BUIP157: Continue Funding a Developer  
Submitted by: Peter Tschipper  
Date: 2020/12/01  
  
Summary:  
  
I still believe this project, over all other cryptos, to have the best chance for significant adoption/scaling, and look forward to continuing to be a part of making it happen, as I have been for the last 5 years.  
  
This BUIP 157 replaces BUIP 134 from last years funding.  
  
Budget:  
  
This BUIP authorizes 100,000 USD a year (paid monthly in crypto) to continue this work. This includes a 4% cost of living bump over last years’ 96,000.  
  
Contributions to this project:  
  
Over the last years aside from the daily grind of fixing bugs, reviewing code and implementing small features, I’ve made the following significant contributions:  
  
Xthinblocks  
  
Parallel Validation  
  
Implemented compact blocks and worked with George Bissias to get the first graphene implementation working on main net. We now seamlessly support all three block relay types in BU.  
  
Created the first Bitcoin implementation of CPFP that has the ability to efficiently mine long unconfirmed chains.  
  
Finished the implementation of dsproofs.  
  
Concatenated transactions which saves significant bandwidth under high load.  
  
Implemented Xpress Validation during mining which significantly improves the performance of mining large blocks. Only 10% of the time is now spend in the validation phase of mining rather than 50%.  
  
Created a priority queue for sending and receiving priority p2p messages. This ensures that block relay is never affected by a backed up transaction queue.  
  
You can view my daily contribs here : <https://github.com/ptschip> and also <https://gitlab.com/ptschip>  
  
  
Current work:  
  
Adding double spend notifications to the BU wallet as well as ZMQ messages.  
  
Unbounded Transaction chains: The final step in removing the unconfirmed chain limit in BU and currently being reviewed in dev. It allows the efficient mempool admission of infinite unconfirmed transaction chains while at the same time preserving CPFP functionality.