Boomerang: Trust in many places.

Boomerang is a decentralized web and mobile application that enables third party escrow for transactions, personal property uses, and business purposes.

For the past decade, internet businesses have thrived on a base not of technology, but of trust. The trust of users that the software will protect their privacy, their efforts, and their assets. Organizations like paypal, the closes thing to a competitor in this space facilitate transactions as a middle-man, protecting users in purchase transactions but charging costly fees, not providing true parity in their administrative process, and steering users into their sales platforms i.e. ebay.

Boomerang is the decentralized answer that necessitates no middle-man, or sales platform. It is a free standing, low cost escrow platform that allows for greater versatility in the type of transactions it can be useful for.

Boomerang puts it's geographically distributed userbase into the driver seat, thereby adding a backbone of democracy to a financial application for the first time in history. Here is how:

Use Case:

Mike wants to lend Greg a 2500 Dollar Gibson guitar, but he knows mistakes can happen. Asking for a deposit might seem rude and questionable from Greg's perspective. So Mike asks Greg to throw him a boomerang!

An agreed upon amount of Greg's money is transferred into a specific account tracked centrally (hashtable/db) and on the Distributed Ledger. Their agreement, regarding the timeframe, condition, photos, and details are stored in a table.

In most cases, Mike will not have to open up a case and upon mutual confirmation, the funds will be returned to Greg. In the event that the Guitar is damaged, destroyed, or the agreement is not upheld in any way, Mike may open up a Claim.
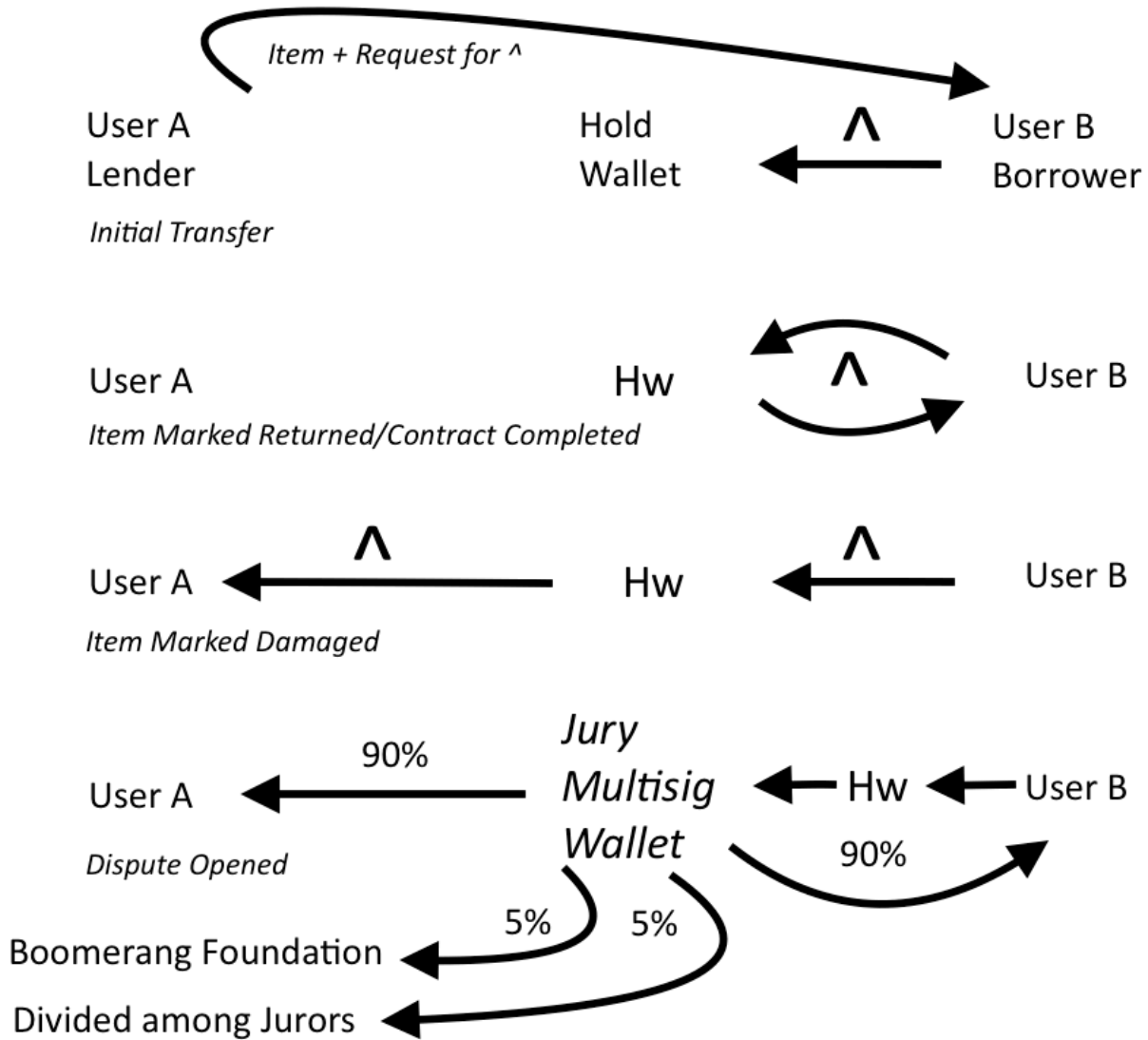
If a claim is opened up, Ten randomly chosen users who have opted-in to 'jury duty' are submitted the information regarding the case, evidence from both parties, and are tasked with deciding who gets the deposit and how much of it.

For a minute or two of their time, they are rewarded with .5% each of the total deposit, costing 5% of the pool. This ensures that there will always be moderators in place, similar to the way distributed ledgers reward nodes for keeping the systems operational.

Currently, the only forms enforcement of electronic or ecommerce transactions are centralized in bodies like paypal, which take a larger cut by default and require an actual Sale to take place. Alternatively, if enough personal information is recorded and the transaction takes place in a singular jurisdiction, courts could handle the matter however time consuming, expensive, and inefficient they may be.

Boomerang is a concept for the enforcement of integrity in a fast-paced, efficient era.

BoomKoin Collateral Situational Cases

Item + Request for ^

| User A | Hold | ^ | User B |
| Lender | Wallet | | Borrower |

Initial Transfer

User A   Hw   ^   User B

Item Marked Returned/Contract Completed

User A   ^   Hw   ^   User B

Item Marked Damaged

Jury
Multisig
User A   90%   *Wallet*   Hw   User B

Dispute Opened

Boomerang Foundation   5%   5%   90%

Divided among Jurors

The Tokensale:

This is an area of the Ethereum space where our software really shines.

The worst ERC20 token sales have only a registration form, and an address to transmit your ethereum to. They then go on MIST (ethereum wallet) or Geth and manually distribute the token to the respective ethereum wallets, which sent ethereum during the sale or pre-sale.

Better token sales already are being done through a web or native application based wallet which may also be the app which the token will later be utilized within. There may also be a well known exchange such as Shapeshift or Changelly in place to speed things up.

Our app has the option of receiving signing authorization of your **existing** ethereum wallet and performing the exchange with one click at current and dynamic ethereum prices. It can do this because of KOIN.

Boomerang is powered by the KOIN tokensale engine, a piece of in house software built in Java. It instantiates the solidity contract for the token (wrapped into java using the web3j library), and then requests the keystore file of the users wallet (from metamask, mist, etc). It secures this information using a cryptographic SALT.

The ethereum received from the token sale is automatically transferred to a ballot-featured multisignature wallet controlled collectively by the Boomerang team and it's trustees/early private investors.

 More on KOIN can be found here.

Post-Sale Structure:

The client app has two distinct ways of altering it's token. It can directly use the walletfile to initiate a transfer of the token through the nearest node. Instead, if there is connectivity to the Boomerang Server, it can request that the app host server (itself a node on the block chain) initiates a transfer of boomtoken to another wallet (within the platform as an escrow transaction or to another user or exchange to cash out into ethereum). These two functions give the platform a robust persistence that is difficult to match.

The logic for the escrow system is a more simple variety of java methods written by our team that only instantiate the solidity contract (now translated into Java by web3j) when performing a token issuance(which we will not reissue), a transfer on the block chain, or a secure judgment protocol where the "jury members" decide on the outcome of a dispute case.