Conflict-Resolution Lifecycles for Governed Decentralized Autonomous Organization Collaboration

Alex Norta, Anis Ben Othman, Kuldar Taveter
Tallinn University of Technology, 12618 Akadeemia tee 15A, Tallinn, Estonia
alex.norta.phd@ieee.org, anis.ben@gmail.com, kuldar.taveter@ttu.ee

ABSTRACT

Recent blockchain-technology related innovations enable the governance of collaborating decentralized autonomous organizations (DAO) to engage in agile business-network collaborations that are based on the novel concept of smart contracting. DAOs utilize service-oriented cloud computing in a loosely-coupled collaboration lifecycle with the main steps of setup, enactment, possible rollbacks and finally, an orderly termination. This lifecycle supports the selection of services provided and used by DAOs, smart contract negotiations, and behavior monitoring during enactment with the potential for breach management. Based on a sound understanding of the collaboration lifecycle in a Governance-as-a-Service (GaaS)-platform, a new type of conflict management must safeguard business-semantics induced consistency rules. This conflict management involves breach detection with recovery aspects. To fill the detected gap, we employ a formal design-notation that comprises the definition of structural and behavioral properties for exploring conflict-related exception- and compensation management during a decentralized collaboration. With the formal approach, we generate a highly dependable DAO-GaaS conflict model that does not collapse under left-behind clutter such as orphaned processes and exponentially growing database entries that require an unacceptable periodic GaaS reset.

Keywords

Decentralized autonomous organization, conflict resolution, e-governance, smart contract, open cloud ecosystem, service orientation, business process, Industry 4.0

1. INTRODUCTION

The emergence of service-oriented cloud computing (SOCC) [36] promises for companies an accelerated e-governance for cross-enterprise-collaboration (CEC) [5] with a seamless, adhoc integration and coordination of information- and business-process flow. The latter orchestrate and choreograph heterogeneous legacy-system infrastructures [20]. Such governance automation of cross-enterprise collaboration enhances overall efficiency and effectiveness. Additionally, a trend-reinforcement occurs with the concept of so-called decentralized autonomous organizations (DAO) that are powered by smart contracts [4, 30] to form agreements with people via the block chain [16]. The ontological concepts and properties for the design of smart-contracting systems [26] we derived from legal principles, economic theory, and theories of reliable and secure protocols. The smart contract itself is a computerized transaction protocol [29] that executes the terms of a contract. The blockchain is a distributed database for independently verifying the chain of ownership of artifacts in hash values that result from cryptographic digests [25].

The SOCC-paradigm facilitates a loose coupling and highly dynamic establishment in the governance of business collaboration. Services are self-describing, business-process grouped logical manifestations of physical resources, i.e., as a set of actions [35, 31] that an organization executes and exposes to the web. To achieve non-repudiation in CEC, the registration of business transactions is of major legal importance for organizations. A business transaction [14] is a well-defined business-function driven consistent change in the state of a business relationship.

While each DAO holds its own business transaction [12, 32], for CEC-governance a transaction conflict-resolution concept is important to ensure collaboration reliability. With the complexity involved, no single transaction model is able to meet all requirements. Instead, it is necessary to cross-organizationally establish transaction frameworks in a way that does not force companies into disclosing an undesirable amount of business internals [5]. While conflict management is addressed on a collaboration-model level [17], it is also important to cater on a Governance-as-a-Service (GaaS) level for conflict management and -resolution. This paper fills the gap by investigating the research question how to govern in a dependable way the flow of business semantics in a meaningfully automated CEC-governance lifecycle? Note that meaningful automation in this sociotechnical context recognizes complex organizational work design with interaction between people and technology in workplaces. Thus, we assume humans want automation for tedious work but retain final decision-making power. Furthermore, dependable [2] means the components that are part of the governance-lifecycle are relied upon to perform exclusively and correctly the system task(s) under defined operational and environmental conditions over a defined period of time. Based on this main research question, we deduce the following sub-questions to establish a separation of concerns. What is the underlying CEC-governance lifecycle and which business semantics flows along it? When exceptional governance scenarios occur, what mechanisms exist in automated CEC for an orderly conflict resolving compensation-rollback and partial-, or complete lifecycle termination? What are the relevant system properties for successfully realizing the CEC-governance lifecycle with a Governance-as-a-Service (GaaS) platform in a Cloud?

The remainder of the paper is structured as follows. Section 2 provides additional information relevant for understanding the business-collaboration context. Section 3 shows the top-level of the formalized CEC-governance lifecycle in which service protocols are visible with their data-exchanges. Furthermore, in Section 4 we show the successful return of business semantics to earlier stages for compensating exceptions in transactional rollbacks within a CEC-governance lifecycle. This section also shows varying types of lifecycle terminations that leave the Cloud ecosystem behind in a clean state so that no clutter-accumulation necessitates a total reset. The latter is not an option when simultaneously business-critical collaborations are in progress. Section 5 shows the feasibility of the approach by listing the results from model checking that are instrumental for implementing a sound CEC-governance lifecycle with conflict-resolution provisions. Section 6 gives related work and finally, Section 7 concludes this manuscript by summarizing the research work, giving the contributions achieved and showing directions for future work.

2. BUSINESS CONTEXT

For comprehending the governance-lifecycle in the sequel, the following frameworks are important to comprehend. The essential way for DAOs-relationships is peer-to-peer (P2P) and therefore, we clarify the corresponding collaboration model in Section 2.1. Furthermore, as contracts are the foundation of business collaboration, we present in Section 2.2 pre-existing concepts and properties for smart contracting.

2.1 P2P-collaboration model

Pertaining to DAO-collaboration, Figure 1 conceptually depicts a configuration. The in-house process of a service consumer is a so-called business-network model (BNM) [27] in the P2P-case. A BNM captures choreographies that are relevant for a business scenario. A BNM contains legally valid template contracts that are service types with assigned roles. Together with the BNM, the service types with their roles are available in a collaboration hub that houses business processes as a service (BPaaS-HUB) [22] in the form of subset process views [5]. The latter address the need to semi-automatically find collaboration parties and learn about their identity, services, and reputation. A BPaaS-HUB enables speedy business-partner discovery and support for on-the-fly background checking with a matching of services.

On the external layer of Figure 1, service offers match with service types from the BNM identically to the contractual sphere of collaborating parties. Additionally, a collaborating

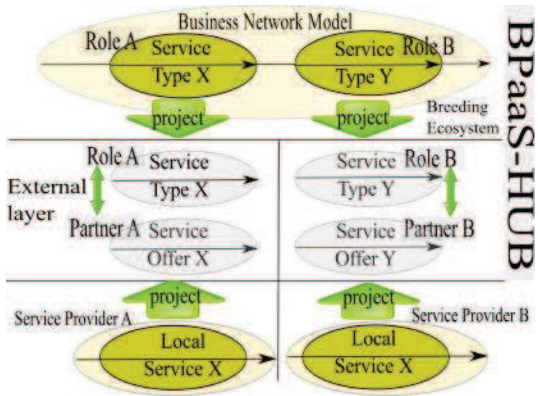

Figure 1: P2P-collaboration with eSourcing.

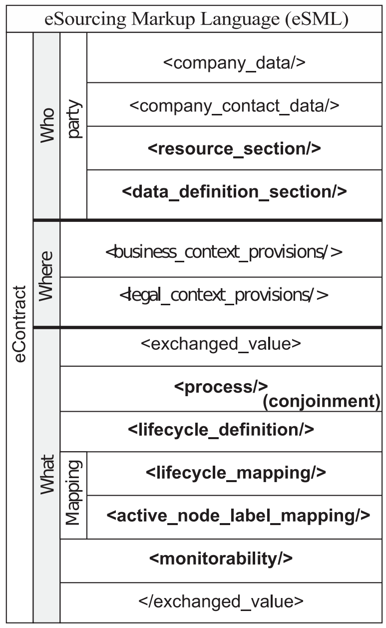

Figure 2: Smart contracting concepts and properties [23].

partner must match into the role specifics associated with a respective service type. We refer the reader to [5] for details about the the tree-based process-view matching to establish a DAO-collaboration configuration.

2.2 Smart contract

We show the top-level structure of a smart contracting language termed eSourcing Markup Language (eSML) [23] in Figure 2. The bold typed eSML-definitions in Figure 2 are extensions and modifications that are not part of the Electronic Contracting Markup Language (ECML) [1] foundation.

The core structure of a smart contract we organize according to the interrogatives *Who* for defining the contracting parties together with their resources and data definitions, *Where* to specify the business- and legal context, and *What* for specifying the exchanged business values. For achieving a consensus, we assume the *What*-interrogative employs matching process views that require cross-organizational alignment for monitorability We refer to [23] for more information about the smart-contracting ontology. Also note that we refer from hereon to a smart contract as an eContract.

We next discuss the DAO-lifecycles for the setup-, enactment- and orderly termination stages in terms of control- and data flow.

3. COLLABORATION LIFEYCLE

We formalize with Colored Petri Nets (CPN) [11] the governance lifecycle. CPN is a language for the design, specification, simulation and verification of systems and has a graphical representation with a set of modules, each containing a network of places, transitions and arcs. The modules interact through well-defined interfaces and the data elements of the overall governance lifecycle are declared for all refinement levels. We use CPN Tools[1] for designing, simulating, performance testing and verifying the models in this paper. Note that way we solve many dependability issues in the design of such a complex system [2]. Table 1 lists relevant token colors [18] with their hierarchic service-refinement availability mentioned in the left column (1 for the top level and 6 for the most detailed refinement). Token colors are present for all lower but not for any higher CPN-refinement-hierarchy levels. The fourth column explains the purpose of a token color for a lifecycle. The integer-type tokens mostly represent an identification number and string-type tokens are either eContract-negotiation outcomes or eContract proposals. Boolean-type tokens represent decision points in the lifecycle.

For the remainder, Section 3.1 shows the top-level formalization of the eContract-setup phase. Section 3.3 elaborates on the enactment phase of the DO-lifecycle and Section 3.2 givs the rollout of a decentralised governance infrastructure. Finally, Section 3.4 shows the governance termination. Note that due to page limitation, we only show a subset of models while we make the complete CPN-model available[2] and also fully document [18] all models.

3.1 Setup phase

The depiction in Figure 3 shows the governance lifeycle in the form of a CPN for forming an eCommunity [13] and starts with the creation of a BNM that contains service offers for validation against service types, and additionally, roles are assigned to the services. Concrete collaborating partners fill these roles during the eContract negotiation.

As Figure 4 shows in a limited screenshot of the actual CPN-module, the partners that slip into roles must vote on agreeing, or rejecting an eContract proposal that is based on a picked BNM. Rejection terminates the eCommunity while having all partners agree, results in a consensual eContract passed on to the next service. A third option during the negotiation phase is the proposal of a contract alternative.

If all partners agree during the *negotiate* stage, the eContract comes into existence that serves as a coordinating agent

level	CPN module	data property	description	type
1	eCommunity lifecycle	sO	service offer that fits a service type	integer
		sOs	service offer source for communication channel establishment	
		sOt	service offer target for communication channel establishment	
		pA	partner of an eCommunity	
		rO	role a partner can fill	
		eC	eCommunity identification	
		eCo	eContract based on which partners of an eCommunity transact	
		n,r,k,p,l,q,s	counter variables	
		assigned	service offer assigned to a service type	boolean
		processed	partner prepared for eContract counteroffer re-distribution	
		decision	for negotiated contract proposal (agree|disagree|counter)	string
		outcome	like decision, but input for eCommunity continuation or termination	
2	create	bNM	business network model that get populated with service types and roles	integer
		m	counter variable	
		sT	service type that populates a bNM	
3	populate	ch	channel of communication between services	integer
4	interoperability checking	rOt	role source for communication channel establishment	integer
		rOs	role target for communication channel establishment	
4	contract extraction	spec	specification of extracted eContract	string
4	agreement finalizing	result	whether all eCommunity partners agree on an eContract proposal or not	boolean
		distributed	contract distributed to partner	
4	disagreeing	z	counter variable	integer
		eCo_new	new eContract from a counteroffer to be negotiated	
2	perform	bnma	business network model agent	integer
		sE_l	local service of a respective eCommunity member	
		mO	monitor for observing policy adherance of eCommunity partners	
		sE	electronic service that is enacted	
		lp	local policy extracted from a local contract copy	
		lpnr	counter of local policies	
		s,x	counter variables	
		lC	local contract for respective eCommunity partners extracted from the eContract that coordinates the first	
4	contract establishment	insert	service inserting to local contract	boolean
		extracted	instances of contract copies for negotiation	
5	governance distribution	errorID	error identity	integer
		error	error for synchronizing main contract and local copies	boolean
5	prepare	eP	published endpoint for allowing services to communicate	integer
6	preparation error	prepErr	preparation error in the context of assigning an electronic service to a service offer	integer
		assignErr	assignment error in the actual assignment of an electronic service to a service offer	
		sEr	service error related to concrete electronic servce, e.g., deadlock	
4	operate	tc	termination criteria, either full for eCommunity or partial for disruptive partner change that rolls back to a negotiation stage	integer
5	enact	startErr	start error when a stopped service is re-started again	integer
6	policy service removal	pAnr	number of partners	integer
4	nondisruptively manage	vl	local vote for replacing a policy-violating partner or reconciling	integer
		lp1	another local policy	
		vote	for policy violation of partner to leave or stay in eCommunity	string
4	nondisruptively choose	pA_new	new partner to replace one who violated a policy	integer

Table 1: Data properties used in the DAO-lifecycle [18].

[28]. In the *enterprise infrastructure distribution*, local eContract copies come into existence for every eCommunity-partner

[1] http://cpntools.org/
[2] CPN-DAO collaboration: http://tinyurl.com/kq7qjvl

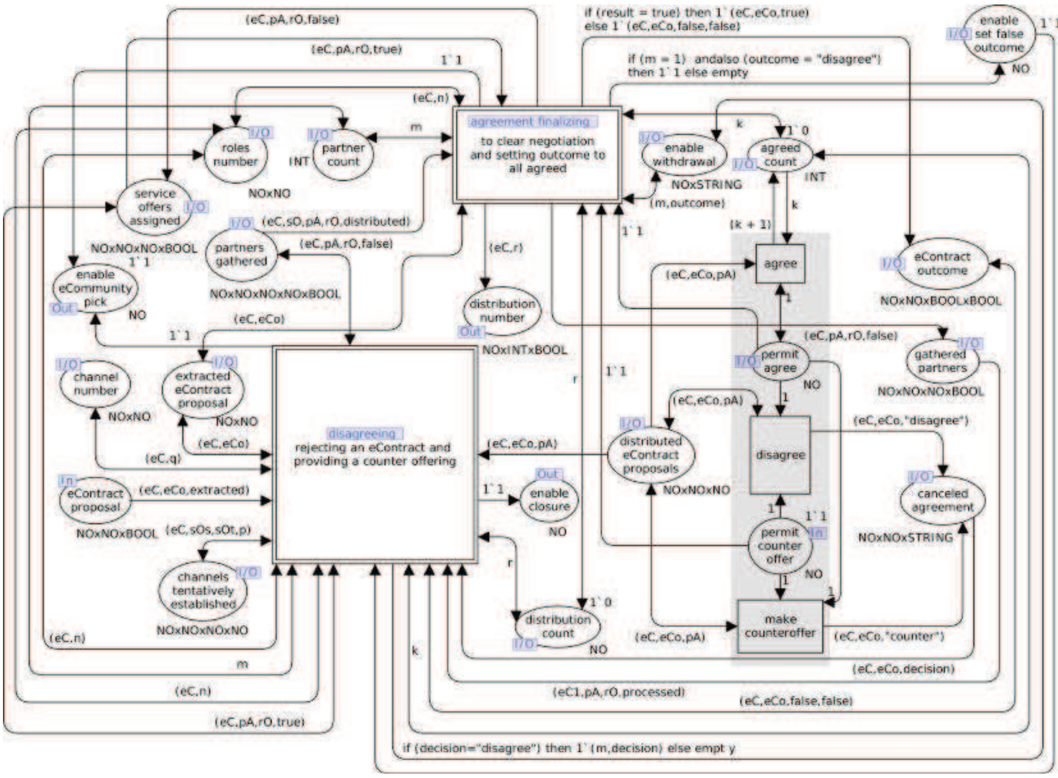

Figure 4: The decision phase in the ***negotiate*** **module [18].**

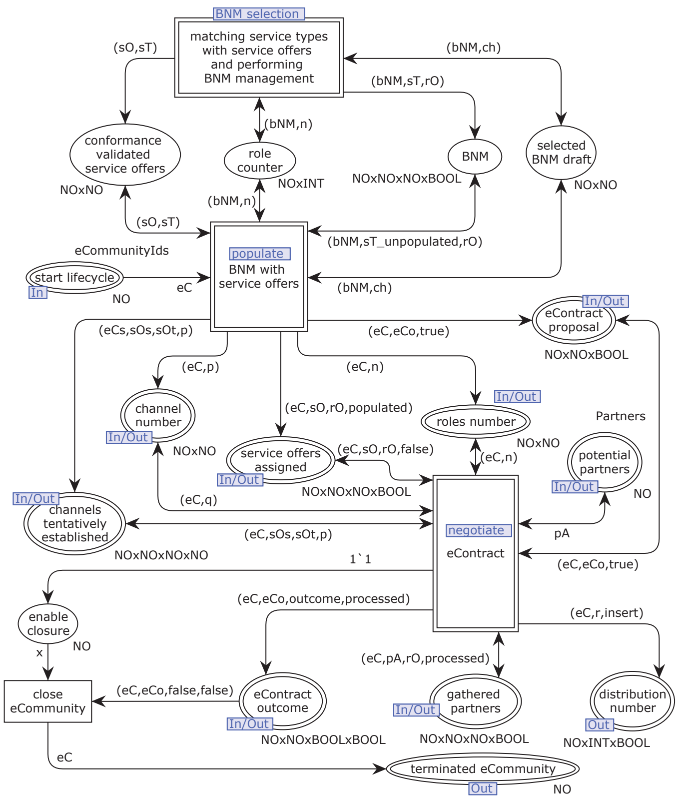

Figure 3: The setup phase of a smart contract in the ***create*** **module [18].**

together with business network model agents (BNMA) and monitors. The extraction stage from the eContract creates sets of policies from the local contracts and assigns each a BNMA and monitor. The final preparation stage populates the lowest technical collaboration-level with matching services and corresponding endpoints for communication channels before enactment.

3.2 Decentralized governance infrastructure

Due to page limitations we give a short overview of the decentralized governance infrastructure (DGI) rollout and refer to [18] for further details. Briefly, every eCommunity-partner receives a local eContract copy that the agreed upon eContract governs. Thus, the latter functions as a controlling agent. Every local eContract copy is the source to extract a respective set of policies, monitors and BNMA for every eCommunity-partner.

Once a DGI is set up, the lowest technical level with locally enactable electronic services are machine enactable together with their service endpoints to enable communication. The business-semantics rollback and compensation options trigger exclusively from the enactment stage of an agreed upon eContract. Thus, during enactment, the scenarios may occur of a policy violation, disruptive or non-disruptive partner change. In the latter case, disruptive means there is a business-semantics rollback to the negotiation phase while non-disruptive means that the DGI remains intact and is taken over by a newly accepted eCommunity-partner.

3.3 Enactment phase

The electronic services that fill the service-offer templates in the eContract are complemented by in-house services. In Figure 5 these latter services reside in the central state *enacting services*. We assume for the gray-shaded part of Figure 5, these services are discrete business-process specifications with a unique start state and tasks relating to each other in sequences or parallelisms that lead to a unique end state [5, 19].

To perpetually enact respective services requires the involvement of the related BNMAs, monitors and policies. Respective services may stop for a period of time and restart again for enactment. Unless an orderly enactment culminates in a regular termination, eCommunity behavior that violates a policy results in triggering a corresponding violation assessment and business-semantics rollback.

3.4 Termination phase

The DAO-governance dissolves in three stages, as Figure 6 shows. The depicted *terminate*-module not only eliminates the technical-level setup of local eContract copies, but also the remainder of the distributed collaboration infrastructure. Correspondingly, first the *policy removal* commences with the termination of all electronic services from the technical level in enactment. Next, the counter of the amount of eCommunity-partners triggers the removing of all policies that are prior extracted from local eContract copies.

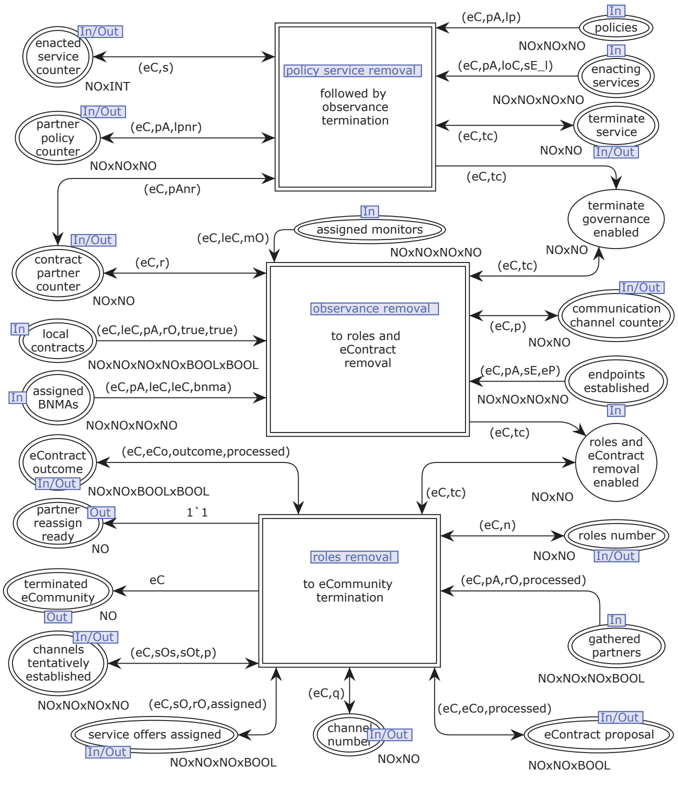

Figure 6: The *terminate* module [18].

The elements for observing the enactment of an eContract are the behavior-monitors and BNMAs. These infrastructure elements for observing the eCommunity-partner behavior are removed as part of the DAO-governance infrastructure together with related local eContract copies. Finally, it is possible to remove the communication channels represented by the established endpoints.

The *roles removal* first removes the roles that are part of the agreed upon eContract and all eCommunity-partners that populate these roles. Finally, the *roles removal* service consumes all tentatively established channels that are realized by communication endpoints on the technical level of electronic services.

4. CONFLICT ROLLBACK

The top-level conflict rollback resembles a Saga [6] transaction being an idea adopted from chained transactions [33] of including a compensation mechanism to roll back. Traditional Sagas divide a long lasting transaction into sequentially executed atomic sub-transactions with ACID properties and each sub-transaction, except the last one, has its own compensating sub-transaction. When any failure arises, the committed sub-transactions are undone by compensating sub-transactions. Unlike chained transactions, Sagas can return a whole transaction back to the very beginning with compensations. Note that failures refer to traditional database settings. However, in the governance lifecycle in this research, it is behavior-controlling policy violations that result in the undoing by sub-transactions. More recently, blockchain technology promises to be the foundation for the next generation of very large and distributed database systems [15], e.g., the blockchain-based DB called Guardtime[3]. Since blockchains as a specific transaction type solve the Byzantine generals' problem [7], the feature of non-repudiation of recorded collaboration events enables effective dicentralized P2P trust management. With this foundation, we specify below a novel e-governance framework that controls the business-semantics flow in a very targeted way.

Three business-semantics rollback scenarios exist that may either be disruptive or calming, and that govern the transition of an eCommunity from one epoch to another. A conceptual depiction of these rollbacks Figure 7 shows. Briefly, disruptive rollbacks imply an eContract renegotiation must start from scratch again. Calming rollbacks imply that the DAOs of an eCommunity see scope to reconcile collaboration issues. In both cases, the eCommunity experiences epoch changes.

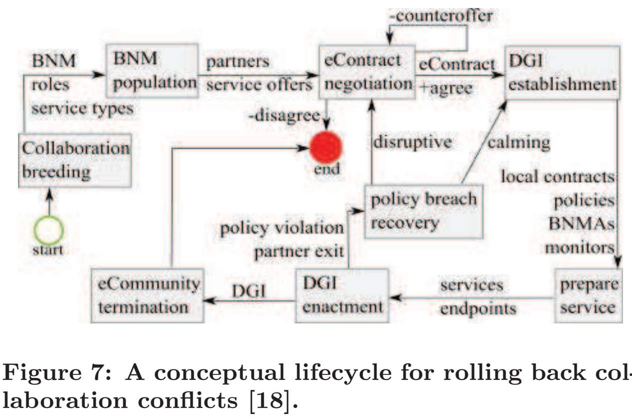

Figure 7: A conceptual lifecycle for rolling back collaboration conflicts [18].

Pertaining to Figure 7, there are one disruptive and three

[3]Guardtime: https://guardtime.com/blog