# **Request Money**

# **1 Basics**

A table with an overview of the support status and applicability.

|  |  |
| --- | --- |
| Status: | e.g. **Supported** |
| Architecture(s): |  |
| Component(s): | Smart Contracts, Mocha, Chai |
| Hardware: | *n/a* |

# **2 Overview**

This feature offers users the ability to request funds from a specific user using an easily identifiable username.

# **3 User details**

A user selects another user from their list of friends or a search and enters the amount of Divi they would like to receive. If the counterparty accepts the request and has enough funds to fulfill it, a transaction is initiated for the amount in question automatically.

# **4 Technical details**

A payment request is essentially a smart contract between two counterparties that is initiated by the requester and confirmed by the requestee.

**Example:**

Andy, who has 0 DIVI in his wallet, requests 100 DIVI from Betty for car repairs. Betty has 200 DIVI in her wallet. The initial request generates a one-time smart contract between the counterparties, but the transaction is not broadcast to the blockchain until Betty accepts the transaction, after which Betty is left with 100 DIVI, and Andy now has 100 DIVI as well.

**Potential Smart Contract frameworks:**

Hyperledger - IBM’s blockchain framework offers easy-to-use, Javascript based blockchain development software that incorporates smart contracts.

Ethereum - The obvious choice for smart contract development, but could be limited because using the Ethereum network would be required.

Request - https://request.network/#/

# **5 Limitations**

Using PIVX or even DASH as the primary fork could present limitations in implementing smart contract technology. We may have to build our smart contract framework into the blockchain.

# **6 Testing**

We can use Mocha along with Chai to run unit tests on smart contracts. If we end up using a pre-built smart contract framework like Ethereum, there are specific testing frameworks that coincide, such as Truffle.

It would be advantageous to test the smart contracts on the test net as well once unit testing has been accomplished.

# **7 Areas for improvement**

TBD

# **8 Known issues**

Smart contracts can lack security and therefore need to be audited heavily before deployment.

# **9 References**

Solidity - <https://solidity.readthedocs.io/en/develop/>

Hyperledger - <https://www.hyperledger.org/>

Request - <https://github.com/RequestNetwork>