Solidpattern

Endbericht | Call 12 | Projekt ID 2232

Inhalt

1 Einleitung / Ausgangslage

Warum ein Pattern für Solidity?

Das Interesse an dezentralisierten Anwendungen nimmt stetig zu. Um eine Business Anwendung auf der Ethereum Blockchain umzusetzen, bedarf es einer gewissen Vorplanung. Eine fehlerhafte, Struktur kann die Anwendung schnell unbrauchbar machen. Anpassungen sind nicht möglich! Der Contract muss erneut geschrieben und im Netzwerk „deployed“ werden. Ein Pattern ähnlich dem bekannten MVC Prinzip würde es Entwicklern ermöglichen, schneller und effizienter neue Anwendungen umzusetzen.

Risiken bei der Umsetzung:

Solidity ist eine junge Programmiersprache und kommt lediglich bei Ethereum zum Einsatz. Dadurch befindet sich die Sprache selbst noch in der Entwicklungsphase. Die Umsetzung von dezentralen Anwendung bringt viele neue Probleme mit sich. Ein Problem stellt die Einschränkung bei der Speicherung von Daten dar. Ein weiteres Problem ist die Unveränderbarkeit des Programmcodes dar - was bei einem "Trustless" System einerseits gegeben sein soll, aber andererseits nicht zwingend den gesamten Programmcode betreffen muss. Ein einfaches Umwälzen vom existierenden Pattern wird aufgrund der neuen Probleme voraussichtlich nicht möglich sein.

State of the Art:

Die Programmiersprache Solidity erlaubt Vererbung, zusätzlich wurde ab der Version 0.4 auch die Möglichkeit der Erstellung von Libraries integriert. Es gibt bis dato keine bekannten oder angesehenen Vorgehensweisen für große modulare Blockchain Anwendungen.

Projektziele:

Dieses Pattern als eine Art Anwendungsgrundlage soll Entwickler dabei unterstützen einfach und schnell auf der Ethereum Blockchain mit einer eigenen Anwendung loszulegen. Im besten Fall entwickelt sich dieses Pattern zu einem bekannten Startpunkt und findet weitere Unterstützer um das Projekt stetig weiter zu entwickeln.

2 Projektbeschreibung

Beschreibung der Projektziele / Zielgruppe und inhaltlicher Überblick über das Projektergebnis (max. 5 Seiten)

Dieses Pattern soll als eine Art Anwendungsgrundlage soll Entwickler dabei unterstützen einfach und schnell auf der Ethereum Blockchain mit einer eigenen Anwendung loszulegen. Im besten Fall entwickelt sich dieses Pattern zu einem bekannten Startpunkt und findet weitere Unterstützer um das Projekt stetig weiter zu entwickeln.

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - *Recherche und eine mögliche Lösung festlegen*

Durch die Recherche und einem Vergleich zwischen zentralen und dezentralen Anforderungen soll sich daraus eine passende Lösung für skalierbare Business Anwendungen ergeben.

Zentral vs. Dezentral

Welche Patterns kommen bei zentralen Anwendungen zum Einsatz? Welche Patterns sind besonders anwenderfreundlich und beliebt? Können diese Patterns auf einer dezentralen Anwendung angewendet werden?

Klassifikation Entwurfsmuster:

Entwurfsmuster lassen sich grundsätzlich durch deren Zweck und deren Wirkungsbereich unterscheiden.

- Erzeugungsmuster (Erzeugen Objekte)
- Strukturmuster (Vereinfachung der Struktur zwischen Klassen)
- Verhaltensmuster (Zusammenspiel und Nachrichtenaustausch zwischen Objekten)

Eine dezentrale Sicht auf MVC:

Heutzutage wird das MVC Prinzip meist mit der Struktur des Frontend-Codes in Verbindung gebracht. Der Grund dafür ist die frühere strikte Trennung zwischen Frondend und Backend. Mittlerweile ist Javascript ein fester Bestandteil jeder modernen Web Anwendung und stellt dadurch oftmals schon den Teil des Controllers dar. Also die View als HTML, der Controller und das Model als Javascript komplett auf der Clientseite dazu findet man je nach Backend erneut Controller und Model als Schnittstelle auf der Serverseite.

Solution

Nachfolgend werden die Vorgehensweise und die daraus resultierenden Ergebnisse dargestellt.

Was zeichnet eine Ökosystem Anwendung aus?
Unternehmen bewegen sich immer mehr in die Automatisierung von Prozessen, dabei tritt oftmals das Problem auf, dass diese Anwendungen von einzelnen Teilnehmern betrieben werden und nicht wie gewünscht vom gesamten Netzwerk bzw. Ökosystem von Teilnehmern. Ökosystemanwendungen versuchen also die Prozesse auf der Ebene der Wertschöpfungskette zu automatisieren und bieten damit einen Rahmen für den Aufbau komplizierter und vernetzter Netzwerke um die Kosten zu senken und neue Möglichkeiten der Wertschöpfung zu eröffnen.

Blockchain Generation
Anhand der neuen dezentralen Anwendungsmöglichkeiten lassen sich die Generationen von Ökosystemanwendungen, wie folgt einteilen:

1. Generation: Einfluss auf die Wirtschaft durch Datenskalierung und Synchronisation von papierbasierten internen Prozessen großer Unternehmen Mithilfe von Datenzentren.
2. Generation: Einfluss auf die Wirtschaft durch Datenskalierung und Synchronisation von papierbasierten internen Prozessen kleinerer Unternehmen Mithilfe von Cloud Anbietern.
3. **Generation: Verwendung von Blockchains und Smart Contracts um abgestimmte Prozesse auf Ökosystemebene zu erzeugen.**

Anforderungsanalyse
Um ein Pattern für eine dezentrale Anwendung zu erstellen, müssen zuerst die Probleme und Herausforderungen definiert werden. Weil eine Blockchain grundsätzlich keine Änderungen zulässt, muss eine Möglichkeit geschaffen werden, nur Teile einer Anwendung zu erneuern und einer bestehenden Anwendung neu zuzuweisen.

Um das zu ermöglichen, muss eine dezentrale Anwendung weitreichend in seine unterschiedlichen Bestandteile aufgeteilt werden.

- Daten und Struktur der Daten
- Logische Operationen
- Controller zwischen Daten und Logik
- Libraries für wiederkehrende Anforderungen

Zusätzlich ist es notwendig eine Art Routing in die Struktur einer dezentralen Anwendung zu bringen. Das Pattern wird deshalb um einen Proxy und Gateway Contract erweitert und ermöglich dadurch eine erhöhte Flexibilität. Dadurch ergibt sich die nachfolgend dargelegte Struktur:

- Proxy Gateway Contract
- Logic Contracts
- Service Contracts
- Controller Contracts

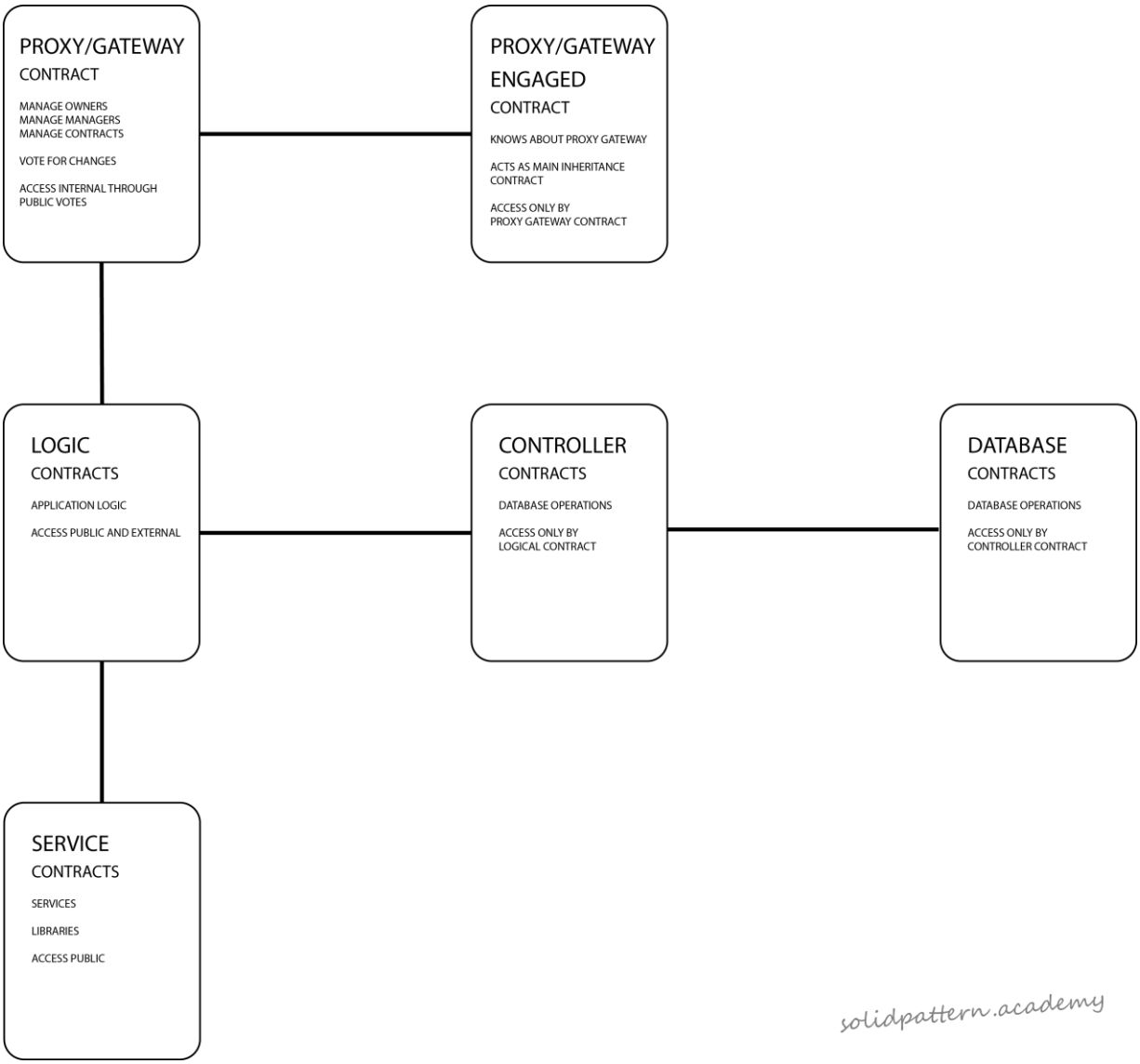
- Database Contracts

3.2 **Arbeitspaket 2** - *Implementierung der besten Lösung*

Wie im Arbeitspaket bereits dargelegt würde die Struktur einer Lösung wie folgt aussehen:

- Proxy Gateway Contract
- Logic Contracts
- Service Contracts
- Controller Contracts
- Database Contracts

Das nachstehende Bild visualisiert das Pattern:



3.3 Arbeitspaket 3 - *Als Referenzimplementierung publizieren*

Das Projekt wurde Inhaltlich aufgearbeitet und als Web Version auf solidpattern.academy veröffentlicht. Der Ablauf, die Gründe und die daraus resultierende Lösung wurden klar strukturiert dargelegt. Der Code und eine entsprechende Beschreibung sind auf der Webseite sowie der Repository einsehbar.

4 Liste Projektendergebnisse

Kurzbeschreibung der erreichten Projektendergebnisse jeweils mit Open Source Lizenz und Webadresse (netidee Vorgaben beachten!)

1	*SolidPattern - Code der Referenzimplementierung*	*MIT*	*https://solidpattern.academy/*
2	*solidpattern.academy - Code der Webseite, High Level View und Beschreibung des Solidity Pattern*	*MIT*	*https://solidpattern.academy/*
3	*Endbericht*	*CC-BY-SA*	*netidee/ solidpattern*
4	*Zusammenfassung*	*CC-BY-SA*	*netidee/ solidpattern*
5	*AnwenderInnen – und EntwicklerInnen-Dokumentation*	*MIT*	*https://github.com/Bit62/SolidPattern*

5 Verwertung der Projektergebnisse in der Praxis

Angaben zur Verwertung der Projektergebnisse in der Praxis

Dieses Pattern dient als Grundlage zur Entwicklung von skalierbaren Business Anwendungen auf der Ethereum Blockchain. Die Entwicklung auf einer Blockchain und das Grundlegende Setup zur Entwicklung wird nur grob als richtweiser beschrieben und wird neben allgemeinen Programmierkenntnissen und Kenntnissen der Programmiersprache Solidity vorausgesetzt.

6 Öffentlichkeitsarbeit/ Vernetzung

Beschreibung der im Rahmen Ihres netidee-Projektes bereits erfolgten bzw. noch geplanten Öffentlichkeitsarbeit oder Vernetzung

Das Projekt steht der Öffentlichkeit über eine eigene Seite (solidpattern.academy) in Deutsch und Englisch zur Einsicht zur Verfügung. Zudem ist das Projekt gut beschrieben auf GitHub der Community zur Verwendung und Bearbeitung bereitgestellt. Skalierbare Ethereum Lösungen gibt es kaum - wer danach Sucht wird unweigerlich auf dieses Projekt treffen.

7 Geplante Aktivitäten nach netidee-Projektende

Sind weiterführende Aktivitäten nach dem netidee-Projektende geplant?

Wird das Projekt von einigen Contract Entwicklern verwendet bzw. folgen Anregungen oder Verbesserungsvorschläge von Anwendern - so hoffe ich auf weitere Unterstützung und Weiterentwicklung des Projektes.

8 Anregungen für Weiterentwicklungen durch Dritte

Welche Nutzungs- und Weiterentwicklungsmöglichkeiten für Dritte ergeben sich durch Ihr netidee-Projekt bzw. empfehlen Sie?

Das Projekt steht der Öffentlichkeit zur Verfügung uns sollte auch genutzt werden. Durch die Lizenz sollten sich Verbesserungen bzw. Anpassungen an neue Solidity Versionen ergeben.