A Decentralized Insurance Marketplace on Blockchains for Achieving a High Level of Open Market-Information Symmetry
(Version: 0.9b)

Risto Rossar¹, Mart Parve¹, Liina Laas-Billson¹, and Alex Norta²

¹ Black Foundation OÜ, Estonia
info@black.insure

² Department of Software Science, Tallinn University of Technology, Estonia
alex.norta.phd@ieee.org

Abstract. The legacy insurance industry is failing to cover risk with capital in a free-market way that is in the interest of the consumer. The status quo is highly discretionary with a small set of agents representing capital who decide about risk coverage in their own interest. Thus, customers and the insurance ecosystem face a wasteful context in the value chain of capital allocation. The advent of blockchain technology such as smart contracts, disruptively changes the status quo towards an improved disintermediated context with rapidly increased speed for bringing insurance products to market. Additionally, blockchain technology also allows for replacing a fraud riddled insurance ecosystem that suffers from qualitative highly bureaucratic decision making. Instead, blockchain technology eliminates inefficient layers of bureaucracy and establishes a novel quantitative organizational governance system based on transparent and traceable rules rooted in mathematics. Black Insurance fills this gap with a distributed blockchain-based insurance platform that overcomes currently existing information asymmetries to reinstate free-market forces for risk coverage with capital in the interest of customers.

Keywords: Insurance, blockchain, smart contract, risk, capital, decentralization, disintermediation, distribution

1 Introduction

The essence of the insurance industry is to allocate capital efficiently and effectively for risk coverage [7]. The challenge is to predict the occurrence of risk ahead of time for assured capital allocation [24]. It is not clear to which extent risk prediction is accurate yielding a mis-allocation of capital as a result. The cause of this mis-allocation is inherent inefficiency in the insurance industry that is offset via high transaction costs. In essence, the decision making for capital allocation does not follow free-market forces and is instead controlled by a small set of colluding agents with high discretionary power. These agents decide over
the capital allocation in their own interest and ignore the customers instead. The discretionary decision-making power is enabled by an overly bureaucratic organization [45] of the insurance industry as it exists today. Thus, the additional consequences of excessive bureaucracy in the insurance industry is the total masking of free-market forces and a disregard of customer interests.

A more detailed check of the cost structure shows\(^3\) that insurance companies have 20% of average costs for administrative expenses and net profit of gross written premium while for small-sized insurers, the average costs are considerably higher than for larger insurers. Overly bureaucratic decision making [4] is also responsible for the high cost structure. Important information travels through layers of bureaucracy to the key decision makers at the top of an insurance marketplace where information arrives late, distorted and filtered. The decisions are then inappropriate and arrive late on the operational level. With respect to the discretionary capital-allocation problem of a small set of agents, research shows that the situation will worsen [20,23]. The trends shows more acquisitions and mergers between existing insurance companies will shrink the set of discretionary agents even more and consequently, free-market forces in the insurance industry predictably further decrease. Literature shows [16] that the problem of information asymmetry in the insurance industry is a given problem and discretionary agents exploit this status quo. Thus, we must infer based on the available evidence [15] that overly bureaucratic governance is the cause of this anti free market information asymmetry problem.

The status quo of the insurance industry shows bringing products to market is hindered by a lot of waste, bureaucracy, high transaction costs that are the hindering free-market forces to unfold for a high degree of customer satisfaction. A gap exists in the insurance industry with respect to a platform for closing the currently existing information asymmetry that is the result of wasteful bureaucracy. This whitepaper fills the gap by exploring the question how to replace the current insurance ecosystem with a novel blockchain-based insurance platform? To establish a separation of concerns, we deduce the following sub-questions: Who are the key stakeholders and their key interests in relationship to the insurance platform? What is the static architecture for the insurance platform? What are the blockchain-supported dynamic interaction protocols within the architecture?

The remainder of the paper is structured as follows. Section 2 introducing a running insurance case that shows the challenges of the status quo, followed by additional presuppositions. Section 3 presents the requirement sets and stakeholders of the BlackInsurance platform. Next, Section 4 derives the distributed system architecture from the requirement sets. Section 5 shows the dynamic governance lifecycle the BlackInsurance architecture enables together with the set of discovered blockchain transactions that enable trustless collaboration between the involved stakeholders. Section 6 specifies the token structure and next, Section 7 evaluates the feasibility of a rapid BlackInsurance platform deployment.

\(^3\) https://medium.com/BlackInsurance/costs-in-the-insurance-industry-today-b86c225eea0b
based on pre-existing blockchain technology. Finally, Section 8 concludes this whitepaper and presents future work.

2 Motivating Example and Preliminaries

We present the status quo of the insurance-industry deficiencies in Section 2.1 with a motivating example. Next, Section 2.2 gives preliminaries that are relevant for the remainder of this whitepaper.

2.1 Motivating Example

Insurance companies control the industry with respect to the products that are sold, pricing and affiliated product terms. Due to established insurance oligopolies that eliminate the need to change structurally, very little innovation occurs. Despite changing insurance-market needs, over-regulation and inefficient bureaucracy have made it near impossible to place insurance products in the market.

Insurance companies are approachable for new insurance products. As Figure 1 depicts, there is a considerable delay, in average 6 to 18 month, before insurance companies decide to cover a broker with capital for a new insurance product. The reasons for such a delay are manifold. First, authorities impose many laws and regulations on new insurance products. Secondly, insurance companies are plagued by outdated and costly, arcane bureaucracy that lacks adequate process automation with information technology. Mostly excel sheets are in use that different departments maintain by themselves that over time comprise very diverging content and figures, rendering decision making very challenging.

![Fig. 1: Status quo of the insurance industry.](image)

The problem of diverging assessments per insurance company and department increases over time due to diverging technological standards of information
technology in use. In addition, a considerable amount of manual bureaucratic labor combined with diverging excel sheets yields more opportunities for errors. Consequently, entrepreneurs and insurance brokers fail in placing new insurance products on the markets for customers who are left unsatisfied. The oligopolistic insurance-industry structure is also a challenge for smaller and retail investors who are not able to place their money into stable insurance portfolios, even though the interest is there. Thus, established insurance companies control the industry and retain the profits. Furthermore, Figure 1 infers that brokers have relevant market know-how as they understand the customers, the industry and the market needs. Thus, brokers able to design well-targeted insurance products that they cannot bring to market, also because of unavailable underwriting.

2.2 Preliminaries

We discuss next the employed methodology for this whitepaper. The development of the BlackInsurance platforms follows a model-driven design (MDD) methodology [6] that emphasizes the system-design process towards a rapid implementation deployment. A set of chained model types provide an increasingly deeper understanding of the BlackInsurance platform. The sequence of development commences with specifying the platform goals, followed by the static architecture-topology design and finally, we give the dynamic behavior protocols that exists between architecture components and respective stakeholders. Thus, a goal-model [43] first describes the requirements of the system, a UML component [25] diagram outlines the static architecture topology and thirdly, a UML sequence diagram specifies the dynamic behavior of the platform comprising key events stored on a blockchain.

Fig. 2: Goal-model notation in (a), followed by component-diagram notation in (c) and finally, BPMN notation in (c).
Goal models as a notation taken from the agent-oriented modeling (AOM) method [43], capture the requirements of the BlackInsurance platform and comprise the simple notation in Figure 2(a). Functional goals of a system are depicted as parallelograms, ‘quality goals’ are synonymously called non-functional requirements in software engineering and depicted as clouds, and agents with specified roles may be human or artificial. The value proposition’ is the root of a hierarchically decomposed AOM goal model and denotes the overall systems goal. Quality goals and roles are attached to functional goals and inherited to lower-level functional goals.

Based on the goal model, we derive a UML component diagram [8] to specify the static structure of the BlackInsurance platform. Figure 2(b) depicts UML-notation elements with labeled, rectangle components. The latter, we further refine with sub-components and assign provided- and required interfaces. Figure 2(b) depicts a provided interfaces as a line with a circle and required interfaces are lines with a cup at the end. Actors interact with components of the BlackInsurance platform and are synonymous to roles in goal models.

The dynamic governance behavior of the BlackInsurance platform we express with the business process modeling notation BPMN [1]. While this notation has grown extensively and comprises a rich set of modeling elements, Figure 2(c) depicts that small set of elements we consider for this whitepaper. A dynamic governance lifecycle commences with a general start element that is depicted as a white-filled circle, and terminates eventually for which we consider a black-filled circle. the governance lifecycles comprise atomic tasks that can not be further decomposed. On the other hand a rounded rectangle with a contained plus sign is a non-atomic sub-process that can be further decomposed with lower-level processes. Finally, sub-processes may also loop until a certain exit condition is met, which is denoted by a directed circle. The depicted elements in Figure 2(c) are connected by directed arcs that we partially label with digital objects that are passed from one element to the next.

With respect to blockchain technology, the BlackInsurance platform must be realized employing decentralized consensus finding for blockchain transactions. One example is proof-of-work (PoW) [46] to assure transparent, distributed traceability and security for on-chain events. For PoW, computationally challenging and -expensive puzzles must be solved, which impairs the scalability for large industry projects. Ethereum [48] uses PoW and is still the de facto smart-contract standard, specifically for issuing so-called ERC20 tokens [30] for initial coin offerings (ICO) [3]. The Solidity smart-contract programming language of Ethereum has no formal foundation for verification [13] with algorithmic tool support. Proof-of-stake (PoS) [12] is a more suitable option for smart contracts [21] where the choice of a new block creator deterministically happens depending on the staked wealth. Proof-of-Authority (PoA) [31] is a variety of staking where a consensus mechanism considers identity as a stake. Other known consensus algorithms are proof-of-bandwidth (PoB) [27], proof-of-elapsed-time (PoET) [17], practical byzantine fault tolerance (PBFT) [2], delegated proof-of-stake (DPoS) [49], and so on.
Pertaining to governance lifecycles we consider for the BlackInsurance platform, the foundation for this whitepaper is research for peer-to-peer (P2P) collaboration [39] in a service-oriented cloud computing context. The lifecycle is formalized in three separate publications, namely the setup-phase in [37], the rollout- and enactment phase in [38] and finally, the rollback- and termination phase in [42].

3 Goals of the BlackInsurance Platform

The AOM method [43] is employed for defining the BlackInsurance-system requirements as it is a socio-technical approach to model dynamic and complex distributed systems comprising human and software agents. We provide the overall goal model in parts where Section 3.1 shows the value proposition together with the first refinement level. Section 3.3 describes the partial goal model of the three left-side decompositions and Section 3.4 comprises the right-side goal refinements.

3.1 Value Proposition

In the depiction of Figure 3, the value proposition being the goal-model root, carries the label *friction-less risk-coverage capacity for insurance brokers*. The functional refinement goals on the first level are *manage a syndicate*, *manage a virtual insurance company*, *exchange tokens*, *viewing performance dashboard*, *manage black platform*. Note that all these first-level refinement goals are further detailed in subsequent goal models below.

Fig. 3: The value proposition and first refinement level of the Black Insurance goal model.
The value proposition in Figure 3 references a set of goal models that hold for the entire goal model that we explain next. Error handling infers there exists system-support help for a user to recover from errors. Error avoiding is support to anticipate and prevent common errors that occur on the BlackInsurance platform. Learnable means that users are able to form syndicates, place insurance products, match both for underwriting, and so on. Scalable means the platform is capable of managing a large amount of simultaneously occurring processes. Secure as a quality goal decomposes into confidentiality, integrity, availability and authorized actions [5]. Confidentiality is the absence of unauthorized disclosure of information, integrity is the absence of improper system alterations and availability the readiness for correct service. Additionally, authorized actions assure that only a stakeholder with appropriate credentials is able to carry out an action using the BlackInsurance platform.

On the first refinement level of functions, Figure 3 shows additional quality criteria the we explain below. Fast means that, e.g., the management of a syndicate is carried out without latency involved that is debilitating to platform usability. Performant means that there exists a high throughput of information artifacts, e.g., high-delivery number of tokens. Highly automated means the carrying out of specific functions should either not involve stakeholders at all, or only to a minimal degree. Private means the information related to a platform activity remains undisclosed. Transparent is an opposite goal in that a functions precondition, activity and postcondition are fully visible to the stakeholders. Modifiable means a system function is responsive to contextual changes. For example, if the data set for a specific insurance changes in a region then a viewing dashboard must be able to respond in the display.

The roles in Figure 3 comprise a syndicate operator who manages a syndicate lifecycle. A broker manages a virtual insurance company and finally, most roles are associated with the function of viewing the performance dashboard, i.e., the syndicate operator, broker, investor black platform administrator and also the reporter agent. Note the latter is an artificial agent who facilitates a highly automated generation of dashboard views. Finally, the black platform administrator also is associated to the function of managing the black platform.

3.2 Left-Branch Goal Refinements

The refinement goal model in Figure 4 shows the hierarchy below the functional goal for managing a syndicate. Note that the quality goals and roles in Figure 3 are inherited down the refined functional goal hierarchy.

The second functional-goal refinement in Figure 4 comprises receiving ICO investment, allocating capacity to products, creating and editing syndicates with authenticated content. For the syndicate, new members should be invited and the syndicate itself must be managed according to a set of policies. Such policies must be issued and define the underwriting rules. Also claims rules must be defined. Furthermore, compensation needs to be payed to the claims fund that is recorded. Finally, it must be possible to also delete policies for syndicate management. Next, the progress in the syndicate lifecycle management must be observable,
i.e., the performance is monitorable by checking information about policies and claims. A syndicate must also have profit-sharing rules defined for sending and receiving tokens. Finally, voting ballots must be created in case of important decision making by a syndicate that requires voting.

Fig. 4: First partial goal-refinements of the Black Insurance system.

Additional quality goals are part of the model in Figure 4 that we explain in Section 3.1. Now these quality goals are assigned in a targeted to specific functional goals in the hierarchy. There exist also additional roles in the refining goal model, being an investor, underwriter, claims handler, profit allocator and a black security token (BST) holder. Furthermore, to satisfy to multiple assignments of the quality goal for highly automated, we assign a set of software agents that
support the human roles that interact with the BlackInsurance platform. These software agents are an accountant, underwriter, claims agent, claims fund wallet, profit sharer, reporter of observed progress, accountant and ICO platform.

The goal model in Figure 4 comprises gray areas that stretch over respective functional goals. These gray zones carry labels for the BST and/or the black utility token (BLCK). Thus, this way the goal models serve to start defining the structure of the token economics for the BlackInsurance platform.

3.3 Central-Branch Goal Refinements

For the model in Section 5 the functional goal of managing a virtual insurance company is further refined. The refinement functions are managing specific insurance products that must be created and edited together with their authenticated content. The capacity for backing a virtual insurance product must be received and along that line, a company created and their details edited. For the virtual insurance company, new members may be invited.

The functional goal for managing policies and claims is further refined. Managing policies involves creating and editing such policies for which a premium must be paid. Furthermore, it is possible to cancel initiated claims. The management of claims comprises their creation and addition of claim evidence. Finally, money may be received to the claims fund. All information about policies and views can be viewed while managing a virtual insurance company involves the sending and receiving of tokens. BLCK tokens are used for paying policy premiums, receiving money to the claims fund and sending, receiving tokens and for the latter, also BST are in use.

Besides a broker who is involved in all functions for managing a virtual insurance company, Figure 5 also comprises as additional roles a syndicate operator. The remaining extra roles are all software agents to satisfy the quality goal for a high degree of automation. New quality goals in Figure 5 are applicable and integrable. Applicable means that the insurance product that the virtual company produces on the BlackInsurance platform matches with the needs of the market needs and also the technology needs of third parties that adopt a respective insurance. Finally, integrable means issues policies and claims do not conflict with each other and create contradictions.
3.4 Right-Branch Goal Refinements

The remaining three refinement branches depicted in Figure 6 cover the exchange of tokens, viewing of the performance dashboard and managing the black platform. The exchange of tokens comprises paying for services in BLCK and exchanging BLCK and BST to fiat and vice versa. Finally, exchanging tokens also includes updating the service-fee schedules.

Viewing the performance dashboard we do not refine further in Figure 6 and also does not require tokenization, while the functional goal of managing the black platform is more sophisticated. The latter comprises the approval or rejection of syndicates and insurance products, sending and receiving tokens that require the minting of BLCK tokens that may also be burnt, and creating ballots for voting procedures.
Additional roles are depicted in Figure 6 that are not present in the top-level goal model of Figure 3 of the root value proposition. These additional roles are artificial software agents that are in support of the human roles who interact with the BlackInsurance platform to satisfy the quality goal of a high degree of automation. The role of BLCK holder is another addition in Figure 6 and associated to the voting for a ballot to manage the black platform.

Fig. 6: Third partial goal-refinements of the Black Insurance system.

4 BlackInsurance Architecture Model

The BlackInsurance architecture we derive from the goal models above. Thus, Section 4.2 explains the mapping heuristics from goal models to the component diagram notation we use for the architecture. Section 4.2 then shows the architecture as a component diagram together with an explanation.

4.1 Mapping Heuristics

The heuristics for deriving a component diagram from the goal models are as follows. The value proposition of the goal model is represented by the entire system architecture while first-level refinement functional goals are the top-level components in Figure 7. Embedded components represent respective second refinement-level functional goals. Due to space limits, we do not consider lower-level functional goals for the architecture and limit ourselves to the first two refinement levels.
Once the components are arranged, it is also derivable to which components actors must be assigned based on the goal-model role associations. The additional information in the component diagram denotes with receiving- and providing interfaces what the directions of information exchanges are between actors and components. We assume that these information exchanges are implemented with application-programming-interface (API) sets. Such providing- and receiving interfaces also show information exchanges between components of the architecture in Figure 7 that goal models can not specify. We use abstract labels for the interfaces that express conceptually the type of information exchange. These labels for information exchange are much larger data sets with complex relationships in a system implementation. Finally, the gray areas of the goal models denoting tokenization is equally mapped into the architecture by gray-shading respective components.

The quality goals we do not map into the component diagram of Figure 7 and are realized in system implementations as architecture styles and -patterns [41]. Still, given the association to functional goal, there exist clear indications what quality goals a component must adhere to when extended and diagnostic system engineering takes place. Again, due to page limitations, we omit such an architecture styles- and patterns study, while this whitepaper established a foundation to subsequently do so.

4.2 Component Diagram

The architecture of the BlackInsurance platform in Figure 7 shows the result of applying the heuristics that Section explain. Since the information flow along providing- and receiving interfaces is an addition that goal models can not express, we focus on this explanation primarily and first describe the interaction of actors with respective components. Next, we explain the information exchanges between components too. For brevity, the information exchanges between embedded components is out of focus for this whitepaper.

Component-to-Actor Information Exchanges

The core component of Figure 7 is the syndicate manager that comprises four tokenized sub-components being the investments manager, profit-share manager, policy administrator and ballot manager. There are eight actors who interface with the syndicate manager and we explain the information exchange starting with the investor that continues in a clockwise direction.

The investor in Figure 7 checks what the total investments are into a syndicate itself and also into insurances a syndicate back with capital to cover insured risk. In return, investments are submitted to the investment manager if an investor chooses so. Furthermore, an investor can also take profit from the component labeled profit-share manager. Next, the syndicate operator submits information about insurance products to the capacity-allocator component and additionally description information to the content manager. Such descriptions comprise details about the syndicate, the members, investments, adopted insurance products, and so on.
The interaction of BST token holders with the syndicate manager is limited to the ballot manager for casting votes while syndicate operators receive performance reports from the progress observer and submit member-permission information to the member manager. The profit allocator submits profit-share rules via a receiving interface to the profit-sharing rules manager component. The policy-administrator component interfaces with two actors being the claims handler who submits claims rules and the underwriter who submits underwriting information. Finally, the syndicate operator also draws investment data and submits ICO terms to the investment manager. The syndicate manager also delivers profits to the profit-share manager.

The virtual insurance company manager comprises one tokenized sub-component labeled policy-and-claims manager and has one actor type exchange information. The broker submits description information to the insurance-product manager, profile facts to the company-profile manager and members-permission information to the embedded component for managing members of a virtual insurance company. Additionally, the broker also submits policy- and claims data while receiving capacity information from the component labeled capacity receiver.

The black platform manager has a ballot-manager component that is tokenized and interfaces with two actor types. The black platform administrator approves syndicates and insurance products via the corresponding respective components. Additionally, the BLCK token holder submits votes to the ballot-manager component.

The tokens exchanger depicted in Figure 7 shows that all contained sub-components are tokenized and has for interacting actors. The broker submits purchase orders to the black-service payment tracker and bids to the exchange manager for both BST and BLCK tokens. Also the investor and syndicate operator submit bids to the exchange manager. The syndicate operator submits purchase-order information to the black-service payment tracker and finally, the black platform administrator submits fees to the black-service fee scheduler.

The last component remaining is the performance dashboard manager that does not show contained sub-components and interfaces with four respective actors. Both the syndicate operator and investor retrieve performance reports. The same holds for the black platform administrator and broker.
Fig. 7: The BlackInsrance platform system architecture.
Inter-Component Information Exchange

As the most complex component of Figure 7 is the syndicate manager, we commence with explaining the inter-component information exchange from this architecture location. First, the syndicate manager’s wallet exchanges bidirectionally tokens with the token exchanger’s embedded exchange manager. Next, the syndicate manager’s wallet then also exchanges equally tokens bidirectionally with the wallets of the virtual insurance company manager and the black platform manager.

Performance data the progress observer provides to the performance dashboard manager and continuing clockwise with the syndicate manager, the content manager provides syndicate descriptions to the syndicate approval manager that in turn provides voting details to the syndicate manager. The latter also receives information about syndicate members from the component labeled members manager.

The policy manager exchanges three types of information sets with the virtual insurance company manager. First, the latter provides underwriting data to the policy manager that itself returns policy and claims data. The policy administrator component also sends BLCK tokens to the capacity receiver of the virtual insurance company manager. Finally, the syndicate manager requests capacity information from the capacity receiver of the virtual insurance company manager.

The virtual insurance company manager in Figure 7 policy and claims data to the component being the performance dashboard manager. The wallet bidirectionally sends and receives tokens to the token-exchanger component and the wallet of the black platform manager. Additionally, the company profile manager requests broker and product information from the component labeled product-approval manager. The latter also provides approval information to the virtual insurance company manager. The token exchanger component’s contained exchange manager bidirectionally sends and BLCK- and BST tokens to the wallet of the black platform manager while the remaining information flow is already covered above.

5 Dynamic Blockchain Protocols

For describing the dynamic behavior of the BlackInsurance system, we first present in Section 5.1 the on-chain transaction that are relevant for preventing litigation. Section 5.2 describes the governance lifecycles of the syndicates, insurance companies, -products and coverage matchings between insurance products and syndicates. Finally, Section 6 complements the governance lifecycles by describing in deeper detail the dynamics of the token structure and economics.
Table 1: Set of on-chain transaction for litigation prevention.

<table>
<thead>
<tr>
<th>Component</th>
<th>Event</th>
<th>Explanation</th>
<th>Stakeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syndicate manager</td>
<td>Create syndicate</td>
<td>Syndicate operator</td>
<td>Syndicate manager</td>
</tr>
<tr>
<td></td>
<td>Invest in contract</td>
<td>Investor, Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invest in fulfillment</td>
<td>Investor, Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BST creation</td>
<td>Syndicate operator</td>
<td>Black admin</td>
</tr>
<tr>
<td></td>
<td>BST issuance</td>
<td>Investor, Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer capacity</td>
<td>Syndicate operator</td>
<td>Broker</td>
</tr>
<tr>
<td></td>
<td>Reserve funds for capacity deal</td>
<td>Syndicate operator</td>
<td>Broker</td>
</tr>
<tr>
<td></td>
<td>Transfer BLCK between wallets</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer BST between wallets</td>
<td>Investor, Syndicate operator</td>
<td>Black admin</td>
</tr>
<tr>
<td></td>
<td>Create registered contract (PoE)</td>
<td>Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create claim rules</td>
<td>Claims handler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create underwriting rules</td>
<td>Underwriter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create profit share rules</td>
<td>Profit allocator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allocate profit for BST</td>
<td>Investor, Syndicate operator, smart contract, oracle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add member to Syndicate team</td>
<td>Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remove member from Syndicate team</td>
<td>Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Settle claim</td>
<td>Claims handler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Underwrite policy</td>
<td>Underwriter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer compensation</td>
<td>Syndicate operator</td>
<td>Broker</td>
</tr>
<tr>
<td></td>
<td>Create voting ballot</td>
<td>Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vote for ballot</td>
<td>Syndicate operator, investor</td>
<td></td>
</tr>
<tr>
<td>Virtual insurance company manager</td>
<td>Create company profile</td>
<td>Broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a product</td>
<td>Broker, Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add member to Company team</td>
<td>Broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Remove member from Company team</td>
<td>Broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accept capacity deal</td>
<td>Syndicate operator, Broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Counteroffer capacity deal</td>
<td>Syndicate operator, Broker</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create policy quota</td>
<td>Broker, Underwriter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Issue policy</td>
<td>Broker, Underwriter</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer premiums</td>
<td>Broker, Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Submit claim</td>
<td>Broker, Claims handler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Submit additional claim evidence</td>
<td>Broker, Claims handler</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer BLCK between wallets</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Update service fee schedule</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pay service fee</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Submit bid/ask</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Accept bid/ask</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Settle trade</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Move tokens through token gateway (atomic swap)</td>
<td>Any platform user, token gateway oracle</td>
<td></td>
</tr>
<tr>
<td>Tokens exchanger</td>
<td>Approve syndicate</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Approve product</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Approve syndicate members</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Approve virtual insurance company</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer BLCK between wallets</td>
<td>Any platform user</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Transfer BST between wallets</td>
<td>Investor, Syndicate operator, Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mint BLCK tokens</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Burn BLCK tokens</td>
<td>Black admin</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create voting ballot</td>
<td>Black admin, Syndicate operator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vote for ballot</td>
<td>Black admin, Broker, Syndicate operator</td>
<td></td>
</tr>
</tbody>
</table>

5.1 On-Chain Transaction Set

The columns in Table 1 list to the very left components that are also part of the architecture model in Figure 7. Note that four components are listed where on-chain transactions occur that we list in the table and the component labeled
performance dashboard manager is omitted. The next column labeled event shows numbers that identify the set of on-chain transactions. The explanation column describes briefly what the objective of a respective transaction is and finally, the column to the very right lists per transaction row the set of involved stakeholders that are part of Figure 7 that depicts the BlackInsurance architecture.

With respect to on-chain transactions in Table 1, the syndicate manager is the most blockchain expensive component, followed by the virtual insurance company manager and the black platform manager. The least blockchain expensive component in Table 1 is the token exchanger, while the performance dashboard manager does not perform any on-chain transactions at all.

5.2 Dynamic Governance

There are three aspects of interleaved, dynamic governance lifecycles that the architecture of Figure 7 supports. These are first, the governance lifecycle of the syndicate that Figure 8 depicts. The governance lifecycle in Figure 9 that depicts the creation of a virtual insurance company together with issued and managed insurance products. The third governance lifecycle in Figure 10 describes the matching of insurance products with syndicates where capital of the latter covers the inherent risks in insurances issued by the former. The notation used for the three partial governance-lifecycle figures is explained in Figure 2 of Section 2.2. Additionally, we use the event ID numbers in the governance lifecycles of Figures 8-10 as inserted numbered, red circles, which is not native to pure BPMN notation.

**Syndicate Lifecycle**

The syndicate lifecycle in Figure 8 follows the example published in [39] and commences with preparing a syndicate network blueprint that is populated with network models and roles. The network models comprise process definitions that a syndicate follows after its establishment. Furthermore, definitions for obligations and rights [40] are also contained in a network model that syndicate members must follow. The network model also comprises in a first instance roles that specify the capabilities, skills, capabilities, etc., that must be matched to populate a role.

Next, syndicate operators and investors may populate the syndicate network model to form a proto-contract. The negotiation sub-process in Figure 8 comprises the delivery of a proto-contract copy to each member, followed by a negotiation cycle that has three outcomes [37]. First, all members consent with the content of the proto-contract and an agreement comes into existence. Second, it is possible that one party issues a counter offer that must be redistributed to all members again for negotiation. Thirdly, if only one member disagrees with a proto-contract then the lifecycle end and reaches the termination point.

Assuming the member syndicate operators and investors agree, a contract comes consensually into existence. Establishing a syndicate involves copying machine-readable versions of the agreement to every syndicate members and establishes a distributed communication configuration [38]. After that, the syndicate
reaches the stage of considering insurance products for covering risk with capital. Note that the gray sub-process labeled propose insurance(s) to syndicate(s) belongs to the lifecycle in Figure 9 for virtual insurance company establishment.

Fig. 8: Lifecycle protocol of the syndicate.

While the syndicate covers risk with its capital, it is possible that conflicts occur. Figure 8 shows as an examples that an obligation of the syndicate contract is violated [40], or a member wants to exit the syndicate. The assumption for the rollback stage is that the syndicate members engage in a voting process to decide about the appropriate response that may be a non-disruptive rollback to the earlier stage of syndicate establishment. Thus, the syndicate constellation remains intact with some form of modification, e.g., an obligation is replaced, a syndicate member is exchanged, and so on. On the other hand, a rollback vote outcome can be disruptive in that the syndicate concludes a violation is so severe that a formation re-negotiation is necessary. Finally, when the business case for covering an insurance risk with capital seizes to exist, the syndicate is carefully terminated for reaching an orderly end. That means, all running processes for the syndicate are cleanly switched off and all unnecessary data deleted, or logged correctly to assure the BlackInsurance platform does not drown in clutter so that the performance eventually collapses.
The syndicate lifecycle in Figure 9 depicts numbered red circles that correspond to the on-chain transaction IDs of Table 1. For the syndicate network preparation, an on-chain transaction establishes proof of existence of the created content. Next, when the network model has contained roles populated with syndicate operators and investors, Transaction 2 records the commitment of these tentative syndicate members. During the cycles of formation negotiations, coupled with potential counteroffer submissions, the investment fulfillment of the syndicate is recorded in Transaction 3. Since the negotiation involves voting procedures, Transactions 20 and 21 record the creation of a ballot is recorded together with the cast votes respectively. Finally, Transaction 42 records the syndicate approval by the Black admin.

The syndicate establishment requires the biggest amount of on-chain transaction with Transaction 1 recording the syndicate creation. Affiliated, Transactions 4 and 5 record the creation and issuance of BST tokens respectively. Next, Transactions 10 to 14 creates a proof of existence for the syndicate content; creates claim-, underwriting- and profit-share rules respectively. Finally, the BST profit allocation for the syndicate is recorded too. The considering of insurance products in Figure 9 proposed by the virtual insurance company, involves Transactions 6 to 10 for recording: the offered capacity deal for risk coverage, reserve funds for a capacity deal, transfers of BST- and BLCK tokens between wallets and proof of existing content respectively.

The looping coverage of insurance risk involves Transactions 17 to 19 for recording on-chain the settling of claims, underwriting policies and transferring compensations. Next, the rollback of conflicts requires in the non-disruptive case optionally involves the recorded adding, or removal of syndicate members with Transactions 15 and 16. Finally, the termination of a syndicate involves Transaction 16, when all syndicate members are removed. The same transaction is also in use when the negotiation of a syndicate formation fails because of a member disagreement.

Virtual-Insurance-Company Lifecycle

In Figure 9, the lifecycle of a virtual insurance company (VIC) follows allegorically quite similarly the syndicate lifecycle in Figure 8. Thus, the lifecycle commences with the VIC network model preparations, comprising inserted role definitions. Brokers next populate the VIC network model before a negotiation phase commences. Note that this formation negotiation is considerably less complex than the assumed formation negotiation of a syndicate in Figure 8. Still, we assume a subset of the negotiation-outcome options exist as for a syndicate, being a consensual agreement of all brokers involved, or a disagreement that immediately terminates the lifecycle.

Once the lifecycle is established, the virtual infrastructure for the VIC must be established in terms of distributed, communicating processes and services. The VIC is then in a position for issuing insurance products that enter the insurance pool on offer for the syndicate in Figure 8 to be adopted. Next, the
issued insurance must be covered by capital for the inherently contained risk, for which the capital again stems from the lifecycle of the syndicate.

Conflicts between a VIC and insurance may stem from a matching conflict with a syndicate that we describe below. If conflicts occur, the non-disruptive rollback either pertains to the insurance product itself, or to the VIC. In the latter case, a rollback occurs to the lifecycle stage of VIC establishment where a re-issuance of an insurance occurs. On the other hand, a non-disruptive rollback may also affect the VIC, which requires a rollback to the lifecycle stage of VIC establishment. Finally, a conflict may also be dealt with in a disruptive way that require a rollback to the lifecycle stage of VIC-formation negotiations. Eventually, a VIC terminates its lifecycle in an orderly way when the business case cedes to exist, which involves also the terminations of affiliated insurance products.

Fig. 9: Lifecycle protocol of a virtual insurance company that issues insurances.

With respect to on-chain transaction in Figure 9, adding the VIC network preparation details to the blockchain involves Transaction 22 that stores in a traceable way the company profile. Next, Transaction 24 affiliated with populating the VIC model roles with concrete brokers records on-chain that a member is added to the company team. The same Transaction 24 is necessary if the VIC formation negotiation requires further adjustments in the team membership. For the VIC establishment, Transaction 43 records the approval and subsequently, Transactions 23 and 28 records during the issuance of insurance products their
creation on-chain together with the creation of policy quota. The Transactions 26 and 27 record accepted capacity deals and counteroffers respectively on the stage towards looping risk coverage by a syndicate. The latter lifecycle stage involves the biggest amount of on-chain transactions. First, Transactions 29 to 33 record the issuance of policies, transfer of premiums, submission of claims and additional claim evidence, and the transfer of BLCK tokens between wallets respectively. Additionally, Transactions 44 and 45 record during risk coverage the approval of the syndicate members and virtual insurance company. In case conflicts occur pertaining to VICs or insurances, Transactions 48 and 49 record the minting and burning of BLCK tokens that is required in conflict resolutions finding. For a non-disruptive rollback to adjust a VIC establishment, Transaction 25 records the removal of members from a company team. Finally, the same Transaction 25 also removes members when the lifecycle termination point is reached.

Matching Lifecycle for Syndicates and Virtual Insurance Companies

The final lifecycle in Figure 10 shows the matching of syndicates and the insurances of VICs that commences with adding insurances to a pool for consideration. The gray sub-process labeled issue insurance products is part of the Figure 9 lifecycle. The insurances are then proposed to the syndicates and passed on to the gray sub-process labeled consider insurance product by a syndicate of Figure 8. Next, a bidding process commences where syndicates and VICs engage in matching attempts that eventually succeed in several cases. Post-matching, capital provided by the syndicates covers risks of matched with insurances. Figure 10 depicts that this coverage affects the lifecycle of Figure 9, as the gray looping sub-process shows with the label risk coverage.

If a conflict occurs during the risk coverage with capital, the matching conflict sub-process exchanges facts with the gray sub-process of VIC/insurance conflict from Figure 9. There are two possible outcomes to such conflicts, one being non-disruptive mitigation due to fixing a respective conflict. On the other hand, a conflict results in a disruptive resolution that requires the splitting of a matched insurance with a syndicate.

The lifecycle in Figure 10 shows additional on-chain transaction IDs that are part of Table 1. Transaction 23 is affiliated with adding insurances to the available pool and records on-chain the creation of a product. Next, for proposing insurances to syndicates, Transaction 6 records offered capacity deals of syndicates and Transaction 36 records on-chain the updated service-fee schedule. After that, during the bidding stage, Transactions 26 and 27 record the acceptance of of capacity deals and also of corresponding counteroffers respectively. The subsequent lifecycle stage of concretely matching insurances with syndicates comprises the biggest affiliated set of on-chain transactions. Transaction 26 records first the accepted capacity deal again, followed by Transaction 37 for recording on-chain payed service fees. Transactions 40 and 41 then record atomic swaps for moving tokens through gateways and approving the syndicate respectively and finally, Transactions 46 and 47 record the transfer of BLCK tokens between wallets that respectively involve different sets of stakeholders. Next, the risk coverage of an
insurance by capital involves Transaction 7 for recording the allocated reserve funds for a capacity deal and Transactions 43 and 39 for recording the adding of brokers and updating the service-fee schedule. Matching conflict notification merely involves Transaction 27 for recording a counteroffer capacity deal. If the match can be fixed with a non-disruptive mitigation then Transaction 38 records on-chain submitted bids while Transactions 48 and 49 record the minting and burning of BLCK tokens respectively. Finally, when only a disruptive splitting of a match is the last option, Transactions 20 and 21 record on-chain the creation of a voting ballot and of concrete votes on that splitting decision respectively.

Fig. 10: Lifecycle protocol of insurance product matching with syndicates.

6 Token Structure and -Economics

Derived from the requirements, the platform architecture and the governance lifecycles above, we are able to define in further detail the token structure and -economics. Thus, Section 6.1 explores in more details the differences and intersections of the two token types the BlackInsurance platform employs. Next, Section 6.2 explores in detail the concrete token economics.
6.1 Two Types of Tokens

The Black Foundation issues two types of tokens: BLCK as a utility token and BST as a security token. BLCKs can be purchased and owned by any user account and is used to pay for platform services without any linkage to profits on the platform. BST, on the other hand, can only be purchased and owned by verified accredited investors. They are structured securities issued on behalf of syndicates to earn profit. The amount and variety of BST depends on the preferences and intentions of parent syndicates. BLCK are issued on a public blockchain and BST on Hyperledger Fabric. Both types of tokens are implemented as smart contracts and comply (compatible) with ERC20⁴ and ERC223⁵ interface specifications.

Black Syndicate Tokens (BST)

The BlackInsurance platform allows investors to co-create investment funds called syndicates. Investors receive tokens from a syndicate in exchange for their investments. These BST are a special class of tokens that are issued when syndicates are launched, or when they are distributing profits to investors. Each syndicate issues their own BST with a unique name and risk-reward profile.

In its most basic form, a BST behaves as a dividend stock, where holders are eligible for dividends paid out in BLCK and amounting to a proportional share of net income of a syndicate. In a more sophisticated case, a syndicate operator is able to define a custom cashflow model for a syndicate token, thereby creating structured financial instruments featuring advanced risk-reward profiles. These instruments may be of arbitrary complexity limited only by the imagination of syndicate operators. They may also offer profits for distribution as bank transfers. Investors are able to view, compute, and project the profitability of any BST before deciding to invest, as cashflow models are accessible to all platform users for analysis and evaluation. BST exist and pay dividends on the BlackInsurance platform only, as moving them through a token gateway out to the public strips them of any special smart contract functionality. BST tokens do not grant any voting rights in a syndicate.

When a syndicate is defining the terms of its BST, it specifies under what conditions the syndicate dissolves and returns funds to the BST holders. These "liquidation events" need to account for regulatory requirements of holding funds in reserve for many years in the event that late claims are submitted after the syndicate closes. BSTs are considered "security tokens" because they accrue profits to their holders and they must be registered with regulators in any jurisdiction in which they are sold. BSTs only represent a claim against the funds controlled by a syndicate and will not give investors any other rights in the ongoing operations of syndicates. In terms of subordination, policyholders have a first claim against funds held by a syndicate if they have incurred damages for which they have policy coverage. After all policy claims, remaining funds are distributed as profit to brokers according to the profit distribution share agreed

⁴ https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md
⁵ https://github.com/Dexaran/ERC223-token-standard
between broker and syndicate, then to the syndicate operator, and finally to the investors. BSTs are strictly investments and cannot be used to pay platform services.

BSTs can only be purchased and owned by verified accredited investors. The Black Foundation is required to maintain KYC-compliance for any account that wants to trade this class of tokens. A prospective investor needs to apply for verification by the Black Foundation and provides all required evidence of investor accreditation. BST can be purchased by investing in a syndicate as a primary market, or by trading with other BST investors on the private platform and on a public blockchain as a secondary market. We allow verified and compliant exchanges to take custody of BST so they can create more liquid markets for BST traders and investors.

Black Platform Tokens (BLCK)

BLCK are used by platform users to execute transactions, pay service fees and to participate in syndicate lifecycle governance. Transactions are enabled by direct token transfers between platform accounts and the exchange of tokens to fiat and crypto currencies and vice versa. These transactions take place on-, or outside the BlackInsurance platform. It is possible to purchase BLCK tokens directly from Black Foundation at the moment of need, or to pre-purchase them on an external crypto exchange, if a platform user wish to do so.

Platform service fees are calculated in the same currency that the insurance premiums are denominated in. When a user makes a payment, the amount of BLCK to charge is calculated based on the respective primary market price of BLCK in the insurance premium currency, for example:

- A broker wants to transact $1,000,000 of gross written premium on the BlackInsurance platform. Blacks agreed services fee is 5%, including 20% discount, of gross written premium. The total service fee payable is, therefore, $50,000 (5% of $1,000,000).
- At that moment, BLCK costs $320 in the primary market. The brokers BLCK account is credited by 156.25 BLCK ($50,000 / $320).
- If the same transaction reoccurs later when BLCK costs $500 in the primary market, then the brokers BLCK account is credited by 100 BLCK ($50,000 / $500).

Participation in syndicate lifecycle governance is possible via staking BLCK tokens at syndicate formation time by syndicate originators. This staking mechanism prevents fraudulent- or unfaithful behavior by syndicate co-creators, guarantee trustworthy lifecycle management and incentivize long-term commitment to syndicate operations. Staking BLCK tokens gives right and voting power to participate in key decision making and syndicate steering at different points of the syndicate lifecycle, spanning from syndicate formation to maintenance and dissolution.

BLCK does not represent any shareholding, participation, right, title, or interest in the Black Foundation, its affiliates, or any other company, enterprise,
or undertaking. Nor do BLCK entitle token holders to any promise of fees, revenue, profits, or investment returns and is not intended to constitute securities in any relevant jurisdiction. To finance the launch phase of the project, the holding company Black Foundation OÜ offers the opportunity to purchase via an ICO 316,250,000 of 575,000,000 BLCK tokens released by a smart contract.

6.2 BLCK Token Economics

Since BST are security tokens, they are essentially structured financial instruments. Their purpose on the platform is to be a digital representation of a specific investment in a specific syndicate. Therefore, the value of a BST token reflects the cashflow model of a respective syndicate, an investment’s revenue, or profit distribution scheme, as well as specific terms and conditions of its existence, operation and dissolution. On the other hand, the BLCK token is an internal pure utility token intended solely for use on the platform and for platform-specific purposes. The token is tightly integrated into the underlying blockchain-based platform infrastructure and its business processes supported by smart contracts.

There must exist technical-, business- and economic reasons for using a dedicated internal utility token. Besides convenience as an internal means of value storage, exchange and accounting, the reasons for using BLCK include the following:

- Potential for higher extensibility of the platform:
  By using a standalone BLCK token for platform operations, we create a potential for higher extensibility of the platform. This allows to eliminate dependence on non-generic features of other crypto tokens, as well as implement our own platform-specific features directly on the token level, if such a necessity occurs.

- Potential for higher portability of BLCK and lower dependence on the blockchain platform:
  Using a standalone BLCK token provides freedom of choice of a new generation of blockchain platforms and smart contract languages when they are in production, should they prove to be more preferable for platform performance and operations.

- Avoiding undesired dependency on volatility of a 3rd party crypto token, or currency:
  If a 3rd party crypto token, or -currency are used instead of BLCK, that would create a dependency on its volatility that does not reflect BlackInsurance business-, or platform performance. Such dependency is therefore strongly undesirable.

- Clear utility function of BLCK:
  The usage of BLCK in syndicate governance structures in addition to transaction processing and service fees, justifies an own internal utility token with intrinsic value.
6.2.1 Utility Token Value-Formation Factors

Before discussing the BLCK token model and its economic behavior, we explore the generic utility token value-formation factors. There are two groups of value-formation factors for a utility token with intrinsic value. One group is market expectation factors and another group is utility factors. Expectations of the Black community are driven by the tokens intrinsic value at each point in time, anticipated future token value, current media coverage, as well as current and future performance of the BlackInsurance platform and Black Foundation. Utility factors are mainly dependent on how much and how actively the token is used in platform operations, and, respectively, on the market adoption of the platform in general. We expect that in the very early phase of the platform lifecycle, i.e. during platform formation and early market adoption phases, the influence of the expectation factors will be higher than of the utility factors. In the medium-term, the weight of both groups of factors are comparatively equal, while in the long-term, when the platform is fully realized, the utility factors increase in importance.

![Figure 11: Qualitative representation of anticipated weights of token value-formation factors at different stages of the platform lifecycle.](image)

Figure 11 presents an approximated qualitative representation of anticipated weights of token value-formation factors at different stages of the platform lifecycle. The y-axis corresponds to the weight. The x-axis corresponds to time, where the left-hand side of the axis corresponds to time zero when the token is issued and the right-hand side corresponds to time \( T_m \) when the platform is mature. The blue line corresponds to the relative weight of the utility factors; the red line corresponds to the relative weight of the expectation factors.

**Expectation Factors**

Expectations can be rational and irrational. Observations show that, currently, community expectations about the value of the majority of crypto tokens and currencies at times irrational. Still, provided that a utility token has a clear use case and applicability on the platform, it is possible to estimate its future intrinsic value and discount it to present time using the desired cost of capital.
In the BLCK model, it is the supply and demand of tokens for transaction and governance purposes that determine the intrinsic long-term value of a token. Expectation factors in our model, therefore, reflect the community’s expectation of and perception about the performance of the platform.

**Utility Factors**

Utility factors reflect the necessity to use BLCK on the platform for platform-specific purposes. These purposes include temporary value storage, payments on the platform (transactions), and staking tokens for platform and syndicate governance purposes. We have identified four governance-dependent categories: syndicate, insurance product, adoption of an insurance product by a syndicate and brokers e-community. For each of these categories, staking is necessary on all the lifecycle phases: formation; e.g., to avoid syndicate formation abuse and to reach consensus about syndicate operation templates; maintenance, e.g. to prevent unfaithful behavior and to resolve operational conflicts; and dissolution to eliminate potential dissolution conflicts. From the utility perspective, BLCK value is therefore determined by supply of and demand for tokens from platform participants: syndicate owners, insurance brokers, and so on.

### 6.2.2 BLCK Token Model

Since BLCK is a utility token that is available to the public on the open market via crypto exchanges, the value and economic behavior is determined by supply of and demand for the token, as well as token velocity, according to the equation of exchange in monetary economics and quantity theory of money:

\[ M \times V = P \times T \]  

where

- \( M \) is total nominal amount of token supply in circulation,
- \( V \) is velocity of tokens (average frequency with which a unit of token is spent during a unit of time, determined by transaction speed and willingness to transact),
- \( P \) is a general token price level, and
- \( T \) is a total nominal amount of transactions per unit of time.

At equilibrium, supply is equal to demand. It is easy to show that, dynamically, equilibrium between supply and demand is achieved according to the equation

\[ dM + dV = dP + dT \]  


where

- $dM$ is the growth rate of nominal amount of token supply (or, alternatively, growth rate of token demand with a negative sign),
- $dV$ is the growth rate of token velocity,
- $dP$ is the growth rate of general price level (price inflation or deflation), and
- $dT$ is the growth rate of transaction volume on the platform.

We show that in an economy with a utility token and some base currency, where goods such as assets, products, services, etc., are denominated in the base currency but transactions happen using a utility token, the dependency between the prices of both is the following:

$$P_t = \frac{T_{base} \times D}{M}$$

where

- $P_t$ is the price of 1 token in base currency,
- $T_{base}$ is the transaction volume in base currency per unit of time (e.g. in a second, day, year),
- $D$ is the transaction cycle duration in units of time (e.g. in seconds, days, years), i.e. how long does a unit of token is being held for transaction purposes by a party.
- $M$ is the amount of tokens in circulation

By combining utility and expectation factors, we may obtain a formula for the present value of a token in base currency in year $N$ as the discounted future utility value of a token when utility factors play the major role. Alternatively, by making certain assumptions about expectations, it is possible to calculate present value of a token in year $N$ as a weighted sum of expectation-based price component and utility-based price component.

3. Token Model Components

From the nature of BLCK token, i.e., for which purpose and how it is used on the platform, we obtain token model components that directly influence token supply, demand and velocity, and therefore, the utility value of a token. They are speculation and investment component, governance component, convenience and fee mitigation component, and transaction volume component.

Speculation and Investment Component

This component reflects market expectations of the future platform development and BLCK demand and the willingness of investors to hold tokens for investment and speculation purposes in response to their expectations. Its influence is strongest in the early period of the platform lifetime when price is formed based on expectations, R&D and marketing performance, but not yet on BLCK
utility. Token holders purchase and hold tokens for speculation and long-term investment. Small amounts of tokens are used for transactions directly on the platform. This keeps supply low and demand high, which, respectively, positively influences the price if expectations are positive. When the platform matures, the role of this component in price formation will decrease.

**Governance Staking Component**
This component reflects the need of different stakeholders (namely, the ones that take role in syndicate governance) to hold BLCK for staking purposes as a guarantee of their honest behavior, long-term commitment to syndicate operations. It also reflects the occasional need to deposit tokens as a warrant for potential conflict resolutions. The influence of this component becomes stronger as the number of syndicates and their size and complexity increases. Staking of BLCK helps to decrease supply and increase demand. Furthermore, it allows to draw token velocity down, which contributes to positive price development.

**Convenience and Fee Mitigation Component**
This component reflects the willingness to hold BLCK instead of fiat, or other crypto tokens, for convenience and in order to avoid exchange fees by accepting market price risk. Early in the platform lifetime, when market price risk is high due to potentially high price volatility, platform participants are expected to prefer to exchange BLCK to fiat, or other crypto-currencies as quickly as possible. Still, as the transaction volume increases and the platform matures, it is reasonable to keep an amount of BLCK on the balance sheet. Therefore, the influence on this component increases as the platform matures. It decreases token velocity that positively affects the price.

**Transaction Volume Component**
This component reflects the effect of transaction volume. The transaction volume determines how many transactions are executed on the platform per unit of time. The more transactions are executed, the more demand there is for the tokens used in transactions as one of the major price formation components and driving factors. In our model, we use gross written premium as a proxy for this parameter by assuming that the higher the gross written premiums are, the more tokens are turned over in various transactions on the platform per unit of time.

The, BLCK economic behavior is subject to the balance between different parameters that represent the mentioned components. These parameters are the amount of unlocked tokens, percentage of tokens held for investment, amount of tokens staked for governance, token possession duration for a transaction cycle, contribute to the supply and demand balance and token velocity to keep supply constrained, increasing demand and decreasing velocity. The interplay of these components, a well-defined utility of BLCK on the platform and its associated intrinsic value contribute to higher economic sustainability of BLCK compared to crypto tokens that lack a clear connection between their utility and intrinsic economic value.
7 Rapid Deployment Feasibility Evaluation

For the BlackInsurance platform architecture in Section 4, we evaluate in a paper-based way the availability of pre-existing and emerging technologies for a rapid system deployment. This technology stack also comprises blockchain products and projects that allow satisfy specific requirements that BlackInsurance platform must satisfy. In case no existing projects are available, research initiatives are discussed that promise to fill potential gaps in the technology stack. Note that the suggestions for rapid system deployment are tentative.

The remaining structure is as follows. Section 7.1 discusses smart-contract platforms and their verifiability. Next, Section 7.2 discusses options for data management on blockchains. Section 7.3 presents research work and industry practices about the authentication of stakeholder identities for secure BlackInsurance platform participation. Finally, Section 7.4, addresses in further detail ongoing relevant research results.

7.1 Smart-Contract Platforms

For the BlackInsurance platform, the earlier discussed Ethereum is a potential candidate. While PoW is sensible for very critical and high value transactions that require on-chain consensus establishment, a significant Ethereum disadvantage is that PoW creates performance and scalability challenges for the BlackInsurance platform if employed for all transactions. As Table 1 shows, the BlackInsurance platform requires a considerable set of blockchain transactions to immutably store critical collaboration events that are potentially litigation relevant.

With respect to a smart-contract platform that successfully validates transactions with PoS, the equally non-permissioned Qtum\(^8\) smart-contract platform uses simple payment verification (SPV) \(^{36}\) and the unspent transaction output protocol (UTXP) \(^{19}\) for supporting lite wallets that only manage transaction headers instead of the entire transaction body. Consequently, Qtum distributed applications (Dapp)\(^9\) run on mobile devices that have low storage and processing capacity. On the other hand, Ethereum has not succeeded in a PoS-version delivery and a realization date is unclear. While PoS is already fully functioning in the Qtum system, both Ethereum and Qtum offer Solidity as a smart-contract development language, which allows for easy adoption by experienced Solidity developers.

Hyperledger \(^{47}\) is a permissioned alternative to Ethereum and Qtum for which Fabric \(^{14}\) is a modular implementation for smart contracts that also offers pluggable extensions for additional functions. More recently, EOS \(^{28}\) is a novel blockchain with the objective of solving the performance issues and lack of self-governance mechanism of other smart-contract systems. Developed as an operating system, EOS aims to allow for building large-scale Dapps comprising delegated PoS (DPoS) \(^{50}\) for consensus finding where stakeholders elect delegates

---

\(^8\) https://qtum.org/ens

\(^9\) https://qtumeco.io/dapps
for generating and validating a block. While the ideas expressed in the EOS whitepaper [28] promise the delivery of the most advanced smart-contract and blockchain-based operating system, the introductions has been marred by many complications\textsuperscript{10}, worst of all, EOS reveals itself even as a system with centralizing tendencies\textsuperscript{11}. An emerging alternative is IOTA\textsuperscript{12} that uses directed acyclic graphs (DAG) instead of a blockchain to allow for highly scaling simultaneous transactions with fast confirmation times irrespectively of their size. Also Hashgraph\textsuperscript{13} is an alternative DAG system comparable to IOTA.

Since the BlackInsurance platform with the architecture of Figure 7 requires a considerable amount of smart contracts, it is important to check with tool support the soundness of these smart contracts before enactment and also eliminate concurrency conflicts, or dependability issues [5]. We discover several pre-existing systems for evaluating contracts that are unfortunately still very unsatisfactory. The available beta-version of the Securify\textsuperscript{14} online service claims to formally verify Solidity smart contracts for insecure code that comprises critical security issues. Securify also still lacks good documentation does not allow to estimate how the online service checks formal properties. Based on an online Securify example, we discover the implemented heuristics that are available in the beta-version check for transaction recordings, recursive calls, insecure coding patterns, unexpected Ether flows and the use of untrusted inputs in security operations. Equally not mature are the Embark-framework\textsuperscript{15} and Populus\textsuperscript{16} for smart-contract development and deployment are currently not mature enough for satisfactory formal verification and evaluation.

A remedy for overcoming the currently existing lack of verification tools for Solidity is to consider a platform that uses functional programming language for smart-contract development. For example, Cardano\textsuperscript{17} uses functional Haskell [44] as programming language for which verification tools are available [33]. Also Aeternity\textsuperscript{18} uses Erlang [34] that also has been well researched [26] for verification. Even the use of matured programming languages such as Java for the NEO\textsuperscript{19} platform pose the opportunity to consider pre-existing verification tools [18]. Such verifiability of smart contracts is specifically important for cases where legal compliance must be demonstrated.

\textsuperscript{12}https://iota.org/
\textsuperscript{13}https://hashgraph.com/
\textsuperscript{14}https://securify.ch/
\textsuperscript{15}https://github.com/iurimatias/embark-framework
\textsuperscript{16}http://populus.readthedocs.io/en/latest/
\textsuperscript{17}https://www.cardano.org/en/home/
\textsuperscript{18}https://aeternity.com/
\textsuperscript{19}https://neo.org/
7.2 Data Management with Blockchains

For blockchain-based mass data management of the BlackInsurance platform, the InterPlanetary File System\(^{20}\) (IPFS)\(^{11}\) is the suitable pre-existing system. IPFS is a peer-to-peer (P2P) and distributed hypermedia protocol where the content is addressable with blockchain use. IPFS is open source, highly performant and a provides a decentralized block-storage model. Thus, IPFS comprises hyperlinks that address data sets in a block-storage model where data is distributed across several computers.

BigchainDB\(^{35}\) is a suitable system for complex operations on large datasets and also for constructing profiles. The main features of BigchainDB are that it is a blockchain database with immutable decentralized control where it is possible to create and move digital assets. An advantage BigchainDB for the BlackInsurance platform is the capability to integrate other decentralized blockchain systems such as IPFS, Ethereum, Qtum etc.

7.3 Identification and Authentication

For the BlackInsurance platform, it is important that the authenticated identity of participating stakeholders is clarified. From the research-side, recent publications about the so-called Authcoin framework\(^{32}\) comprises flexible means of authenticating an individual’s identity via issued challenges that must be satisfied by the claimant of an identity. Important is also the identification key of participants on the BlackInsurance platform. For example, it is possible to split identification keys and store the parts at different locations. ZeroPass\(^{21}\) is a blockchain service that performs the splitting of identification keys. One key half remains with the user while the entire private key never enters the ZeroPass server. The ZeroPass located key half is cryptographically secured. Additionally, the user may share his half of the key with third parties via granted access rights.

An alternative to key-splitting for the BlackInsurance platform is the use of multi-signature in which multiple users sign a transaction ahead of the broadcasting to the blockchain. Several existing crypto wallets\(^{22}\) already use multisignature mechanisms. For example, when an investor registers on the BlackInsurance platform, he generates a second private key at the same time that is stored on the BlackInsurance platform. Thus, a transaction consequently requires a multi-signature of both the BlackInsurance platform and the investor. Employing multiple servers for storing several private keys respectively yields substantially enhanced security provisions\(^{29}\). Since such distributions of multiple keys may result in multi-signature latency, defined rules may clarify, e.g., when three servers and an investor hold copies of private keys, having the investor and only two servers sign a transaction is sufficient.

\(^{20}\) https://ipfs.io/

\(^{21}\) https://www.zeropass.io/

7.4 Research Work

The BlackInsurance platform aims for satisfying the quality goal of high automation, as the goal models in Section 3 depict. Consequently, also the architecture in Figure 7 shows that artificial software agents operate in a decentralized way with a high degree of scalability. We assume that the dedicated agent types for the BlackInsurance platform are created with a limited lifecycle that ends when the assigned task is successfully completed, or canceled. Briefly, an artificial agent [43] are either purely software, or hybrid as a combination of hardware together with embedded software. In both cases, an artificial agent comprises a set of sensors to perceive contextual events, an internal knowledge base that stores complex information structures, a controller to support algorithmic reasoning with input from the knowledge base and finally, actuators that project defined actions into the context an agent interacts with. To implement multi-agent systems, the so-called JADE platform [9, 10] is well established for developing so-called believe-desire-intention agents for the programming of anthropomorphic properties.

With respect to further research trends that are relevant for a full implementation of the BlackInsurance platform, the verification of smart-contracts is a topic of recent publications. The discussed technology stack of this section suggests that several blockchain applications with diverse smart-contract types are part of a BlackInsurance platform deployment. Citation [?] discusses that the development of secured smart contracts is challenging since programs and pseudonymous users call public methods of third-party programs. Thus, the combination of trusted and untrusted programs results in a lack of security. For smart contracts that are written in Solidity, a security check is translated to a functional programming language termed F* with which it is possible to analyze and verify the functional correctness and runtime safety of smart contracts.

Given the currently limited availability of Solidity smart-contract verification tool support, the deployment of a BlackInsurance platform implementation requires the strict adoption of best development practices. In [22], a set of heuristics aim at decreasing common development mistakes. Observing students who develop faulty smart contracts in programming classes is the source for specifying best-practice heuristics. Empirical studies show that applying these heuristics results in a considerable reduction of security issues in smart-contract code. These pitfalls are errors in encoding the state machines, failures to use cryptography, misaligning incentives and finally Solidity-specific mistakes such as call-stack-, blockhash- and incentive bugs.

8 Conclusion

This whitepaper presents the BlackInsurance platform that uses blockchain technology and smart contracts for the rapid deployment of syndicate-provided capital for the coverage of risks that insurance products cover. For that, we provide a running case about the traditional insurance-industry problems, followed by the
BlackInsurance requirement goal model that also arrange the relationship of stakeholders with the platform. We next derive the BlackInsurance architecture from the goal models where the data-exchange channels are specified between architecture components and stakeholders. Next, the dynamic system-protocol study commences with the set of component-specific on-chain transactions and three governance lifecycles for the establishment of syndicates and insurance-products issuing virtual insurance companies, and the matching between syndicates and respective insurance products. These governance lifecycles allow for the assignment of the on-chain transactions. We study also in detail the token structure and economics based on the requirements of the BlackInsurance platform and its dynamic behavior protocols. Finally, we evaluate the availability of pre-existing technologies for the rapid implementation and deployment of the BlackInsurance platform focusing on important aspects such as smart-contract platforms, blockchain-based data management, solutions for identity authentication and finally, available research results realizing a high automation degree with the use of artificial agents and research that backs the lacking availability of tools for assuring the security of smart contracts.

To answer the questions posed in the introduction of this whitepaper, the key stakeholders of the BlackInsurance platform are the syndicate manager who takes care that investors are gathered and a syndicate operator assigned for the formation of a syndicate. Related to the latter are underwriters who define underwriting rules for a syndicate and issues policies. A claims handler approves or rejects claims, defines claims rules and is also involved in paying compensations to claim funds. The profit allocator defines the syndicate-related profit-sharing rules and related BST token holders are affiliated to votes for ballots related to syndicates. Brokers manage virtual insurance companies where syndicate operators are involved in receiving risk-coverage capacity. Additionally, Black platform administrators are involved in recording payments related to virtual insurance companies. The updating of service-fee schedules for token exchanges involves too the black platform administrator. Paying black services in BLCK tokens requires the syndicate operator, broker and investor. Also the actual exchange of BLCK and BST tokens to fiat and vice versa require the syndicate operator, broker and investor. The latter three stakeholders together with the black platform administrator may view key performance indicators on a dashbord. Finally, the black platform administrator manages the BlackInsurance platform where related the BLCK token holder is associated to casting votes for ballots.

The goal-model derived system architecture shows that the component for syndicate management is the complex one comprising the biggest amount of embedded sub-components, associated stakeholders and information-exchange channels with other platform components. Again derived from the goal model, the syndicate manager also involves an intensive degree of tokenization. Concretely, the contained components of investment manager, profit-share manager, policy administration and ballot manager involve tokenization. Intensive information exchange the syndicate conducts with the component of virtual insurance company
manager. The latter is tokenized in only one embedded sub-component of policy- and claims manager. The component of black platform manager that is tokenized in the ballot-manager sub-component. The black platform manager facilitates the matching between established syndicates and insurances that are issued by the virtual insurance company. Next, the sub-components of the tokens exchanger are all tokenized. Specifically, the sub-component of the exchange manager interfaces with the syndicate manager, virtual insurance company manager and the black platform manager for exchanging BLCK and BST between each other and with fiat currency.

The study of dynamic on-chain exchange protocols between stakeholders and components of the BlackInsurance platform shows that the syndicate manager is also the most complex in terms of on-chain transaction amount. The virtual insurance manager and black platform manager are nearly equally intensive in terms of on-chain transactions. Together with the tokens exchanger, we estimate there is a need for currently conceptually listed 49 on-chain transactions, which any other transactions should be kept off-chain and be managed by the artificial agents of the BlackInsurance platform. The governance lifecycles for the syndicate and the virtual insurance company are quite similar in that both commence with developing collaboration models first that are then populated and negotiated by the participants with the possible outcomes of agreement, or disagreement, while a syndicate member may also issue a counteroffer. In both cases an establishment process is required after an agreement and the government lifecycles manage occurring conflicts either with non-disruptive rollbacks to a re-establishment, or a disruptive rollback for a re-negotiated establishment. Finally, the matching lifecycle pools issued insurances that then syndicates consider for risk-coverage with capital. The syndicate and virtual insurance company bid until a matching occurs and capital coverage for insurance risk commences. Occurring matching conflicts can either be fixed with non-disruptive mitigation, or a split occurs.

Finally, we show that a deployment of the BlackInsurance platform requires a technology stack that comprises diverse blockchain systems. Several smart-contract platforms are available that have respective advantages and disadvantages. Essential is to understand with which smart-contract platforms verified compliance is achievable versus others that offer for certain aspect of the BlackInsurance architecture suitable consensus mechanisms, mobile usability, and so on. We also consider it important to use separate blockchain-systems for data management and identity authentication. It is not credible to assume that only one blockchain system is able to cover all aspects of the BlackInsurance platform. Finally, we stress that the topic of smart-contract verification for soundness and security is still an ongoing research issue as the currently available tools are not satisfactory.

Acknowledgement

We thank the Dimitar Petkov for his valuable input in developing the token structure.
References


