

Developers need steady income in the long run to fund development and to compete with traditional services. How about: The developers charge fees but instead of instant payments the fees are transferred to a kind of DAO. The developers then make a proposal stating the amount of money they need and what kind of development they want to fund. Then users can decide to accept that proposal or to reject. In case of rejection the money would be transferred back to the users.

#### Advantages

- The problem of collecting fees is that somebody just forks the code and deploys the smart contract without fees. One reason against that is that users want sustainability and innovation and therefore support the development team. This reason would be reinforced by that user have a say in that regard and can actively support or reject proposals. For example users can reject a proposal when it is clear that a project turns into a cash cow where there is little to no further development.
- Accountability and transparency would improve a lot.
- The fact that users are kind of investors is an incentive alignment that is obviously very healthy.
- People who use the service more often and pay more fees have higher voting power. People who use the service one time have little voting power. This is a good incentive structure.

#### Disadvantages

- Higher complexity and therefore security risk \* Higher costs to just simply collecting fees

This is just a simple draft and I'm wondering how complex it would be to implement something like that at the current state? Are there dapps like Aragon that could support something like this?