income, users can also get bonus payments in an electronic currency in proportion to their Prestige Scores. This functionality is the foundation for a reputation based economy.

Quantified Prestige Networks are software based services which can be run centralized or decentralized. They can be used within certain organizations or be open for all, globally. In the first case, the QPN is a useful tool for such an organization. In the latter case, the QPN would represent a global reputation based economic system, or at least a global reputation system.

Another very interesting use of a QPN is to reward the creators of any kind of product, or even whole organizations by allocating Esteem Points to an associated **Esteem Splitter**, which forwards and distributes the Esteem Points to the creators of a product or the members of an organization. This is much more comfortable than allocating Esteem Points to individual users.

Part III.

How does a Qualified Prestige Network work?

A QPN has three basic components and is connected to an arbitrary number of Prestige Guided Applications (PGAs). As basic components we have:

1. A mechanism for **Esteem** allocation with which users allocate Esteem Points (EPs) to other users.
2. An optional **Trust System** which helps to prevent exploitation of the system.
3. The reputation index **Prestige** (\mathbb{P}). The more Esteem Points a user gets from others, the more Prestige he has.

1. A few comparisons

To get a clearer idea how Qualified Prestige works, let's compare it with a few similar systems: A budget, karma systems, and Klout.

1.1. Comparison between QP and budgets

Imagine you set up a household budget: You determine how much you spend within a month in different areas, say food, clothes, transportation, entertainment, and so on. Because you have a limited amount of money to spend you need to get clear about your priorities and allocate your money to those different areas accordingly. If you are unhappy with your current budget, you can reallocate your funds to areas which are more important to you, but you have to spend less on some selected other areas in compensation.

In a QPN you have a fixed amount of Esteem Points which you can allocate to different network users, products, or organizations you like. Similarly to your household budget you can reallocate your Esteem Points at any time. Usually this would mean that some people who have got some amount of Esteem Points from you get less, and some get more after the reallocation.

While your monetary income that determines the total size of your household budget may change over time, the number of Esteem Points you can distribute is fixed forever. Once you have already allocated all of your Esteem Points, you really need to reallocate some of your EPs if you want to allocate more EPs to someone.

1.2. Comparison between QP and karma systems

Let's take a look on karma systems like those on Reddit. They roughly work like this: There is a message board on which users write posts. Other users can upvote or downvote those posts. The karma score of a user depends on the number of upvotes minus the number of downvotes that user has got from others for all of her posts. All votes have the same fixed value, which is usually 1 (or -1 for downvotes).

A QPN could be seen as a more complex version of a positive karma system in which there are only upvotes. But users have more freedom in a QPN, because they don't need to bind their "upvotes" to posts, they can simply give EPs to users they find generally awesome for any kind of reason. And "upvotes" don't need to have the same value, but their value depends on the number of EPs which are allocated to a certain user.

In a QPN the "karma score" is called Prestige (\mathbb{P}) and it depends on those Esteem Point "upvotes", but also on a few other factors.

1.3. Comparison between QP and Klout

Klout is a service that analyzes your popularity in different social media networks and compresses that into the Klout Score which is supposed to be a measure for influence. While Klout collects user generated data from different services to compute the Klout Score, the Prestige Scores in a QPN only depend on the activity within that QPN. Because Klout uses a whole lot of data, it is not very transparent how your Klout Score arises from all that information. Compared to that complexity, your Prestige Score in a QPN arises in a not too complicated way from the Esteem Points you get, user Trust Levels and user Distribution Levels.

Even though the Klout Score is mainly based on the activity within other social networks, Klout has its own internal reputation metric, the “+K”, which behaves like a karma upvote, but is bound to specific areas of expertise. You have a daily quota of +K points which you can give other Klout users. Just choose an area of expertise that user has chosen and give your “+K” vote in that area if you think that user is really an expert in that field. The “+K” has similarities to the EPs you can allocate to users in a QPN. However, EPs don’t have a daily quota, but simply a global quota which doesn’t change ever. If your quota is used up, you need to “take EPs back” in order to spend them on other users.

Part IV.

Quantified Prestige Light

Quantified Prestige Light (QPL) is a simple version of Quantified Prestige that lacks advanced features. It is intended to be used by small groups of people (no more than 200), who know and trust each other to some degree. Since QPL is a very basic version of QP, it is easier to understand, so it is explained here before proceeding to the more advanced Full QP.

The basic idea is that each user of a QPN has a **Prestige** score that depends on how much other users in the network esteem that user. If we want a quantifiable Prestige score, it suggests itself to use a quantified notion of esteem, which is simply called **Esteem** and is explained in the next section.

2. Esteem

Now that we are looking for a way to quantify Esteem, let’s take a look at some simple, but problematic approaches:

Motivation 1: Quantifying esteem - binary

The most simple way to quantify esteem is to make it binary: Either you hold another person in esteem or not. While it is possible to set up a reputation system like that, it couldn’t be used to express subtle degrees of approval. To see why this would be an unreasonable restriction just imagine that you only could pay a single Dollar bill to any person ever, no matter what you buy from that person, whether that’s a car or a single apple. Clearly, it’s not possible to build a smoothly working economy on such a basis.

Motivation 2: Quantifying esteem - “ratings”

Now what if users could rate other users like you can rate products on many websites - on a scale between 1 and 5 stars? The problem with that approach is that getting anything less than 4 stars would feel rather bad, making “micropayments” of esteem rather awkward. Getting some esteem should be seen as better than getting none, but in a “rating” framework that won’t work well.

Motivation 3: Quantifying esteem - unlimited points

So, what about giving points to other users? Let’s for a moment imagine that each user could give an arbitrary number of “esteem points” to other users, and that there is no limit to the number of esteem points that each user can give away, and that each point counted for as much as each other point. Such a scheme would soon lead to massive inflation of these esteem points, since users would constantly outcompete others with respect to their “esteem influence” by giving away higher and higher amounts of esteem points. That would make the system more than useless!

In Quantified Prestige Light each user has a *limited* number of **Esteem points (EPs)**. The default number η of EPs is 12000. This is a fixed quota which actually doesn’t change over time! The only thing that a user can do with EPs is to *allocate* them to other users, and to take them back again at any time when desired. When another user allocates EPs to you, these points do not increase your own number of EPs which you can allocate to others - the EPs you got from the other user remain in her possession and can be taken back again at any time. In other words: You can only allocate your own EPs, not the EPs you receive from others. You also cannot allocate EPs to yourself! It is not possible to trade EPs in any way that allows one user to get more (or less) EPs to allocate.

Rationale 1: Why 12000?

It's a big number (it's over 9000!) that has great divisibility properties. Consider that

$$12000 = 12 \cdot 10^3 = (2^2 \cdot 3) \cdot (2 \cdot 5)^3 = 2^5 \cdot 3 \cdot 5^3$$

can be divided by

2, 3, 4, 5, 6, 8, 10, 12, 15, 16, 20, 24, 25, 30, 32, 40, 48, 50, 60, 75, 80, 96, 100, 120,

and larger numbers. If there are 120 people in your QPL besides you and you feel egalitarian, you could simply allocate 100 EPs to each of them. Afterwards you can still take back some EPs from people you don't esteem very much and reallocate them to users you hold in really high esteem.

There are two basic types of EPs **allocated EPs** and **unallocated EPs**, and together they should add up to 12000, or whatever the value of η in the actual system implementation turns out to be. Common sense tells us that you should only be able to allocate unallocated EPs. Allocated EPs are "used up", so to speak. However, you can unallocate allocated EPs at any time, so they become new unallocated EPs which you can spend freely. In many cases this would be quite bothersome, so the system can bypass this necessity to unallocate EPs manually via automatic EP reallocation, which is explained in the following subsection:

2.1. Automatic EP Reallocation

Assume you have allocated 100 EPs to user X and you want to increase that number to 1000 EPs. Further assume that you happen only to have 500 unallocated EPs left. If you don't want to unallocate the missing 400 EPs manually from someone, automatic EP reallocation will come to the rescue. What happens is that the 400 missing EPs will be unallocated from the whole of all users you are currently allocating EPs to, except from user X . Each user will lose the same fraction of EP she gets from you.

How is that fraction calculated? First note that you have 11500 allocated EPs. You already allocate 100 EP to user X , and he's not supposed to lose anything, so we need to see what happens to the 11400 EPs that you allocate to other users. At the end these users will need to share $12000 - 1000 = 11000$ among themselves, because then X will get 1000 EPs. So, the fraction of EPs that each of those other users will retain is $\varrho_\alpha = \frac{11000}{11400} \approx 96.5\%$ and is called the **reallocation factor**. In other words, each of them will lose the fraction $\frac{400}{11400}$, or about 3.5% of her former EPs. In general, this fraction could be characterized as "missing EPs" / "total old EPs of other users".

2.1.1. Simultaneous Automatic EP Reallocation

There's also the possibility to allocate different amounts of EPs to many different users simultaneously. Also in this setting there's automatic reallocation. You can set the new desired EPs of some selected users. In fact, you can set the new EP count to the old EP count if you want that the Esteem of that selected user doesn't change through automatic reallocation. Only EPs from unselected users will be taken. Each unselected user will lose the fraction "missing EPs" / "total old EPs of unselected users" of her EPs. Here, the "missing EPs" are the total sum of the newly set EPs for the selected users minus the sum of the EPs those users already get from you and minus the unallocated EPs you happened to have left at the beginning.

Shiny Formula 1: EP Reallocation Formula

To formalize this matter, let's introduce some notation. Each user has a user id number, and we call the user with the user id i simply "user i " for brevity. Let E^i be the number of EPs you allocate to user i at the moment and \vec{E}^i the number of EPs user i will get from you after automatic reallocation. Note that the i 's on top are upper indexes and **not** exponents! Upper indexes always denote values related to *other* users in QP, while lower indexes refer to values associated with the users themselves. For each selected user with id j you choose a number E_{new}^j of EPs. Collect all the id numbers of unselected users in a set U and the id numbers of selected users in a set S . Finally, let E^0 be the number of unallocated EPs you have at the beginning. When we define the EP difference value

$$\vec{E} = \sum_{j \in S} E_{\text{new}}^j - \sum_{j \in S} E^j = \sum_{j \in S} (E_{\text{new}}^j - E^j),$$

then the general automatic reallocation formula is:

$$\vec{E}^i = \begin{cases} E^i & , \text{ if } i \in U \text{ and } \vec{E} \leq E^0 \\ \left(1 - \frac{\vec{E} - E^0}{\sum_{j \in U} E_j}\right) E^i & , \text{ if } i \in U \text{ and } \vec{E} > E^0 \\ E_{\text{new}}^i & , \text{ if } i \in S \end{cases}$$

It's clear that the only really interesting thing that happens is that the values E^i for $i \in U$ get multiplied with the reallocation factor

$$\rho_\alpha = 1 - \frac{\vec{E} - E^0}{\sum_{j \in U} E_j},$$

if $\vec{E} > E^0$, that is if you don't have enough unallocated EPs left, so that real automatic reallocation is required.

Example 1

The general formula will be exemplified in this simple example, which is illustrated by the following table:

i	E^i	E_{new}^i	\vec{E}^i
1	2000	—	800
2	3000	—	1200
3	1500	1500	1500
4	2500	2500	2500
5	1000	2000	2000
6	0	4000	4000

How were the values 800 and 1200 in the top right calculated? The users 1 and 2 were unselected, so their EPs were multiplied by the reallocation factor ϱ_α which we need to compute. First, we calculate the EP difference value

$$\begin{aligned}\vec{E} &= \sum_{j \in \{3,4,5,6\}} (E_{\text{new}}^j - E^j) \\ &= (1500 - 1500) + (2500 - 2500) + (2000 - 1000) + (4000 - 0) \\ &= 5000.\end{aligned}$$

At the begining there are 10000 allocated EPs, so there are

$$E^0 = 12000 - 10000 = 2000$$

unallocated EPs. And since

$$\sum_{j \in \{1,2\}} E^j = 2000 + 3000 = 5000,$$

the reallocation factor is

$$\varrho_\alpha = 1 - \frac{\vec{E} - E^0}{\sum_{j \in U} E^j} = 1 - \frac{5000 - 2000}{5000} = \frac{2}{5} = 40\%,$$

which is the factor that 2000 and 3000 need to be multiplied with to get the values 800 and 1200.

Comment 1: Fading Esteem as Side Effect of Automatic Reallocation

Automatic reallocation has the “side-effect” that the EPs a certain user gets from you tend to decline after a while with this system, which reflects the fact that popularity slowly fades, unless it gets “refreshed” regularly. So, you would need to renew your EPs for that user from time to time, if you want him to have a relatively large fraction of your EPs - unless you use the advanced features of the Full version of QP, or you don’t use the automatic reallocation system at all.

3. Prestige

Now that we know how Esteem works in QPL understanding Prestige is simple. The Prestige score of a user is basically the sum of all EPs he gets from other users. There’s only a little issue of reputation systems that needs to be addressed first:

3.1. Trust

In reputation systems there’s the danger of Sybil attacks, in which a single person would create many different user accounts in order to gain more influence or even reputation in the system. Quantified Prestige addresses this problem by introducing the concept of **Trust**. In the Light version the concept is binary: A user is trusted or not, and there’s nothing in between. What is meant with “Trust” is the situation that a user is believed to be a real person who is not trying to exploit the system with Sybil attacks or other illegitimate manipulations.

A user gets trusted by registering on the QPL network and then getting trusted by an administrator of that network. This mechanism assumes that new users are known by at least one administrator, or that users are able to verify their identity in a more objective way, for example by using an already relatively trustworthy online identity, like a Google or Facebook account.

The set of all registered users is partitioned into a set of trusted users T_1 and a set of untrusted users T_0 . Therefore, the term “ $i \in T_1$ ” means that the user with id i is trusted.

3.2. Prestige Computation

Shiny Formula 2: QPL Prestige Formula

Let the number E_j^i denote the Esteem Points that user j allocates to user i . The **Prestige score** \mathbb{P}_i of user i is then:

$$\mathbb{P}_i = \begin{cases} \sum_{j \in T_1} E_j^i & , \text{ if } i \in T_1 \\ 0 & , \text{ if } i \in T_0 \end{cases}$$

Note that E_j^i defaults to 0 when it is not defined (by user j), or if $j = i$. That means user i can't allocate Esteem Points to himself in order to get more Prestige.

Untrusted users have neither influence nor Prestige, though they can start allocating EPs to other users in the hope that they will be trusted soon, so that these EPs become effective.

And that's already the whole QPL system. It's designed to be easy to understand. The user won't get confused by the more advanced concepts and features of the Full version of QP.

Prestige scores have interesting applications, and since QPL provides Prestige scores they can be used by Prestige Guided Applications, which are described in the part after the next part.

Rationale 2: Why the Full Version of QP is Awesome

So, if the Light version of QP suffices for interesting applications why even bother with the Full version at all? There are a couple of reasons:

1. The Full version has advanced features that provide better control over the automatic EP reallocation mechanism.
2. It enables much finer and more comfortable control of EP allocation.
3. This finer system can also be used to allocate EPs to whole groups of users, for example the creators of a certain product.
4. The Full version of Quantified Prestige is optimized for large networks.
5. Users are incentivized to allocate EPs to a large number of other users.
6. The Trust system of the Full version is much more sophisticated and flexible than that of the Light version.

Part V.

Quantified Prestige - Full Version

If you have skipped the previous part of this documentation please read it now, because this part assumes familiarity with the Light version of Quantified Prestige.

4. Esteem

The basics of Esteem Points are the same as in the Light version: Each user has a total of $\eta = 12000$ (default value) Esteem Points and can allocate them to other users, as well as taking them back at any time. What's different is that there's an optional *advanced EP system* which enables finer control over automatic EP reallocation, and that EP allocation in general is made much more flexible with different kinds of so-called **allocators**.

Each user has a total of η (can be any positive rational number; default: $\eta = 6000$) **Esteem Points (EPs)** which he can allocate to other users. What's important is that the amount of EPs of each user is fixed at η . Nobody can get more EPs or lose some of them. The only thing you can do with EPs is allocate them to other people (but not yourself). You can reallocate them at any time you want to. It is not possible to trade EPs in any way that allows one user to get more (or less) EPs to allocate.

4.1. Default System

There are basically two types of EPs: **Allocated EPs** and **unallocated EPs**.

Allocated EPs are those which you have allocated to another user. Unallocated EPs are those which you haven't allocated to anyone yet. You can turn unallocated EPs into allocated EPs by taking some of them and allocating them to another user. Say, you have 2000 unallocated EPs and allocate 500 of them to the user with ID-number i (in this example we take the default value $\eta = 6000$). Then you have 1500 unallocated EPs afterwards and 4500 allocated EPs (since they need to sum up to 6000 EPs in total). Obviously, EPs can be either allocated or unallocated. They cannot be both at once (unless you somehow use some weird quantum mechanical trick).

You can unallocate some of the EPs that you already allocated to a certain user in order to get more unallocated EPs. Those can be reallocated to any other user or just kept idle.

If you want to allocate EPs to a user and you don't have enough unallocated EPs, the default option is to take unallocated EPs from all other users proportionally, which means that each of those users will lose the same fraction of EPs allocated to him. For example, if you only have 1000 unallocated EPs left and want to allocate 2000 EPs to a user with ID j , first she gets the 1000 unallocated EPs and then 1000 EPs which you

have already allocated. The users which got those allocated EPs had in total 5000 EPs, but now have to share only 4000 EPs among them. The result is that each of those users will only have $4000/5000 = 4/5 = 80\%$ left of those of his former EPs which you allocated to him.

Of course, this automatic reallocation option can be switched off. The reason why it's the default option is to eliminate the need for manual reallocation of EPs once (almost) all EPs are allocated. There's also the effect that Prestige Scores tend to decline after a while with this system, which reflects the fact that popularity fades after a while. So, you would need to renew your EPs for a certain user regularly, if you want him to have a large fraction of your EPs - unless you use the advanced features, or don't use the automatic reallocation system at all.

4.2. Advanced System

4.2.1. Subtypes of allocated EPs

Whenever you give EPs to a participant, you have the option to choose the subtype of EPs you want to allocate to her. There are three different subtypes of allocated EPs:

1. **Stable EPs**, which are immune to automatic reallocation.
2. **Flexible EPs**, which are the default option and behave as described in the preceding paragraph.
3. **Volatile EPs**, which are used up first when automatic reallocation happens.

Flexible and volatile EPs are also called unstable EPs, as they tend to be shifted around a lot, once automatic reallocation of EPs kicks in. The only difference between all those subtypes is their behavior under automatic reallocation. How that works out becomes clear when considering some examples.

1. Suppose you have 500 unallocated EPs and 1000 volatile EPs. If you allocate 1300 EPs to a participant, first the 500 unallocated EPs are used up. Still, 800 EPs need to be taken from the allocated EPs in order to make the reallocation possible. Those EPs will be taken exclusively from the volatile EPs, because you have enough of them. Your volatile EPs will be reduced to 200 uniformly, which means that each user which got EPs from you will only have $20\% = 200/1000$ of those volatile EPs left after the reallocation. Stable and flexible EPs aren't touched at all in this example.
2. Let's imagine that you want to allocate 2000 EPs to a participant in the situation above. You have 500 unallocated EPs, 1000 volatile EPs and 1500 flexible EPs. Now, the unallocated and volatile EPs are all used up for the reallocation. This means that everyone who got volatile EPs from you loses them at once. But you still need to take 500 allocated EPs in order to complete the transaction. Those are taken from the flexible EPs, all of which are reduced to $2/3 = (1500 - 500)/1500$

of their former value. So, you will have 1000 of those previous flexible EPs left. Again, your stable EPs remain untouched by this transaction.

3. In the situation of the previous point you cannot reallocate more than 3000 EPs (as that's the total of your unallocated and unstable EPs), unless you turn some of your stable EPs into unstable or unallocated ones.

4.2.2. User Circles

You can create **User Circles** in order to manage your EPs better. Externally, user circles are handled exactly as regular users you have in your distribution list, so you can allocate stable, flexible, or volatile EPs to circles. It's just that internally a user group behaves similarly to the distribution list as a whole. Every circle c has η_c (default: $\eta_c = \eta = 6000$, we will use that default in the following example) internal esteem points, or IEPs. You can allocate different types (stable, flexible, or volatile) IEPs to the users within a circle. If the circle gets a certain amount of EPs, those are split up according to the IEPs. For example, if a circle gets 600 EPs, then a single user within that circle has 200 IEPs, she will only get $200 \cdot (600/6000) = 200 \cdot (1/10) = 20$ EPs. The number "EPs that a group gets" divided by "6000" is the **Distribution Ratio** of that circle. In the previous example the distribution ratio was 1/10. So group members will get "IEPs · distribution ratio" EPs.

If automatic reallocation of EPs happens on the global level, the distribution ratio of your circles will decrease (unless you are actually allocating EPs to that circle, in which case it will increase). Global automatic reallocation doesn't mess with IEPs, but if the distribution ratio of a circle decreases, the EPs every member of that circle gets allocated from you will also decrease.

Automatic reallocation within a circle works like the regular automatic reallocation on the global level, as described in the previous subsection, but restricted to the members of the circle. So, internal automatic reallocation will change nothing outside of that circle.

Circles are private, in the sense that others don't see which circles you have created and in which of your circles they are in (unless you allow them to see that information).

4.2.3. Splitters

It is possible to allocate EPs to **Splitters**, which behave similarly to user circles, but have public visibility: Everyone sees all splitters, while only you (and the people you allowed them to) see your private user circles. The purpose of splitters is to make allocation of EPs to creators of a single product or to whole organizations more comfortable. A user who makes a splitter defines an internal esteem distribution with IEPs which behave like the IEPs in circles. That means that this user determines who gets which share of the EPs which get allocated to the splitter.

A group who collaborated to create a product can set up a splitter to make it easy for others to allocate EPs in a fair way to all those who worked on that product. Of course, that group will need to discuss which distribution is "fair" and create a splitter

accordingly. Similarly, whole organizations can set up a splitter to give each member a specifically defined share of incoming EPs.

4.2.4. The general concept of an Allocator

The general distribution list of a user, circles, and splitters all share very similar mechanics. All of them can be seen as **Allocators**, which have the purpose of allocating incoming EPs to other Allocators, or to users. So, Allocators only have the purpose of "moving around" EPs. An EP distribution list of a user is called an **Initial Allocator**. The number of allocated EPs of that users go into the Initial Allocators and go out to User Circles, Splitters, or individual users. Each Allocator bundles all the incoming EPs which get allocated to that Allocator and "forwards" them to other Allocators and users, in proportion to the IEPs that an Allocator or user gets. Those IEPs are defined by the **Allocator Manager(s)**. Users are always managers of their own Initial Allocator, Circles, and the Splitters they have set up. But they can also grant other users management rights for specific Allocators. Only the creator of an Allocator can appoint (or fire) managers of that Allocator. Ordinary Allocator Managers can't do that.

4.3. User defined Esteem Points

User are not bound to use the default values η or η_c , but use their own numbers $\eta(i)$ or $\eta_c(i)$. In order not to disturb the system by that change, the QPN only takes the ratio of allocated (I)EPs into account, but publicly displays everything as if it would calculate with the default values for η or η_c .

4.4. The Esteem Matrix

The most important data of a QPN are the EP relations. Which user allocates how much EPs to which other users? For two users with IDs i respectively j the variable $E_i'^j$ denotes how much EPs i allocates to j . Esteem allocation is bottom-up in the sense that EPs go from the lower index to the upper index. Because of the possibility of customized EP quotas $\eta(i)$, the system rather calculates with the normed variables

$$E_i^j := \frac{1}{\eta(i)} E_i'^j.$$

All those numbers for all users are saved in the quadratic Esteem Matrix $E = (E_i^j)_{i,j}$ (not to be confused with the identity matrix $\mathbb{1}$). The i -th row of that matrix shows how much normed EPs user i gets allocated from other users. The j -th column of the matrix displays how much normed EPs user j allocates to other users.

This matrix is a compact representation of the most important data of a QPN, but it shouldn't be publicly visible, because it would reveal which user allocates how much EPs to which other users even if some users prefer to keep their EP allocation pattern private. If QPN administrators have access to the Esteem Matrix data they should disclose that fact.

5. Trust

There is an issue that makes it rather necessary to include a trust system. If multiple full accounts per user were allowed, this would pose a couple of problems:

1. A user could create multiple accounts which allocates esteem to each other in order to get a lot Prestige.
2. That user would have more influence, because he could allocate a real multiple of η EPs.

Especially the first point is quite critical. If this kind of behavior cannot be minimized, the system would soon become worthless. Users aren't supposed to generate Prestige for themselves directly in that cheap way. The second point is also quite important, but not so critical.

To prevent the creation of multiple full accounts for a single user or similar schemes which can be used to exploit the system, a trust system is used. This trust system consists of

1. A **native trust system** which generates trust scores only based on data which is intrinsic to the system
2. Or an **external trust system** which uses data from outside the system to verify legitimate users
3. Or **both**.

5.1. Trust Scores, Trust Levels and Trust Factors

Each users i has a Trust Score $T(i)$ which determines her Trust Level $T_\Lambda(i)$ in the following way: There is a Trust Treshold Value T_α (default: $T_\alpha = 100$). If the Trust Score $T(i)$ is below T_α the Trust Level $T_\Lambda(i)$ is 0, and it's at least 1, if $T(i) \geq T_\alpha$. The Trust Level always increases by one, if the Trust Score doubles, after having reached the Trust Threshold Value. So, it's 2 for a Trust Score of $2T_\alpha$, 3 for a Trust Score of $4T_\alpha$, 4 for a Trust Score of $8T_\alpha$ and so on. A closed formula for that is:

$$T_\Lambda(i) = \max \left(0, \left\lfloor \log_2 \left(\frac{2T(i)}{T_\alpha} \right) \right\rfloor \right).$$

Trust Scores, and therefore also Trust Levels are a measure of trustworthiness. Another interpretation of trust is that it's the probability that a user is legitimate and doesn't plan to exploit the system. The Trust Factor most closely resembles that probability. It is computed by the Trust Level of the respective user. Albeit it might more accurate to compute it from the Trust Score directly, this would cause problems with the computation of Prestige (similar to those mentioned in the last paragraph of the Esteem Trust section and the section "Why your Prestige doesn't depend on the Prestige of your

Supporters“). So while a Trust Factor of $\max\left(0, 1 - \frac{1}{T(i)}\right)$ might be most reasonable, the formula this system uses for the Trust Factor $T_{\Phi}(i)$ of the user i is

$$T_{\Phi}(i) = 1 - \frac{1}{2^{T_{\Lambda}(i)}}.$$

The Trust Factors are the only variables from the Trust System that factor into the computation of Prestige Scores directly.

5.2. The Native Trust System

Each user i has a Native Trust Score $T_N(i)$. This score has two components:

1. The **Top Trust** $T_T(i)$, which is especially important when the system is young.
2. **Esteem Trust** $T_E(i)$, which is based on the Esteem allocation relations between users. This component is important in more mature systems.

A user’s Native Trust Score is then simply the sum of her Top Trust and her Esteem Trust:

$$T_N(i) = T_T(i) + T_E(i).$$

5.2.1. Top Trust

This type of trust comes from the top, more specifically the **top users**. All administrators of the system are top users. In addition, the ν non-administrator users with the highest Prestige scores are also top users. (default: $\nu = 10$) With this definition of top users in mind, top trust is described by a few rules:

1. Every top user gets a bonus of τ_A to their top trust T_T . (default: $\tau_A = 6000$)
2. The previous bonus is removed once a top user ceases to be a top user.
3. Each top user can grant τ_{Λ} top trust to any other user. (default: $\tau_{\Lambda} = 1000$)
4. The respective top user can take that top trust grant back at any time.
5. Also top trust grants from top users disappears once the respective top user loses his top user status.
6. Top users can start a mistrust petition against any user. If two other top users sign the mistrust petition, the top trust of the user who is the target of that petition gets reduced by τ_V . (default: $\tau_V = \tau_{\Lambda} = 1000$)
7. The malus from a mistrust petition gets removed once one of the signing users loses top user status or revokes her support of the petition.

5.2.2. Esteem Trust

This type of trust is based on the Esteem relations between users. The idea behind Esteem Trust is "independent verification" of trustworthiness. That means that a user gets Esteem Trust if several "independent" users deem that user trustworthy. To simplify matters for users, only the EPs which one user allocated to another user are counted as indicator of trustworthiness.

For that purpose we consider the Esteem Network consisting of the users of a QPN as nodes and their non-trivial EP-relations as directed links.

We say that a pair (i, j) of users with IDs i and j are **2-unlinked**, iff there is no path of length 2 or less in the Esteem Network connecting them - any directed links are treated as undirected links, so the direction of links doesn't play a role here. For three users with IDs i, j , and k we say that a pair (i, j) is **2-unlinked up to k** , iff there is no path of length 2 or less connecting them in the reduced Esteem Network without k (and all links between k and other nodes).

Remark: It is straightforward to define the concept of n -unlinkedness (up to k), but that is not used in this system. It would be computationally very expensive to check the Esteem Network for more than 2-unlinkedness, at least for large QPNs.

The idea is that for a user k getting independent verification of trustworthiness means that there is a pair (i, j) of users, both of whom allocate EPs to k , and who are unlinked up to k . This makes it very hard for users to get Esteem Trust from clusters of users. Even though it's natural to expect getting trust by a cluster, there's the problem that clusters could be corrupt and would give out trust to users who shouldn't be trusted. Independent verification bypasses that potential problem.

Now, we need a function that translates Esteem into Trust. Given $x = E_i^j$, what degree of trust $f(x)$ should that imply? Let's start with a few sensible demands on the function f :

1. No Esteem means no Trust: $f(0) = 0$.
2. Maximum Esteem means maximum Trust, which is set to 1 = 100%: $f(1) = 1$
3. The function f is non-negative, continuous, and increasing.
4. A doubling of Esteem always should have the same effect, more precisely:

$$\frac{f(2x)}{f(x)} \equiv c \quad , \text{ for a constant number } c.$$

5. Diminishing returns: The gains in Trust decrease with growing Esteem, or mathematically: $f'(x) < 0$ for all x between 0 and 1.

It is not hard to see that any function of the form

$$f(x) = x^a$$

for a strictly between 0 and 1 suffices all those conditions, with c being 2^a . In the absence of further criteria, the most natural choice for a is the golden middle: $a = \frac{1}{2}$. In this case the function f becomes:

$$f(x) = \sqrt{x}.$$

In the spirit of independent verification by pairs of users, let's make a little thought experiment and say that k gains Trust, iff i and j are 2-unlinked up to k and are asked independently whether k is trustworthy and they say that k is trustworthy with probabilities $f(E_i^k)$ and $f(E_j^k)$ respectively. Then the probability that k gains Trust is simply $f(E_i^k) \cdot f(E_j^k)$. This motivates the following preliminary formula for k 's Trust Score:

$$T_{E,\text{preliminary}}(k) = \sum_{(i,j) \in U_k} \sqrt{E_i^k} \cdot \sqrt{E_j^k},$$

where U_k is the set of pairs of users who are not k and who are 2-unlinked up to k .

That formula doesn't consider whether the users i and j are trusted or not. This can be remediated by factoring in the Trust Levels of i and j . Also the Trust formula should be proportional to T_α , which makes it irrelevant how high that threshold value actually is, apart from aesthetical considerations (small Trust Scores could feel disappointingly low). What really matters is a factor T_ε (default: $T_\varepsilon = 100$) that determines how easy it is to get Esteem Trust. These considerations inspire the final Trust Score formula:

$$T_E(k) = T_\varepsilon T_\alpha \sum_{(i,j) \in U_k} \sqrt{E_i^k} \cdot T_\Lambda(i) \cdot \sqrt{E_j^k} \cdot T_\Lambda(j).$$

Trust Levels play a large role when it comes to propagating Trust to others. One issue that requires some attention is that the Trust Scores of k depend on the Trust Levels of other users. If the Trust Levels of 2-unlinked users i and j increase, then the Trust Score of k will also increase, so that k 's Trust Level will increase too, which would mean that k gives more Trust to others, which may lead to an increase in their Trust Levels again. So, the system will experience Trust Level update cascades, which may propagate for a while. But since your Trust Score needs to double in order to gain a Trust Level, longer cascades are not very likely. Actually, this is the rationale behind the concept of Trust Levels. If the Trust of k depended directly on the Trust Scores of other users, the update cascades would be unlikely to stop finally, since Trust could be propagated in cycles which have no reason for stopping. By introducing quantized Trust Levels and making them increase very slowly with increasing Trust Scores, the problem becomes computationally feasible.

There's a nasty side of Esteem Trust: It can decrease even if nobody decreases the amount of EPs she allocates to you. The reason for this is that as the Esteem Network expands and new links are created, it may happen that two users who once were 2-unlinked up to a user k get connected directly or over a different user l . In that case, user k won't get Esteem Trust from that pair of users anymore. So, in order to get an increasing Esteem Trust Score, you need to get EPs from more and more users who aren't already connected to those users you get EPs from already.

5.3. Using External Trust Systems

External Trust Systems are systems which operate outside the narrow confines of a QPN. They can be webs of trust, ID verification services which might even use biometric data, or any kind of other schemes for trust and identity verification.

For External Trust Systems there are a few different options how they can interact with a QPN. First of all, it could assign an External Trust Score $T_X(i)$ to the generic user i . That Score can be added to the Native Trust Score $T_N(i)$ to arrive at the total Trust Score $T(i) = T_N(i) + T_X(i)$. It is also possible not to use the native Trust System. That is an approach which suggests itself if the External Trust System is highly reliable. In that situation there are basically three different options for the External Trust System to interact with the QPN:

- The External Trust System produces the Trust Score $T_X(i)$ as mentioned above.
- It determines the Trust Level $T_\Lambda(i)$ directly.
- Or it provides a Trust Factor $T_\Phi(i)$ without considering things like Trust Scores or Trust Levels at all.

Of course it's also totally legitimate to rely on the Native Trust System exclusively. It's also possible to set $T_\Phi \equiv 1$ in the case that no Trust System is used at all. That's an approach which should work best for small community QPNs.

6. Prestige

The reputation index Prestige is a number which is mostly determined by the Esteem a user gets from others. But Prestige also depends on a few other factors which are elaborated in this section.

6.1. Received Esteem Points

The Esteem Points another users allocates to you are your Received Eistribution Points, or REPs. The more Esteem Points you receive, the better.

6.2. Spread Factor

Prestige is supposed to be more meaningful if users Esteem a lot of other users. Otherwise, if you don't Esteem many other users, only your top favorites would get EPs and everyone else gets nothing from you. That's not really fair, because there are probably many people which deserve at least some recognition and Prestige. And that's why there is an incentive for granting significant amounts of EPs to many different users: The **Spread Factor**. For a single user with ID i the Spread Factor is denoted by $S_F(i)$

The Spread Factor is the rounded down square root of the number of users you allocate at least 20 EPs to, but it's capped at 10 and won't grow any further from that point

on. So, let n be the number of participants who get at least 20 EPs from you, then your Spread Factor $S_F(i)$ (assuming you have the user-ID i) can be found in the following table:

n	$S_F(i)(n)$
1 – 3	1
4 – 8	2
9 – 15	3
16 – 24	4
25 – 35	5
36 – 48	6
49 – 63	7
64 – 80	8
81 – 99	9
100 – 300	10

This looks quite ad-hoc, but it’s not easy to find a really good compromise between different requirements for a useful Spread Factor:

- It shouldn’t be easy to get a high Spread Factor when allocating only very few EPs to many other users.
- On the other hand it shouldn’t be too hard to get a reasonably high Spread Factor with a “reasonable” usage of the QPN.
- You shouldn’t be forced to use a specific EP allocation pattern to get a maximal Spread Factor.
- The Spread Factor should be transparent in the sense that it’s very easy to see how it arises from your usage of the system. Using statistical concepts like “variations“ might be fancier, but it would introduce unnecessary complexity, apart from giving an unreasonable advantage to people who are more into mathematics and statistics.

6.3. How the Prestige Score is calculated

Suppose you get 100 REPs from a user who has a Spread Factor of 5, you have a Spread Factor of 3, and both of you have a Trust Factor of $\frac{1}{2}$. Then you will get $\frac{1}{2} \cdot 3 \cdot \frac{1}{2} \cdot 5 \cdot 100 = 375$ Prestige from that user. So, the Factors of the users you get RDPs from, as well as your own Factors play an important role in determining the Prestige you have in the end. Your total Prestige Score is simply the sum of all Prestige points you get from all users which have allocated EPs to you. To arrive at the Prestige Score of a user the Prestige points that user gets from other users are simply added up. The Prestige Formula therefore is

$$\mathbb{P}(i) = \eta_\eta T_\Lambda(i) \cdot S_F(i) \cdot \sum_{j \neq i} T_\Lambda(j) \cdot S_F(j) \cdot E_j^i,$$

where the sum goes over all users of the QPN other than i , and η_η (default: $\eta_\eta = \eta = 6000$) is a scaling factor that makes the Prestige Scores conveniently large.

6.4. Why your Prestige doesn't depend on the Prestige Scores of your supporters directly.

At first, I considered making the Prestige you get from a user dependent on the Prestige of that user. So, users with a lot of Prestige would be able to grant others lots of Prestige. You could argue that such a system would be less democratic than the system described above, and that having a lot of Prestige doesn't automatically mean that they can esteem others "better" in some way. But there's also a more basic (less philosophical) reason why I didn't use that scheme: It faces some nasty mathematical problems.

To understand the rest of this section, you need some knowledge about a mathematical discipline called linear algebra.

Let's say we simplify the formula for the Prestige Score by leaving out the Spread Factors, Trust Factors and η_η (or setting all of them to 1), but making it dependent on the Prestige Scores of those users who allocate EPs to you. A sensible approach would be just to multiply the EPs E_j^i you receive from user j with her Prestige Score \mathbb{P}_j and sum over all users:

$$\mathbb{P}(i) = \sum_j E_j^i \cdot \mathbb{P}_j$$

For simplicity's sake the sum goes over all users, which doesn't change anything, because $E_i^i = 0$. Let's define the Prestige Vector \mathbb{P} as $(\mathbb{P}(i))_i$, the collection of all individual Prestige Scores. The equation above can be simplified using the conventions of matrix multiplication. It simply becomes

$$\mathbb{P} = E \cdot \mathbb{P}.$$

This says nothing else than that \mathbb{P} needs to be a fixed point of mapping

$$\tilde{E} : \mathbf{R}^N \rightarrow \mathbf{R}^N, v \mapsto E \cdot v,$$

where N is the number of users in the QPN. Therefore, in order to compute the Prestige Vector \mathbb{P} , we need to make sure that there exists a non-trivial solution to this fixed point problem (obviously $\mathbb{P} = 0$ is always a solution). It would also be very nice if the solution was unique. For this purpose we can rewrite the formula to

$$(E - \mathbf{1}) \cdot \mathbb{P} = 0,$$

where $\mathbf{1}$ is the identity matrix, so $\mathbf{1} \cdot \mathbb{P} = \mathbb{P}$. Therefore, the fixed points we look for are exactly the elements of the kernel of the linear mapping

$$T : \mathbf{R}^N \rightarrow \mathbf{R}^N, v \mapsto (E - \mathbf{1}) \cdot v.$$

Because kernels of linear mappings are always linear subspaces, there are an infinite number of solutions once there is a non-trivial solution which is a really awkward situation.

But let's assume we could choose a suitable non-trivial solution. When are there any non-trivial solutions at all? The kernel of T must be non-trivial, which means that the matrix $E' := E - \mathbf{1}$ must not have full rank. Or equivalently, the determinant of E'

must be exactly 0. As the product of the entries on the main diagonal of E' is 1 or -1 (due to the fact that users cannot allocate EPs to themselves), all other terms in the determinant formula

$$\det(E') = \prod_{\sigma \in \Sigma_N} \text{sign}(\sigma) \sum_{i=1}^N E_i^{\sigma(i)},$$

where Σ_N is the set of all permutations of the set $\{1, \dots, N\}$, must sum up to -1 , respectively 1, which almost certainly does not happen.

Consequently, the modified formula usually cannot hold. And even if there is a non-trivial solution there are infinitely many of them, so we would need to use some additional criteria to single out the "most sensible" one.

It may be possible to get a solution with a suitable manipulation of the Esteem Matrix, but any manipulation which would suffice for that purpose would distort the Esteem relations between users to some degree. If it always can be achieved that the rank of E' is $N - 1$, the space of solutions is 1-dimensional. In that case it would suffice to choose a solution vector with a desired norm. This produces two solutions, but one of them contains negative numbers, so it is discarded. However, if the solution space has higher dimension it would be harder to single out a meaningful solution.

So, the most sensible way to have a system where your Prestige Score depends on the Prestige Scores of other users is to apply a transformation to the matrix E' that always changes the rank of the matrix to $N - 1$ and which distorts the Esteem relations as marginally as possible. However, even then such a scheme would require more computational effort, because after each Esteem transaction the matrix would have to be transformed and then a set of linear equations would need to be solved.

Nevertheless, there is an easy possibility to use the Prestige data to some degree for the computation of the Prestige Vector. It is presented in the next subsection:

6.5. Prestige dependent Prestige computation

If you really want Prestige Scores which depend on other Prestige Scores, then you can actually get that by letting go of the idea of a direct dependence. Instead you introduce a Prestige Level $\mathbb{P}_\Lambda(i)$ for each user. This Prestige Level would be computed from the Prestige Score $\mathbb{P}(i)$ in some way. Then you can take the modified Prestige Formula

$$\mathbb{P}(i) = \eta_\eta T_\Lambda(i) \cdot S_F(i) \cdot \sum_{j \neq i} \mathbb{P}_\Lambda(j) \cdot T_\Lambda(j) \cdot S_F(j) \cdot E_j^i.$$

Similarly to the section about Trust Levels, it would make sense to let the Prestige Level depend logarithmically on the Prestige Scores, because that reduces the length and frequency of Prestige update cascades, which simply cannot be avoided in a system in which Prestige depends on Prestige. A formula that is similar to the Trust Level formula might be reasonable:

$$\mathbb{P}_\Lambda(i) = \max \left(1, \left\lfloor \log_2 \left(\frac{4\mathbb{P}(i)}{\mathbb{P}_\alpha} \right) \right\rfloor \right),$$

where the number \mathbb{P}_α is the Prestige Level Threshold (default: $\mathbb{P}_\alpha = 10000$), which determines how quickly you start to rise in Prestige Levels. Once your Prestige Score reaches that value, your Prestige Level would go from 1 to 2.

However, using this approach your Prestige Score would only depend logarithmically on the Prestige Scores of others. To get a linear dependence, either Prestige Levels would need to depend linearly on Prestige Scores, or the factor $\mathbb{P}_\Lambda(i)$ in the modified Prestige Formula would need to be replaced by $2^{\mathbb{P}_\Lambda(i)}$. In the first case, the Prestige update cascades would be more frequent and would require more computation, and in the second case there would be unreasonably large jumps (doublings) in Prestige Scores once the Prestige Level of a user increases.

Part VI.

Prestige Guided Applications (PGAs)

Having a positive reputation index like Prestige is good in itself, but such an index can also be used for further purposes. This part presents some potentially promising applications:

1. **Electronic Currency Distribution Funds (ECDFs)**, which pay out money in an electronic currency to the users or one or more QPNs in proportion to their Prestige Scores.
2. **Fluido**, the native currency that has been developed as natural complement to the Prestige index. It works similarly to an Electronic Currency Distribution Fund, but has some unique advantages.
3. **Prestige Poll Applications (PPAs)** which could be seen as generalization of conventional polls, as they allow users to set up a whole preference distribution instead of simple votes for a single option. The strength of votes may depend on one or more Prestige indexes. But a PPA can also work as a stand-alone application.

Usually, a PGA depends on the data of one or more QPNs. The only data that should be accessible from the outside on a QPN for that purpose is the the Prestige Scores, Trust Levels, and Trust Factors of each user. Everything else should stay within the QPN unless a user explicitly wants to share more data.

7. Electronic Currency Distribution Funds (ECDFs)

Of course, an ECDF needs an electronic currency. Currently, the archetype for such a currency is Bitcoin. For this kind of PGA it doesn't really matter which currency is

chosen. So, we are dealing with a generic electronic currency which we call \mathbb{X} - the unit of \mathbb{X} shall also be called \mathbb{X} .

What is an ECDF good for? It's easier to understand that once you know how it actually works. The answer to the previous question is therefore given after the technical description of the ECDF.

7.1. How does an ECDF work?

An ECDF needs to be coupled to one or more QPNs. The situation with only a single QPN is easy: The Prestige Scores of that QPN are used. When using more than one QPN, the ECDF needs a compound index called \mathbb{P}_c which is based on the Prestige indexes $\mathbb{P}_1, \dots, \mathbb{P}_n$ of n different QPNs. Each Prestige index gets a Weight w_j which determines as how important that single Prestige index is seen as compared to the others. There's also an optional scaling factor ω for which the most natural choice is simply the sum of all w_j s. The default compound Prestige index is then:

$$\mathbb{P}_c = \frac{1}{\omega} \sum_{j=1}^n w_j \mathbb{P}_j.$$

Of course, it's also possible to define a different compound Prestige index, which can be any real, non-negative function f of the individual Prestige indexes:

$$\mathbb{P}_c = f(\mathbb{P}_1, \dots, \mathbb{P}_n).$$

Another complication when using different Prestige indexes is that users have to create a separate account for the ECDF which links all their Prestige Scores (Scores which aren't linked to the ECDF account are 0 by default). When there's only one QPN the users can simply use their QPN account.

Each ECDF of course needs a fund of \mathbb{X} , which is simply called the \mathbb{X} -Fund. The question how the \mathbb{X} -Fund is filled up with \mathbb{X} is answered in the next subsection. For now, let's see how it is paid out to the users of the ECDF.

A \mathbb{X} -Fund payout occurs regularly every d days (default: $d = 30$) or whenever an ECDF admin decides it's payout time. How much is paid out every time in total? There are several different options for the payout amount α :

1. A fixed amount α_F .
2. A fixed percentage π_F of all the \mathbb{X} in the Fund.
3. A percentage π'_F of the difference between the current amount of \mathbb{X} in the Fund and the amount that was there directly before the last payout.
4. A percentage π_F^* of all the \mathbb{X} above a certain Saving Amount σ_F .

The next thing to think about is whether there should be a basic income β_i for each generic user i or not. The sum of all basic incomes of all K users is denoted by $\hat{\beta} = \sum_{i=1}^K \beta_i$. If $\alpha \leq \hat{\beta}$, then each user gets a payout of

$$\alpha_i = \beta_i \cdot \frac{\alpha}{\hat{\beta}}.$$

Otherwise the payout is:

$$\alpha_i = \beta_i + (\alpha - \hat{\beta}) \cdot \frac{\mathbb{P}_c(i)}{\sum_{k=1}^K \mathbb{P}_c(k)}.$$

The whole point of an ECDF is that payouts depend on Prestige Scores. This serves to implement a kind of meritocracy. At least it's an extension of the general QPN rewarding system.

7.2. Who pays for the Fund and what is it good for?

It could be that you have a large amount of money you want to give away for whatever reason. Then you could set up an ECDF to distribute that money. Otherwise it is assumed that there is some kind of income stream which fills the ECDF, or even several different income streams. Of what nature could such income streams be?

- Donations to the organization managing the ECDF.
- A part of the profits of a company.
- Revenues from financial products the ECDF manager(s) invest in. This point could be seen as part of the previous one however.

In each case the question which incentives exist for paying into an ECDF arises. Let's begin with possible incentives for donations to an ECDF:

- An ECDF is innovative and cool. Donation to it means that you are associated with innovation and coolness.
- Paying into an ECDF means supporting a community. That potentially brings prestige, or even \mathbb{P} .
- If you are a beneficiary of the ECDF eventually you might even get more \mathbb{X} back from the ECDF than you donate to it, if your donations cause a sufficient rise of your \mathbb{P} . Of course, if that's your only motivation to donate to the ECDF, and other think alike, it may degenerate into a Ponzi scheme.
- It's easier to donate to an ECDF than to all beneficiaries of the ECDF directly.
- If the ECDF managing organization is a non-profit donations might be tax-deductible.

- The distribution of \mathbb{X} in an ECDF depends on the democratic processes in one or more QPNs. This may be interpreted as crowdsourced meritocracy.

Why could it be useful for a company to invest into an ECDF?

- An ECDF is a convenient bonus payment system.
- It also has the potential to be the only payment system you need to implement!
- The size of the bonuses paid out through the ECDF is determined by the users of the linked QPNs. That's simply a bottom-up worker evaluation which may reduce the workload of management and human resources departments.
- ECDFs may be used to reward external collaborators or even customers, if they are part of a potentially external QPN that is associated with your company.

8. Fluido Currencies (FCAs)

The native electronic currency that fits Prestige best is the Fluido (\mathbb{F}). A Fluido Currency PGA is simply called a **Fluido Currency Application (FCA)** and it implements a Fluido Currency which is based on one or more QPNs. In the case of $K > 1$ QPNs, a compound Prestige Score $\mathbb{P}_C = \sum_{k=1}^K \mathbb{P}_k$ is used, just as in the ECDF case. For brevity's sake we call the compound Prestige Score simply \mathbb{P} .

While a FCA has some similarities to an ECDF, it works quite differently as there is no central \mathbb{F} -Fund at all! Every single \mathbb{F} is created directly in the account of a user in proportion to her \mathbb{P} . Nobody has to pay into the system. The whole basis of the FCA is \mathbb{P} .

There are two different systems for \mathbb{F} creation, a discrete and a continuous one. In the discrete system \mathbb{F} is created regularly after a fixed time interval, for example 24 hours. Whereas, in the continuous system \mathbb{F} is created continuously all the time, like a river flows into the sea continuously all the time. So, in the continuous system you would get more and more \mathbb{F} every microsecond, if you have some \mathbb{P} .

While the discrete system could be implemented it has some disadvantages compared to the continuous one, but it is used to motivate the continuous system.

8.1. The discrete system

The basic idea is simple: The amount of \mathbb{P} you have equals your annual income in \mathbb{F} . If you want to get Fluido more often, say every day, you would get $\frac{\mathbb{P}(i)}{365}$ Fluido as daily income ($\mathbb{P}(i)$ is your Prestige Score, as you are the user with the generic ID i). So far, so easy. A problem of this scheme is that it is rather inflationary, as more and more money is produced all the time. To counter this effect, you first need to understand the principle of demurrage.

8.1.1. Demurrage

Demurrage is like a tax on having money in a certain currency. A currency with demurrage is devalued regularly, which means that the amount of money is reduced systematically, and not due to the relation between the amount of money in circulation and the amount of available goods and services. One possible way to implement demurrage is to require owners of a currency to pay a fee of some percentage of the value of their banknotes. For example, the Chiemgauer is a local currency with demurrage, and to maintain the validity of a Chiemgauer banknote you have to pay 2 % of the value of that banknote every three months. Demurrage is a tool that is supposed to prevent hoarding and increase the circulation speed of money. I think that historically demurrage wasn't very popular, because it's more convenient to get the same effect with inflation.

In Prestige Fluido there is a similar mechanism. In the discrete system with unconditional demurrage, the amount of Fluido on your account is reduced by a certain fraction regularly. This fraction is the **demurrage factor** δ (default: $\delta = 5\%$) per year. It works like a negative interest rate of -5% in the default case. The reasoning behind the default value of 5% per year is that this is about the maximum of the global growth in GDP during the last 35 years. At least for a normal currency the idea is that hoarding money becomes unattractive if the demurrage fee is higher than the economic growth rate.

First your Fluido Count $\mathbb{F}(i)$ is devalued and then you get more Fluido in proportion to your Prestige Score. (It would be possible to exchange the order and first create Fluido and then devalue everything, even the freshly created Fluido, but that system would make less sense.)

To see how the demurrage works out in a FCA, let's assume for simplicity that you have a fixed Prestige Score $\mathbb{P}(i)$, that your \mathbb{F} is devalued only at the end of every year by 5% , and that it stays idle on your account all the time. In the first year there would be $\mathbb{F}_1^0(i) = \mathbb{P}(i)$ Fluido on your account, because Prestige basically is your yearly income in Fluido (at least if you don't have Fluido yet) and \mathbb{F} is created exactly at the beginning of the year. If we set $\delta = 0.05$ and define the **remanence factor** $\varrho := 1 - \delta = 0.95$ (which tells you what fraction of your Fluido will remain with you after a year if your Prestige is 0), then at the beginning of the second year, you would have $\mathbb{F}_2^0(i) = \mathbb{P}(i) + \varrho \cdot \mathbb{P}(i) = (1 + \varrho)\mathbb{P}(i)$ Fluido. In the beginning of the third year your Fluido count would be

$$\mathbb{F}_3^0(i) = \varrho(1 + \varrho)\mathbb{P}(i) + \mathbb{P}(i) = (1 + \varrho + \varrho^2)\mathbb{P}(i).$$

And in the beginning of the fourth year:

$$\mathbb{F}_4^0(i) = \varrho(1 + \varrho + \varrho^2)\mathbb{P}(i) + \mathbb{P}(i) = (1 + \varrho + \varrho^2 + \varrho^3)\mathbb{P}(i)$$

It's easy to see how this goes on. You always add another $\varrho^{Y-1}\mathbb{P}(i)$, where Y is the year at whose beginning you check your account. This is known as finite geometric series and there's a nice formula for it. At the beginning of the n -th year the amount of Fluido on your account would be

$$\boxed{\mathbb{F}_n^0(i) = \sum_{j=0}^{n-1} \varrho^j \mathbb{P}(i) = \frac{1-\varrho^n}{1-\varrho} \mathbb{P}(i) = \frac{1-\varrho^n}{\delta} \mathbb{P}(i) = 20 \cdot (1 - (0.95)^n) \mathbb{P}(i)}$$

As n goes to infinity this converges to

$$\mathbb{F}_\infty^0(i) = \frac{1}{\delta} \mathbb{P}(i) = \frac{1}{0.05} \mathbb{P}(i) = 20\mathbb{P}(i)$$

Here's a table with the values of $\mathbb{F}_n(i)$ in multiples of $\mathbb{P}(i)$:

Year	1	2	3	4	5	6	7	8	9	10
$\frac{\mathbb{F}_n^0(i)}{\mathbb{P}(i)}$	1	1.95	2.85	3.71	4.52	5.30	6.03	6.73	7.40	8.03

Year	15	20	25	30	40	50	75	100
$\frac{\mathbb{F}_n^0(i)}{\mathbb{P}(i)}$	10.73	12.83	14.45	15.71	17.43	18.46	19.57	19.88

8.1.2. Evaporation

Actually, the real discrete system for a FCA doesn't use demurrage exactly as described above, but rather uses a conditional system of demurrage which I call **Evaporation**.

You may have up to $\sigma_{\mathbb{P}}(i)$ (default: $\sigma_{\mathbb{P}}(i) = 5$ for all i) times your Prestige Score $\mathbb{P}(i)$ (as always you are the generic user with the generic ID i) in Fluido on your account without losing anything due to demurrage. Conditional demurrage is only applied to the amount of Fluido that's above $\sigma_{\mathbb{P}}(i)$ times your Prestige Score in Fluido. So, if we stay in the default case and you have $10\mathbb{P}(i)$ Fluido on your account, the first $5\mathbb{P}(i)$ of them are safe and the 5% default evaporation only applies to the other $5\mathbb{P}(i)$, so there's an effective **Annual Evaporation Rate** of only 2.5%. And if you somehow manage to have exactly $25\mathbb{P}(i)$ Fluido on your account, your effective annual evaporation rate is

$$\frac{20 \cdot 0.05}{25} = \frac{1}{25} = 0.04 = 4\%,$$

which exactly eats up $25 \cdot 0.04\mathbb{P}(i) = 1\mathbb{P}(i)$ Fluido, so no new \mathbb{F} is generated in your account, but effectively you also lose nothing. The amount $\mathbb{F}(i)$ of \mathbb{F} on your account only decreases due to evaporation if you have more than $25\mathbb{P}(i)$ \mathbb{F} . In terms of system constants the critical value at which $\mathbb{F}(i)$ remains stationary is

$$\mathbb{F}_\infty(i) = \left(\sigma_{\mathbb{P}}(i) + \frac{1}{\delta} \right) \mathbb{P}(i) = \sigma_{\mathbb{P}}(i)\mathbb{P}(i) + \mathbb{F}_\infty^0(i).$$

With evaporation, the formula for $\mathbb{F}_n(i)$ if your Prestige doesn't change and you remain inactive is slightly more complicated. We assume that $\sigma_{\mathbb{P}}(i)$ is an integer.

$$\mathbb{F}_n(i) = \begin{cases} n\mathbb{P}(i) & , \text{ if } n \leq \sigma_{\mathbb{P}}(i) \\ \sigma_{\mathbb{P}}(i)\mathbb{P}(i) + \frac{1 - e^{n - \sigma_{\mathbb{P}}(i)}}{\delta} \mathbb{P}(i) = \sigma_{\mathbb{P}}(i)\mathbb{P}(i) + \mathbb{F}_{n - \sigma_{\mathbb{P}}(i)}^0(i) & , \text{ if } n \geq \sigma_{\mathbb{P}}(i) \end{cases}$$

This time the exponent is $n - \sigma_{\mathbb{P}}(i)$, because evaporation only starts $\sigma_{\mathbb{P}}(i)$ years after the first year.

Here's the table with the values of $\mathbb{F}_n(i)$ in multiples of $\mathbb{P}(i)$ for this situation and using default values:

Year	1	2	3	4	5	6	7	8	9	10
$\frac{\mathbb{F}_n(i)}{\mathbb{P}(i)}$	1	2	3	4	5	6	6.95	7.85	8.71	9.52

Year	15	20	25	30	40	50	75	100
$\frac{\mathbb{F}_n(i)}{\mathbb{P}(i)}$	13.03	15.73	17.83	19.45	21.68	23.01	24.45	24.85

8.1.3. Adjusting the timescale

Actually using a timescale of a whole year for the discrete system is far too long, because Prestige changes much faster than that. If you want to change the timescale to a $\frac{1}{m}$ th of a year, you need to replace the demurrage factor ϱ with ϱ' , where

$$\varrho'^m = \varrho, \text{ so } \varrho' = \sqrt[m]{\varrho}.$$

Because in one m th of a year, you only get one m th of your Prestige Score $\mathbb{P}(i)$ in Fluido, the Fluido formula for the system with unconditional demurrage then becomes:

$$\mathbb{F}_n^0(i) = \sum_{j=0}^{n-1} \varrho'^j \frac{1}{m} \mathbb{P}(i) = \frac{1-\varrho'^n}{1-\varrho'} \frac{1}{m} \mathbb{P}(i),$$

where n denotes the n th m th fraction of a year.

The Fluido formula for the system with Evaporation becomes:

$$\mathbb{F}_n(i) = \begin{cases} \frac{n}{m} \mathbb{P}(i) & , \text{ if } n \leq m\sigma_{\mathbb{P}}(i) \\ \sigma_{\mathbb{P}}(i)\mathbb{P}(i) + \frac{1-\varrho'^{n-(m\sigma_{\mathbb{P}}(i))}}{1-\varrho'} \frac{1}{m} \mathbb{P}(i) & , \text{ if } n \geq m\sigma_{\mathbb{P}}(i) \\ = \sigma_{\mathbb{P}}(i)\mathbb{P}(i) + \mathbb{F}_{n-m\sigma_{\mathbb{P}}(i)}^0(i) & \end{cases}$$

8.1.4. Problems of the discrete system

Whenever the time comes that the demurrage fee is applied, there's an incentive to get rid of as many of your \mathbb{F} which exceed $\sigma_{\mathbb{P}}(i)\mathbb{P}(i)$ as possible (in order to minimize your losses from demurrage). One of the most obvious ways to do that is to sell some of your Fluido. But shortly before the devaluation happens, the demand for Fluido is low, because buyers probably know that they will soon lose a fraction of the Fluido they want to buy. This effect creates unnecessary fluctuations of the Fluido exchange rates. And it's also a general inconvenience for buyers and sellers, because the discrete model incentivizes unnecessary tactical Fluido trading, which is mostly just a waste of time and effort.

Using the discrete model is also problematic, because the same time that the demurrage fee is applied (and only at that time) the Prestige Scores become active by creating new Fluido. So, if you are unlucky and have a low Prestige Score just at that time, it doesn't help you if you have a higher Prestige Score at other times.

Minimizing those problems by changing the timescale to minutes, seconds, or even milliseconds would enforce unnecessarily frequent updates of all Fluido Counts, while still not solving the above problems completely.

The ideal solution is to update the Fluido Count just in time when a relevant Fluido transaction happens. But that's best done with a . . .

8.2. Continuous system

In this system Fluido is generated continuously. Regardless how recently you checked your Fluido Count the last time, it always changes (that is if you have a suitable amount of Prestige). This system is mathematically more complicated than the discrete case, but it makes more sense and has less problems.

8.2.1. Getting to the basic temporary Fluido formula

Remember, Fluido is first devaluated and then created by Prestige. So, what happens within a small time interval Δt (sufficiently small, so that your Prestige Score doesn't change and there are no external Fluido transactions happen), if you start with a Fluido count $\mathbb{F}(i)(t)$? First of all, let's abbreviate $\mathbb{F}(i)$ with F for convenience. For the beginning, let's consider the case with unconditional demurrage. First, that amount reduced to a certain fraction $\phi(\Delta t)$, and then a small amount of new Fluido $a\Delta t$ is created:

$$F(t + \Delta t) = \phi(\Delta t)F(t) + a\Delta t.$$

Dividing by Δt leads to

$$\frac{F(t + \Delta t) - F(t)}{\Delta t} = \frac{\phi(\Delta t) - 1}{\Delta t} F(t) + a.$$

If Δt goes to 0, this becomes

$$F'(t) = \lim_{\Delta t \rightarrow 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} = \phi'(0)F(t) + a,$$

where $\phi'(0)$ is just a simple constant, which we can abbreviate with b , so we get the ordinary differential equation

$$F'(t) = bF(t) + a.$$

The general solution of this ODE is

$$F(t) = ce^{bt} - \frac{a}{b},$$

where c is just a simple constant and e is Euler's constant ($e = 2.71828\dots$). That this function is actually a solution can be verified easily by differentiating it:

$$F'(t) = bce^{bt} = b \left(ce^{bt} - \frac{a}{b} + \frac{a}{b} \right) = b \left(F(t) + \frac{a}{b} \right) = bF(t) + a.$$

Now, what are the right constants a, b and c ? Let's do that by comparing this function with the discrete case. If t goes to infinity, $F(t)$ ought to go to

$$F_\infty := \mathbb{F}_\infty^0(i) = \frac{1}{1 - \varrho} \mathbb{P}(i)$$

But if b was positive, the exponential term e^{bt} would go to infinity, so b must be negative. Therefore, the exponential term goes to 0 and we get:

$$\lim_{t \rightarrow \infty} F(t) = -\frac{a}{b} = F_\infty.$$

If we start with F_0 Fluido at $t = 0$, we get

$$F_0 = F(0) = c - \frac{a}{b} = c + F_\infty,$$

so $c = F_0 - F_\infty$. With this, we can capture an intermediate result:

$$F(t) = ce^{bt} - \frac{a}{b} = (F_0 - F_\infty)e^{bt} + F_\infty$$

For $F_0 = F_\infty$ this means $F(t) = F_\infty$ for all $t \geq 0$.

In order to find out what e^{bt} is, we now consider the case $F_0 \neq F_\infty$. As timescale we chose a year, so after a year we would have $\varrho F_0 + \mathbb{P}(i)$ Fluido, as $\mathbb{P}(i)$ is the amount of \mathbb{F} that are generated every year:

$$\varrho F_0 + \mathbb{P}(i) = F(1) = (F_0 - F_\infty)e^b + F_\infty.$$

From the definition of F_∞ we get $\mathbb{P}(i) = (1 - \varrho)F_\infty$, so we have:

$$\varrho F_0 + (1 - \varrho)F_\infty - F_\infty = (F_0 - F_\infty)e^b,$$

which simplifies to

$$\varrho(F_0 - F_\infty) = (F_0 - F_\infty)e^b.$$

Because we have $F_0 \neq F_\infty$, we can divide by $F_0 - F_\infty$, which provides the simple equation:

$$e^b = \varrho,$$

from which we get

$$e^{bt} = (e^b)^t = \varrho^t.$$

Thus, we can write the Fluido formula in "limit form" for the case with unconditional demurrage:

$$\boxed{F(t) = (F_0 - F_\infty)\varrho^t + F_\infty.}$$

Essentially, this formula means that the difference between $F(t)$ and F_∞ decreases exponentially - no matter whether $F_0 > F_\infty$ or $F_0 < F_\infty$! For $F_0 > F_\infty$ this actually means that $F(t)$ is decreasing over time. If $F_0 = F_\infty$, then we simply get $F(t) = F_\infty$ for all $t \geq 0$, in accordance with our previous observation.

Remember that F_∞ is nothing else than $\frac{\mathbb{P}}{1-\varrho}$, so we can express the Fluido formula in dependence of \mathbb{P} :

$$F(t) = \left(F_0 - \frac{\mathbb{P}}{1-\varrho} \right) \varrho^t + \frac{\mathbb{P}}{1-\varrho} = F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \mathbb{P}.$$

Therefore, we get the Fluido formula in "Prestige form":

$$\boxed{F(t) = F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \mathbb{P}.}$$

Note that the factor $\frac{1-\varrho^t}{1-\varrho}$ is always positive, so if we replace \mathbb{P} with $\bar{\mathbb{P}} > \mathbb{P}$ we get a modified function $\bar{F}(t)$:

$$\bar{F}(t) = F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \bar{\mathbb{P}} > F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \mathbb{P} = F(t), \quad \text{for all } t \geq 0$$

This simply means that it is always a good idea to increase \mathbb{P} , no matter whether we are dealing with increasing or decreasing functions $F(t)$, respectively $\bar{F}(t)$!

Formally, $F(t)$ can be seen as continuous (even smooth) function of \mathbb{P} , but that view could be misleading, because $F(t)$ was computed assuming that \mathbb{P} remains fixed for all $t \geq 0$. In reality, \mathbb{P} will change over time, but when we compute $F(t)$, we need a constant \mathbb{P} .

8.2.2. The basic global Fluido formula

What happens when \mathbb{P} actually changes at different times? If there are times $t_0 = 0, t_1, t_2, \dots, t_n$ at which \mathbb{P} changes and we have constant Prestige scores \mathbb{P}_{I_0} for $t_0 \leq t < t_1$, \mathbb{P}_{I_1} for $t_1 \leq t < t_2$, \mathbb{P}_{I_2} for $t_2 \leq t < t_3$, and so on, we use different Fluido functions $F_{I_j}(t)$ for the different time intervalls $I_j = [t_j, t_{j+1}[$ (where $t_j \leq t < t_{j+1}$ and $j = 0, \dots, n$). We can set $t_{n+1} = \infty$ if we assume that t_j is the last time that the Prestige score ever changes. The first function is the original Fluido function up to the time t_1 :

$$F_{I_0}(t) = F(t) = F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \mathbb{P}_{I_0}, \quad \text{where } t_0 \leq t < t_1$$

At t_1 we compute the current Fluido count as

$$F_{t_1} := F_{I_0}(t_1) := \lim_{t \rightarrow t_1} F_{I_0}(t) = F_0 \varrho^{t_1} + \frac{1-\varrho^{t_1}}{1-\varrho} \mathbb{P}_{I_0}$$

and start with that value in the next function

$$F_{I_1}(t) = F_{t_1} \varrho^{t-t_1} + \frac{1-\varrho^{t-t_1}}{1-\varrho} \mathbb{P}_{I_1}, \quad \text{where } t_1 \leq t < t_2.$$

It is actually necessary to effectively "reset" the starting time for F_{I_1} to 0 by replacing t with $t - t_1$, because otherwise the function $F_{I_1}(t)$ would "assume" that we have started

with F_{t_1} and \mathbb{P}_{I_1} at $t = 0$, which would be pretty wrong!

At t_2 we compute the next Fluido count as

$$F_{t_2} := F_{I_1}(t_2) := \lim_{t \rightarrow t_2} F_{I_1}(t) = F_{t_1} \varrho^{t_2-t_1} + \frac{1 - \varrho^{t_2-t_1}}{1 - \varrho} \mathbb{P}_{I_1}$$

and start with that value in the function $F_{I_2}(t)$. The general procedure should be clear now. We set

$$F_{I_j}(t) = F_{t_j} \varrho^{t-t_j} + \frac{1 - \varrho^{t-t_j}}{1 - \varrho} \mathbb{P}_{I_j}, \text{ where } t_j \leq t < t_{j+1},$$

and

$$F_{t_j} := F_{I_{j-1}}(t_j) := \lim_{t \rightarrow t_j} F_{I_1}(t) = F_{t_{j-1}} \varrho^{t_j-t_{j-1}} + \frac{1 - \varrho^{t_j-t_{j-1}}}{1 - \varrho} \mathbb{P}_{I_{j-1}}, \text{ for } j = 1, \dots, n.$$

These functions defined on intervalls agree in their endpoints, so we get a continuous total Fluido function defined as

$$\mathbb{F}(t) = F_{I_j}(t), \text{ if } t_j \leq t < t_{j+1}.$$

This is the basic global Fluido formula. We have set up the functions $F_{I_j}(t)$ in “Prestige form”, because their Prestige dependence is obvious in that form. However, it is possible to write them in “Limit form”. In general, the “Limit form“ has the advantage that it makes understanding how the function develops over time much easier, because t only appears in one place in the ”Limit form“, whereas it appears in multiple places in the ”Prestige form“. That’s why, from now on, we will only use the ”Limit form“.

Because now we are dealing with variable Prestige scores, we are actually dealing with a piecewise constant function $\mathbb{P}(t)$ where the constants \mathbb{P}_{I_j} are the different values of $\mathbb{P}(t)$ on the different time intervals I_j . This can be written as

$$\mathbb{P}|_{I_j}(t) \equiv \mathbb{P}_{I_j},$$

where $\mathbb{P}|_{I_j}(t)$ is the restriction of the function $\mathbb{P}(t)$ onto the interval I_j .

Now that we are dealing with time dependent Prestige scores, the Fluido limit F_∞ , due to its dependence on \mathbb{P} , also becomes a time dependend function $F_\infty(t)$. Like the function $\mathbb{P}(t)$, the function $F_\infty(t)$ is piecewise constant. Therefore, we can define **temporary Fluido limits** on the different time intervals I_j as

$$F_\infty|_{I_j} := F_\infty|_{I_j}(t) = \frac{1}{1 - \varrho} \mathbb{P}|_{I_j}(t) = \frac{1}{1 - \varrho} \mathbb{P}_{I_j}.$$

By using that equation, we can write the functions $F_{I_j}(t)$ in “Limit form”:

$$F_{I_j}(t) = F_{t_j} \varrho^{t-t_j} + (1 - \varrho^{t-t_j}) F_\infty|_{I_j} = (F_{t_j} - F_\infty|_{I_j}) \varrho^{t-t_j} + F_\infty|_{I_j}, \text{ where } t_j \leq t < t_{j+1},$$

and

$$F_{t_j} = \left(F_{t_{j-1}} - F_{\infty|I_{t_{j-1}}} \right) \varrho^{t_j - t_{j-1}} + F_{\infty|I_{t_{j-1}}}.$$

Summing it up, we get the basic global Fluido formula in “Limit form”:

$\mathbb{F}(t) = (F_{t_j} - F_{\infty I_j})\varrho^{t-t_j} + F_{\infty I_j}$, if $t_j \leq t < t_{j+1}$
$F_{\infty I_j} = \frac{1}{1-\varrho}\mathbb{P}_{I_j} \equiv \frac{1}{1-\varrho}\mathbb{P}(t)$, for $t_j \leq t < t_{j+1}$,
$F_{t_j} = \left(F_{t_{j-1}} - F_{\infty I_{t_{j-1}}} \right) \varrho^{t_j - t_{j-1}} + F_{\infty I_{t_{j-1}}}$, for $j = 0, \dots, n$

8.2.3. Fluido computation with evaporation

For the system with evaporation it is convenient to split the problem into two parts: First, determine whether $F_0 < \sigma_{\mathbb{P}}$. Then, if the answer is yes, apply linear growth with \mathbb{P} as inclination. And if the answer is no, then we use the standard Fluido formula with F_{∞} replaced with

$$F_{\infty}^{\sigma} := F_{\infty} + \sigma_{\mathbb{P}},$$

because the safe amount $\sigma_{\mathbb{P}}$ is added to the Fluido amount $F(t) - \sigma_{\mathbb{P}}$ that is prone to evaporation and converges to F_{∞} .

If $F_0 < \sigma_{\mathbb{P}}$ and as long as $F(t) < \sigma_{\mathbb{P}}$ use $F(t) = F_0 + \mathbb{P}t$ Otherwise use $F(t) = (F_0 - F_{\infty}^{\sigma})\varrho^t + F_{\infty}^{\sigma}$, where $F_{\infty}^{\sigma} = F_{\infty} + \sigma_{\mathbb{P}}$

How does the last equation look in “Prestige form”?

$$F(t) = \left(F_0 - \frac{\mathbb{P}}{1-\varrho} - \sigma_{\mathbb{P}} \right) \varrho^t + \frac{\mathbb{P}}{1-\varrho} + \sigma_{\mathbb{P}} = F_0 \varrho^t + \frac{1-\varrho^t}{1-\varrho} \mathbb{P} + (1-\varrho^t) \sigma_{\mathbb{P}}$$

Unfortunately, this formula doesn’t grant us any new significant information, so let’s not bother with the “Prestige form” anymore.

8.2.4. Continuous Fluido transfer: Inflows and Outflows

It’s possible to move Fluido from one account to another instantly, but the continuous system opens up a new option: Transferring Fluido continuously! An incoming continuous Fluido transfer from another user is called **Inflow**, and an outgoing continuous Fluido transfer to another user is called **Outflow**.

Quite surprisingly, it is rather simple to implement this into the continuous system. For that purpose you need to consider that an Inflow just works like the continuous generation of Fluido by Prestige. For example, if you have 1000 \mathbb{P} and an inflow of 500 \mathbb{F} per year, for the FCA that’s the same as if you had 1500 \mathbb{P} - with the difference that the Inflow doesn’t increase your safe amount $\sigma_{\mathbb{P}}$ of Fluido which don’t evaporate. Since an Inflow behaves partially like real Prestige, the 500 \mathbb{F} you get per year count as **Flow Prestige** $\tilde{\mathbb{P}} = 500$, which is added to your **Real Prestige** $\mathbb{P} = 1000$ to get an **Effective Prestige** $\hat{\mathbb{P}} = \mathbb{P} + \tilde{\mathbb{P}} = 1500$.

Therefore, the Fluidio fomula above already respects Inflows and Outflows (which count as negative Prestige), if you replace \mathbb{P} with $\hat{\mathbb{P}}$, and F_∞^σ with

$$\hat{F}_\infty^\sigma = \frac{1}{1 - \varrho} \hat{\mathbb{P}} + \sigma_{\mathbb{P}} = \frac{1}{1 - \varrho} (\mathbb{P} + \tilde{\mathbb{P}}) + \sigma_{\mathbb{P}}.$$

Note that the safe amount $\sigma_{\mathbb{P}}$ is still computed with the Real Prestige value \mathbb{P} .

However, something new happens here: It is possible to spend more \mathbb{F} with Outflows than are generated by your Prestige and Inflows together. So, it is necessary to check whether you are still over the safe amount $\sigma_{\mathbb{P}}$. Also, once your Fluidio Count falls to 0, all your Outflows are stopped automatically, because having a negative amount of \mathbb{F} is not allowed.

Finally, the Fluidio computation formulas which also implement inflows and outflows are:

If $0 \leq F_0 < \sigma_{\mathbb{P}}$ and as long as $0 \leq F(t) < \sigma_{\mathbb{P}}$ use $F(t) = F_0 + \hat{\mathbb{P}}t$ If $F_0 \geq \sigma_{\mathbb{P}}$ and as long as $F(t) \geq \sigma_{\mathbb{P}}$ use $F(t) = (F_0 - \hat{F}_\infty^\sigma)\varrho^t + \hat{F}_\infty^\sigma.$
--

8.2.5. Event times

What makes the last formulas a bit problematic is the clause “and as long as”, because they imply that the system should keep track about the current Fluidio count $F(t)$ at all times. But that is neither practical, nor necessary. Instead, the system relies on **Event times**, which are **Transaction times** or **Threshold times**. Transaction times are the times at which the (effective) Prestige score changes, or at which a discrete Fluidio transfer happens, meaning that Fluidio are transferred all at once, rather than continuously. Let’s denote the different Transaction times with $T_0 = 0, \dots, T_n$, with n being the number of Transactions that happened so far (at $t = 0$ nothing really happens).

Threshold times are the times at which the momentary Fluidio count $F(t)$ actually reaches one of the two thresholds 0 or $\sigma_{\mathbb{P}}$. There are three cases in which that can happen:

1. Your Fluidio count $F(t)$ is below $\sigma_{\mathbb{P}}$ and your effective Prestige $\hat{\mathbb{P}}$ is positive. The Threshold time τ is defined as the time when $F(t)$ reaches $\sigma_{\mathbb{P}}$: $F(\tau) = \sigma_{\mathbb{P}}$.
2. Your Fluidio count $F(t)$ is below $\sigma_{\mathbb{P}}$ and your effective Prestige $\hat{\mathbb{P}}$ is *negative*. The Threshold time τ is defined as the time when $F(t)$ reaches 0: $F(\tau) = 0$.
3. Your Fluidio count $F(t)$ is *above* $\sigma_{\mathbb{P}}$ and your effective Prestige $\hat{\mathbb{P}}$ is *negative*. The Threshold time τ is defined as the time when $F(t)$ reaches $\sigma_{\mathbb{P}}$: $F(\tau) = \sigma_{\mathbb{P}}$.

Let’s compute τ in each of those cases, first using the temporary Prestige formulas (thus assuming that $\hat{\mathbb{P}}$ doesn’t change and no discrete Fluidio transfer happens until the time τ is reached. Keep in mind that if $F(t) < \sigma_{\mathbb{P}}$ we use the linear formula $F(t) = F_0 + \hat{\mathbb{P}}t$ and when $F(t) \geq \sigma_{\mathbb{P}}$ we use the exponential formula $F(t) = (F_0 - \hat{F}_\infty^\sigma)\varrho^t + \hat{F}_\infty^\sigma$.

1.

$$\begin{aligned} \sigma_{\mathbb{P}} = F(\tau) &= F_0 + \hat{\mathbb{P}}\tau \\ \Leftrightarrow \tau &= \frac{\sigma_{\mathbb{P}} - F_0}{\hat{\mathbb{P}}} \end{aligned}$$

Note that division by $\hat{\mathbb{P}}$ is allowed, because $\hat{\mathbb{P}} > 0$. Furthermore, the enumerator on the right is positive, because $F_0 < \sigma_{\mathbb{P}}$.

2.

$$\begin{aligned} 0 = F(\tau) &= F_0 + \hat{\mathbb{P}}\tau \\ \Leftrightarrow \tau &= -\frac{F_0}{\hat{\mathbb{P}}} \end{aligned}$$

In this case, τ is positive, because $\hat{\mathbb{P}}$ is negative!

3. First, we note that $F_0 - \hat{F}_{\infty}^{\sigma}$ is positive, because $F_0 - \sigma_{\mathbb{P}} > 0$ and $\hat{\mathbb{P}} < 0$:

$$F_0 - \hat{F}_{\infty}^{\sigma} = F_0 - \frac{1}{1 - \varrho} \hat{\mathbb{P}} - \sigma_{\mathbb{P}} > \frac{1}{1 - \varrho} \hat{\mathbb{P}} > 0$$

Second, it is the case that $\sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma}$ is positive, also because $\hat{\mathbb{P}} < 0$:

$$\sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma} = \sigma_{\mathbb{P}} - \frac{1}{1 - \varrho} \hat{\mathbb{P}} - \sigma_{\mathbb{P}} = -\frac{1}{1 - \varrho} \hat{\mathbb{P}} > 0$$

Next, we need to know that ϱ^{τ} can be written as $\exp(\ln(\varrho^{\tau}))$, where \exp is the exponential function $t \mapsto e^t$ (e being Euler's constant) and \ln is the natural logarithm (the logarithm to the base e). The functions \exp and \ln are inverses of each other. With this in mind, we can finally compute τ :

$$\begin{aligned} \sigma_{\mathbb{P}} = F(\tau) &= (F_0 - \hat{F}_{\infty}^{\sigma})\varrho^{\tau} + \hat{F}_{\infty}^{\sigma} \\ \Leftrightarrow \sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma} &= (F_0 - \hat{F}_{\infty}^{\sigma}) \exp(\ln(\varrho^{\tau})) \\ \Leftrightarrow \frac{\sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma}}{F_0 - \hat{F}_{\infty}^{\sigma}} &= \exp(\tau \ln(\varrho)) \\ \Leftrightarrow \ln\left(\frac{\sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma}}{F_0 - \hat{F}_{\infty}^{\sigma}}\right) &= \tau \ln(\varrho) \\ \Leftrightarrow \tau &= \frac{1}{\ln(\varrho)} \ln\left(\frac{\sigma_{\mathbb{P}} - \hat{F}_{\infty}^{\sigma}}{F_0 - \hat{F}_{\infty}^{\sigma}}\right) \end{aligned}$$

Here, the fraction $\frac{1}{\ln(\varrho)}$ is a negative system constant, because $\varrho < 1$. The fraction in the second \ln term is positive, but smaller than 1, because $\sigma_{\mathbb{P}} < F_0$. Therefore, the second \ln term is also negative, which makes τ a positive number.

In reality, we can't assume that $\hat{\mathbb{P}}$ doesn't change or no discrete Fluidio transfer happens before τ is actually reached, so we need to be careful. Actually dealing with the next *expected* Threshold time, which we emphasize by writing τ_e instead of τ . Also, we need to know how the functions in the global version look like. First of all, we need to make sense of how the Event times t_0, t_1, \dots work. Assume that the last Event time was t_n , the last Transaction time was T_k , and there will be a next Transaction time T_{k+1} . If $t_n + \tau_e < T_{k+1}$, then the next Event time is actually $t_{n+1} = \tau_e$. Otherwise, it is $t_{n+1} = T_{k+1}$.

To get the correct formulas for τ_e in the three different cases above, we simply need to replace a few letters:

- F_0 needs to be replaced with F_{t_n} , the amount of Fluido at the time t_n .
- \mathbb{P} needs to be replaced with $\hat{\mathbb{P}}_{I_n}$, where I_n is the time interval $[t_n, t_{n+1}[$. So, $\hat{\mathbb{P}}_{I_n}$ is the effective Prestige score during the current time interval.
- $\sigma_{\mathbb{P}}$ needs to be replaced with

$$\sigma_n := \sigma_{\mathbb{P}_{I_n}}.$$

- F_{∞} needs to be replaced with $\hat{F}_{\infty}^{\sigma}|_{I_n}$, the temporary Fluido limit during the time interval I_n :

$$\hat{F}_{\infty}^{\sigma}|_{I_n} = \frac{1}{1-\varrho} \hat{\mathbb{P}}_{I_n} + \sigma_n$$

- In each case, the time t_n needs to be added, since we usually don't start at $t_0 = 0$, but at t_n .

So, we can compute the next expected Threshold time τ_e :

$$\tau_e = \begin{cases} \frac{\sigma_n - F_{t_n}}{\hat{\mathbb{P}}_{I_n}} + t_n & , \text{ if } F_n < \sigma_n \text{ and } \hat{\mathbb{P}}_{I_n} > 0 \\ -\frac{F_{t_n}}{\hat{\mathbb{P}}_{I_n}} + t_n & , \text{ if } F_n < \sigma_n \text{ and } \hat{\mathbb{P}}_{I_n} < 0 \\ \frac{1}{\ln(\varrho)} \ln\left(\frac{\sigma_n - \hat{F}_{\infty}^{\sigma}|_{I_n}}{F_{t_n} - \hat{F}_{\infty}^{\sigma}|_{I_n}}\right) + t_n & , \text{ if } F_n \geq \sigma_n \text{ and } \hat{\mathbb{P}}_{I_n} < 0 \end{cases}$$

8.2.6. Final Fluido formulas

For $t_n \leq t < t_{n+1}$: $\mathbb{F}(t) = \begin{cases} F_{t_n} + \hat{\mathbb{P}}_{I_n} \cdot (t - t_n) & , \text{ if } F_{t_n} < \sigma_n \\ (F_{t_n} - \hat{F}_{\infty}^{\sigma} _{I_n})\varrho^{t-t_n} + \hat{F}_{\infty}^{\sigma} _{I_n} & , \text{ if } F_{t_n} \geq \sigma_n \end{cases}$
$\hat{F}_{\infty}^{\sigma} _{I_n} = \frac{1}{1-\varrho} \hat{\mathbb{P}}_{I_n} + \sigma_n$ $F_{t_n} = \begin{cases} F_{t_{n-1}} + \hat{\mathbb{P}}_{I_{n-1}} \cdot (t_n - t_{n-1}) & , \text{ if } F_{t_{n-1}} < \sigma_{n-1} \\ (F_{t_{n-1}} - \hat{F}_{\infty}^{\sigma} _{I_{n-1}})\varrho^{t_n-t_{n-1}} + \hat{F}_{\infty}^{\sigma} _{I_{n-1}} & , \text{ if } F_{t_{n-1}} \geq \sigma_{n-1} \end{cases}$

9. Prestige Polls

A **Prestige Poll Application (PPA)** is a PGA that makes it possible for the users of one or more QPNs to create and participate in polls. As in other PGAs a compound Prestige Score \mathbb{P}_C is used, and it's again abbreviated by \mathbb{P} . In a PPA users can create four different types of polls:

1. Simple Polls (SPs): A normal poll in which users can choose exactly one option. For each option the number of votes is added up to show the result.
2. Checkbox Polls (CPs): Users need to click on checkboxes to signal their agreement to certain options.

As with the SPs the number of votes is added up in the end.

3. Allocation Polls (APs): By default users have a certain number ν of Voting Points (VPs) (default: $\nu = 6000$) they can allocate to different options. Like in the with Esteem allocation, also fractions of VPs are allowed. Also, users can choose a different number $\nu(i)$ of VPs for themselves. Actually only the fraction of VPs a user gives to an option counts, so you can't cheat by increasing the number of VPs you have.

For scoring we assume that there are K options, N users and that the number of VPs the user i allocates to a option k is denoted by $v_k(i)$. The score of option k is then

$$S_k = \sum_{i=1}^N \frac{v_k(i)}{\nu(i)} \nu.$$

4. Prestige Polls (PPs): This one works like a normal poll with the difference that the vote of a user is weighted with her Prestige Score $\mathbb{P}(i)$.

The option k has the score:

$$S_k = \sum_{i=1}^N \chi_k(i) \mathbb{P}(i),$$

where $\chi_k(i)$ is 1 if user i voted for option k and 0 otherwise.

5. Allocation Prestige Polls (APPs): It's the natural combination of APs and PPs. Again, users have their number $\nu(i)$ of VPs, which get normed to ν anyway for the computation. They allocate their VPs to the different options and their votes are weighted with their Prestige Scores.

Scoring works similarly to the AP case, but is weighted with Prestige Scores (the factor ν is not in this formula in order to get reasonably small scores):

$$S_k = \sum_{i=1}^N \frac{v_k(i)}{\nu(i)} \mathbb{P}(i).$$

Part VII.

User Interface

A reasonable user interface of a QPN should have the following elements:

- The User List, which displays all users, their Prestige Scores and Trust Levels. It should also display User Circles and Splitters with their collective Prestige Scores. Should be sortable with respect to names, Prestige Scores, and Trust Levels.
- An Esteem Table for each user, which serves for allocating EPs to selected users.

- A Profile for each user.
- Settings for each user. With the possibility to change privacy default options.
- The Messaging System which contains Notifications and allows for sending IMs to selected users. Notifications can be Global Notifications by the QPN admins, or Action Notifications which users can broadcast within a Channel they can set up for themselves and other users they invite into that Channel. Users get Action Notifications if they are subscribed to the Channels in which the Notification is broadcasted.

Part VIII.

Appendix

A. Reasoning behind reputation economies

Classically, in economics there is a distinction between physical goods and intangible services. In our digital age this simple distinction is too coarse. A third category of goods needs to be considered: **Digital goods**. These digital goods can be defined as objects which can be copied and distributed for a negligible cost. This attribute distinguishes them from physical goods and services. Some examples of digital goods are:

- Software
- Patents
- Texts saved in digital form, e.g. ebooks, scanned and digitized books, static web-pages, scientific papers
- Audiobooks
- Music
- Photos, graphics and images
- Videos
- Objects in virtual worlds
- Digital data in general
- Genetic codes

Treating digital goods either as material goods or as services is inappropriate and creates serious problems.

What's the reason for these problems? In our current economic system the replication and distribution of digital goods often is severely restricted. That's rather bad from the consumers' perspective. However, distributing digital goods for free is an economic model that looks problematic for producers, as they need to invest work and possibly money to create their goods in the first place. They still can make quite a profit if they sell services and material goods for which the free digital goods act as promotion. Nevertheless, adding those complementary goods and services requires a lot of additional effort and money. This makes it very difficult to invest a lot of time into creating digital goods. It can be argued that this represents a significant loss for the vast majority of consumers.

Theoretically, a donation economy could fix that problem. Producers would create digital goods, give them away for free, and get rewarded with donations. In reality, this concept doesn't work out very well, as donating is not an overly popular form of payment. Schemes in which consumers are free to decide about the price they want to pay for a digital product on their own (with the possibility of paying a price of 0) look more promising - even if the difference is only psychological. So, that's a relatively good system, but there's still the fact that paying for something is usually irreversible, unless there's a money-back guarantee. Now let's assume you pay for a product that you thought was rather good, but disappointed you somewhat, but not enough to demand your money back. You rather feel like it would be appropriate to demand half of the price back, because the product's not terribly bad after all. Actually doing so would feel rather awkward in the current system. After all, you paid for a product freely and it's not broken - you just didn't know in advance that it would disappoint your hopes.

What's the solution here? Paying for a product what you want and also having a "get $x\%$ of your money back, where you choose freely what x is"-guarantee? Indeed, in very many cases that would be a pretty good solution (at least for the customers). Nevertheless, as nice as this sounds, it would be cumbersome to implement such a solution for all kinds of digital products. If you wanted to get really serious with it, you'd need to include a payment plug-in into every text you write, every photo you take, every video you make, and every line of code you type. Apart from being technical overkill, it would also be annoying and somewhat detrimental to the experience and enjoyment of the product itself.

If you pay for a product that's available for free, you actually want to reward the creator(s) of that product and not the product itself. So, it would be much more convenient to implement a system that enables consumers to reward creators of digital products without making the detour through the payment for a specific product. Since donations aren't the sexiest solution, something different is needed: A reputation system with direct economic consequences. And that's what Quantified Prestige is.

With the Quantified Prestige system you simply reward people for the things they do and create by increasing their Prestige score. Participants of the system who have a high Prestige score get gratified with a high *continuous income* in the digital currency Fluidio. This currency works similarly to any other currency: You can trade it, you can

accumulate it, you can buy goods and services with it (if payment with Fluido is seen as valid option).

So, Quantified Prestige is a system that couples positive reputation with money. This opens up completely new possibilities: For example you can write an article and get rewarded for it with Prestige (and indirectly also with Fluido) even though you didn't ask for payment or donations! How convenient is that? You'd only need a single Quantified Prestige account.

Even though the primary motivation for the implementation of a system like Quantified Prestige was to make it easier to reward producers of digital goods, it isn't restricted to that purpose. By its hybrid nature it's even more flexible and versatile than the current forms of money. As Fluido behaves in many respects like a normal currency, it is possible to buy regular goods and services with it, once the required infrastructure for this kind of payment is in place. But it's also possible to increase the regular income of someone you like simply by increasing her Prestige Score. You can even transfer Fluido to her *continuously* - not every year, or every month, or every day, but *every second*! And it's just as simple as transferring Fluido the regular way.

All these new options mean that Quantified Prestige represents the **next level of economy!**

B. Possible issues of Quantified Prestige

B.1. Wealthy participants could simply buy Prestige

This would work by paying other participants for allocating EPs to you. Such Prestige might be quite undeserved. But it's difficult to distinguish illegitimate from legitimate transactions. After all you would probably thank a friend for helping you out with financial problems. It's reasonable that generous persons are rewarded with high Prestige. So, where do you draw the line between these meaningful transactions and transaction made only out of greed for Prestige? Helping others for selfish reasons might not be ideal, but it still does some good after all.

If the user base really takes offense at a transaction that look like "Prestige purchasing" it can react by pointing out that issue. Then users would be free to reduce the EPs they allocate to the buyer - or even the Prestige sellers. In extreme cases it might be considered to ban participants who behave in an unethical way, so that they end up with no Prestige at all.

B.2. Celebrities and widely popular persons are favored in this system

Indeed, this is likely. However, it doesn't need to be seen as problem. If someone is popular, one could reason that he is good at satisfying a real desire, even if it's not a very noble desire, for example the desire for shallow entertainment. What are the long-term consequences of not rewarding those who fulfill such desires? It's hard to say.

Those desires would be harder to fulfill in that case. This would create an incentive to seek other ways of keeping up a relatively high level of well-being. Those alternatives could be healthier, but they could also be less healthy. As humans tend to prefer easy alternatives, I'd argue that it's more likely that they'd switch to less healthy alternatives.

B.3. General popularity doesn't mean that someone or something is actually good

Sure, but neither do other methods guarantee that someone or something is actually good. Quantified Prestige just adds an additional democratic layer to the economy. Of course, there are other meaningful metrics to determine "goodness", but implementing them into Quantified Prestige would distort the purpose of the system as tool for expressing and rewarding popular esteem. And not using such a tool would only be justified if it was shown that it does more bad than good.

B.4. This system could favor already wealthy persons

Quantified Prestige ought to favor persons who do good. If those wealthy persons which are favored do good, then that's fine. And if they don't, it is to be expected that their Prestige Scores would be relatively low.

B.5. Minorities could be discriminated against

That is, members of minorities might be less likely to get EPs from others, just because they are members of that minority. Thus, minorities would have an unfair disadvantage. Unfortunately, this is a real problem that can hardly be prevented. Nevertheless, it's possible to compensate for such biases by voluntary affirmative action, that is giving members of minorities slightly more EPs than usual, because they are members of a minority that gets discriminated against.

B.6. Multiple user accounts per person could distort Prestige Scores

Yes, that's a serious problem that could threaten the meaningfulness of Prestige and Fluido. Therefore, it is justified to use very serious identity verification methods to minimize the distortion coming from this problem. One possibility would be to send an activation code by snail mail. Users without valid activation code would only have restricted possibilities to grant Prestige to others. And there's also the whole Native Trust System to prevent multi-accounts from accumulating Prestige.

B.7. Fluido is just another "fiat currency" that's backed by nothing

Fluido is not backed by tangible goods, but receives its value through its initial democratic legitimation as expression of popular esteem, and through its relative scarcity.

Even though Fluido is produced continuously, it doesn't need to lose its value. Due to Evaporation, the average maximum amount of F per person is limited to 25 times the average Prestige Score in the default case. And the average Prestige Score per person can never exceed 600000 when using default parameters. In any case, every currency only has value because people think it has value. Having a high positive reputation is generally seen as quite valuable. So, Prestige has a clear value as representation of positive reputation. Fluido can be seen as Prestige summed up over time. So, Fluido has value as tradable accumulated manifestation of positive reputation. That's something that isn't wholly true for classical currencies, because you don't have to think highly of someone you purchase a good or get a credit from, and vice versa. Consequently, it is not true that classical currencies have a clear link to positive reputation. These advantages of Fluido ought to be enough to be seen as valuable by lots of people.

C. Where do these ideas come from?

Even before I finished high school I was somewhat intrigued about the difference between goods that can be replicated easily and those which can't. It seemed quite artificial to me that you need to pay a price for the former that resembles the price of usual mass-produced goods, which is still relatively high when considering that you only buy information which can be distributed for free. However, I didn't see a good solution to this issue.

It was only years later that I realized that a full-fledged reputation economy would be a really elegant solution. People could produce and distribute digital products in order to gain reputation, which could somehow be used to pay for all kinds of goods and services. But I didn't have a good idea how that would actually work. After I read the science fiction novels **Accelerando** by *Charles Stross* and **Down and Out in the Magic Kingdom** by *Cory Doctorow* I thought it would work roughly as pictured in those stories. Unfortunately, those novels didn't feature precise descriptions how their reputation systems actually work.

In the meanwhile, I stumbled upon alternative concepts like basic income guarantees and currencies with demurrage (the latter was an idea of the perhaps underappreciated economist *Silvio Gesell*). When I started to think about how to implement a reputation based economy, I noticed that these idea fit into my schemes quite naturally and easily - actually my first idea was to apply the basic income and the demurrage directly to the reputation score. My first ideas of a reputation economy only featured a reputation index, but no additional digital currency. Only later on I realized that including a currency that is generated by reputation would be much more practical than a reputation score alone. I called that combined system **Repo Fluido**.

There were many different versions of Repo Fluido, as I always changed the system once I noticed a major flaw. During the November of 2011 I worked a lot on developing that system, so that it finally arrived in the first hypothetically usable form. The last major change before I released the first version of the documentation was to call the system **Prestige Fluido** (due to a comment that "Repo Fluido" sounds like a laxative),

because “Prestige” is a more positive and descriptive term than “Repo”, since reputation can be positive as well as negative.

In 2012 I discussed my system with a few fellow transhumanists. Rüdiger Koch suggested that I should combine Prestige with Bitcoin rather than with Fluidio, because using Bitcoin is an already established electronic currency, and a quite promising one. Later on, I revamped the system to become more modular, so that a Prestige index can be used to guide multiple different electronic currencies, and one electronic currency can be guided by multiple Prestige indexes. Also, the need for implementing a Trust System became apparent, because Rüdiger suggested to make the whole system decentralized. The technical details of a decentralized system architecture for Quantified Prestige still have to be resolved. Calling the system **Quantified Prestige** was also a reaction to the new modular approach with the QPN as core element.

D. License

This documentation is distributed under a CC BY 4.0 license.