[DRAFT] Efficient information and incentives propagation in multi-level communities

Anton Akentiev

Anytype Labs

`antonkent@anytype.io`

November 8, 2022

Abstract

In this work, we propose an approach to efficiently allocate and propagate incentives to the vertices of a social network of hierarchical communities. Information from one community in the hierarchy can be moved to another level community. As a result, such a network becomes an efficient data/knowledge curation system.

Each "community" in this paper can be treated as a group of people creating, consuming and curating information on a specific topic. Reddit's `https://reddit.com/r/rust/` board is a good example of such a community. Another great example is a `https://vas3k.club/` - a closed community that provides curated list of contents for a different topics like "how to get into cryptocurrencies", "how to become a good Rust programmer" or even "best places to visit in Japan".

In order to maintain a high quality of information and grow the community we propose a simple, yet powerful incentivisation mechanism.
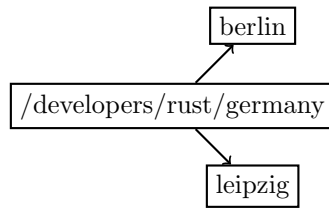
1 Introduction

There are many different systems designed for knowledge management [1] and data curation. Usually manual data curation is incentivized with some sort of reward. This can be non-monetary rewards like Reddit "likes", or monetary rewards such as that are used in token curated registries [2,3].

Incentive Networks is an approach in which a participant's reward depends not only on his own contribution; but also in part on the contributions made by his social contacts or friends [4].

In this paper, we describe an incentive network that motivates users to form hierarchical communities and to propagate data in order to categorize it. Each level on the path can be a separate community with its own members and curators.

Example:

```
            ┌────────┐
            │ berlin │
            └────────┘
                ▲
┌──────────────────────────┐
│ /developers/rust/germany │
└──────────────────────────┘
                │
                ▼
           ┌─────────┐
           │ leipzig │
           └─────────┘
```

Let's assume that each community is curated and moderated effectively on its level - relevant content is *added*, irrelevant content is *removed*.

1.1   System

Our proposed system consists of 3 key components:

1. Data propagation mechanism

2. Incentivization mechanism

3. Dispute resolution mechanism.

1.2   Out of the scope of this paper

For clarity these topics are left aside:

- How a community is bootstrapped and updated. We assume that communities already exist in some form and there are some ways to enter or exit them

- How communities establish hierarchy. We assume it is easy to launch a community on top of the existing one

- How communities are "mirrored". As an example: there can be many communities dedicated to Rust developers in Berlin. One of them can have such a path: */developers/rust/germany/berlin* and another one can be located at path */germany/berlin/developers/rust*

- How data is curated and sorted inside of each community. Different incentivisation mechanisms like Token Curated Registries or just a simple Reddit-like content propagation algorithm can be used to curate and sort data in the particular community

- How the proposed system can be automated using "auto propagation" bots or other algorithms to reduce human intervention

- How community members distribute rewards inside the community between its members

- How communities track origins of the content

- How communities provide Sybil-resistance

- How communities eliminate duplicates and normalize routes

- How community members are elected, especially in low level communities like "root", "earth" or "internet".

2 Information Propagation

2.1 Move Up and Move Down

Imagine there is a "Rust Berlin 2022 conference.pdf" record in *berlin*'s community. Obviously it is a very specific record that is relevant only to *berlin*, but not relevant to *leipzig*.

```
berlin ──── Rust Berlin 2022 conference.pdf
   ▲
   │
/developers/rust/germany
   │
   ▼
leipzig
```
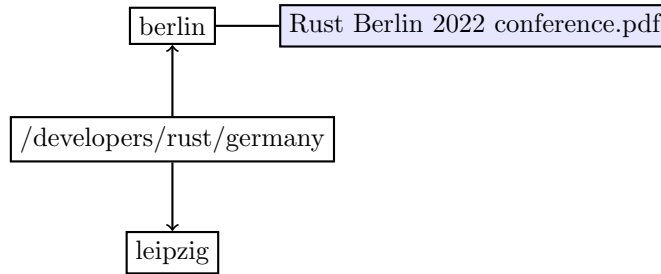
Imagine that there is also a "Curated list of Rust meetups in Germany.pdf" record in *berlin*'s community.

```
Rust Berlin 2022 conference.pdf
        ╲
       berlin ──── Curated list of Rust meetups in Germany.pdf
          ▲
          │
  /developers/rust/germany
          │
          ▼
       leipzig
```
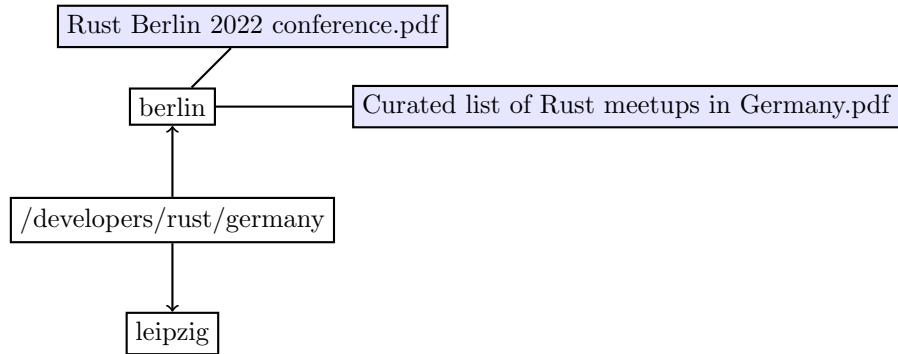
In this case, a record can be propagated further down to *germany*. Let us call this action *move down*. Such action can be done by either the higher level community (i.e. *berlin*) or by lower level community (i.e. *germany*).

```
Rust Berlin 2022 conference.pdf
        ╲
       berlin
          ▲
          │
  /developers/rust/germany ── Curated list of Rust meetups in Germany.pdf
          │
          ▼
       leipzig
```
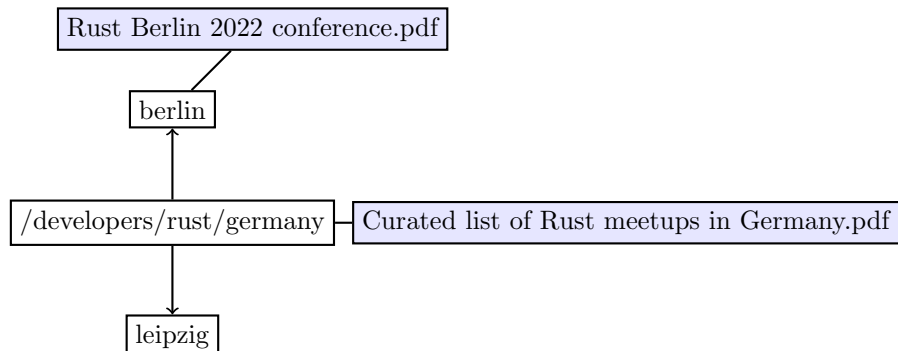
In some cases we will want to move data to the more specific community if

it is no longer relevant. Let's call this action *move up*.

Usually it is not in the interest of the higher level community to move data lower, because in this case the community will not collect incentives received by these data records (more on that later). It is only in the interest of the lower level community to collect more and more content from the top level connections. We believe that a balanced incentivization model will motivate honest and good behavior. Otherwise, if your community is bloated with erroneous or unrelated data, users will leave it.

As a result, community members can not only *add*, *remove*, *hide* or *sort* data in their community, but also to *move data* to more relevant communities. This is how our curation mechanism works. And this should be reinforced with incentives. Usually extra incentives solve a problem of low user participation.

Rule: community members can move data up or down one step in the hierarchy. It is not possible to move data to an arbitrary graph vertice.

3 Multilevel Incentives

Actions like *add*, *remove*, *hide* and *sort* should be incentivized in some way, but it is out of the scope of the current paper. However, data propagation incentives are critical for our case.

Our goal is to:

1. Incentivise users to launch their community on top of existing hierarchy

2. Incentivise community members to add relevant data to the community

3. Incentivise more specific (higher level) community members to move data from lower level communities to increase their reward

4. Incentivise more generic (lower level) community members to move data from top level communities to increase their reward

5. Incentivise honest behavior.

Different examples of incentives (also known as rewards) are as follows: tipping, payment for the paywalled contents, reputation increase, likes, global token incentives/distribution, etc.

For the sake of simplicity, let's select the first option: platform users will send tips for the interesting and relevant content.

Rule: rewards should be distributed proportionally to the level in the hierarchy, from most specific to less specific communities.

3.1 Example

User finds a great CV in the */developers/rust/germany/berlin* and wishes to send a $1 as a tip to the */berlin* community. Reward distribution is as follows:

- 3% to /

- 5% to /developers

- 8% to /developers/rust

- 13% to /developers/rust/germany

- 71% to /developers/rust/germany/berlin

Note: even if the work of adding and preparing data was done by the top level community, lower level communities still receive some percentage of fees. It is obvious that 3% of one million tips mean huge profit for the / community. This network effect [5] will help us to bootstrap a broad hierarchy of communities with members willing to participate in the curation process.

On the other hand, it should be very easy to launch a community on top of existing one, which will force users to manually propagate information upward, which improves categorization of the data. As an example: you can create a */seniors* community on top of */berlin* and to move some CVs up to get more rewards.

By changing distribution rules and constants we will be able to build a balanced system in which: being lower in the hierarchy means lower fee cut but higher inflowing stream, e.g.: 3% of 1000000 tips. being higher in the hierarchy means higher fee cut but lower inflowing stream, e.g.: 71% of 100 tips.

Possible improvement: in the described approach, moved data does not track the origin. This means that if content was moved from one community to another, the system will not distribute extra incentives to the original community that has added that content in the first place. This mechanism can be introduced later, but it's outside of our scope.

Possible improvement: reward rates can vary depending on how many staked tokens are locked in each community. The more tokens a community locks, the more percentage of the reward it receives. This improvement is also outside of our scope.

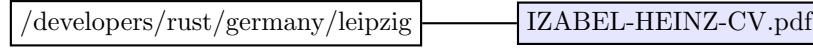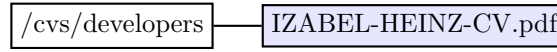## 3.2 What if multiple copies of data exist?

There are three different situations in which copies of the same record can exist:

1. Copies in Unrelated communities

2. Copies in Sibling communities

3. Copies in Parent-child communities

3.2.1 Unrelated communities

Copies exist in unrelated communities. This is totally fine as long as data is relevant to each community. Also, this situation can motivate users to create more specific communities on top of the low level hierarchy.
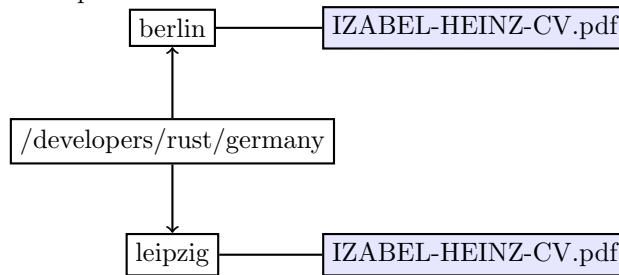
Example:

```
┌──────────────────┐      ┌─────────────────────────┐
│ /cvs/developers  ├──────┤  IZABEL-HEINZ-CV.pdf    │
└──────────────────┘      └─────────────────────────┘


┌──────────────────────────────────┐   ┌─────────────────────────┐
│ /developers/rust/germany/leipzig ├───┤  IZABEL-HEINZ-CV.pdf    │
└──────────────────────────────────┘   └─────────────────────────┘
```

3.2.2 Sibling communities

Sibling nodes are nodes on the same hierarchical level under the same parent node. In this case data may be propagated one level down and only one copy should be left as a result. This can also be automated.
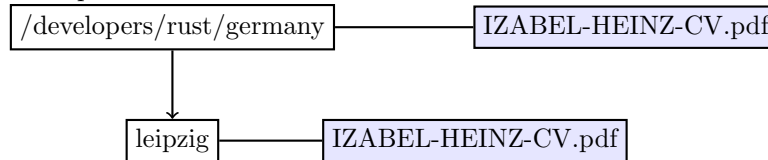
Example:

```
┌────────┐          ┌─────────────────────────┐
│ berlin ├──────────┤  IZABEL-HEINZ-CV.pdf    │
└────────┘          └─────────────────────────┘
     ▲
     │
┌──────────────────────────┐
│ /developers/rust/germany │
└──────────────────────────┘
     │
     ▼
┌────────┐          ┌─────────────────────────┐
│ leipzig├──────────┤  IZABEL-HEINZ-CV.pdf    │
└────────┘          └─────────────────────────┘
```

3.2.3 Parent-child communities

Copies exist on both - parent and children levels of hierarchy. By default low level data should be selected as the correct duplicate. But the Dispute Resolution mechanism can be used if a higher level community can prove that data is more relevant and is more specific to their community.

Example:

```
┌──────────────────────────┐      ┌─────────────────────────┐
│ /developers/rust/germany ├──────┤  IZABEL-HEINZ-CV.pdf    │
└──────────────────────────┘      └─────────────────────────┘
            │
            ▼
      ┌────────┐      ┌─────────────────────────┐
      │ leipzig├──────┤  IZABEL-HEINZ-CV.pdf    │
      └────────┘      └─────────────────────────┘
```

4   Dispute Resolution

As long as each community is incentivized with rewards, communities *can* try to move all data to them, in order to collect the greatest reward. We mentioned above that communities that provide data of low quality will organically get less attention by users of the system, so such a system can feature no special mechanism to prevent the adversarial behavior.

Still, we believe that a special mechanism is required. It will enforce the rules of the system and will increase the overall quality of the content on the platform.

Dispute Resolution in its most simple form can be done by a moderation team run by the top-level community first. Next we will describe an approach influenced by token curated registries [2, 3]:

1 - Staking: a UserA launches a new community by submitting a staking deposit. By doing that he is now able to curate contents and move data up and down.

2 - Data propagation attempt: UserA moves some data up or down from another community.

3 - Signaling: If another UserB (not necessary from the same community) feels that action of a UserA is adversarial or wrong, she can issue a challenge, by submitting an equal amount of tokens. We call this user - a "challenger". This initiates a voting period.

4 - Voting: Moderators then decide whether the challenger was right. There are 2 sides that moderators can take: "YES, challenger is right" or "NO, challenger is wrong".

5 - Finalizing: After the voting period concludes, tokens are settled as follows:

If the challenger succeeds:

- UserB receives his deposit back in full

- $x\%$ of the UserA's deposit is distributed to the UserB as a reward for winning in dispute resolution

- $100 - x\%$ of the UserA's deposit is distributed to the moderators on the winning side (YES)

If the challenger fails:

- UserA receives his deposit back in full

- $x\%$ of the UserB's deposit is distributed to the UserA as a reward for winning in dispute resolution

- $100 - x\%$ of the UserB's deposit is distributed to the moderators on the winning side (NO)

6 - Data propagation: as an additional result of the dispute resolution process, data is finally moved to the proper community; either up or down.

5 Conclusion

In this work, we proposed a simple system that allows efficient data curation with the help of incentivized graphs of communities. That description should be taken into account as a simple primer of our idea. Please refer to the "out of the scope of this paper" section for more information on what components are still missing and should be considered in the real-world implementation of our idea.

References

[1] Murray E. Jennex, *What is Knowledge Management?*, `https://www.researchgate.net/publication/314500732_What_is_Knowledge_Management`, 2007

[2] Clark R., *Framework-based Token Curated Registries*, `https://hackernoon.com/framework-based-token-curated-registries-9691e83c2c4c`, 2018

[3] Mike Goldin, *City Walls and Bo-Taoshi: Exploring the Power of Token-Curated Registries*, `https://medium.com/@ilovebagels/token-curated-registries-1-0-61a232f8dac7`, 2017

[4] Yuezhou Lv, Thomas Moscibroda, *Incentive Networks*, `https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/AAAI_2015.pdf`, 2015

[5] *Wikipedia: Network Effect*, `https://en.wikipedia.org/wiki/Network_effect`