

Aigang

Blockchain protocol for digital insurance

Insurance for Internet-of-Things using DAO and smart contracts

Release 0.3

Updates from Release 0.2 (2017 July):

- **New sections:** 10. [Example of Free Reserve Calculation](#) 11. [Solvency II Standard Formula for Insurance DAO](#) 11.1 [Non-Life Underwriting Risk](#) 11.2 [Premium and Reserve Risk](#) 11.3 [Catastrophe Risk](#) 11.4 [Counterparty Default Risk](#) 11.5 [Operational Risk](#) 11.6 [Overall Solvency Capital Requirement](#)
- **Updated sections:** 6. [Investing in DAO Insurance and Token Model](#) 12. [Pricing simulation with Machine Learning](#)

Contacts:

team@aigang.network

Web:

<https://aigang.network>

Slack:

<http://slack.aigang.network>

Telegram:

<https://t.me/aigangnetwork>

2017 August

Table of contents

- [1. Executive summary](#)
- [2. Smart Policy](#)
- [3. Smart Pricing](#)
 - [3.1 Risk Premium](#)
 - [3.2 Office Premium](#)
- [4. Smart Claims](#)
- [5. Profit Distribution](#)
- [6. Investing in DAO Insurance and Token Model](#)
- [7. Peer to peer investment platform](#)
- [8. Device data tracking and claim process](#)
- [9. Aigang Protocol Showcase: Functional Battery Insurance for Smartphones](#)
- [10. Example of Free Reserve Calculation](#)
- [11. Solvency II Standard Formula for Insurance DAO](#)
 - [11.1 Non-Life Underwriting Risk](#)
 - [11.2 Premium and Reserve Risk](#)
 - [11.3 Catastrophe Risk](#)
 - [11.4 Counterparty Default Risk](#)
 - [11.5 Operational Risk](#)
 - [11.6 Overall Solvency Capital Requirement](#)
- [12. Pricing Simulation with Machine Learning](#)
- [13. Architecture of the protocol](#)
- [14. Roadmap](#)
- [15. Leadership & Core Team](#)
- [16. Competitive landscape](#)
- [17. Conclusion](#)

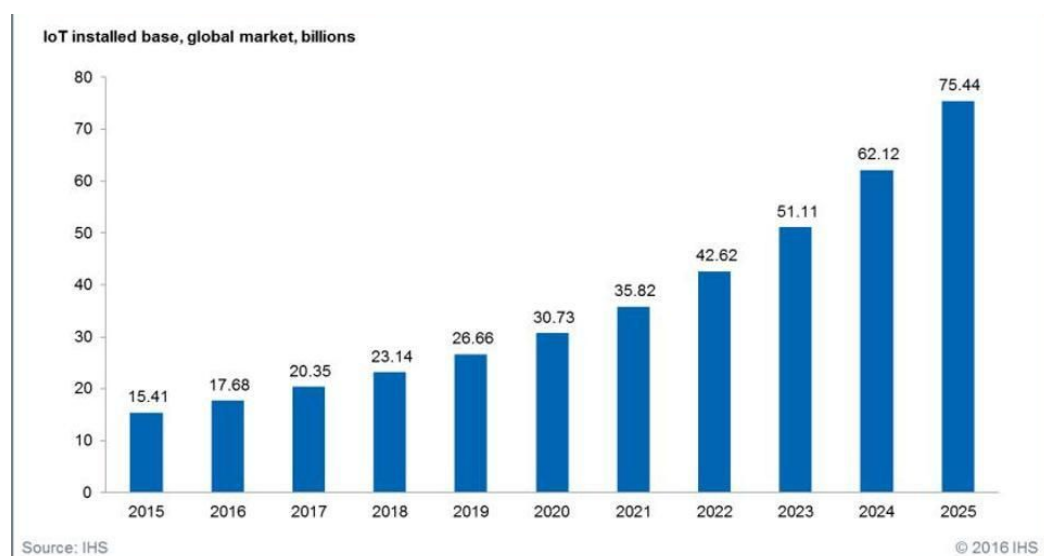
1. Executive summary

The blockchain is a decentralised database that maintains a continuously growing list of ordered records. The blockchain was first used in 2009 as the basis for the digital currency Bitcoin. Since then the technology has evolved drastically. Now it is one of the cutting-edge innovations even named the greatest revolution since the Internet. The main advantages of the blockchain technology underpinning its success are disintermediation (no central agent is required to approve transactions), immutability (all transactions cannot be altered or deleted), reliability (database is replicated on a large network of servers and does not have a central point of failure) and transparency (changes to public blockchains are publicly viewable by all parties).

The second generation of blockchain (referred to as “Blockchain 2.0”) [\[1\]](#) provides a platform for smart contracts which embed software code that contains defined rules and can execute code based on those rules. This enables a formation of a Decentralised Autonomous Organisation (DAO) which functions without human employees. This greatly extends the potential applications beyond digital currencies especially in the financial industry. The blockchain has just scratched the surface of the global insurance industry transformation. As potential fields of application in insurance considered today are new methods for distribution and payment, increased effectiveness in pricing, claims handling and fraud detection, reinsurance, reducing administrative costs and new products and services for growth, such as the Internet of Things (IoT).

The IoT is a fast growing phenomenon which is projected to grow to 75 billion objects by 2025. This provides an emerging market for automated insurance as those devices could detect the broken part and initiate the claim themselves, e.g. smartphone after crash could detect not functioning microphone, drone could detect its broken accelerometer or Amazon Echo could detect broken glass in the office. This paper discusses the framework for incorporating insurance for IoT as a DAO powered by smart contracts in a blockchain. It discusses the main aspects of the insurance DAO smart contract, namely Policy, Pricing, Claims and Profit Distribution.

Figure 1. “The IoT market growth” [\[2\]](#)



The growing adoption of IoT products in developed and developing economies, value-added services for insurance industries, cost reduction of premium policies and automation are the main drivers for global IoT adoption in the industry. According to the report "IoT Insurance Market - Global Forecast to 2022" conducted by [MarketsandMarkets](#), the global IoT insurance market is expected to be worth USD 42.76 Billion by 2022, growing at a CAGR of 65.89% between 2016 and 2022 [4]. The IoT will have a huge impact on various markets: e.g. automotive, transportation, agriculture, consumer electronics. Global positioning systems, in-built sensors and other detectors would increase the need for blockchain technologies in the insurance industry as those systems can gather valuable data, interpret it and perform needed actions autonomously. For instance, the growing usage of drones and Wi-Fi enabled devices in China, India and Japan [5] opens up new possibilities to insure the devices themselves and the environment they are in based on the gathered data.

2. Smart Policy

In standard wording, it is important to consider appropriate exclusions that would limit the coverage. The implementation of the Bitcoin, blockchain was the first ever solution to the double spending problem. In the same way the blockchain can ensure the uniqueness and validity of the insurance policies and provide the offer and acceptance property which is essential for a contract to be legally binding.

With Blockchain 2.0 further benefits arise if insurance operations such as underwriting and claims handling can be automated by the rules embedded in the smart contract. The blockchain could then be used to automatically sign new policies, evaluate provided data of damage, trigger the repair process and claim payment. In order to achieve that the insurance cover has to be standardised as much as possible.

The traditional insurance policy is the document which governs the relationship between the insured and the insurer. It sets out the rights and obligations of both parties and the rules for making transactions. Similarly, the Smart Policy would form the basis of the insurance DAO but by including software algorithms it would also automatically govern and perform its operations such as; issuing new policies, handling claims, distributing profits, etc.

Conventional insurance policy consists of two parts – standard wording and a schedule. Standard wording is the same for all policyholders and describes the cover provided as well as rules, rights and obligations of the parties. Of particular importance are the terms and conditions under which an insurer is liable to pay insurance claims and must be carefully worded to cover all possible circumstances under which payment will and will not be made. The cover is usually split into perils, some of which can be included by default others can be optional.

Sample perils for the IoT would be:

- Extended Warranty (replacement or repair of the device beyond the standard warranty period provided by the manufacturer).
- Product Liability (damage to the insured, insured's property or third parties caused by

malfunction of the device).

- Damage by Third Parties (damage to the device, insured, insured's property or other parties caused by malicious actions of third parties such as hackers).

Provided and enable automated verification by software algorithms. Examples of situations where payment should be avoided include:

- Policyholder has an advantage by possessing more information about the likelihood of a claim
- Claim event is under the control of the policyholder
- Claim event would be difficult to verify
- Loss can be considered as depreciation

Without appropriate exclusions the probability of a claim being filed or inaccurate risk assessment would be very high. Exclusions can also be used when the risk is covered by a third party, such as the manufacturer who guarantees repair or replacement within the standard warranty period.

The second part of the policy form is the schedule, which may vary between policyholders. The schedule may include:

- Details of insured object
- Excess or deductible
- Sum insured
- Optional covers
- Premium and payment schedule

If wider cover is to be provided than can be handled by automated algorithms, part of the business could be reinsured. For example, the part to be ceded to the reinsurer could be determined by peril (e.g. retaining property damage claims and ceding liability) or by size, e.g. retaining smaller claims and ceding claims above retention limit.

In addition to standard wording and the schedule the Smart Policy would virtually include a third part consisting of the software algorithms executing all the operations of the insurance DAO.

3. Smart Pricing

Underwriting of an insurance contract normally consists of the following steps:

- Assessing whether the risk can be accepted
- Setting the conditions
- Setting the premium

In our case the first step should be straightforward since the insurance product can be bundled with the IoT device by default or acceptance can be verified by checking a list of required conditions and device data which can be performed with help of software within the device, machine learning algorithms (based on historical data) and smart contracts. If required, the blockchain could receive (using Oracle's) data from external data sources to retrieve needed extra information. The second point is also straightforward since policy terms and conditions should be standardised for all

policies. So in this part we focus on premium calculation, which can be split into two parts – Risk Premium and Office Premium.

3.1 Risk Premium

The risk premium is the pure cost of claims, which is not known in advance and has to be estimated using mathematical techniques. At start up the risk premium will have to be based on assumptions or benchmarks available from reinsurers or the market. Once the blockchain is in operation data from its own claims experience can be used to determine the risk premium. The data can be represented as:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \cdots & x_{2n} \\ x_{31} & x_{32} & x_{33} & \cdots & x_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m1} & x_{m1} & \cdots & x_{mn} \end{pmatrix} \text{ and } \mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{pmatrix} \quad (1)$$

\mathbf{X} is the matrix of predictors potentially to be used as rating factors. x_{ij} is the data element representing the value of j^{th} characteristic of the i^{th} policy. \mathbf{Y} is the vector representing the dependent variable we wish to predict for pricing, e.g. number of claims or claim amount. y_i is the data element representing the value of the variable for the i^{th} policy.

Evaluation methods for the risk premium range from simple approaches like average burning cost (claim amount per unit of exposure) to predictive modelling like GLM (Generalised Linear Model) which is widely used by general insurers. The aim of the model is to identify the factors which have the most impact on the amount of risk taken. It can be represented by the following equation:

$$Y = g^{-1}(X\beta) = g^{-1}(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n) \quad (2)$$

where \mathbf{Y} is the dependent variable, $\mathbf{X}\beta$ is the linear predictor (a linear combination of unknown parameters β) and g is the link function allowing for non-linear relationship. \mathbf{X} and \mathbf{Y} are the data matrixes from (1) and the model estimates the β coefficients which are then used to predict \mathbf{Y} when pricing future policies. Machine learning techniques can be potentially employed to automate this process.

The model is deployed by expressing the Risk Premium as follows:

$$\text{Risk Premium} = \text{Base Premium} \times \beta_1 \times \beta_2 \times \dots \times \beta_n \quad (3)$$

where *Base Premium*

is a constant and $\beta_1, \beta_2, \dots, \beta_n$

are coefficients looked up depending on factor values for a particular policy.

3.2 Office Premium

Office Premium includes Risk Premium and loadings for expenses. Calculation of the expense loading component is straightforward as it simply adds all the assumed expenses to be loaded on top of the risk premium. Final premium can then be calculated as follows:

$$\text{Office Premium} = \frac{\text{Risk Premium} + \sum_{i=1}^n M_i}{1 - \sum_{i=1}^m P_i} \quad (4)$$

Where M_1, M_2, \dots, M_n would be loadings expressed as fixed monetary amounts and P_1, P_2, \dots, P_m loadings expressed as percentages of Office Premium, such as:

- Reinsurance cost
- Claim handling costs
- Other administration costs
- Commission (if policy is sold through an intermediary)
- Cost of capital (to generate return to shareholders)

Another important component in price calculation is the measure of exposure, i.e. the unit of cover for which we are calculating the price. Assuming that usual duration of the policy would be one year and risk would be evenly distributed throughout the year, the price could be calculated for one year of cover. If in some cases a different cover period would be required the premium can be adjusted proportionally:

$$\text{Short Term Premium} = \text{Annual Premium} \times e$$

$$\text{and } e = \frac{\text{days of policy duration}}{365,25} \quad (5)$$

In case of premature policy cancellation the premium for the remaining period can be calculated similarly and returned to the policyholder.

4. Smart Claims

There are several options for the basis on which policies can be written:

- Claims Occurring Basis (claim event date must be within policy duration period for the claim to be valid)
- Claims Made Basis (claim should be notified within policy duration period for the claim to be valid)

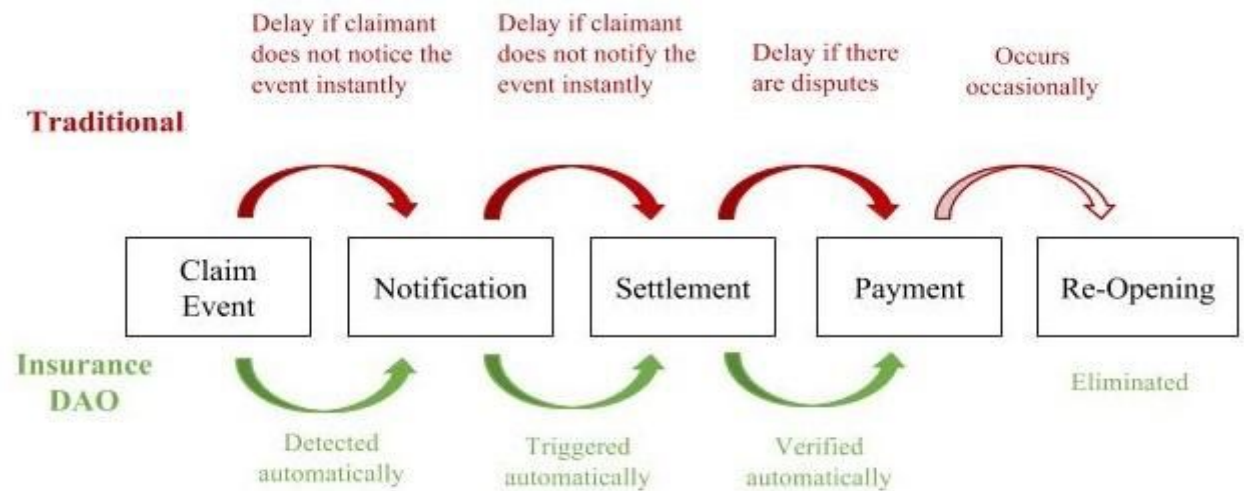
In traditional insurance both options have their advantages and disadvantages and the basis is chosen depending on the cover provided. For example, in some insurance classes the claim event date may not be known or difficult to verify which makes a claims occurring basis hard to apply and a claims made basis can be more practical. In the case of the insurance DAO it should also depend on the cover provided but since claim event detection and notification will be automated the difference between the two approaches may be less relevant than in traditional insurance.

Usually a claim process consists of the following stages:

- (1) **Claim Event.** Occurrence of the actual damage event. In traditional insurance the duty to detect the event lies on the insured person who may not be aware of the claim event yet and notices it only after some time, which creates delay. In the insurance DAO it would be possible to automate event detection, e.g. autonomous vehicle crash or device malfunction could be automatically detected and pushed to the blockchain network.
- (2) **Notification.** Insured person notifies the insurance company of the claim event. Again the insured may not report the claim to the insurer at the same moment if he has a more important business to deal with or feels that the claim is too small and not important. In the insurance DAO notification would be instantaneous for the cases where the claim event is detected automatically. In some cases it may still rely on the policyholder but the following procedures should be triggered automatically in any case.
- (3) **Settlement.** Verification of claim validity and details required to determine the claim amount. In traditional insurance this stage relies heavily on human factors. Various conditions can be interpreted subjectively and cause disputes which may lead to court. If the insurance cover can be standardised in such a way that claims can be validated automatically by the software within the IoT device and third party integrations, claims settlement can be fully objective and autonomous in the insurance DAO.
- (4) **Payment.** In most cases this is the final stage of the claims process. It can be either actual payment or replacement of the damaged item. In an insurance DAO the software code can make the money transfer or trigger the repair process automatically.
- (5) **Re-Opening.** Occurs occasionally if new information comes to light which requires adjustment of the settled claim. Re-opening of a claim inherently indicates a dispute or complexity of the claim so in the case of the insurance DAO the process should be optimised in such a way that reopening is not possible or such claims are ceded to a reinsurer.

Data sources could connect to the blockchain (using oracles) if required to verify claim validity or other details at any stage.

Figure 2. Traditional vs. DAO claim process.



In traditional insurance stages (1), (2) and (3) are prone to significant delays which can take from weeks to months or years. These delays create the need for the traditional insurance company to hold reserves (estimates for the claims that are not yet reported or fully settled) which involve complex calculations and create uncertainty in the company's financial result.

In the case of the insurance DAO the process should be simplified and automated so that occurrence of a claim event can be automatically identified triggering the subsequent processes handled automatically by software embedded in the smart contract. Ideally we should be able to jump to the payment phase instantly or if there are cases where the algorithm detects that the claim is more complex and cannot be handled instantly they could be ceded automatically to the reinsurer for further processing.

5. Profit Distribution

The simplified P&L for a smart contract DAO could be represented as follows:

	Premiums Earned
-	Claims Paid
-	Reinsurance Costs
-	Expenses Paid
<hr/>	
=	Total Profit

Full insurance company P&L has two more parts – changes in claims reserves and investment income. Claim reserves arise due to delays in various stages of the claims process. As noted in the claims section this paper assumes that the claims process can be automated to such extent that reserves can be ignored.

Investment income arises because insurance premiums are collected in advance and claims are paid out later thus creating a cashflow which can be invested to generate extra return. In such case the P&L should also include investment income which is added to the pure underwriting result above to get the total profit attributable to shareholders. In the case of the insurance DAO this may be seen as an unnecessary complication introducing additional risks related to investment and liquidity.

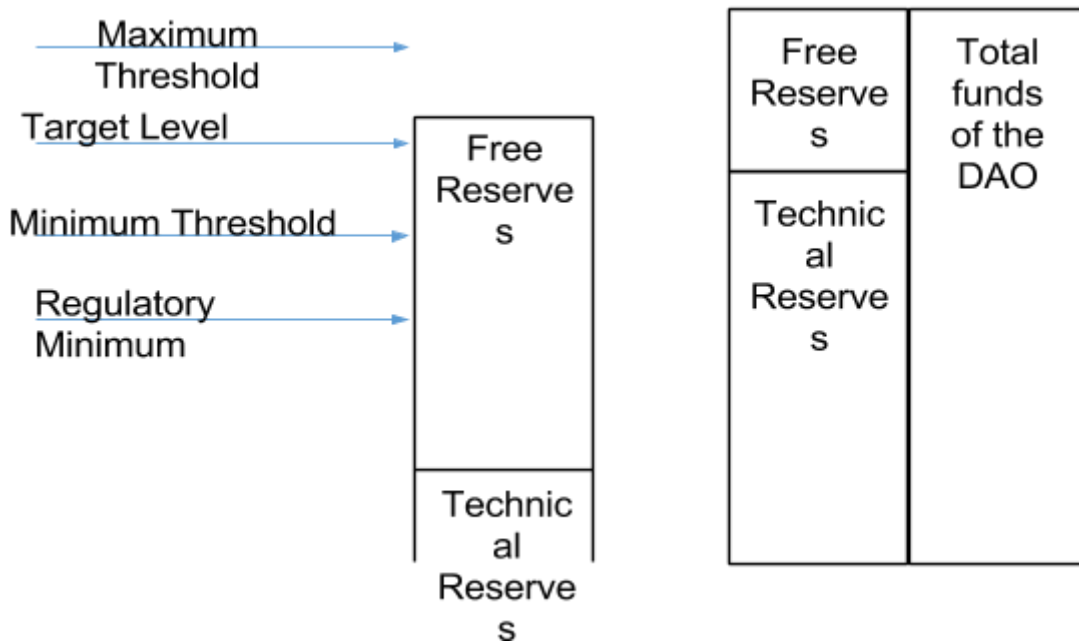
At any point in time the total funds available to the DAO can be divided into two parts:

Technical Reserves – amount required to pay claims for unexpired policies and expenses. This represents the DAO's liabilities to policyholders and other parties.

Free Reserves – amount above technical reserves which is used when technical reserves turn out to be insufficient. This is the shareholder capital.

P&L is calculated assuming some accounting period (e.g. a year) in which case the profit would be released from the technical reserves and transferred to the free reserves at the end of the period. In the case of the insurance DAO this would happen continuously. The blockchain would always know the capital position by continuously updating the values of the technical reserves and the free reserves.

To ensure solvency of the insurance DAO the free reserves must be maintained to be no less than a certain level, e.g. under the Solvency II regime in Europe the required level is such that the probability of default (when technical reserves plus free reserves are insufficient to meet the liabilities) would not be higher than 0.5%. Normally, companies hold free reserves well above the regulatory minimum, which is defined by a target level, e.g. 200% of the regulatory minimum.



Under normal circumstances the level of free reserves would fluctuate around the target level. This can be managed by setting the Minimum Threshold and Maximum Threshold levels.

If the free reserves decrease to breach the minimum threshold additional capital can be raised by issuing shareholder tokens through the investment platform. Alternatively the capital position can be improved by issuing debt capital. Also the system can be programmed to cease issuing new policies in such an event until the target capital level is restored.

If the free reserves breach the maximum threshold the excess capital can be distributed as dividends to shareholders bringing the free reserves down to target level, i.e.

$$Total Dividend = Maximum Threshold - Target Level \quad (6)$$

The dividend would be distributed to the shareholders in proportion to their holding:

$$\pi_i = \frac{Shares\ held\ by\ investor\ i}{Total\ shares} \quad (7)$$

$$Total Dividend (TD) \times \pi \rightarrow \begin{pmatrix} TD \times \pi_1 \\ TD \times \pi_2 \\ \vdots \\ TD \times \pi_n \end{pmatrix} \rightarrow \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \quad (8)$$

The risk carried by the shareholders (the fluctuations in the level of free reserves) can be reduced through reinsurance arrangements.

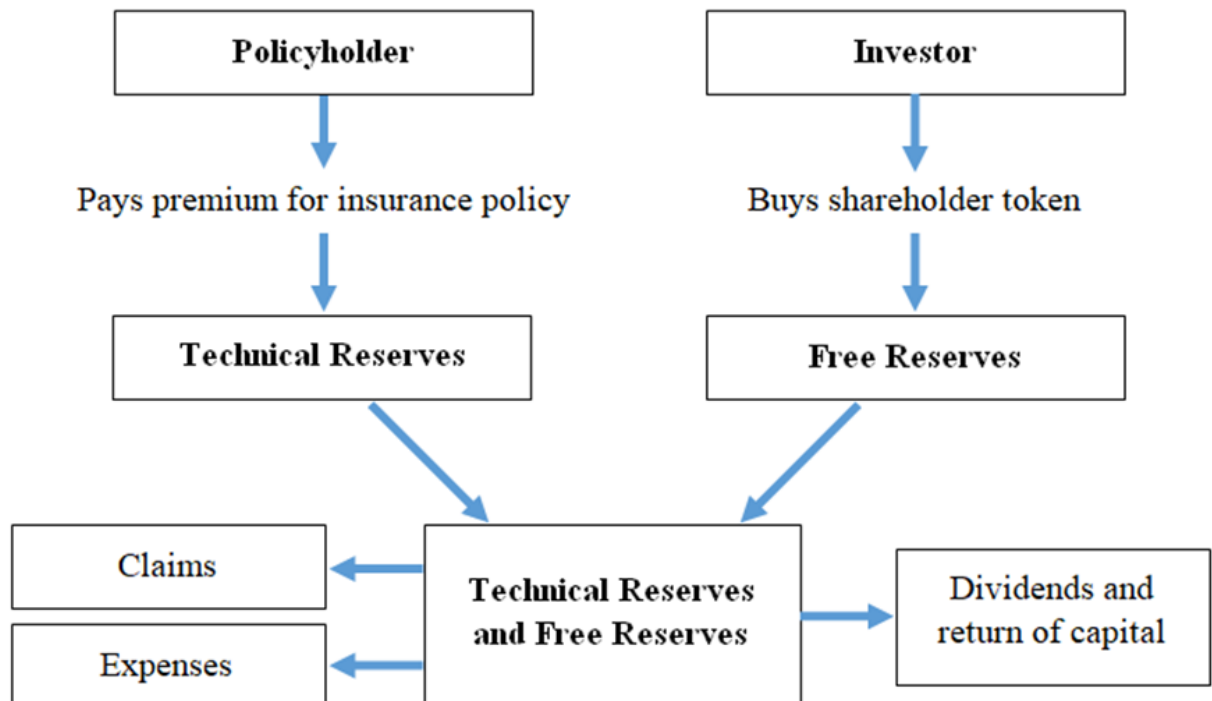
6. Investing in DAO Insurance and Token Model

This section provides an explanation and simplified example how the proposed framework will work in practice as a peer to peer insurance platform with two profiles – policyholder and investor.

Insurance will be organised in separate pools. Each pool will represent an issue of some insurance product for a certain predefined amount of premium, e.g. \$1,000,000 premium from phone battery insurance. Investors will be able to diversify their portfolio by investing in pools of different products with different risk levels and potential profit, e.g. phone battery, drone, smart car, etc.

Policyholders pay premiums and receive claims in case of damage but they carry no risk for the performance of the pool as a whole. Therefore additional funds are required from investors to form the free reserves. As premiums broadly represent the cost of average expected outcome having additional funds in the form of free reserves ensures that even if the outcome turns out worse than expected the pool remains solvent with high probability, e.g. 99.5%. As investors provide these additional funds which are required in case experience is different than expected, they carry the risk and for that receive a return.

Each pool will function according to the following scheme:



Let's take for example a \$1,000,000 pool for phone battery insurance (e.g. 100,000 policies \$10 each).

If target level of free reserves is 20% of premium then investors need to provide \$200,000. To gather these funds 200,000 shareholder tokens are issued through the investor platform with \$1 nominal. When the funds are collected the pool can start to issue policies.

Every insurance product at a different time will represent separate pool. The issued token will represent the USD value of that pool, as insurance statistical models cannot be calculated and won't work with volatile currencies. Everytime new pool is created, investors are invited to send their investments directly to specialized smart contract and get tokens in return. Ownership of these various Aigang platform issued tokens represents your right to claim profits from the pool at the profit sharing period and claim invested amount after the whole pool insurance period ends.

Assuming for simplicity that all policies have a duration of one year and start at once. Also assume that after half a year free reserves reach a certain threshold which triggers release of free reserves above target level. In reality this calculation will be performed continuously and release of free reserves can happen at any time depending on portfolio performance.

Let's say that premiums were set according to the following assumptions:

- 80% loss ratio
- 18% expense ratio
- 2% cost of capital (targeting 10% return of \$200,000 invested capital which gives 2% of \$1,000,000 premium)
- Actual claims and expenses turned out as expected

After half a year \$500,000 of premium will be earned and remaining \$500,000 held as technical reserves. Also half of claims and expenses will be paid. If 2% cost of capital was loaded in the premium and experience was as expected 10% return can be paid as dividends. Additionally only \$500,000 of premium is remaining to be earned and only \$100,000 capital is required to support this pool, so remaining \$100,000 of capital can also be returned to investors resulting in \$110,000 funds returned.

After one year all policies have expired and remaining free reserves are returned to shareholders.

Invested amount	\$200,000	\$1 per share
Premiums paid by policyholders	\$1,000,000	
Funds at outset	\$1,200,000	
Claims during half year	(\$400,000)	
Expenses during half year	(\$90,000)	
Funds at half year	\$710,000	
<i>Unearned premiums (Technical Reserve)</i>	<i>\$500,000</i>	
<i>Free Reserve</i>	<i>\$210,000</i>	
<i>Target Free Reserve</i>	<i>\$100,000</i>	
Funds returned to investors	(\$110,000)	\$0.55 per share
Funds after distribution	\$600,000	
Claims during half year	(\$400,000)	
Expenses during half year	(\$90,000)	
Funds at end of year	\$110,000	
Funds returned to investors	(\$110,000)	\$0.55 per share
Total funds returned to investors	\$220,000	\$1.1 per share
Return	10%	

In reality not all policies will be issued at once, so the duration of the pool before all free reserves are released will be longer than one year.

A secondary market will be available for investors who wish to trade their tokens earlier. All insurance investors will have their tokens assigned to their wallet. As soon as all funds for the pool are collected, tokens become tradeable. Meaning that in case they need money earlier, or they do not want to take risk, they can sell their tokens to others investors directly on the price they agree. This enables people to invest in various insurance products reserves and not get stuck, but be able anytime to sell its tokens directly via blockchain without any intermediary.

Combining the proposed peer to peer framework and the secondary market for investors creates a new asset class which was previously available only for large investors investing in traditional insurance companies. This can transform insurance in a similar way as peer to peer online platforms transformed the investments from retail investors into consumer and business loans.

7. Peer to peer investment platform

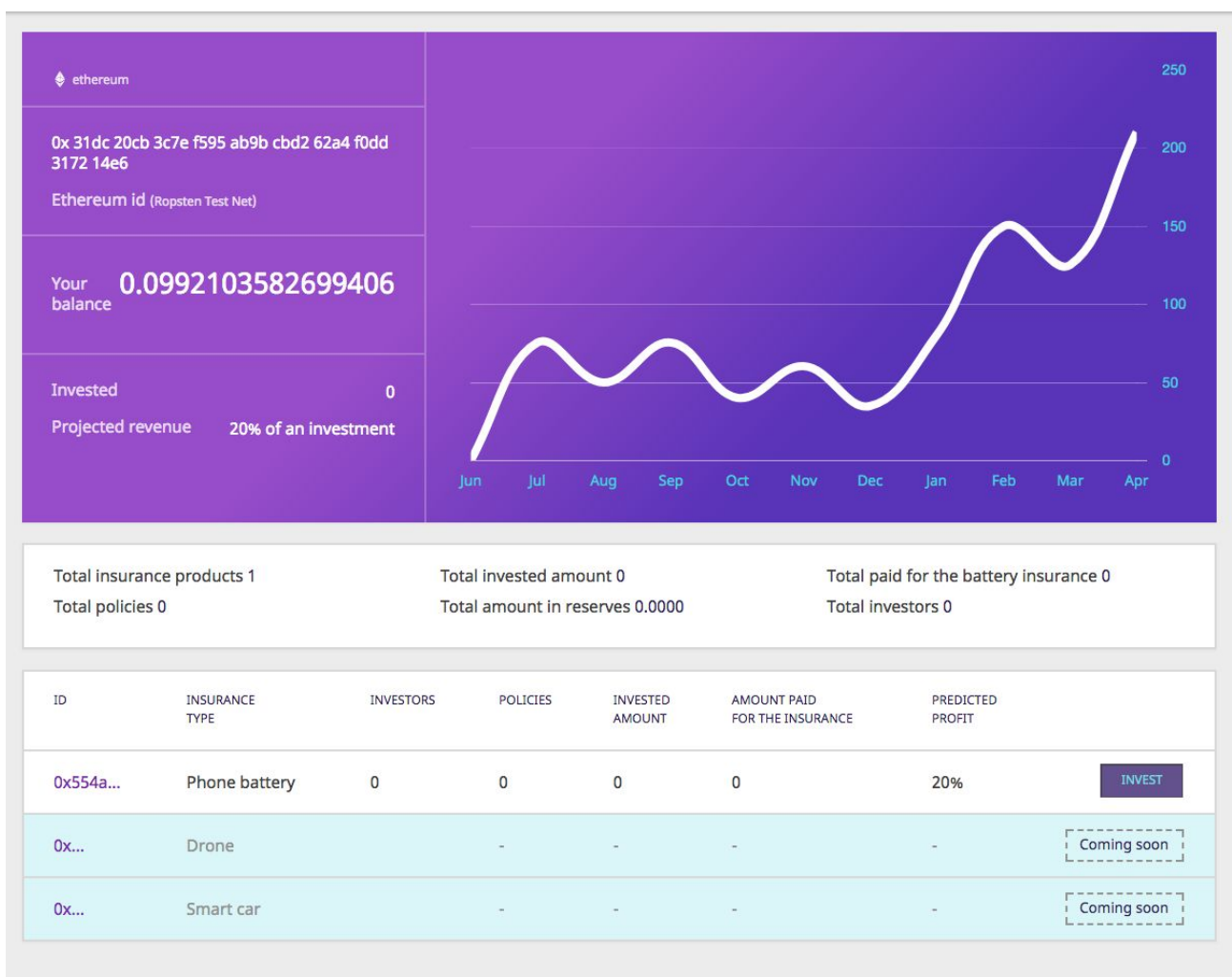
The peer to peer investment platform will be developed by Aigang team. The key targets for this platform is:

1. Perform calculation and build statistical models for needed reserves and risk assessment.
2. Tokenize the reserves.
3. List all insurance product pools with assessed risk, predicted profit, etc.
4. Enable investors purchase tokens for various insurance product pools.
5. Display whole investor profile: funds, projected profits, risk and the status of each pool.
6. Enable secondary market trading.
7. Collect and monitor premiums, process claims and payouts.

This platform will be the core of our protocol, as it will enable full peer to peer insurance vision. It will cover many complex insurance parts and will help us to automate the whole process. Early alpha can be accessed here - <https://investment.aigang.network/>



Investment_{Demo}



8. Device data tracking and claim process

Current insurance industry lacks accurate data about the devices they insure. Everything is based on trust of people and some general lookup of the damaged item. This leads to a huge frauds in the whole industry. Based on Insurance Information Institute calculations, property/casualty fraud amounted to about \$34 billion each year. [\[3\]](#) And this amount is increasing as more and more expensive items are being insured. Here we saw a huge untapped potential to build a completely new insurance category for IoT devices based on the data they gather. The software will be built for devices or product makers API's will be used to retrieve data about the device state. The whole process could be divided into 3 parts:

1. Data collection and monitoring. We will constantly monitor IoT device health and its surroundings based on sensors in the device. The data will be pulled to our backend servers for storing and processing. This will enable our algorithms to have a historical view of the device and act accordingly on the actual data.
2. Event triggering. In case of an accident, device will automatically register the event or if that's not possible (limited by device software), policyholder will trigger the event manually.
3. Data validation. On the event, our system will pull the data from device and gather the state and sensor data at the time of the event and register the claim.

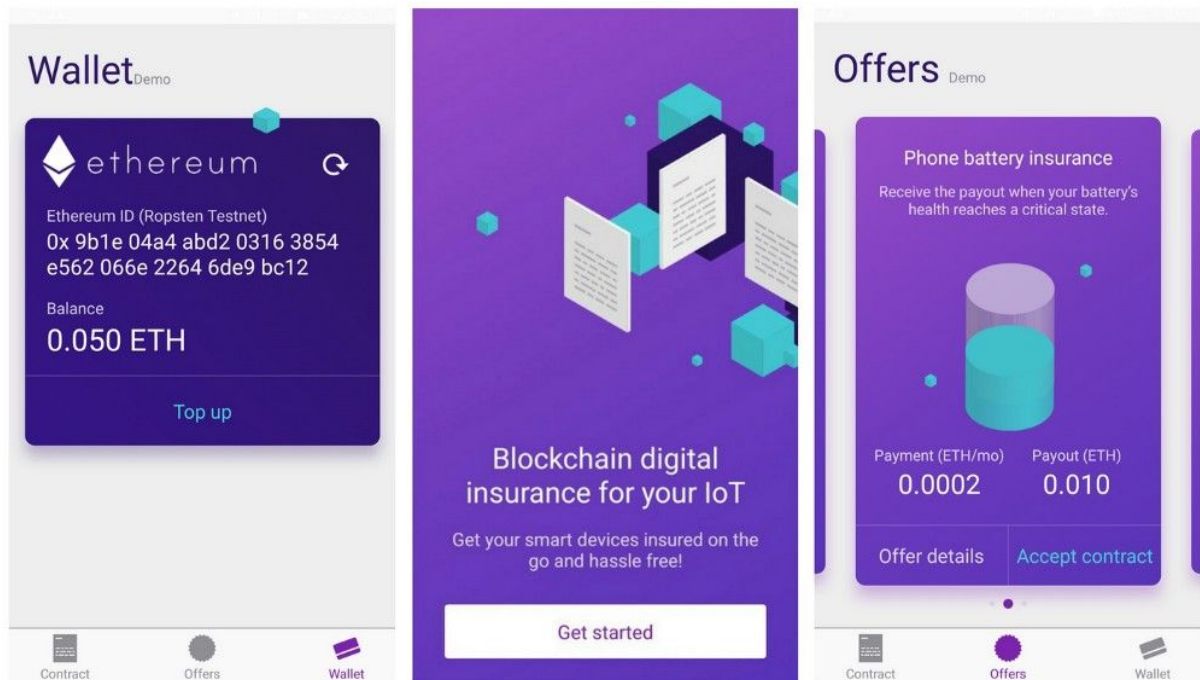
Using historical data, machine learning and the data at the time of the event, our algorithms will be able to detect the damage, calculate the payout and finalize the claim. We completely understand that there are possibilities for hacking, providing fake data and technical limitations in case devices are completely dead. Therefore, we believe, that our system will minimize the risk of frauds using historical data of the device which is hard to be tampered, and statistical models that will make insurance profitable for investors. In the beginning there will be a need to have a support team which would solve edge cases, monitor frauds till the protocol can be moved to fully autonomous insurance driven by AI and blockchain.

9. Aigang Protocol Showcase: Functional Battery Insurance for Smartphones

For a showcase of blockchain protocol for digital insurance, we selected the one of the most common IoT devices - smartphone. Nowadays, they offer a variety of built in sensors that could be accessed by a software. One of the most common technical issue, smartphone owners face, is battery malfunction. And manufacturers offering shorter warranty periods, force the owners to replace the their phone batteries. This lead Aigang to build its MVP (minimum viable product) and solve this issue by offering such insurance. Since it is possible to understand the state of the battery by accessing the smartphones sensors, the gradual degradation can be detected and risk could be assessed based on the real-time data. Once the battery reaches a critical stake (a certain threshold set) the payout is automatically processed and executed by smart contracts.

It is possible to fully automate such a process by storing it on blockchain using smart contracts and validate payouts. Judging the current state of the battery, Aigang assesses different risk levels with different payment plans. If the smartphone already requires a new battery - it does not allow to insure that device. This solidifies a process that can be fair to both parties - insurer and insuree. Currently, MVP functions on Ethereum Testnet and only accepts payments through Ropsten Testnet. However, it is planned to transfer to Ethereum Mainnet and allow the commercial use of such insurance.

This is how the Aigang MVP looks like to the end user.



10. Example of Free Reserve Calculation

Continuing with our phone battery insurance example, let's assume that we have a pool of n policies, each paying out a claim of \$100 in case of battery failure. The number of claims on policy i is a random variable X_i . Strictly for this example X_i is likely to be either 0 in case of no claims or 1 in case of a claim. This would suggest the Bernoulli distribution for X_i and Binomial distribution for the aggregate number of claims. However for the results to be extendable beyond this simple example let's assume the Poisson distribution for X_i , which allows more than one claim per policy and is widely used for modelling claim frequency in general insurance:

$$X_i \sim \text{Pois}(\lambda), i = 1, \dots, n$$

Where X_i are independent random variables and λ is the intensity parameter of the Poisson distribution representing the average claims frequency. The aggregate number of claims Z would then be:

$$Z = \sum_{i=1}^n X_i \sim \text{Pois}(n\lambda)$$

Which also has a Poisson distribution but with n times higher intensity. To simulate the results over time we also need to add the time dimension which brings us to the Poisson process $Z(t)$ which is a collection of Poisson random variables representing the aggregate number of claims up to each moment in time t . It is known that:

$$Z(t) \sim \text{Pois}(n\lambda t)$$

and the waiting times between claims have the exponential distribution:

$$T_j \sim \text{Exp}(n\lambda), j = 1, 2, 3, \dots$$

Where T_j is the time between claim $j-1$ and j . This gives us a stochastic model to simulate the performance of the pool over time. The simulation is performed by generating the waiting times T as exponential random variables and then deriving the aggregate number of claims:

$$Z(t) = k, \text{ when } T_1 + \dots + T_k \leq t < T_1 + \dots + T_{k+1}$$

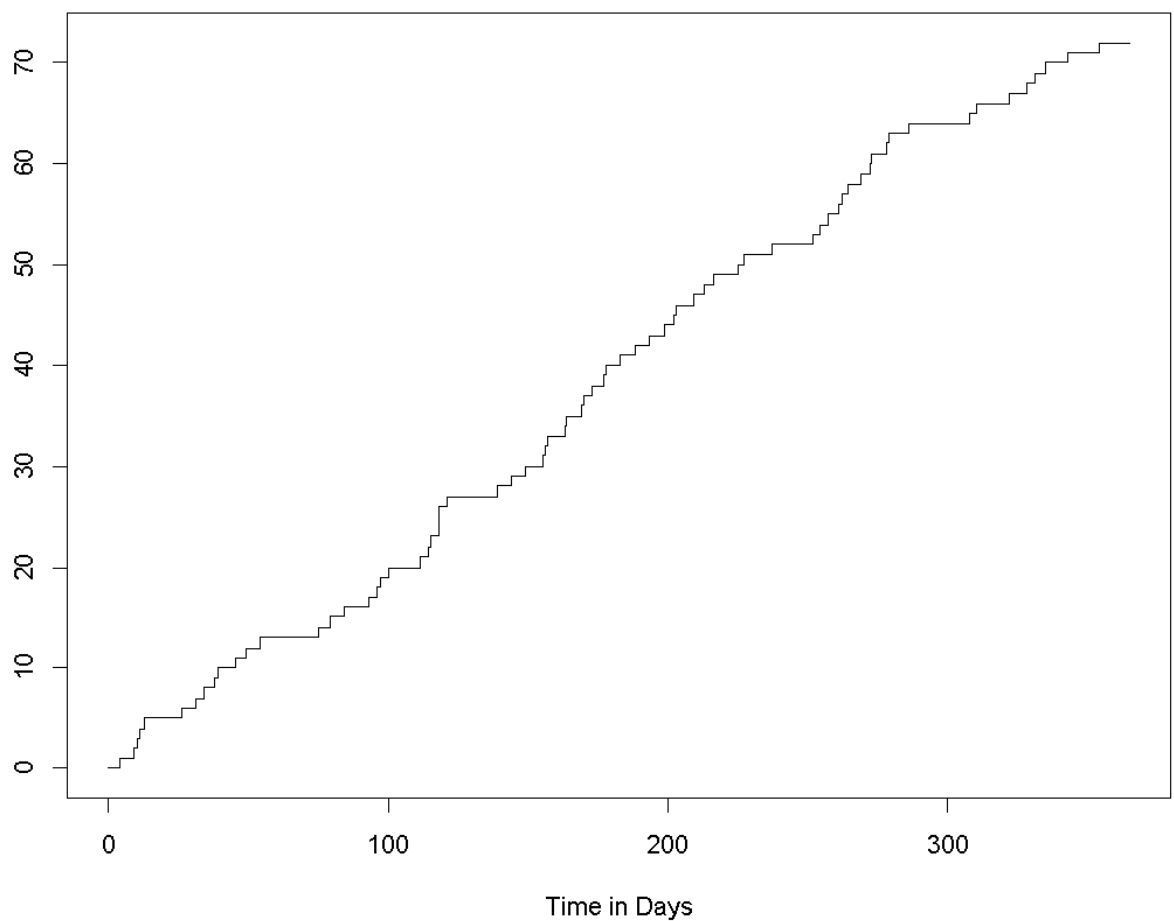
Parameters for simulation:

Number of policies *	$n = 1,000$
Premium per policy	\$10
Size of claim	\$100
Claim frequency	$\lambda = 0.08$
Loss ratio assumed in premium	80%
Expense loading in premium	18%
Cost of capital loading in premium	2%

* Number of policies taken here is smaller than in previous example so that random features are more visible in the graphs. However all the results can be easily scaled up proportionally to a larger portfolio of \$1,000,000.

It is assumed for simplicity that all n policies start at the same time $t = 0$ and have a duration of 1 year. A single simulation of the aggregate claim number performed with R is presented in the following graph, where each step of the process path represents a claim occurring at that particular time:

Aggregate Number of Claims



Multiplying the aggregate number of claims by the claim amount we get the aggregate amount of claims:

$$C(t) = \$100 * Z(t)$$

Earned premium is calculated as follows:

$$PE(t) = \$10 * n * t/365$$

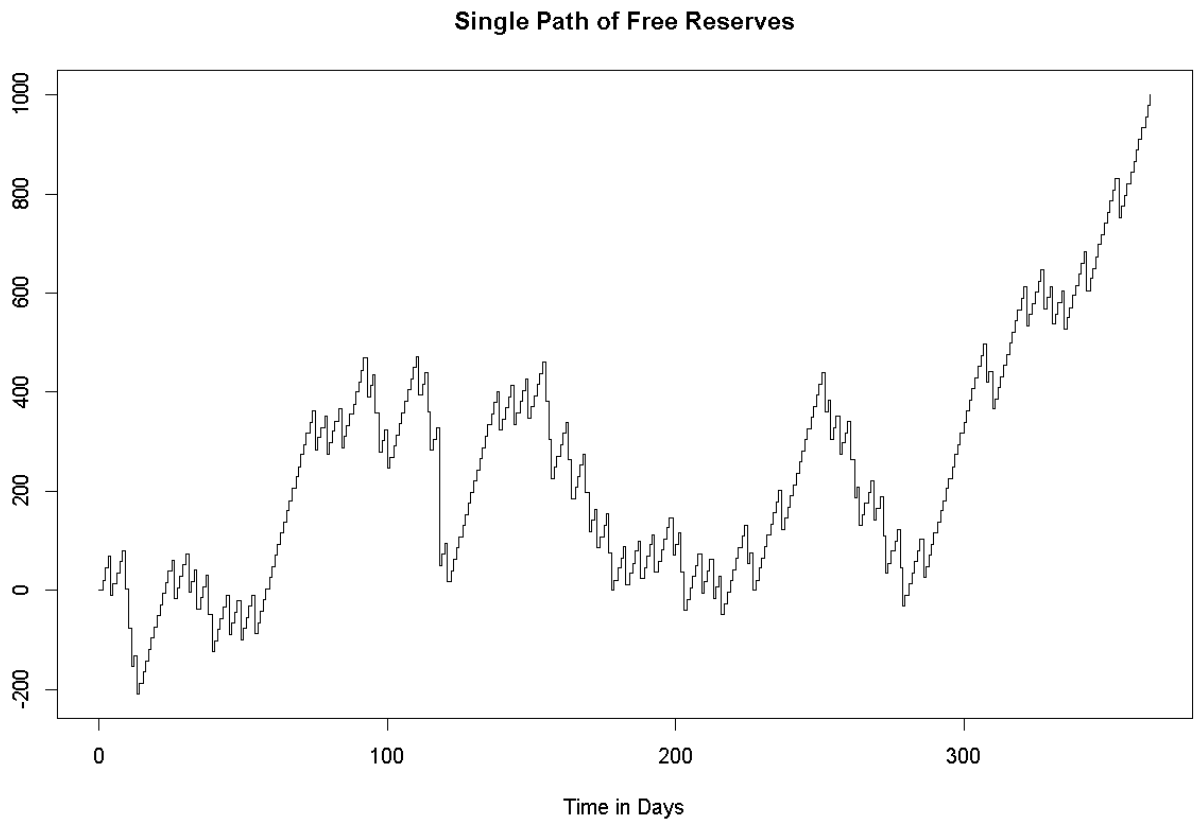
Expenses are assumed to be incurred uniformly, i.e. as premium is earned:

$$E(t) = 0.18 * PE(t)$$

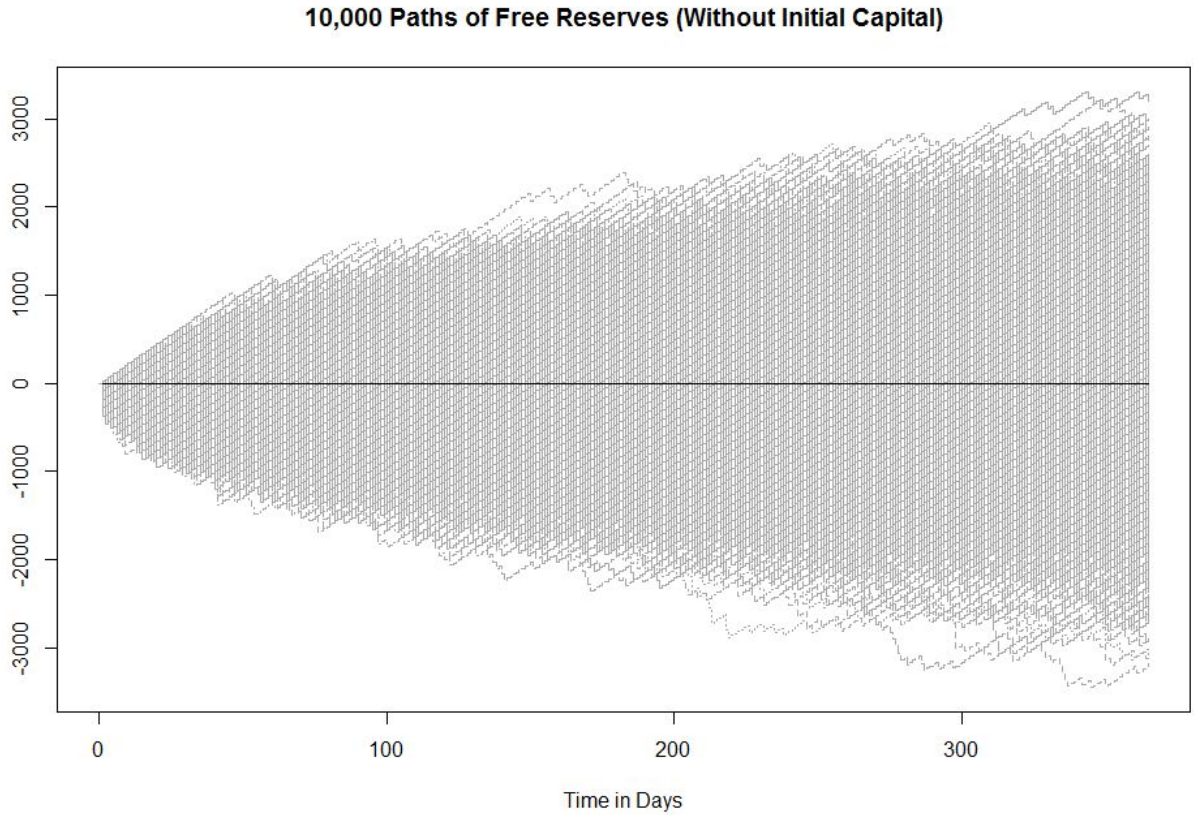
The level of Free Reserves is calculated as follows:

$$FR(t) = PE(t) - E(t) - C(t)$$

A typical path of the level of free reserves is provided in the following graph of R simulation where the path goes up as premiums are earned and down as claims are paid:



This is just a single simulation – one of many possible paths that the free reserves may follow. Full distribution can be obtained by running multiple simulations of the same process. The following graph illustrates 10,000 paths of free reserves simulated in R:



If at any time t the level of free reserves hits 0 the pool becomes insolvent. In the above graphs we observe that this happens a number of times. The key question is what initial level of free reserves $FR(0)$ is required to ensure the probability of ruin is low, e.g. 0.5% which is required by Solvency 2 regime in Europe. The level of free reserves then becomes:

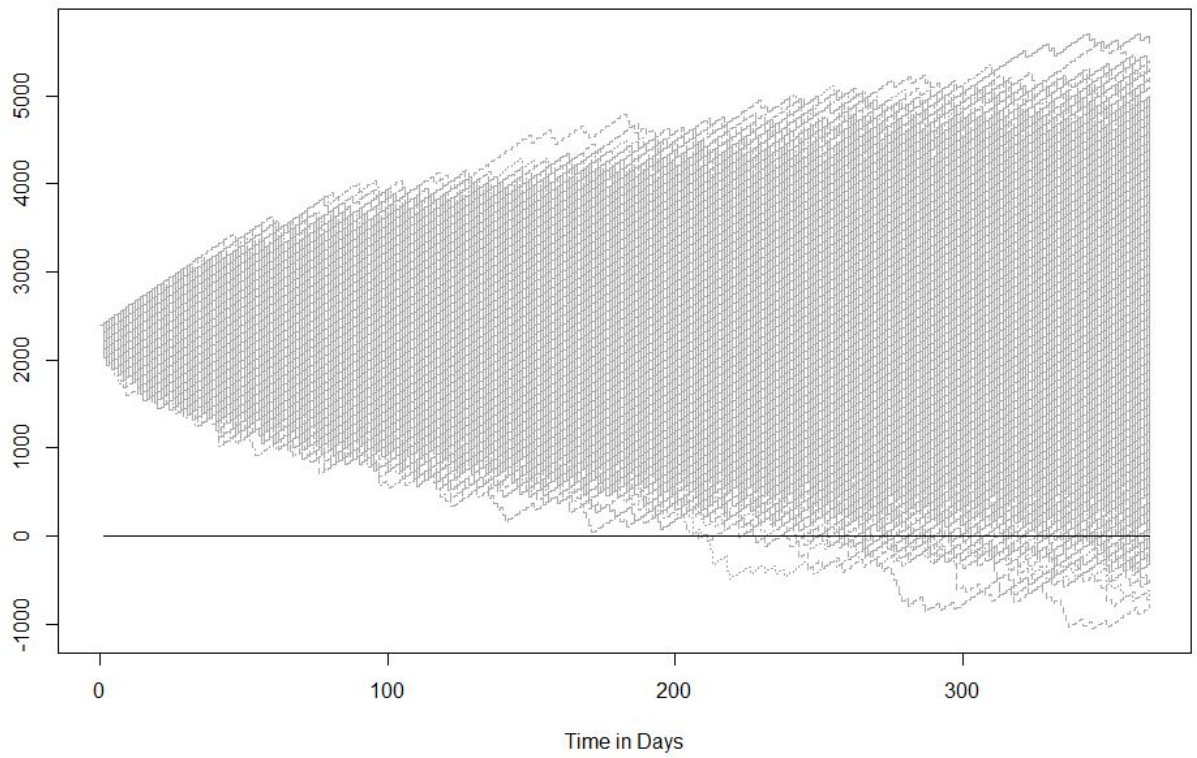
$$FR(t) = FR(0) + PE(t) - E(t) - C(t)$$

Where $FR(0)$ is the initial level of free reserves provided by investors. We need to find the value of $FR(0)$ such that:

$$P(FR(t) < 0; 0 \leq t \leq 365) < 0.5\%$$

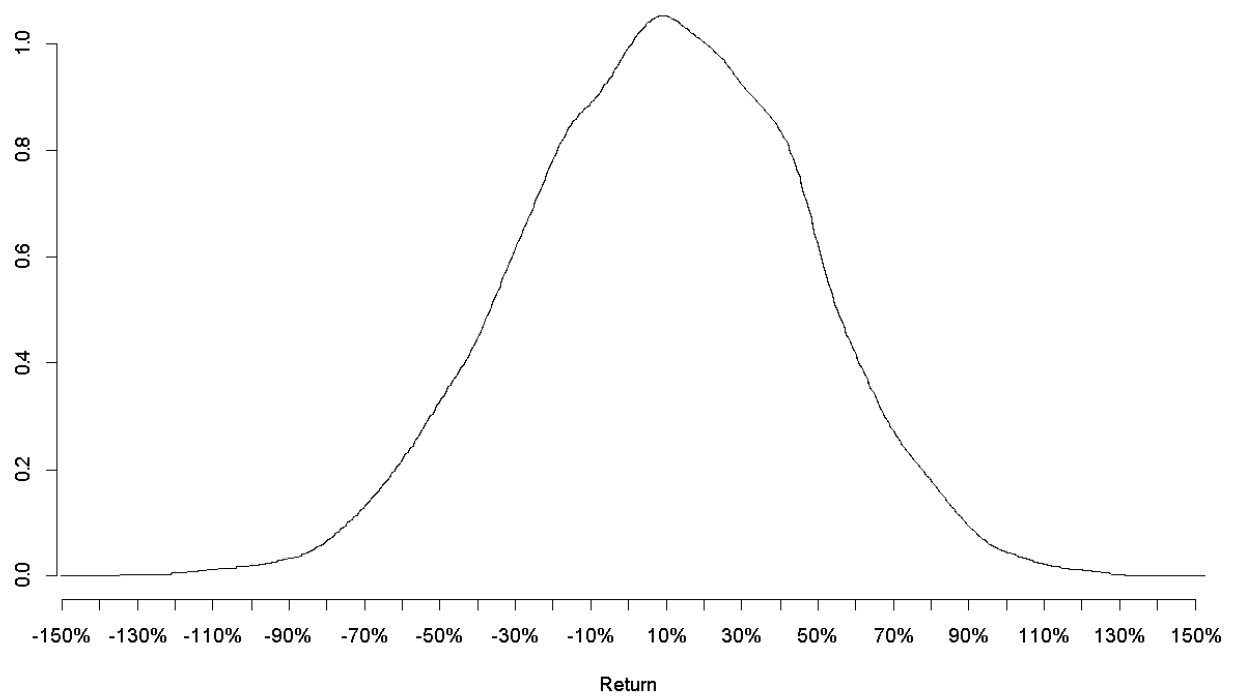
It can be determined in R by increasing $FR(0)$ iteratively until probability becomes less than 0.5%. For this example it yields \$2400 (or \$240,000 for a portfolio of \$1,000,000), i.e. 24% of written premium. With this initial capital the process paths generated in R are shifted accordingly so that only very few of them hit 0.

10,000 Paths of Free Reserves (With Initial Capital)



Return to shareholders is distributed as follows with mean return of 8.7%.

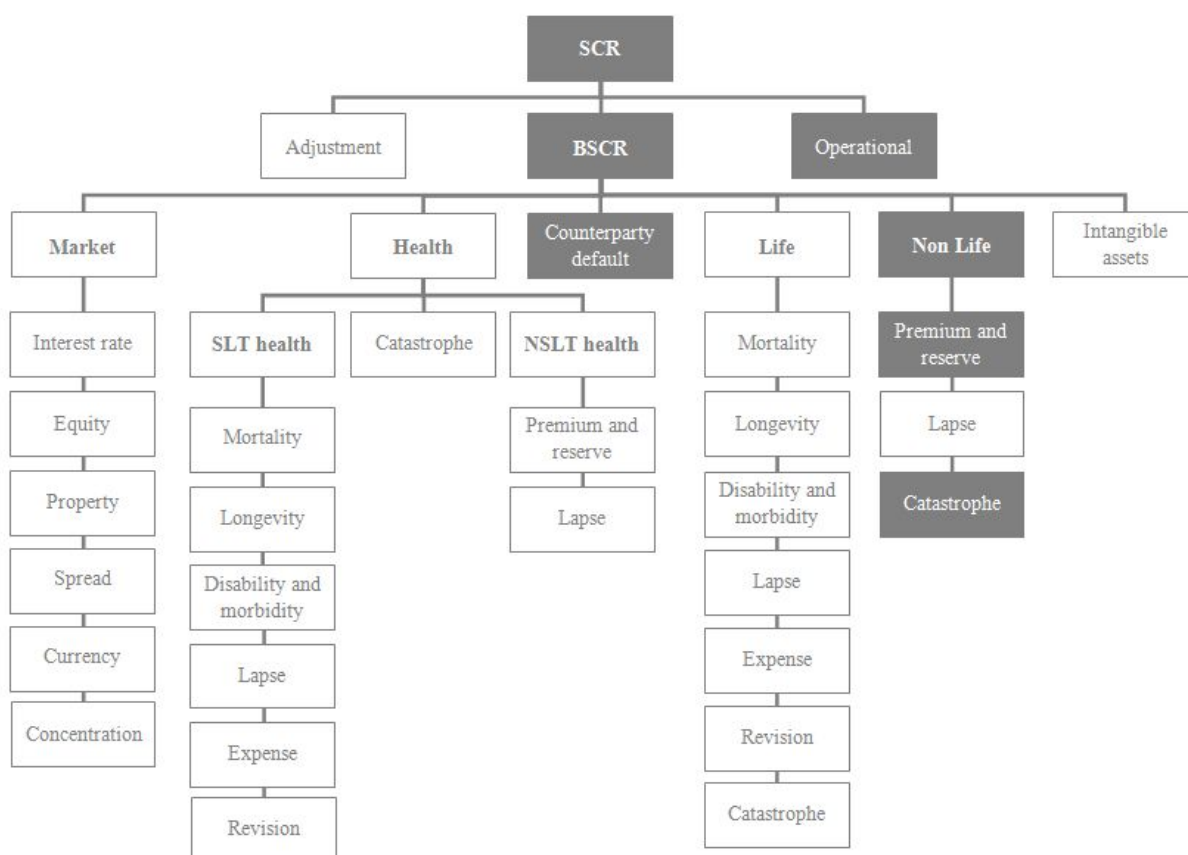
Probability Density of Return



11. Solvency II Standard Formula for Insurance DAO

In this section we provide an example calculation of the capital (free reserve) requirement that may be applied to an automated insurance DAO under Solvency II regime which governs solvency requirements for insurance companies in Europe. In other territories different requirements would apply, however Solvency II was chosen for this example due to its risk based approach which may provide valuable guidance even if the DAO would not be governed by this particular regime. There are two alternative ways to calculate the capital requirement under Solvency II – follow the standard formula or develop own internal model approved by the regulator. Since the latter alternative is more complicated, resource consuming and thus only feasible for very large undertakings, we focus here on the standard formula.

Solvency capital requirement (hereafter – SCR) or economic capital to be held by insurance undertakings is determined as to ensure that ruin occurs no more often than once in every 200 cases or, alternatively, that those undertakings will still be in a position, with a probability of at least 99.5%, to meet their obligations to policyholders and beneficiaries over the following 12 months. In general case according to standard formula SCR composition is the following:



However, in the case of automated insurance DAO the structure of SCR becomes far simpler as only some components apply (highlighted in the above chart). Namely 3 risk modules need to be assessed: Non-life underwriting risk, Counterparty default risk and Operational risk. The following assumptions were made for simplicity and consistency with other sections of this white paper:

- insurance premiums are paid at inception, there are no due payments from the policyholders;
- insurance cover begins the same day as insurance premium is paid;
- all insurance premiums collected are held as cash at bank;
- there is no reinsurance in place;
- premiums, claims and assets are all in the same currency;
- interest rate risk due to technical reserve discounting is negligible;
- there are no deferred taxes or intangible assets.

In this particular case SCR would consist of Operational risk charge and Basic Solvency Capital Requirement (BSCR), which comprises Non-life underwriting as well as Counterparty default risk modules. Those risk modules are aggregated using correlation coefficients according to the following formula:

$$SCR = \sqrt{SCR_{non-life}^2 + 2 * 0.5 * SCR_{non-life} * SCR_{default} + SCR_{default}^2} + SCR_{operational} \quad (1)$$

11.1 Non-Life Underwriting Risk

The main risk driver in the SCR is the non-life underwriting risk which in turn is equal to the following:

$$SCR_{non-life} = \sqrt{\sum_{i,j} CorrNL_{(i,j)} * SCR_i * SCR_j} \quad (2)$$

where:

- SCR_i and SCR_j denote the capital requirements for risk sub-modules i and j respectively;
- $CorrNL_{(i,j)}$ denotes the correlation parameter for non-life underwriting risk for sub-modules i and j of the following correlation matrix:

i	j	Non-life premium and reserve	Non-life catastrophe
Non-life premium and reserve		1	0.25
Non-life catastrophe		0.25	1

11.2 Premium and Reserve Risk

The capital requirement for non-life premium and reserve risk is equal to the following:

$$SCR_{nl\,prem\,res} = 3 * \sigma_{nl} * V_{nl} \quad (3)$$

where:

- σ_{nl} denotes the standard deviation for non-life premium and reserve risk;
- V_{nl} denotes the volume measure for non-life premium and reserve risk.

The volume measure for non-life premium and reserve risk is equal to the sum of the volume measures for premium and reserve risk. In general case for all segments the volume measure of a particular segment s is equal to the following:

$$V_s = (V_{(prem,s)} + V_{(res,s)}) * (0.75 + 0.25 * DIV_s) \quad (4)$$

where:

- $V_{(prem,s)}$ denotes the volume measure for premium risk of segment s ;
- $V_{(res,s)}$ denotes the volume measure for reserve risk of segment s ;

- DIV_s denotes the factor for geographical diversification of segment s .

However, since in our automated insurance there are no claims reserves, reserve risk component is eliminated. Also DIV_s is assumed to equal 1 for simplicity. Therefore calculation of volume measure becomes straightforward:

$$V_s = V_{(prem,s)} \quad (5)$$

It should be mentioned that insurance undertakings segment their insurance obligations into homogeneous risk groups, as a minimum by lines of business. There are totally 12 lines of business defined for direct non-life insurance. For the non-life underwriting risk calculation each of those segments has particular premium and reserve risk standard deviation prescribed. These segments are denoted s in the formulas above. Segmentation of non-life obligations and corresponding standard deviations for premium risk are provided in the table below:

Segment	Standard Deviation for Premium Risk
Motor vehicle liability insurance	10%
Other motor insurance	8%
Marine, aviation and transport insurance	15%
Fire and other damage to property insurance	8%
General liability insurance	14%
Credit and suretyship insurance	12%
Legal expenses insurance	7%
Assistance	9%
Miscellaneous financial loss insurance	13%

In this example we assume that phone battery insurance falls under miscellaneous financial loss segment.

The volume measure for premium risk of a particular segment s equals to the following:

$$V_{(prem,s)} = \max[P_s; P_{(last,s)}] + FP_{(existing,s)} + FP_{(future,s)} \quad (6)$$

where:

- P_s denotes an estimate of the premiums to be earned by the insurance undertaking in the segment s during the following 12 months;

- $P_{(last,s)}$ denotes the premiums earned by the insurance undertaking in the segment s during the last 12 months;
- $FP_{(existing,s)}$ denotes the expected present value of premiums to be earned by the insurance undertaking in the segment s in the following 12 months for existing contracts;
- $FP_{(future,s)}$ denotes the expected present value of premiums to be earned by the insurance undertaking in the segment s for contracts where the initial recognition date falls in the following 12 months but excluding the premiums to be earned during the 12 months after the initial recognition date.

For newly established insurance undertaking with growing premium volumes $P_{(last,s)}$ component would be irrelevant. $FP_{(future,s)}$ component can also be ignored as in our example only one year contracts are offered. If we were to make an assumption that all contracts start and premiums are paid at the beginning of the year and all contracts end at the end of the year, then the volume measure would be even simpler to calculate – it would equal premiums to be earned during the year. Therefore the formula can be simplified to the following:

$$V_{(prem,s)} = P_s \quad (7)$$

Another measure that is required for the capital requirement for non-life premium and reserve risk calculation is the standard deviation. It is equal to the following:

$$\sigma_{nl} = \frac{1}{V_{nl}} * \sqrt{\sum_{s,t} CorrS_{(s,t)} * \sigma_s * V_s * \sigma_t * V_t} \quad (8)$$

where:

- V_{nl} denotes the volume measure for non-life premium and reserve risk;
- the sum covers all possible combinations (s,t) of the segments;
- $CorrS_{(s,t)}$ denotes the correlation parameter for non-life premium and reserve risk for segment s and segment t ;
- σ_s and σ_t denote standard deviations for non-life premium and reserve risk of segments s and t respectively;
- V_s and V_t denote volume measures for premium and reserve risk of segments s and t respectively.

Since in our example there is only 1 segment and no volume measure for reserve risk, standard deviation for non-life premium and reserve risk calculation is straightforward:

$$\sigma_{nl} = \sigma_s \quad (9)$$

Finally, capital requirement for non-life premium and reserve risk is calculated as follows:

$$SCR_{nl\text{ } prem\text{ } res} = 3 * \sigma_s * P_s \quad (10)$$

Using the standard deviation parameter for miscellaneous financial loss segment, which is specified to be 13%, and \$1,000,000 premium in our example we get the following result:

$$SCR_{nl\text{ } prem\text{ } res} = 3 * 0.13 * 1,000,000 = \$390,000 \quad (11)$$

11.3 Catastrophe Risk

Previously discussed non-life premium and reserve risk is not the only component of non-life underwriting risk, catastrophe risk should also be assessed. In general case it might not be trivial but in our example neither natural catastrophe scenarios nor man-made scenarios are relevant. There is only factor-based formula for other non-life catastrophe risk, i.e.:

$$L_{other} = \sqrt{(c_1 * P_1 + c_2 * P_2)^2 + (c_3 * P_3)^2 + (c_4 * P_4)^2 + (c_5 * P_5)^2} \quad (12)$$

where:

- P_1, P_2, P_3, P_4 and P_5 denote estimates of the gross premium expected to be earned by the insurance undertaking during the following 12 months in relation to the groups of insurance obligations 1 to 5;
- c_1, c_2, c_3, c_4 and c_5 denote the risk factors for the groups of insurance obligations 1 to 5.

Miscellaneous financial loss line of business falls under group 3 with risk factor c_3 being equal to 40%, therefore:

$$SCR_{nl\text{ } CAT} = 0.4 * P \quad (13)$$

In our example capital requirement for the non-life catastrophe risk is as follows:

$$SCR_{nl\text{ } CAT} = 0.4 * 1,000,000 = \$400,000 \quad (14)$$

Aggregating premium and reserve risk as well as catastrophe risk component we get the final SCR for non-life underwriting risk. It can be shown that in our case (2) simplifies to the following:

$$SCR_{Non-life} = \sqrt{0.3901 * P_s^2} \quad (15)$$

Using the \$1,000,000 premium from our example we get:

$$SCR_{Non-life} = \sqrt{0.3901 * 1,000,000^2} = \$624,580 \quad (16)$$

It should be noted that due to diversification effect we get lower total non-life underwriting risk result compared to simple sum of premium and reserve and catastrophe risks, which would be equal to \$790.000.

11.4 Counterparty Default Risk

Another component of BSCR is the counterparty default risk which is calculated as follows:

$$SCR_{def} = \sqrt{SCR_{(def,1)}^2 + 1,5 * SCR_{(def,1)} * SCR_{(def,2)} + SCR_{(def,2)}^2} \quad (17)$$

where

- $SCR_{(def,1)}$ denotes the capital requirement for counterparty default risk on type 1 exposures (mainly those counterparties that are rated);
- $SCR_{(def,2)}$ denotes the capital requirement for counterparty default risk on type 2 exposures (mainly those counterparties that are numerous and not rated);

Since in our example the only counterparty default risk comes from cash at bank, which is considered type 1 exposure, the calculation becomes trivial:

$$SCR_{def} = SCR_{(def,1)} = x * \sigma \quad (18)$$

where:

- parameter x depends on the standard deviation of the loss distribution of type 1 exposures. If the standard deviation (σ) is lower than or equal to 7 % of the total losses-given-default on all type 1 exposures, parameter x equals to 3. In case the standard deviation (σ) is higher than 7 % and lower or equal to 20 % of the total losses-given-default on all type 1 exposures, parameter x equals 5;
- σ denotes the standard deviation of the loss distribution of type 1 exposures;
- where the standard deviation of the loss distribution of type 1 exposures is higher than 20 % of the total losses-given-default on all type 1 exposures, the capital requirement for counterparty default risk on type 1 exposures is equal to the total losses-given-default on all type 1 exposures.

The standard deviation of the loss distribution of type 1 exposures is equal to the following:

$$\sigma = \sqrt{V} \quad (19)$$

where V denotes the variance of the loss distribution of type 1 exposures, equal to the sum of V_{inter} and V_{intra} .

V_{inter} equals to the following:

$$V_{inter} = \sum_{(j,k)} \frac{PD_k * (1 - PD_k) * PD_j * (1 - PD_j)}{1.25 * (PD_k + PD_j) - PD_k * PD_j} * TLGD_k * TLGD_j \quad (20)$$

where:

- the sum covers all possible combinations (j,k) of different probabilities of default on single name exposures;
- $TLGD_j$ and $TLGD_k$ denote the sum of losses-given-default on type 1 exposures from counterparties bearing a probability of default PD_j and PD_k respectively.

Assuming that all cash is held at one bank, having rating j (e.g. A), V_{inter} calculation can be simplified as follows:

$$V_{inter} = \frac{PD_j^2 * (1 - PD_j)^2}{2.5 * PD_j - PD_j^2} * LGD_j^2 \quad (21)$$

V_{intra} equals to the following:

$$V_{intra} = \sum_j \frac{1.5 * PD_j * (1 - PD_j)}{2.5 - PD_j} * \sum_{PD_j} LGD_j^2 \quad (22)$$

where:

- the first sum covers all different probabilities of default on single name exposures;
- the second sum covers all single name exposures that have a probability of default equal to PD_j ;
- LGD_i denotes the loss-given-default on the single name exposure i .

Having only one single name exposure with rating j this can be simplified as follows:

$$V_{intra} = \frac{1.5 * PD_j * (1 - PD_j)}{2.5 - PD_j} * LGD_j^2 \quad (23)$$

The final formula for counterparty default risk is then as follows:

$$SCR_{def} = x * \sqrt{\frac{PD_j^2 * (1 - PD_j)^2}{2.5 * PD_j - PD_j^2} * LGD_j^2 + \frac{1.5 * PD_j * (1 - PD_j)}{2.5 - PD_j} * LGD_j^2} \quad (24)$$

Assuming that cash is held at bank having A rating and using the \$1,000,000 premium from our example, we get the following result for counterparty default risk:

$$SCR_{def} = 3 * \sqrt{\frac{0.0005^2 * 0.9995^2}{2.5 * 0.0005 - 0.0005^2} * 1,000,000^2 + \frac{1.5 * 0.0005 * 0.9995}{2.5 - 0.0005} * 1,000,000^2} = \$67,065 \quad (25)$$

11.5 Operational Risk

Finally the last SCR component to be evaluated is the operational risk. The capital requirement for the operational risk module is equal to the following:

$$SCR_{operational} = \min(0.3 * BSCR; Op) + 0.25 * Exp_{ul} \quad (26)$$

where:

- $BSCR$ denotes the Basic Solvency Capital Requirement;
- Op denotes the basic capital requirement for operational risk charge;
- Exp_{ul} denotes the amount of expenses incurred during the previous 12 months in respect of life insurance contracts where the investment risk is borne by policyholders.

The last component of the formula can be ignored since no unit linked business is written in our example. Basic capital requirement for operational risk Op is calculated as follows:

$$Op = \max(Op_{premiums}; Op_{reserves}) \quad (27)$$

where:

- $Op_{premiums}$ denotes the capital requirement for operational risk based on earned premiums;
- $Op_{reserves}$ denotes the capital requirement for operational risk based on technical reserves.

The capital requirement for operational risk based on earned premiums $Op_{premiums}$ is calculated as follows:

$$\begin{aligned} Op_{premiums} = & 0.04 * (Earn_{life} - Earn_{life-ul}) + 0.03 * Earn_{non-life} \\ & + \max(0; 0.04 * (Earn_{life} - 1.2 * pEarn_{life-ul} - (Earn_{life-ul} - 1.2 * pEarn_{life-ul}))) \\ & + \max(0; 0.03 * (Earn_{non-life} - 1.2 * pEarn_{non-life})) \end{aligned} \quad (28)$$

where:

- $Earn_{life}$ denotes the premiums earned during the last 12 months for life insurance obligations;
- $Earn_{life-ul}$ denotes the premiums earned during the last 12 months for life insurance obligations where the investment risk is borne by the policyholders;
- $Earn_{non-life}$ denotes the premiums earned during the last 12 months for non-life insurance obligations;
- $pEarn_{life}$ denotes the premiums earned during the 12 months prior to the last 12 months for life insurance obligations;
- $pEarn_{life-ul}$ denotes the premiums earned during the 12 months prior to the last 12 months for life insurance obligations where the investment risk is borne by the policyholders;
- $pEarn_{non-life}$ denotes the premium earned during the 12 months prior to the last 12 months for non-life insurance obligations.

It should be noted that generally one would expect that $Op_{premiums}$ would be greater than $Op_{reserves}$. In our example there are no claims provision, the business is expected to grow, therefore it is reasonable to assume that $Op_{premiums}$ would be driving the operational risk calculation. However, since newly established undertaking would have no premiums earned last year, operational risk calculation at least initially should be based on $Op_{reserves}$. The capital requirement for operational risk based on technical provisions is calculated as follows:

$$Op_{reserves} = 0.0045 * \max(0; TP_{life} - TP_{life-ul}) + 0.03 * TP_{non-life} \quad (29)$$

where:

- TP_{life} denotes the technical provisions for life insurance obligations;
- $TP_{life-ul}$ denotes the technical provisions for life insurance obligations where the investment risk is borne by the policyholders;
- $TP_{non-life}$ denotes the technical provisions for non-life insurance obligations.

Due to the fact that no life business is foreseen and non-life technical provisions consist only of premium provision which at the beginning is assumed to be equal to premiums written, operational risk calculation can be simplified the following way:

$$SCR_{operational} = \min(0.3 * BSCR; 0.03 * TP_{non-life}) \quad (30)$$

With the data from our example we get the following result:

$$SCR_{operational} = \min(0.3 * 660,670; 0.03 * 1,000,000) = \$30,000 \quad (31)$$

11.6 Overall Solvency Capital Requirement

Having covered all the relevant components of the standard formula we can now aggregate them according to (1) as follows:

$$SCR = \sqrt{624,580^2 + 624,580 * 67,065 + 67,065^2} + 30,000 = \$690,670 \quad (32)$$

The table below summarises our calculation of Solvency Capital Requirement according to the standard formula simplified for the case of insurance DAO with a \$1,000,000 portfolio of single product (phone battery insurance):

Component	Capital Requirement
Premium and reserve risk ($SCR_{nl\ prem\ res}$)	\$390,000
Catastrophe risk ($SCR_{nl\ CAT}$)	\$400,000
Diversification effect	(\$165,420)
Non-life underwriting risk ($SCR_{non-life}$)	\$624,580
Counterparty default risk ($SCR_{default}$)	\$67,065
Diversification effect	(\$30,975)
Basic Solvency Capital Requirement ($BSCR$)	\$660,670
Operational risk ($SCR_{operational}$)	\$30,000
Solvency Capital Requirement (SCR)	\$690,670

As presented in the table above Solvency II standard formula results in a considerably higher capital charge (\$690,670 or ~70% of premium) than our calculation in the previous section (24% of premium). To some extent this is expected as standard formula takes into account more risk components, i.e. catastrophe, counterparty default and operational risks, which were not considered in the previous calculation. However there is also a number of reasons why we would expect the capital charge to be lower than the standard formula result. Phone battery insurance has a fixed claim amount and only claim frequency is variable which should result in a lower variance. Also a 40% charge is for catastrophe risk which comes from a rather simplistic factor based calculation. In phone battery insurance catastrophic events resulting in a very large number of claims are not likely since such events as systematic faulty battery failures would be covered by manufacturer's warranty. Furthermore it should be noted than in this case the calculation was done for a single product in a single territory. If the DAO sells more types of product in more territories the overall capital charge would be further reduced due to segment and geographical diversification effects. Overall it would be reasonable to expect the capital charge to be in the range of 20-50%.

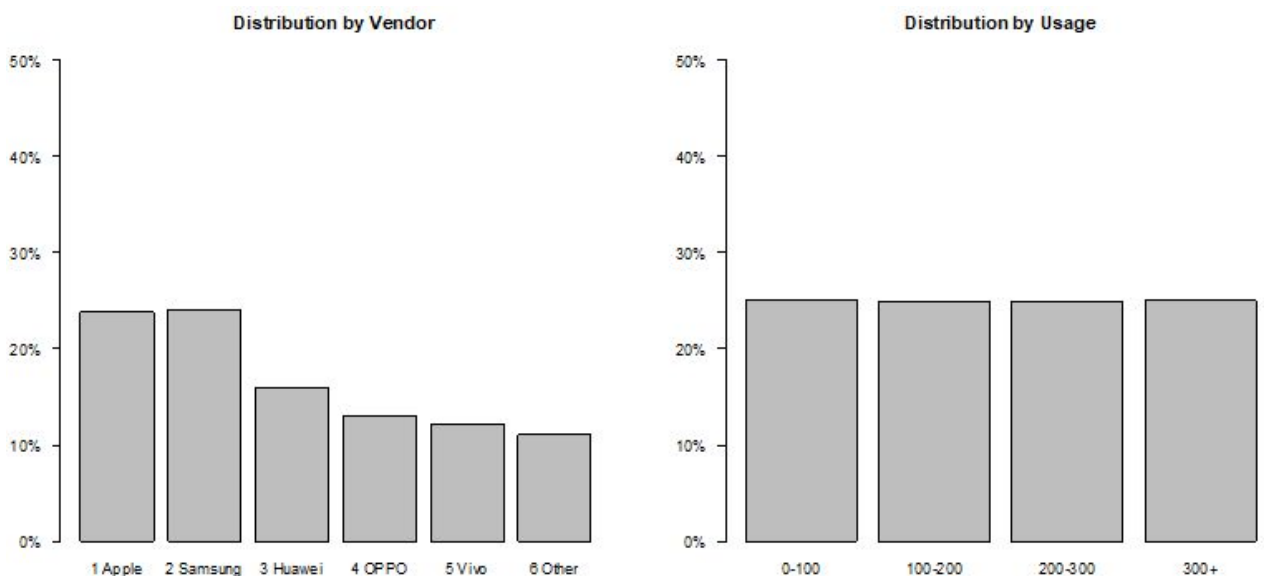
12. Pricing Simulation with Machine Learning

Together with other conditions that will be embedded in the smart contract to govern risk selection pricing will be key in ensuring the profitability of the portfolio and appropriate return to investors. It is important not only to set the appropriate average level of price, but also make sure that the pricing structure captures the significant risk factors of each individual policy. The risk of policies available to insure in the market will not be the same. There will be policies with higher and lower probabilities of claim.

The key function of pricing is to protect the portfolio from adverse changes in the mix of risk factors (the proportion of worse risks increasing). In other words this is called selection against the insurer or anti-selection. Anti-selection can obviously occur in a competitive market where one insurer with better pricing structure selects against the other insurer. But even being the only player in the market does not protect from anti-selection because it can also occur as selection not to insure at all. If offered price is the same for everyone it will appear too high for people with better risk and attractive for people with worse risk so the insurer ends up insuring only worse risks.

Machine learning techniques are used to capture the significant risk factors for pricing. To illustrate the concept for the purposes of this white paper we will use artificial data generated in R. For simplicity let's assume that there are only two risk factors – Vendor and Usage (again following the example of smartphone battery insurance). In reality machine learning deals with much higher number of risk factors which can also be related with each other through complex interactions.

Let's assume that the market of smartphone battery insurance policies potentially to be incepted is distributed as follows by Vendor and Usage:



Furthermore, let's assume that the risk differs as follows:

- Batteries of more established brands (Apple, Samsung, Huawei) are 20% less likely to fail than other brands.

- Compared to newer batteries (0-100 charge cycles), batteries charged 100-200, 200-300 and 300+ cycles are 10%, 20% and 30% more likely to fail respectively.

Please note that the numbers are only assumptions for the illustrative purpose of this white paper and may not be true in reality.

Artificial data exhibiting these features is generated by treating the above assumptions as GLM (Generalised Linear Model) coefficients and then reverse engineering the GLM. Machine learning seeks to identify risk factors by learning from the data and distinguishing significant patterns from the random noise. Applying this process from the other end we can produce artificial data exhibiting the above features, i.e. putting random noise on top of the pattern described above.

By applying the GLM coefficients each policy in the dataset is assigned an average claims frequency λ_i which depends on Vendor and Usage as described above. Random number of claims Y_i is then generated for each policy:

$$Y_i \sim \text{Pois}(\lambda_i)$$

We can now apply machine learning on this data. In any function estimation problem we wish to find a regression function $f(x)$, that minimizes the expectation of some loss function $\Psi(y, f)$:

$$\hat{f}(x) = \underset{f(x)}{\operatorname{argmin}} E_{y,x} \Psi(y, f(x))$$

There is a wide variety of machine learning algorithms developed to do that. GLM is just one of these algorithms which is widely used in general insurance pricing. For this example let's choose a different algorithm called GBM (Gradient Booster Machine) which is also used in modern pricing. The output of GBM algorithm from the H2O package widely used by machine learning community is presented below.

Model Details:
=====

H2ORegressionModel: gbm
Model Key: GBM_model_R_1500798425455_7

Model Summary:

	number_of_trees	number_of_internal_trees	model_size_in_bytes	min_depth	max_depth	mean_depth
min_leaves	max_leaves					
1	20	20	5626	2	5	4.00000
4	16					
	mean_leaves					
1	10.70000					

H2ORegressionMetrics: gbm
** Reported on training data. **

MSE: 0.0803715
RMSE: 0.2834987
MAE: 0.1484591
RMSLE: 0.1921301
Mean Residual Deviance : 0.0803715

Scoring History:

	timestamp	duration	number_of_trees	training_rmse	training_mae	training_deviance
1	2017-07-23 12:29:17	0.047 sec	0	0.28372	0.14866	0.08050
2	2017-07-23 12:29:17	0.140 sec	1	0.28368	0.14864	0.08047

3	2017-07-23 12:29:17	0.249 sec	2	0.28364	0.14862	0.08045
4	2017-07-23 12:29:17	0.343 sec	3	0.28362	0.14860	0.08044
5	2017-07-23 12:29:17	0.421 sec	4	0.28359	0.14858	0.08042

	timestamp	duration	number_of_trees	training_rmse	training_mae	training_deviance
16	2017-07-23 12:29:18	1.139 sec	15	0.28350	0.14848	0.08037
17	2017-07-23 12:29:18	1.201 sec	16	0.28350	0.14847	0.08037
18	2017-07-23 12:29:18	1.263 sec	17	0.28350	0.14847	0.08037
19	2017-07-23 12:29:18	1.310 sec	18	0.28350	0.14846	0.08037
20	2017-07-23 12:29:18	1.357 sec	19	0.28350	0.14846	0.08037
21	2017-07-23 12:29:18	1.419 sec	20	0.28350	0.14846	0.08037

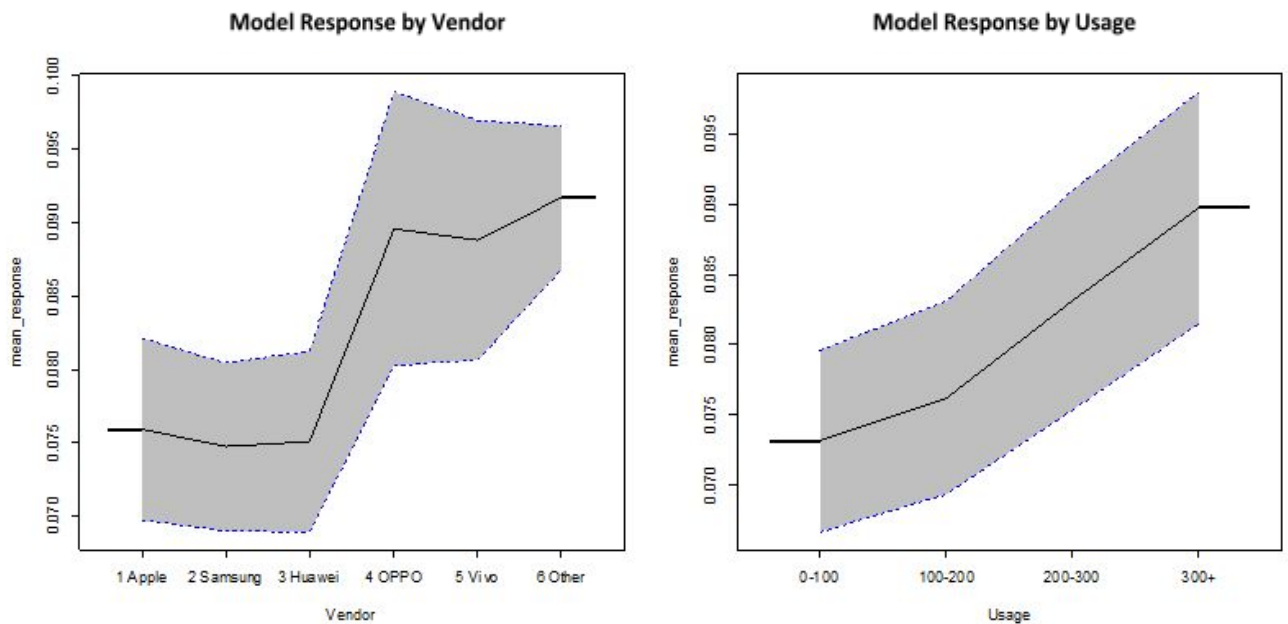
Variable Importances: (Extract with `h2o.varimp`)

=====

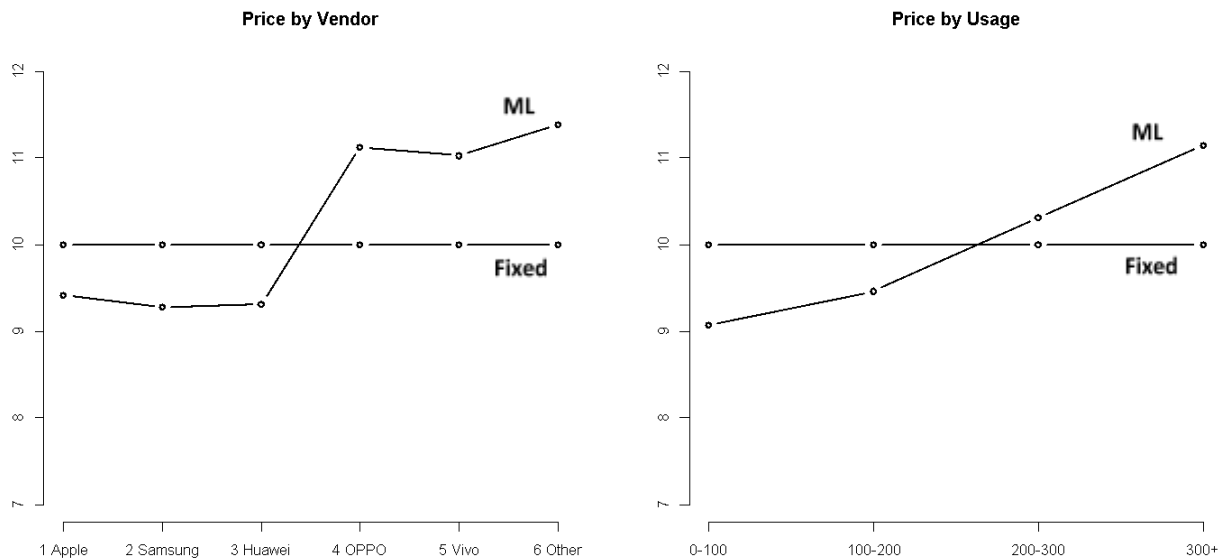
Variable Importances:

	variable	relative_importance	scaled_importance	percentage
1	Vendor	74.634300	1.000000	0.558408
2	Usage	59.021095	0.790804	0.441592

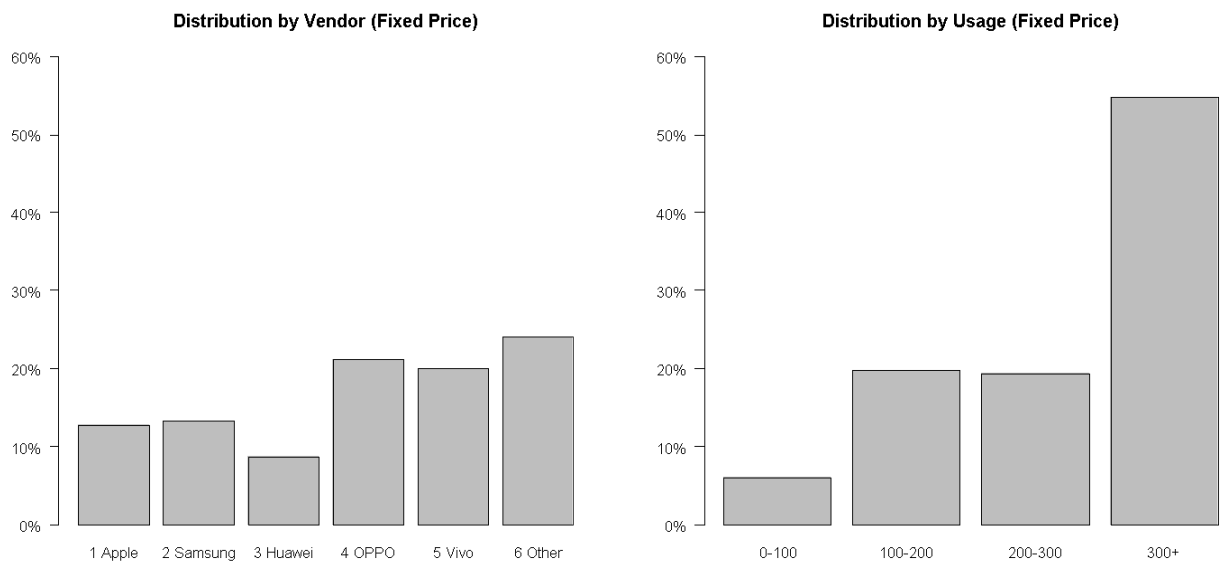
GBM is one of the most powerful machine learning algorithms which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. In the following graphs we observe that it captured the risk features in our data as expected:



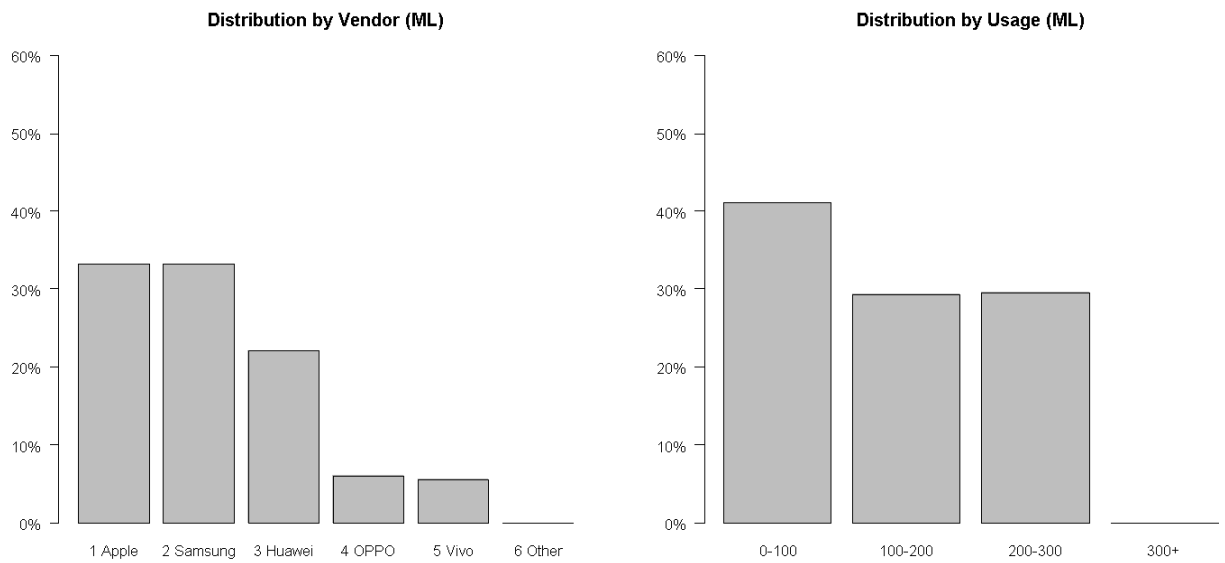
Let us now compare the performance of two competing portfolios – one offering the price derived from machine learning (ML) vs. fixed price for all policies.



Assuming each policy is incepted by the portfolio offering lower price the portfolio attracted by fixed price would be as follows:



As expected the fixed price portfolio attracted a higher proportion of less established brands and more used batteries which are worse risks. On the opposite the ML portfolio attracted more policies of established bands and less used batteries:



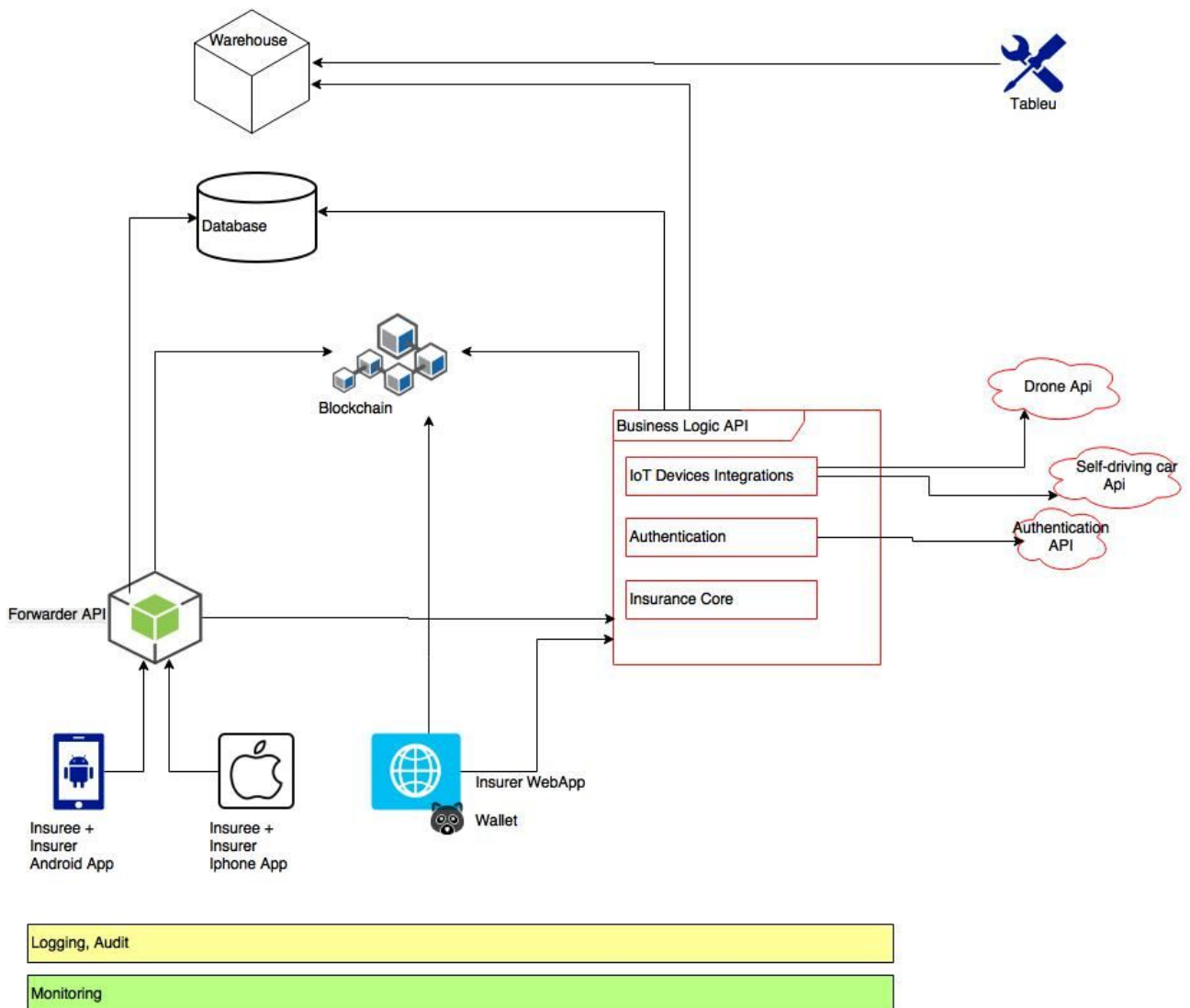
Simulated loss ratios were as follows:

Market portfolio: 80.6%
Fixed price portfolio: 90.6%
ML portfolio: 79.2%

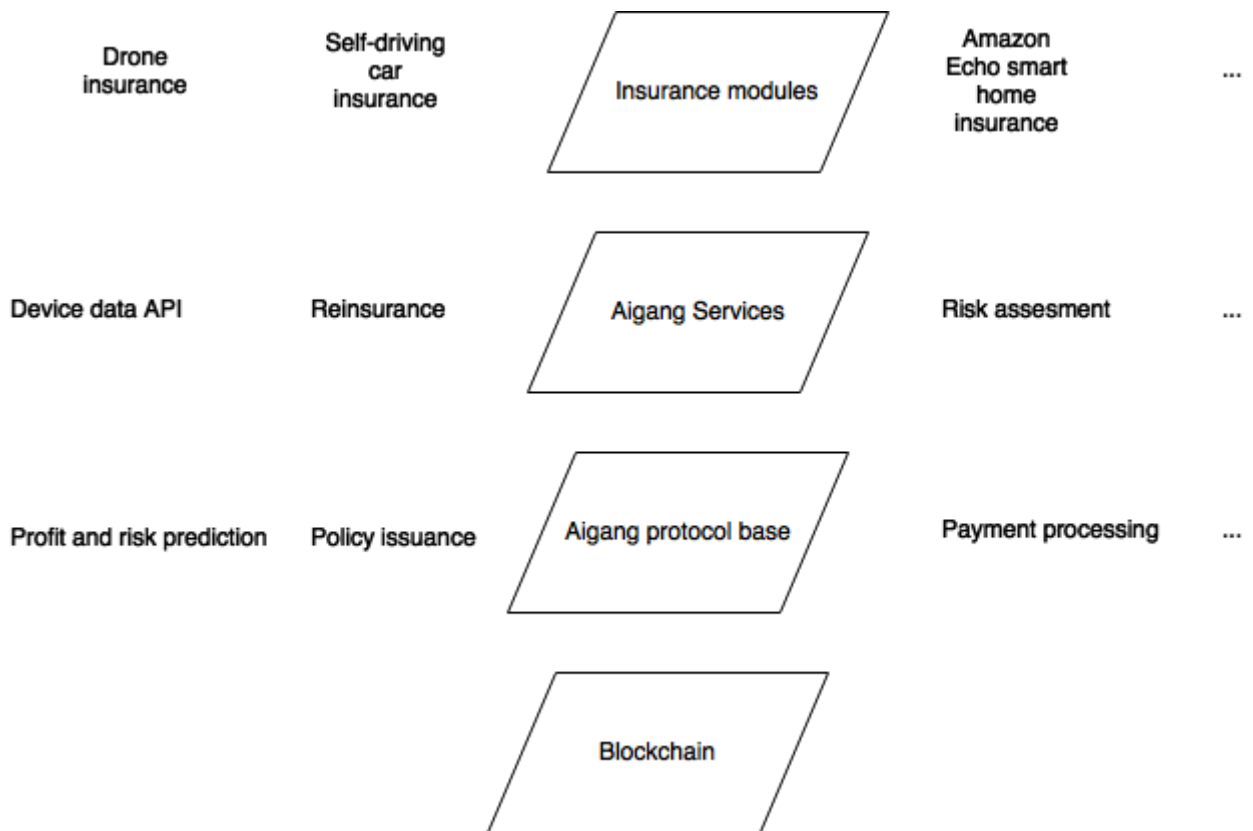
ML portfolio selects against the fixed price and attracts all the better than average risks thus experiencing 10 p.p. better loss ratio.

13. Architecture of the protocol

Based on our vision, we proposed architectural model of whole protocol and platform we aim to build. It contains many parts, and will require decent amount of time to build it. However, the main reason for that is ability to make fully autonomous insurance. The scheme below represents platform architecture.



The technical goal of our team is not to build all of the insurance products for various devices ourselves. We aim to build a protocol on a blockchain which would enable community, companies, developers build insurance modules themselves, using our infrastructure. That's how we see the future of blockchain insurance protocol. The scheme below represents layers of our system.



14. Roadmap

- **Phase 1:** MVP development v1.0. Ethereum blockchain smart contracts for policy issuing, risk assessment, claim processing. User interface for the insuree to manage all insured devices. Smart device tracking software use case development for issuing claims automatically. Blockchain environment: Ethereum Testnet. **Completed** (Demo apps can be downloaded from here - <https://aigang.network/#section-downloadapp>).
- **Phase 2:** MVP development v2.0. Ethereum blockchain smart contracts update for profit calculations, reserves formation and tokenization of it. User interface for the investor to manage portfolio of investments and invest into insurance product pools. Backend infrastructure for off-chain data collection and calculation. Blockchain environment: Ethereum Testnet. **In progress** (Early alpha can be accessed here - <https://investment.aigang.network/>).
- **Phase 3:** Beta version. Release for several developed insurance products. Opening beta version of the platform for investors and people to insure. Reinsurers integration. Blockchain environment: Ethereum Mainnet.
- **Phase 4:** Release v1.0. Stable and functioning software for several insurance products. Opening platform for module developers and introducing new insurance products. Blockchain environment: TBA.
- **Phase 5:** Release v2.0. Fully automated Insurance Protocol with insurance products (modules) for people to insure and investors to fund. Blockchain environment: TBA.

15. Leadership & Core Team

Augustas Staras, Business development

<https://www.linkedin.com/in/augustasstaras/>

Augustas has 10+ years experience in creating, growing & managing online and consumer businesses. He works in the emerging digital finance industry and has co-founded a peer-to-peer lending and investment platform with 15K+ users. He also works with blockchain projects, helping them achieve fundraising objectives.

Lukas Kairys, Smart contract developer

<https://www.linkedin.com/in/lkairys/>

<https://github.com/LukasKairys>

Lukas has experience as technical founder at startups, where he learned his ropes by working on software development and product management. He completed his Bachelor in Computer Sciences with a deep dive in the Ethereum blockchain, creating a prototype for an electronic voting system with smart contracts as a basis for voting.

Aidas Ignatavicius, Chief Actuary

Aidas has 10+ years experience working as an Actuary for a leading insurance company in Europe. His responsibilities include designing, validating and testing various pricing models for insurance products and policies. He has an MSc in Mathematics and spends his time to become a fully accredited actuary at the Institute and Faculty of Actuaries, UK.

Reda Markeviciute, Insurance Product & Policy

<https://www.linkedin.com/in/redamarkeviciute/>

Reda has more than 10 years experience in various finance, digital and marketplace businesses. She previously worked as a product manager and an actuary at SEB Life Insurance, one of the biggest insurance companies in Scandinavia and the Baltics. She has experience in product management, fundraising & business development. Also, an avid traveller!

Jonas Matkevicius, Marketing Manager

<https://www.linkedin.com/in/jonasmatkevicius/>

Jonas is a certified inbound marketer with a deep passion to process automation. Experienced in SEO, content marketing & inbound leads generation. He has a BSc in International Business. He worked as a marketing manager in a sales consultancy startup. He has also cofounded a few startups and was chosen as one of the few to participate in EUXCEL 2015 startup scrum.

Darius Devenas, Full Stack Developer

<https://www.linkedin.com/in/darius-devenas/>

<https://github.com/dariusdev>

Darius has 8+ years experience in enterprise software solutions. He works as a Senior Software Engineer in Adform - reporting platform for media agencies, trading desks and advertisers. He builds reliable platforms for critical business processes. He is responsible for Aigang security.

Mindaugas Jucius

<https://www.linkedin.com/in/mindaugas-jucius-915929bb/>

<https://github.com/MindaugasJucius>

Software developer with a huge passion to iOS architecture. Graduate in Computer Software Engineering. He helped to build application that has over 15 million users. Has experience working on grand scale projects and establishing Point of Sale systems.

Naglis Zemaitis

<https://www.linkedin.com/in/naglis-%C5%BEemaitis-22826993/>

<https://github.com/galisamas>

Cofounder of mobile applications and design company. Built secure and intuitive Android applications for banks, massive festivals and other business projects. He also has a BSc in Computer Software Engineering.

16. Competitive landscape

- Friendsurance (friendsurance.com), Lemonade (lemonade.com), P2P protect (tongjubao.com/en), Trov (<https://www.trov.com/>) - Centralized insurance brokers, returning part of premium to policyholders who did not submit any claims. Acts as a third party broker or insurance company. No automation.
- Dynamisapp (dynamisapp.com), Teambrella (teambrella.com) - peer to peer based insurance solutions implemented on the blockchain. Requires policyholders to act as evaluators. Does not eliminate the human factor.
- Rainvow (rainvow.org) - automated solution to insure against various weather conditions implemented on the blockchain. Focused on a very narrow use case and acts as an insurance company itself.
- Etherisc (etherisc.com) - blockchain peer to peer insurance solution. Team focuses on insurance using external data (weather, flight delays, etc.) to validate claims and issue payouts. Aigang's vision is fully automated insurance for IoT devices using its sensors and software.
- Gnosis (gnosis.pm) - blockchain based prediction market. Plans to use prediction for insurance pricing calculation and claims processing. Gnosis could act as a module on our protocol to create new insurance products with its community and algorithms.

17. Conclusion

IoT is a fast growing phenomena which creates an emerging market for insurance as none of the IoT devices can be 100% safe. This paper discusses the framework for an insurance DAO which leverages second generation blockchain technology to exploit this growth opportunity. Provided that insurance cover can be standardised to such extent that core functions of insurance operation such as underwriting and claims handling can be automated by the smart contract, this creates a new generation of insurance. The vision is to build a protocol layer for insurance peer to peer network. Furthermore, modules could be developed by other contributors who would enable various innovative IoT insurance products based on our protocol.

Infographic 1. “Comparison of traditional insurance vs. digital insurance as DAO ”

