InsuraLink

A modular open-source IoT framework that employs Chainlink middleware to execute real-time peer to peer insurance smart contracts built on Ethereum. On-chain parametric deterministic insurance contracts are set to minimise the transaction costs, specifically the coordination costs, when claiming for insurance. Data-driven insurance contracts allow for claims to be instantly approved or disapproved, which lowers the processing time for agent to effectively zero. Become reconnected with the risk in your immediate network by engaging in peer to peer insurance markets.

For corporate and enterprise solutions where open-source contracts are undesirable due to privacy concerns, private ledgers and side chains can be employed to ensure confidentiality as well as scalability.

Hackathon Features

Real World Data Events

- Multiple IoT sensors. For security and IoT reliability, using multi-factor authentication for the IoT layer is recommended.
    - Tilt sensor: 45 degrees tilt
    - Temperature sensor: 70 degrees fahrenheit
    - Humidity sensor: less than 1% humidity
    - G-force sensor: if more than 3 gs (if the pizza is dropped)
- 3 IoT hubs are used to decentralise the data inputs.
- The IoT sensors serve data in HTTPS endpoints to mimic other APIs, accessible at any time by external services.

Chainlink Middleware

- External Initiator server that polls the IoT data
  External initiator starts a Chainlink job if a data threshold is met.
- Raspberry Pi Functionality
- 2 Chainlink nodes used to decentralise oracle inputs
- Connects to Ethereum Ropsten RPC

Ethereum Smart Contracts

- Solidity smart contracts deployed on Ropsten
- DAI ERC20 token used as payments

Frontend functionality

- Interactive data flow diagram
- Real-time data feeds
- Pay in DAI
- Interactive UX that allows users to engage with all key components.

The P2P Insurance Market

Become reconnected with the risk in your immediate network by engaging in peer to peer insurance markets. The coordination costs of engaging in insurance contracts is effectively lowered by the InsuraLink market, allowing local communities to more readily insure peers within their immediate networks. Risk can therefore be localised, transparent and distributed among incentivised network stakeholders.

Future Implementations

Number and Frequency of Payments

A future addition to the solidity contracts would be to add in the number and frequency of payments that occur on the insurance contract. This would allow for both the buyer and seller to write and issue more specific contracts that are better suited to their utilities.

Data Security Concerns

The full-stack design of the insurance contract is built with security and transparency in mind. Multi-factor authentication for IoT devices will help to protect against this. Deterministic contracts are only as valuable as their inputs. Multifactor authentication and other IoT security protocols must be considered further in order to produce a scaleable, peer to peer solution.

Dispute Function

It is important to firstly mention that tribunal systems are ineffective if the data has been tampered with in the first place or the sensor is faulty. Although in the case where the sensor is believed to be secure, a “Dispute” function can be called which retriggers the temperature sensor on the spot to execute a Chainlink job run.

Past this point of dispute, if unsatisfied the disputer can call a second and last “Dispute” function which post the on-chain IoT data logs to an on-chain oracle tribunal (such as Augur) where the outcome can be determined by incentivised voters. There are incentive mechanisms designed into protocols such as Augur to deter bribery, but there is always the risk that the profit from corruption will outweigh the cost of corruption.

On-chain data

The IoT data is written on-chain using the Skale file storage system in order to produce an immutable record of data that can be used in order to dispute contract outcomes post-execution. This data could also be referenced by the insurance smart contracts.