

WunderPools

G. Fricke, M. Löchner, S.Tschurilin

October 5, 2022, Berlin

Contents

1	Einleitung	2
2	Formalisierungen	3
3	Pool-Erzeugung	4
4	Pool-Lifetime	7
5	Pool-Liquidierung	11
6	Pool-Vertrag	13
7	WunderPools-Economics	15

1 Einleitung

Die Idee hinter den sogenannten *Wunder-Pools* ist das Bündeln von Liquidität mehrerer User/Teilnehmer bzw. eine Art 'Treuehandverwahrung' in einem gemeinsamen Pool. Die Anwendungsfälle solcher Pools können sehr zahlreich sein. Um im Folgenden nur einige Beispiele zu nennen:

- Pool für ein gemeinsames (Geburtstags-)Geschenk.
- Haushaltskasse (z.B. in einer WG oder im Urlaub).
- Vereinskasse.
- Gemeinsame Invests in (Crypto-)Assets.
- Kicktipp-Pool (der über die gesamte Saison verwahrt werden muss).
- Wetten unter Freunden (z. B. 'Schaffe ich den Marathon?').
- Ausgleichspool für Auslagen von Geld an Freunde & Bekannte (Splitwise).

Das besondere an dem in den folgenden Abschnitten genauer zu beschreibenden Modell ist sein sehr allgemein gehaltener Ansatz, mit dem sich gleichzeitig Cases umsetzen lassen, die auf den ersten Blick sehr verschieden zu sein scheinen. Genauer genommen lassen sich solche Pools mit speziellen *DAO-Strukturen* beschreiben.

Abgesehen von der den Pools zugrundeliegenden Geschäftslogik besteht der zentrale Ansatz unserer *Wunder-Pools* darin, dem User ein rundes Produkt anzubieten - und zwar gänzlich unabhängig davon, welcher der oben genannten Cases nun tatsächlich umgesetzt werden soll. An dieser Stelle möchten wir uns daher ganz explizit von dem Status quo der heute gängigen UX in der Web3-Welt abgrenzen.

Ganz grob beschrieben, streben wir in etwa folgende Geschäftslogik an:

- Ein User erstellt einen Pool (in unserer eigens designten Wunder-Pool-UI).
- Derselbe User wählt die gewünschte *Pool-Art*, ein etwaiges dazugehöriges Regelwerk und fordert andere User auf, dem Pool beizutreten. Idealerweise erfolgt die Einladung mittels Suche nach der Wunder-ID bzw. eines sprechenden Namens des einzuladenden Teilnehmers (und nicht etwa anhand seiner Ethereum-Adresse oder sonstigem).
- Die eingeladenen Teilnehmer erhalten die Einladung (in der WunderPass-App oder der Wunder-Pool-Applikation) und können entscheiden, ob sie dem Pool beitreten möchten oder nicht.

- In der Regel ist der definierte Einsatz sofort beim Beitritt des Pools zu entrichten und geht direkt in die Pool-Treasury. In einigen Cases kann der Einsatz evtl. auch zu einem späteren Zeitpunkt erfolgen oder gar ganz entfallen (z.B. beim Case *Splitwise*).
- Der eingerichtete Geldpool kann nun als Gemeinschaftsvermögen/ -konto zu u. a. folgenden Zwecken verwendet werden:
 - zum gemeinschaftlichen Investieren in (Crypto-)Assets,
 - zum Verwahren "*in Treuhand*" bei einer oder mehreren abgeschlossenen Werten (oder auch z. B. Kicktipp)
 - etc.
- Der Pool wird liquidiert und das gemeinschaftliche Geld (nach einem aus dem vorher gemeinsam festgelegten Regelwerk folgenden Verteilungsschlüssel) auf alle Pool-Mitglieder verteilt. Die Liquidierung selbst kann entweder ebenfalls durch das Regelwerk auf einen bestimmten Zeitpunkt und/oder Ereignis terminiert sein (z.B. Ende einer BuLi-Saison beim Case *Kicktipp*) oder aber durch die Teilnehmer beschlossen werden (mittels einer DAO-Abstimmung). Die Errechnung des genannten Verteilungsschlüssels möchten wir möglichst allgemein halten und übertragen diese Verantwortlichkeit einem *abstrakten Oracle*, welches es stets Case-spezifisch zu definieren (und zu implementieren) gilt.

Product-Sicht

Abschließend sei noch einmal betont, dass wir das/die aus den Wunder-Pools hervorgehende(n) Product(s) (mittelfristig) alternativlos user-friendly sehen. Ohne notwendigen Bezug zur Crypto-Szene, ohne MetaMask und ohne kryptische hexadezimale Wallet-Adressen. Stattdessen clean und simpel.

2 Formalisierungen

Zunächst einmal benötigen wir einige formale Werkzeuge und bedienen uns dafür folgender Definition:

Definition 1

Im folgenden setzen wir voraus, die Nutzung der Pools seitens der User erfordert zwingend den Besitz eines WunderPass (bzw. Wunder-ID) und betrachten von daher auch nur solche User.

$$U := \{u_1; u_2; \dots; u_n \mid u_i \text{ besitzt eine Wunder-ID}\}$$

Wir unterstellen zudem, die geforderte Existenz einer Wunder-ID gehe mit dem Besitz von unterschiedlichen Wallets bzw. anderen durch die Wunder-ID implizierten Dingen einher. So hat jeder User u_i z.B. eine Telefonnummer mit seiner Wunder-ID verknüpft (anhand derer er mittels Kontakte-Scan auf dem Smartphone als Inhaber einer Wunder-ID und damit potenzieller Pool-Teilnehmer erkannt werden kann und soll). Des weiteren kann u_i einen [NFT-Pass](#) besitzen und/oder unseren [WunderPool-Utility-Token \(WPT\)](#).

Wir schreiben unsere [NFT-Pässe](#) formal als (eine geordnete) Menge aller bisher geminteter NFT-Pässe:

$$WPN := \{wpn_1; wpn_2; \dots\} \text{ mit} \\ wpn_i := (s_i, w_i, m_i)$$

Dabei repräsentiert s_i den Status des NFT-Passes, m_i sein Muster und w_i das sich auf ihm abgebildete Weltwunder.

Den Besitz eines Pass-NFTs beschreiben wir durch die Funktion

$$\omega : U \rightarrow \mathcal{P}(WPN)$$

$$\omega(u) := \{wpn \in WPN \mid \text{User } u \text{ besitzt den Pass-NFT } wpn\}.$$

$$\left(\omega(u) = \emptyset \text{ falls der User } u \text{ keinen Pass-NFT besitzt.} \right)$$

Analog dazu definieren wir auch den Besitz am WPT eines Users - mit dem Unterschied, dass der Funktionsbereich dieser Funktion - aufgrund der Fungibilität - von einer Potenzmenge auf einen simplen numerischen Wert zusammenfällt:

$$\varphi : U \rightarrow \mathbb{Q}$$

$$\varphi(u) := \text{Balance des Users } u \text{ am ERC20-Token WPT.}$$

Die Notwendigkeit bzw. die Vorteile der vorangegangenen Annahmen und Definitionen sollten sich größtenteils aus der Lektüre der Kapitel des übergeordneten [WunderPass-Whitepapers](#) erschließen, dessen lediglich partiellen Bestandteil das hiesige *Pools-Kapitel* ausmacht.

3 Pool-Erzeugung

Die Erzeugung eines Pools findet in zwei Phasen statt: Der *Initialisierungs-Phase* und der *Joining-Phase*.

Initialisierungs-Phase

Die Initialisierungs-Phase läuft in etwa in folgenden Schritten ab:

- Ein Initiator (Admin) $u_A \in U$ erzeugt den Pool in einer dafür vorgesehenen *Pool-Applikation* (vergleichbar mit z. B. der Erstellung einer WhatsApp-Gruppe). Der Initiator u_A ist dabei selbst ein Teilnehmer des Pools. Unsere klare Absicht hierbei ist jedoch keine "gesonderten" Pool-Teilnehmer zu haben bzw. mit besonderen Rechten auszustatten. Die Unterscheidung zwischen dem Admin u_A und anderen Pool-Teilnehmern $u \in U$ ist idealerweise - sofern es denn der spezielle Case zulässt - nur für die Initialisierungs-Phase von Nöten und kann anschließend entfallen.
- Der Admin definiert das Regelwerk für den zu erstellenden Pool:
 - Art des Pools (Invest-Pool, Wette, Spende, Kicktipp, Splitwise etc.)
 - privater oder öffentlich zugänglicher Pool
 - etwaige Obergrenze an Teilnehmern
 - Einsatz (minimaler, maximaler oder exakter Einsatz pro Teilnehmer und Währung des Einsatzes)
 - Auszahlungslogik (per Abstimmung oder Adresse eines Oracle-Smart-Contracts, der abhängig seiner Contract-Logik einen Auszahlungsschlüssel bereitstellt)

Joining-Phase

Die Teilnahme-Phase besteht grob aus folgenden Schritten:

- Der Admin verliert seine Sonderstellung und wird stattdessen zum ersten Teilnehmer seines eigens initiierten Pools.
- Der (ursprüngliche) Admin lädt Teilnehmer ein, sich am kreierte Pool zu beteiligen. Die Beteiligung erfordert dabei einen WunderPass (= Wunder-ID) seitens des Teilnehmers. Idealerweise sind die Wunder-IDs mit Telefonnummern verknüpft, mittels welcher sich die einzuladenden User in den Kontakten des Admins erkennbar als potenzielle Teilnehmer wiederfinden.
- Alternativ kann der (ursprüngliche) Admin - wie auch jeder andere bereits beigetretene Teilnehmer - einen Teilnahme-Link an (weitere) potenzielle Teilnehmer verschicken.

- Jeder adressierte User erhält die Einladung inklusive aller relevanten Informationen zum beizutretenden Pool (insbesondere des benötigten Einsatz) in seiner Wunder-Pool-Applikation, und muss diese lediglich entweder bestätigen oder ablehnen (*Pull-Prinzip*). Insbesondere braucht der User für den Beitritt zum Pool kein MetaMask oder sonstiges Hilfsmittel (*Push-Prinzip*; wie aktuell bei DAOs üblich).
- Auch der Einsatz des neuen Teilnehmers muss nicht aktiv entrichtet (manuell an eine Wallet gesendet) werden, sondern wird stattdessen im Zuge des vorigen Schritts nach Bestätigung der Teilnahme am Pool automatisch eingezogen.

Wir fassen die bisher erzielten Ergebnisse etwas formaler zusammen:

Definition 2

Ein (jungfräulicher) Pool im Sinne der oben aufgezählten Eigenschaften und Anforderungen lässt sich formal schreiben als

$$Pool := (\mathcal{U}, \mathcal{R}, \mathcal{T}, \mathcal{G}) \text{ mit}$$

$\mathcal{U} = \{u_1; u_2; \dots; u_n\} \subseteq U$ die Menge der n Pool-Teilnehmer,
 \mathcal{R} das Regelset des Pools, was es gesondert zu formalisieren gilt,
 $\mathcal{T} = \{s_1; \dots; s_n\}$ mit $s_i \in \mathbb{Q}$ die Treasury des Pools und
 $\mathcal{G} = \{g_1; \dots; g_n\}$ mit $g_i \in \mathbb{N}$ die Governance des Pools.

Dabei beschreibt jedes s_i den Einsatz des Teilnehmers $u_i \in \mathcal{U}$ (s für Stake). Dieser Einsatz liegt dabei in einem vom Regelset \mathcal{R} definierten Intervall $\mathcal{I} \subseteq \mathbb{Q}$.

Damit haben wir bereits an dieser Stelle einen kleinen Teil der noch fehlenden Formalisierung von \mathcal{R} identifiziert. Bei genauer Betrachtung fehlt uns noch die Einheit der Einsätze s_i . Diese wird sehr wahrscheinlich *USDT* sein oder ein anderer Stable-Coin.

Zudem beachte man bereits an dieser Stelle, die Definition von \mathcal{T} werde im Verlaufe der Lifetime eines Pools nicht so simpel bleiben können, als lediglich aus den eingebrachten Einsätzen der Teilnehmer zu bestehen. Die Pool-Treasury bedeutet nämlich mehr als nur die Menge der initialen Stakes. Etwaige Invests aus der Treasury heraus würden nämlich ebenfalls in der Treasury landen.

Die g_i dagegen beschreiben ganz simpel die Anzahl der Governance-Tokens pro User $u_i \in \mathcal{U}$. Man kann diese auch als Gesellschaftsanteile einer GbR betrachten. Das Stammkapital dieser Gesellschaft würde sich in diesem Vergleich auf

$$\kappa := \sum_{i=1}^n g_i$$

belaufen. Der prozentuale Stimmrecht-Anteil eines Users $u_i \in \mathcal{U}$ ergäbe sich hieraus als

$$\rho_i = \frac{g_i}{\kappa}, \forall i = 1, 2, \dots, n.$$

Die zusammengetragenen Anforderungen für die Initialisierung eines WunderPools lassen sofort deutlich erkennen, **diese Pools könnten mittels DAO-ähnlicher Strukturen implementiert werden**. Dies erscheint insofern umso logischer und konsequent, als dass wir bereits erkannt haben, die Pools stellen gesellschaftsrechtlich GbRs dar - also Gesellschaften und/oder Organisationen. Diese Erkenntnis wollen wir noch einmal als eine formale Annahme formulieren:

Annahme 1: Ein WunderPool bildet de facto eine GbR ab

Sei $\mathcal{P} := (\mathcal{U}, \mathcal{R}, \mathcal{T}, \mathcal{G})$ ein WunderPool wie in Definition 2 beschrieben. Wir ziehen die Analogie zu einer **Gesellschaft** bürgerlichen Rechts:

- Die Menge \mathcal{U} der Pool-Teilnehmer repräsentiert den **Gesellschafterkreis der Gesellschaft**.
- \mathcal{G} beschreibt den **Cap-Table der Gesellschaft**.
- Das Pool-Regelwerk \mathcal{R} ist nichts anderes als der **Gesellschaftervertrag zur Gesellschaft**.
- Die Pool-Treasury \mathcal{T} modelliert zuletzt das **Gesellschaftskonto und/oder -depot der Gesellschaft**.

4 Pool-Lifetime

Eine (allgemeine) funktionale Beschreibung derjenigen WunderPool-Funktionalität, die der Überschrift der gegenständigen Sektion gerecht wird, ist insofern sehr schwierig, als dass sich diese deutlich schwerer auf unterschiedliche Pool-Cases verallgemeinern lässt. Wie anfangs in dem Kapitel **Einleitung** ist die möglichste Verallgemeinerung aller Cases oberste Prämisse gewesen. Hier müssen wir versuchen zu verallgemeinern, was zu verallgemeinert geht, und den Rest eben Case-spezifisch lösen.

Wir schauen zurück auf die anfangs in Kapitel **Einleitung** hervorgehobenen Anwendungsfälle der WunderPools. Und zwar jetzt mit explizitem Blick auf ihre *Lifetime*:

- **Social Investing:** Dies ist mit der klarste Case für eine relevante Lifetime eines Pools. Während der Lifetime werden mögliche Invests vorgeschlagen, zur Abstimmung gestellt und im Erfolgsfall abgewickelt. Die Möglichkeiten zur Erweiterung von Investmöglichkeiten (Staking, Lending, Liquidity-Providing, Yield Farming, Aktien, ETFs etc.) scheinen schier unendlich. In diesem Case unterliegt **die Dauer der Lifetime auch keinerlei natürlicher Grenzen** - diese Art von Pool kann theoretisch ewig existieren.
- **Geschenk-Pool:** In diesem Case besteht die Daseinsberechtigung des Pools im Grunde lediglich darin, bequem und einfach Geld einzusammeln und evtl. bis zum Kauf des Geschenks "*in Treuhand*" zu verwahren. Sind alle gewünschten Teilnehmer beigetreten (und damit gleichbedeutend deren Beitrag zum Geschenk entrichtet), hat der Pool de facto bereits seinen Zweck erfüllt. Man kann zwar argumentieren, man könne die Auswahl des Geschenks mit DAO-Mitteln zur Abstimmung stellen, dies bliebe jedoch an den Haaren herbeigezogen, solange das Geschenk kein auf der Blockchain erwerbbares Asset darstellt. **Die Dauer der Lifetime der Pools in diesem Case sind also klar begrenzt:** Spätestens bis zu dem Moment des Kaufs des Geschenks.
- **Kicktipp-Pool:** Dies ist der Bilderbuch-Case für den Pool im Sinne der Treuhand-Verwahrung (eines Spieleinsatzes) über einen längeren Zeitraum. Hier wird eingezahlt, über einen gewissen Zeitraum (außerhalb des Pools) gespielt und am Ende - je nach Ergebnis - wieder ausgezahlt. Das Geld wird vom Pool also lediglich verwahrt und umverteilt. In der sogenannten *Lifetime* des Pools passiert faktisch gar nichts. Man könnte sich sicherlich kreative Möglichkeiten zur Interaktion mit dem Pool überlegen (wie z.B. Abstimmungen über etwaige Regeländerungen oder über das Nachtragen von verspätet abgegebenen Tipps), dies beträfe aber nie die relevante Kernfunktionalität des Pools innerhalb dieses Cases. Die defacto '*leere Lifetime*' des Pools endet in diesem Case mit Ablauf der Spielzeit, für die die Kicktipp-Runde eingerichtet wurde. Ihre **Dauer ist also begrenzt**.
- **Wetten:** Dieser Case verhält sich sehr analog zum *Kicktipp-Case*. Dazu muss jedoch klargestellt sein, dass wir den Case als eine einzige Wette (zwischen zwei oder mehr Leuten) verstehen, bei der der Pool der Treuhand-Verwahrung dient, und nicht etwa eine "Wett-Gruppe", wo immer mal wieder neue Wetten vorgeschlagen und umgesetzt werden. Der Pool dieses Cases bildet also eine einzige Wette ab und seine **Lifetime endet in dem Moment, wo das Ergebnis der Wette feststeht**.
- **Splitwise:** Dies ist der außergewöhnlichste aller Cases. Hier existieren de facto weder eine echte Treasury noch eine Lifetime. Für Splitwise wird erst die Umverteilung interessant, wobei hier genau genommen der Betrag von 0 auf die Teilnehmer

umverteilt wird. Da hier aber - im Gegensatz zu allen obigen Cases - auch negative Withdraws zulässig sind (also genau genommen eine Einzahlung von denjenigen Teilnehmern, die anderen Teilnehmern etwas schulden), klingt die Umverteilung des Betrags 0 plötzlich doch nicht mehr so abwegig. Die 0 signalisiert nur die Forderung, die verteilten Beträge (Schulden und Auslagen mit entsprechendem Vorzeichen) müssten sich am Ende auf 0 summieren. Da der Pool in diesem Case faktisch gar keine Lifetime besitzt, ist **die Dauer der Lifetime konsequenterweise begrenzt**.

Zusammenfassend halten wir fest, die Dauer der Pool-Lifetime sei nur für den *Social-Investing-Case* theoretisch unbegrenzt. Bei allen anderen Cases wird der Pool nach einer bestimmten Zeit oder bei Eintreten eines bestimmten Ereignisses obsolet und muss/sollte anschließend aufgelöst werden. Und auch hinsichtlich relevanter Funktionalität während der zugehörigen *Lifetime* scheint der *Social-Investing-Case* ebenfalls der einzig interessante zu sein.

Eine Verallgemeinerung erscheint also - zumindest für die zuletzt genannten vier Cases - evtl. doch im Rahmen des Möglichen.

Zu guter Letzt sollte der Umstand nicht unerwähnt bleiben, etwaiges Austreten bestehender Pool-Teilnehmer bzw. das Eintreten neuer stellen keine irrelevanten Szenarien dar, die sich ebenfalls während der vermeintlichen *Pool-Lifetime* abspielen würden.

Abschließend formalisieren wir erneut die erarbeiteten Gedanken - und zwar mit besonderem Blick auf Definition 2 und Annahme 1:

Conclusion 1: Bestehen und Geschäftstätigkeit eines WunderPools als GbR (Lifetime)

Sei $\mathcal{P} := (\mathcal{U}, \mathcal{R}, \mathcal{T}, \mathcal{G})$ ein WunderPool wie in Definition 2 und das Verständnis davon stark an Annahme 1 angelehnt.

Wir möchten gerne auch die Lifetime eines WunderPools in die GbR-Analogie überführen und unterscheiden dabei zwischen **der Geschäftstätigkeit / Unternehmensgegenstand der Gesellschaft selbst** auf der einen Seite und **den internen Gesellschaftsstrukturen** auf der anderen.

Geschäftstätigkeit:

Rein abstrakt betrachtet, versuchen hierbei alle Gesellschafter $u \in \mathcal{U}$ - legitimiert durch deren Anteile aus \mathcal{G} und restriktiert mittels Gesellschaftervertrags \mathcal{R} - durch strategisches Verhalten das gemeinschaftliche Gesellschaftsvermögen \mathcal{T} zu optimieren bzw. dieses zumindest optimal für ein zweckgebundenes gemeinsames Ziel einzusetzen.

Die internen Gesellschafter- und Gesellschaftstrukturen - repräsentiert durch die

Größen \mathcal{U} , \mathcal{G} und \mathcal{R} - bleiben in diesem Kontext in aller Regel unberührt.

Interne Strukturen:

In diesem speziellen Kontext sind gegenteilig zum ersten genau die konträren Größen \mathcal{U} , \mathcal{G} und \mathcal{R} adressiert. Hierbei sind die Gesellschafter $u \in \mathcal{U}$ - erneut legitimiert durch deren Anteile aus \mathcal{G} und wieder restriktiert mittels Gesellschaftervertrags \mathcal{R} - dazu befähigt und ggf. daran interessiert, Veränderungen an \mathcal{U} , \mathcal{G} und \mathcal{R} zu erzwingen. Beispielfhaft sind dabei folgende Cases denkbar:

- **Kapitalerhöhung:** Hierbei würden die Gesellschafter anhand ihrer Stimmrechte aus \mathcal{G} (und etwaiger Zusatzvereinbarungen aus \mathcal{R}) über die Aufnahme eines neuen Gesellschafters in die Gesellschaft abstimmen. Eine Zustimmung hätte mindestens eine Veränderung der Größen \mathcal{U} und \mathcal{G} zur Folge - und zwar in beiden Fällen eine Vergrößerung. In aller Regel würde bei einer Kapitalerhöhung ebenso die Gesellschaftskasse \mathcal{T} wachsen. Und in manchen Fällen wäre ebenso eine Veränderung des Gesellschaftervertrags \mathcal{R} zu erwarten.
- **Auszahlen eines bestehenden Gesellschafters:** Von der Logik her ein ähnlicher Case wie der erste, nur dass hierbei ein oder mehrere Gesellschafter die Gesellschaft verlassen, die Menge \mathcal{U} also schrumpft. Anders als der erste Case ist der gegenständliche jedoch differenzierter hinsichtlich des **Wie** zu betrachten. Während bei der Kapitalerhöhung einfach neue Anteile kreiert werden, die schlichtweg von neuen Gesellschaftern übernommen werden, gibt es im aktuellen Fall mehrere gängige Varianten:
 - Die Anteile des/der ausscheidenden Gesellschafter(s) werden von der Gesellschaft selbst übernommen und anschließend vernichtet. Hierbei verringert sich also $\sum_{g \in \mathcal{G}} g$ konsequenterweise. In diesem Fall muss in aller Regel die Gemeinschaftskasse \mathcal{T} zur Auszahlung herangezogen werden. Dies kann insofern etwas tricky werden, als dass die Treasury \mathcal{T} nicht zwingend ausschließlich aus Fiat-Vermögen bestehen muss, und stattdessen ggf. Assets liquidiert werden müssten.
 - Die Anteile des/der ausscheidenden Gesellschafter(s) werden von den verbleibenden Gesellschaftern übernommen. Hierbei bliebe $\sum_{g \in \mathcal{G}} g$ unverändert. Die finanzielle Abwicklung würde de facto *außerhalb der Gesellschaft* stattfinden. Insbesondere bliebe die Treasury \mathcal{T} von der Transaktion unberührt, was die Implementierungslogik vehement vereinfacht.
 - Der technisch simplest Case wäre sicherlich der Verkauf der Anteile des/der ausscheidenden Gesellschafter(s) am Sekundärmarkt - die ja als ERC20-Tokens einfach handelbar wären. Auch hierbei bliebe sowie $\sum_{g \in \mathcal{G}} g$ als auch \mathcal{T} als auch wahrscheinlich \mathcal{R} unverändert. Genau genommen ist dieses Sub-Szenario ein Spezialfall des nächsten Case.

- **Anteilsverkauf / -übertragung:** In diesem Szenario würde ein Gesellschafter $u_i \in \mathcal{U}$ einen Teil oder alle seiner $g_i \in \mathcal{G}$ Anteile am Sekundärmarkt verkaufen. Der Gesellschaftervertrag \mathcal{R} könnte ihn zwar theoretisch daran hindern bzw. die anderen Gesellschafter in eine solche Entscheidung einbeziehen. Dies würde jedoch mit erheblicher Komplexität einhergehen, weshalb wir *per default* erst einmal davon ausgehen wollen, die Gesellschaftsanteile sind als ERC20-Governance-Tokens frei handelbar und unterliegen keinen Einschränkungen. In diesem Fall wäre die innere Gesellschaftsstruktur einem ausschließlich äußeren Einfluss unterstellt, dem sie nicht Herr wäre. (Unkontrolliert) betroffen wären die Größen \mathcal{U} und \mathcal{G} .
- **Änderung des Gesellschaftervertrags:** Hierbei würden die Gesellschafter anhand ihrer Stimmrechte aus \mathcal{G} (und etwaiger Zusatzvereinbarungen aus \mathcal{R}) über gewisse Änderungen an \mathcal{R} abstimmen. Welche Änderungen hierbei möglich wären, könnte wiederum mittels des vor der Änderung geltenden Regelsets \mathcal{R} gemäßregelt sein. Aufgrund dieser Rekursivität müsste \mathcal{R} wahrscheinlich einige unveränderliche Elemente beinhalten. Zu der definitiv komplexesten Größe unserer WunderPools \mathcal{R} siehe auch das Kapitel **Pool-Vertrag**.

5 Pool-Liquidierung

Für eine etwaige Pool-Liquidierung stellen sich exakt zwei Fragen: ”**Wann** wird liquidiert?” und ”**Wie** wird liquidiert?” Das **Wann** ist hierbei schnell geklärt. Es gibt grob folgende drei Möglichkeiten, von denen **genau eine** durch das in Definition 2 definierte Regelset \mathcal{R} zu benennen ist:

- \mathcal{R} legt einen exakten Zeitpunkt fest, zu dem der Pool liquidiert werden soll.
- \mathcal{R} definiert ein bestimmtes Ereignis, bei deren Eintreten der Pool liquidiert werden soll.
- \mathcal{R} regelt, dass die Pool-Liquidierung per (DAO-)Abstimmung beschlossen werden muss.

Bullet 2 klingt hier leider noch nicht ausreichend abstrakt. Daher abstrahieren wir die genannten drei Forderungen in einer einzigen:

Definition 3: Liquidierungsentscheidungs-Oracle

Das in Definition 2 definierte Regelset \mathcal{R} definiert ein Oracle, welches zu jedem

Zeitpunkt die Frage beantworten kann, ob der Pool zum jetzigen Zeitpunkt liquidiert werden soll oder nicht.

Dieses Oracle kann beliebig einfach gestrickt sein (z.B. im Falle des obigen Bullet 1 einfach anhand " $SYSDATE \leq T_{END}$ " über das Fortbestehen des Pools entscheidet) oder aber auch beliebig komplex. Dies braucht uns aber an dieser Stelle nicht weiter interessieren.

Und da die Abstraktion mittels Oracle so bequem scheint, tun wir das Gleiche ebenfalls für das oben genannte **Wie**:

Definition 4: Auszahlungsschlüssel-Oracle

Seien $\mathcal{P} = (\mathcal{U}, \mathcal{R}, \mathcal{T}, \mathcal{G})$ der Pool und $\mathcal{U} = \{u_1; u_2; \dots; u_n\}$ die Menge seiner n Teilnehmer wie in Definition 2 beschrieben und $v_{\mathcal{T}}$ der sich zum Liquidierungszeitpunkt in der Pool-Treasury \mathcal{T} befindende Value.

Falls der Pool lediglich als Treuhand-Verwahrung diene (also über die Zeit keine Veränderung der Treasury stattfand) ergibt sich $v_{\mathcal{T}}$ als

$$v_{\mathcal{T}} = \sum_{i=1}^n s_i \text{ mit } s_i \text{ wie in Definition 2}$$

Wir definieren einen Auszahlungsvektor als

$$\varphi_{\mathcal{P}} = [\varphi_1, \varphi_1, \dots, \varphi_n] \text{ mit } \sum_{i=1}^n \varphi_i = v_{\mathcal{T}}$$

Die φ_i beschreiben also die absoluten Anteile der Teilnehmers u_i an der Pool-Treasury \mathcal{T} . Und $\frac{\varphi_i}{v_{\mathcal{T}}}$ dann logischerweise die prozentualen.

Definiert/Konkretisiert werden müssten die *Liquidierungsentscheidung*- und *Auszahlungsschlüssel-Oracle* in dem Pool-Regelset \mathcal{R}

Im Folgenden einige Beispiele für denkbare *Auszahlungsschlüssel-Oracle*. Zur Vereinfachung nehmen wir dazu an, die Pool-Treasury \mathcal{T} enthielte ausschließlich Funds derselben Fiat-Währung, weshalb der oben genannte Value $v_{\mathcal{T}}$ gänzlich intuitiv ersichtlich sei.

Beispiel 1: Pro-Rata-Auszahlung

Dies stellt eigentlich den intuitivsten aller denkbaren Auszahlungsschlüssel dar. Jeder Pool-Teilnehmer bekommt genau den prozentualen Anteil an $v_{\mathcal{T}}$ ausgezahlt, der seinem Anteil an der Pool-Governance \mathcal{G} entspricht.

Der obige Auszahlungsvektor würde sich in diesem Fall ganz simple als

$$\varphi_i = v_{\mathcal{T}} \cdot \frac{g_i}{\sum_{j=1}^n g_j}$$

ergeben. Wobei $\mathcal{G} = \{g_1; g_2; \dots; g_n\}$ wäre.

Beispiel 2: Auszahlung nach abgeschlossener Kicktipp-Tipprunde

Angenommen so ein WunderPool würde für die Verwahrung der Spieleinsätze einer Kicktipp-Tipprunde verwendet werden. Nach abgeschlossene Spielzeit sollte der gesamte Wettpool an die besten Tipper ausgezahlt werden. Wer die besten Tipper waren und wie das Geld konkret unter diesen verteilt wird, mocken wir hinter einem *Kicktipp-Oracle*, der diese Daten wie auch immer (von extern) beschafft. Bei einem Wettpool von insgesamt 1000 Euro und acht Mitspielern könnte das vom Oracle gelieferte Ergebnis z. B. wie folgt aussehen:

$$\varphi_{\mathcal{P}} = [0, 100, 0, 0, 700, 0, 200, 0]$$

Beispiel 3: Splitwise-Abrechnung

Nutzt man die WunderPools-Abstahierung, um eine Splitwise-Abrechnung unter Wunder-Usern innerhalb derselben Splitwise-Gruppe abzubilden, könnte die Splitwise-API dazu genutzt werden, ein *Splitwise-Oracle* zu implementieren. Dieses könnte bei acht Spitwise-Usern z. B. folgendes Ergebnis liefern:

$$\varphi_{\mathcal{P}} = [80, 50, -50, -30, 0, -40, 20, -30]$$

Man beachte, dass bei diesem Case die φ_i auch negativ sein können und sich in Summe auf 0 addieren: $\sum_{i=1}^n \varphi_i = 0$.

Beispiel 4: Random Gambling

Keine besonders sinnvolle aber theoretisch dennoch denkbare Auszahlungsstrategie wäre eine völlig zufällige Verteilung von $v_{\mathcal{T}}$ unter allen Pool-Teilnehmern. Hierbei wäre unser Oracle ein einfacher Zufallsgenerator, der uns eine Zufallsverteilung $P = \{P_1; P_2; \dots; P_n\}$ für unsere n Pool-Teilnehmer mit $\sum_{i=1}^n P_i = 1$ liefert.

Der Auszahlungsvektor würde sich in diesem Fall ganz simple als

$$\varphi_i = v_{\mathcal{T}} \cdot P_i$$

ergeben.

6 Pool-Vertrag

Die in Definition 2 erstmals eingeführte Größe \mathcal{R} begegnete uns in den letzten Kapiteln unzählige Male. Wir nannten sie *Regelset*, *Pool-* bzw. *DAO-Vereinbarung* oder in Anlehnung an die in Annahme 1 formulierte Analogie auch ***Gesellschaftervereinbarung*** oder ***-vertrag***.

So richtig formalisiert haben wir \mathcal{R} jedoch bisher nirgends. Stattdessen wurde benannt, was \mathcal{R} zu regeln hat oder was es enthalten kann oder muss. Die fehlende Formalisierung ist kein Versäumnis sonder schlichtweg kaum möglich, was aber auch gleichzeitig wenig überraschen mag, wenn man bedenkt, wie beliebig komplex, heterogen und zahlenmäßig unbeschränkt die Regelungen in einem *Gesellschaftervertrag* nur sein können. Man kann darin prinzipiell alles regeln oder fast gar nichts.

Um erneut bei der Analogie aus in Annahme 1 zu bleiben, ist \mathcal{R} nicht anderes als ein **Shareholder's Agreement**, woran man sich beim Design von \mathcal{R} ganz gut inspirieren lassen kann.

Es folgt eine Sammlung von Dingen, die durch \mathcal{R} geregelt werden müssen oder können und die wir im Verlaufe der vorangehenden Abschnitte ohnehin bereits als Bestandteile von \mathcal{R} erkannt haben, bzw. die darüber hinaus zu bedenken sind oder zu bedenken sein könnten. Die Liste hat dabei weder einen Anspruch auf Vollständigkeit noch auf wasserdichte formale Exaktheit. Sie ist eher als gedankliche Anregung für die Struktur und die Anforderungen an \mathcal{R} zu verstehen.

- Pool-Art (*Geschäftsgegenstand der Gesellschaft*)
- Vorgaben zum Teilnehmer-Kreis \mathcal{U} (*Gesellschafterkreis*)
 - öffentlicher vs. privater Pool
 - min/max Teilnehmer
 - etwaige Teilnahmebedingungen (z. B. Exklusivität durch NFT-Besitz)
- Vorgaben zur Pool-Treasury / Teilnahmeeinsatz \mathcal{T} :
 - Währung (zB *USDT*)
 - Intervall \mathcal{I} für Einsatz $s_i \in \mathcal{I}$ (vergleiche Definition 2)
- Vorgaben zur Governance \mathcal{G} (Anteile, Stimmen, Mehrheiten):
 - initiale Vergabe von Shares
 - Mehrheiten & (Sperr)minoritäten
 - Voting-Regeln
- Regelungen zum nachträglichen Beitreten des Pools (*Kapitalerhöhung; neue Gesellschafter*)
- Regelungen zum vorzeitigen Verlassen des Pools

- Regelungen zu Anteilsverkäufen / -abtretungen
- Regelungen zur Liquidierung des Pools
 - Definition der *Liquidierungsentscheidung-Oracle* (vergleiche Definition 3)
 - Definition der *Auszahlungsschlüssel-Oracle* (vergleiche Definition 4)
- Regelungen zu nachträglichen Änderungen an \mathcal{R} selbst.
- etc.

7 WunderPools-Economics

Aufgrund des beachtlichen inhaltlichen Umfangs haben wir uns dazu entschlossen, dem *Economics-Part* ein eigenes [Teil-Paper](#) zu spendieren.