Testaufgabe TheGraph

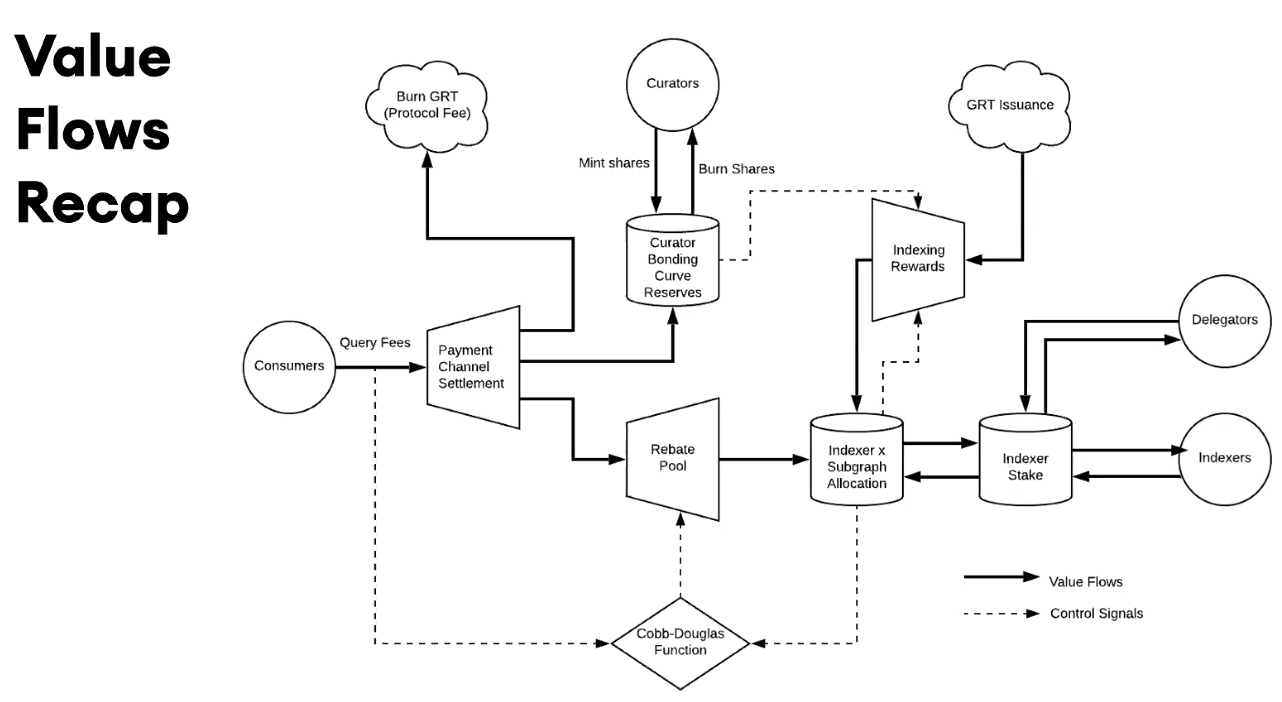
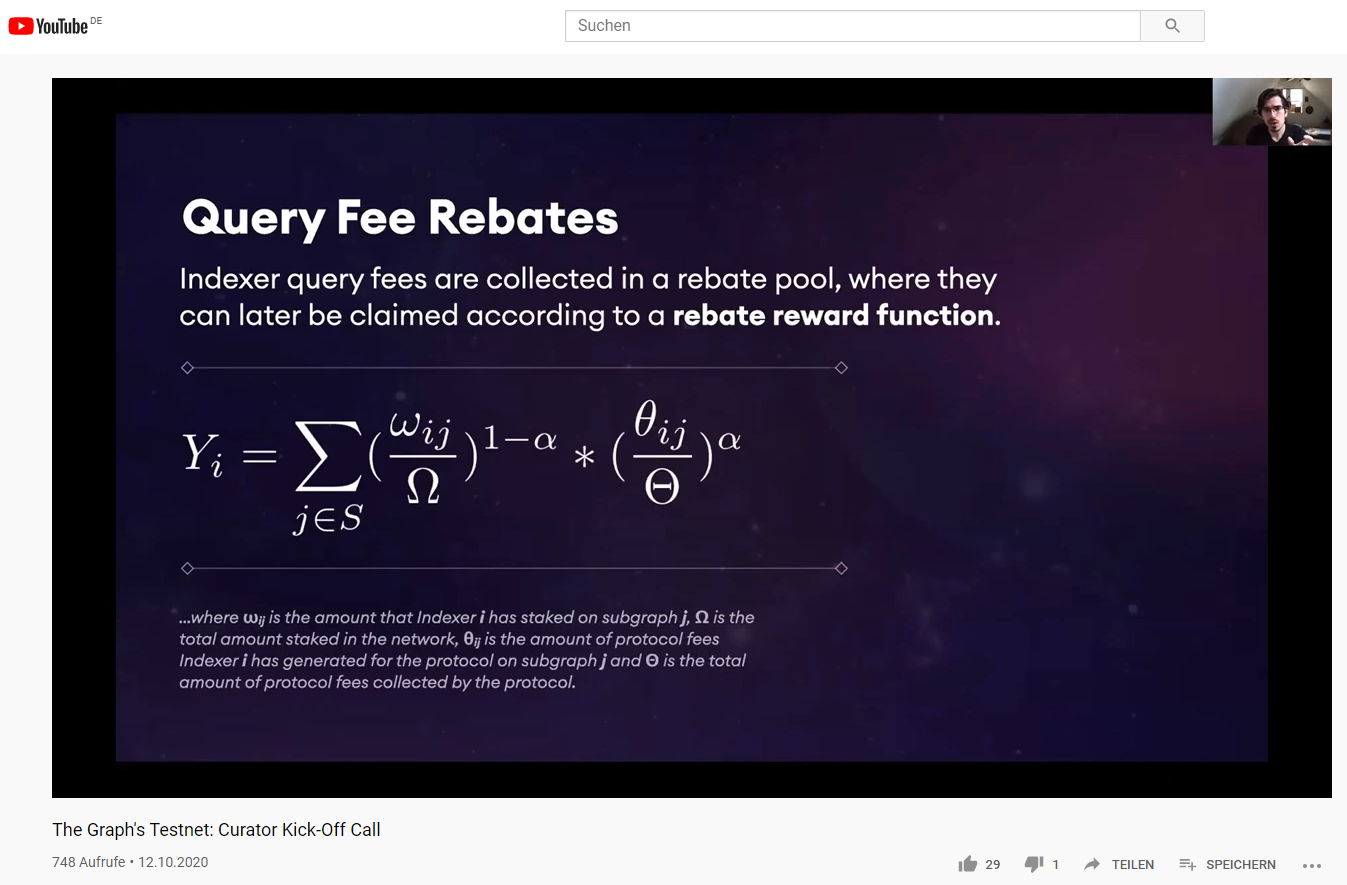
Wir möchten durch die Testaufgabe einerseits deine Stärken, als auch deine Fähigkeiten in Elasticsearch abprüfen. Daher haben wir uns eine geteilte Aufgabe überlegt. Im ersten Teil geht es um die Modellierung der Cobb-Douglas Funktion im TheGraph Network. Die Cobb-Douglas Funktion regelt die Menge an Gewinnen aus dem Rebate Pool. Hierzu wird der Stake auf einem Subgraphen in (optimaler) Relation zu den generierten Query Fees betrachtet.

Im zweiten Teil der Aufgabe geht es um die Analyse der Gaskosten im The Graph Network. Das heißt, es soll betrachtet werden, wie viel Gas eine Allocation, das Staking und andere Operationen im The Graph Network kosten. Hierzu müssen mit Elasticsearch die relevanten Events extrahiert werden und deren Gaskosten (möglicherweise mit der Transaktionshash) überprüft werden.

Für den ersten Teil soll der Metasubgraph von The Graph mit Hilfe von Python gequeried werden (siehe Ressourcen) und eine lineare Optimierung anhand der Cobb-Douglas Funktion erstellt werden, die uns Aufschluss darüber gibt, wieviele Queries wir beim aktuellen Stake bedienen müssen, um einen hohen prozentualen Anteil am Rebate Pool zu erhalten.   
Also:

* Queries in GraphQL erstellen und in Python anwenden und visualisieren
* Lineare Optimierung durchführen:
* Wie viele Queries müssten beim aktuellen Stake bedient werden für ein optimales ergebnis
* Wie müsste der Stake angepasst werden, um bei der derzeitigen Query Anzahl ein optimales Ergebnis zu erhalten?
* **Arbeits- und Denkweise beim Vorgehen erläutert und beschreiben.** Welche Schritte sind nötig, welche wurden unternommen, wo könnten Hürden entstehen?
* Die Informationen aus dem Subgraph (GraphQL) sollen möglichst **validiert** werden, in dem Informationen zum Stake über Elasticsearch mithilfe unserer API abgeholt und abgeglichen werden mit den Ergebnissen vom Meta Subgraph. (D.h. stimmen die Werte vom Stake, von der Anzahl an Query Fees etc. überein und **können diese Daten überhaupt generisch abgeholt werden?** Prozess dokumentieren).

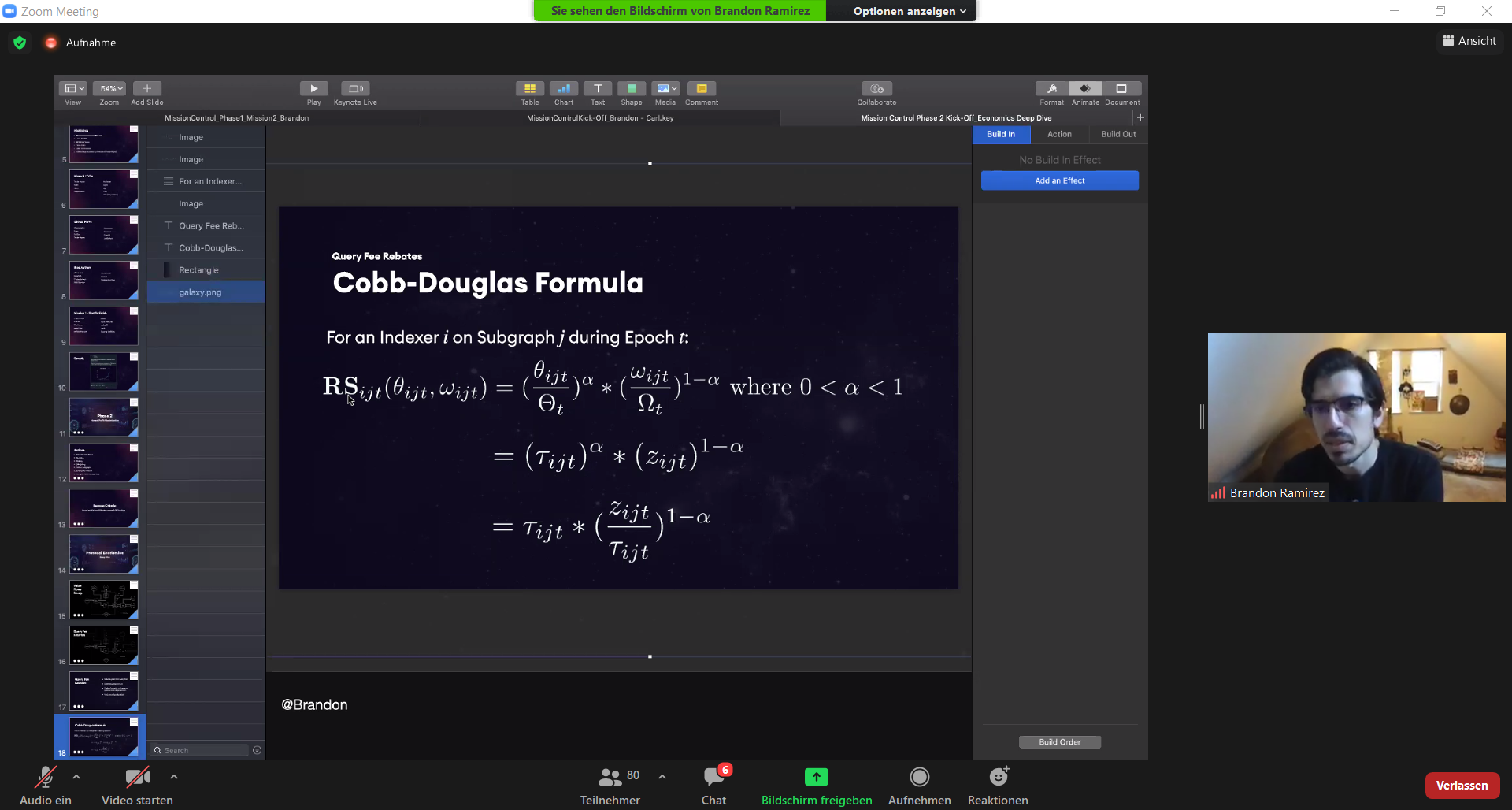
Anbei sind Ressourcen und Beispiele von TheGraph. Mehr Informationen können in der Dokumentation von TheGraph gefunden werden. Wichtig ist die Dokumentation der Vorgehensweise. Bei Fragen einfach melden! (Es geht ja nicht nur um die Inhalte, sondern wichtiger Bestandteil der Textaufgabe ist auch Herangehensweise und Kommunikation.)  
 <https://graphiql-online.com/graphiql> ist ein nettes Frontend für GraphQL Abfragen (API: <https://gateway.network.thegraph.com/network> ) und ermöglicht es, Queries leichter zusammenzustellen. Diese Queries sollen in Python übernommen werden. Es sollen wichtige Elemente die für die Optimierung relevant sind, visualisiert werden.



#### **Cobbs-Douglas Production Function**

In addition to query fees and indexing rewards, there is a Rebate Pool that rewards all network participants based on their contributions to The Graph Network. The rebate pool is designed to encourage Indexers to allocate stake in rough proportion to the amount of query fees they earn for the network.

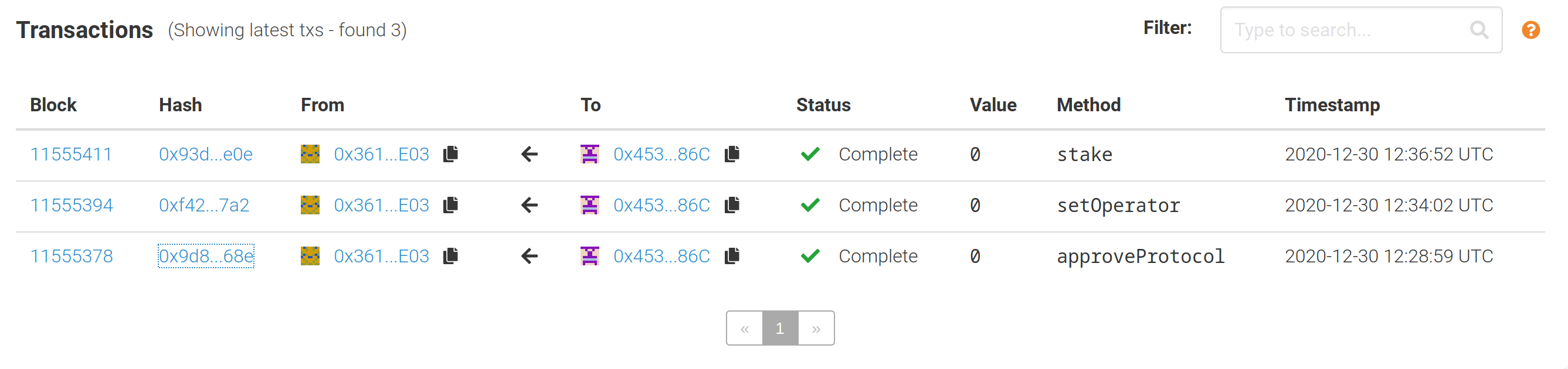
A portion of query fees contributed to the **Rebate Pool** are distributed as **rebate rewards** using the **Cobbs-Douglas Production Function**, a function of contributions to the pool and their allocation of stake on a subgraph where the query fees were generated. This reward function has the property that when Indexers allocate stake in proportion to their share of contribution of fees to the rebate pool, they will receive back exactly 100% of their contributed fees back as a rebate. This is also the optimal allocation.

[https://thegraph.com/blog/the-graph-grt-token-economics](https://thegraph.com/blog/the-graph-grt-token-economics**)

Ressourcen:

* <https://www.moesif.com/blog/technical/graphql/Getting-Started-with-Python-GraphQL-Part1/>
* <https://graphlets.io/>
* <https://graphiql-online.com/graphiql> -> Subgraph Meta: <https://gateway.network.thegraph.com/network>
* Anyblock The Graph Adresse mit Events: **0x453B5E165Cf98FF60167cCd3560EBf8D436ca86C**

Für die erste Teilaufgabe am besten unsere Dokumentation konsultieren wie man Events/Transaktions queried. Unser Blockexplorer ist nützlich um einen Überblick über events/transaktionen zu erhalten: siehe <https://explorer.anyblock.tools/ethereum/ethereum/mainnet/address/0x453b5e165cf98ff60167ccd3560ebf8d436ca86c/>

Warum ← zwischen from und to?

How to decode input data?

Teil 1: