Towards Faulty Claim Impact Analysis & Mitigation in Verifiable-Credential-Driven Collaborations

Bertalan Zoltán Péter, Imre Kocsis
Budapest University of Technology and Economics
Department of Artificial Intelligence and Systems Engineering
Budapest, Hungary
Email: {bpeter@edu,kocsis.imre@vik}.bme.hu

***Abstract*—Verifiable credentials (VCs) are being adopted in various domains, providing a decentralized, standardized, and trustworthy ecosystem for issuing cryptographically verifiable documents comprising one or more claims. Using VCs to represent (some) digital documents in business processes has strong integrity, traceability, and auditability benefits. This is especially the case when several VCs are combined into *chains;* actors perform the next step of the process, possibly involving the issuance of another VC, when they are presented with a verifiable chain of VCs from previous steps. However, while VCs can be cryptographically verified as being authentically issued and structurally valid, the truthfulness of the actual claims they contain cannot be verified through the ecosystem. With chained VCs, claims found false or erroneous within a VC may directly or indirectly affect other VCs issued on the basis of those claims.**

This paper presents a novel perspective on the impact analysis of erroneous or false claims on chained VCs and proposes fault-tolerance techniques for reducing associated risks. We show the potential applications of these techniques in the context of a case study in the supply chain domain.

***Index Terms*—dependability, error propagation analysis, fault-tolerance, logistics, risk reduction, supply chain, verifiable credential**

I. Introduction

Verifiable credentials (VCs), often combined with decentralized identifiers (DIDs), represent cryptographically signed statements made by trusted entities in a decentralized manner and following the philosophy of self-sovereign identity (SSI). According to the W3C standard *Verifiable Credentials Data Model* [1], VCs encapsulate one or more *claims* made about a subject and are signed by an *issuer.* The *holder* of a VC can present it to a *verifier* (as a standardized verifiable presentation (VP)) who can verify the validity of the credential without any communication with the issuer.

These credentials are increasingly widely adopted in several domains; EU project examples built with VCs include EBSI VECTOR [2] targeting educational and social security purposes, or TRACE4EU [3] ensuring supply chain transparency.

While standalone VCs provide value on their own, their true potential is realized when several VCs are combined into chains and form what we refer to as *credential systems* [4]. In business processes, it is natural an entity only performs a particular activity in the process, provided some 'dependencies' or requirements of that activity have already been completed verifiably. For example, shipping can only begin once items have been prepared for pickup. When a process involves VCs, this generally maps to only issuing certain VCs provided some other VCs have been successfully presented and verified.

Unfortunately, while VCs assure the authenticity of the issuer and the integrity of the credential itself by cryptographic methods, they do not inherently ensure the truthfulness of the *claims* themselves. This is an issue to consider in itself, but it also implies additional challenges to tackle when there are dependent VCs that have been issued – directly or transitively – on the basis of such claims. Returning to our supply chain example, if an initial claim made by a supplier regarding their stock of a certain item was false, then the VCs issued by the consumer regarding the purchase order and any other VCs issued based on that order down the line are also affected.

The risk of false or invalid claims is not merely theoretical. For example, electronic attestations of attributes (EAAs) defined by the the electronic Identification, Authentication, and trust Services (e-IDAS) regulation 'shall not be denied legal effect or admissibility as evidence in legal proceedings on the sole ground that it is in electronic form or that it does not meet the requirements for qualified electronic attestations of attributes' [5] – suggesting that not only public sector bodies and other qualified trust service providers can issue credentials possessing legal effects. Furthermore, with the advent of artificial intelligence (AI) and especially large language modelss (LLMs), it can be expected that some claims will be made by AI agents which are well-known to commonly produce hallucinations, potentially leading to false claims.

We observe that one way to battle the issue of fake claims in credential systems is to apply and adapt analyses and techniques from classic dependable computing, such as error propagation analysis (EPA) [6]–[8] for systematic impact analysis, the introduction of redundancy and diversity requirements.

In this paper, we present a first theoretical overview of mapping dependability concepts to the credential systems context with a special focus on EPA and adaptable defences. We describe our approach in the context of a running example in the logistics domain.

The work of Bertalan Zoltán Péter supported by the Doctoral Excellence Fellowship Programme (DCEP) is funded by the National Research Development and Innovation Fund of the Ministry of Culture and Innovation and the Budapest University of Technology and Economics under a grant agreement with the National Research, Development and Innovation Office.

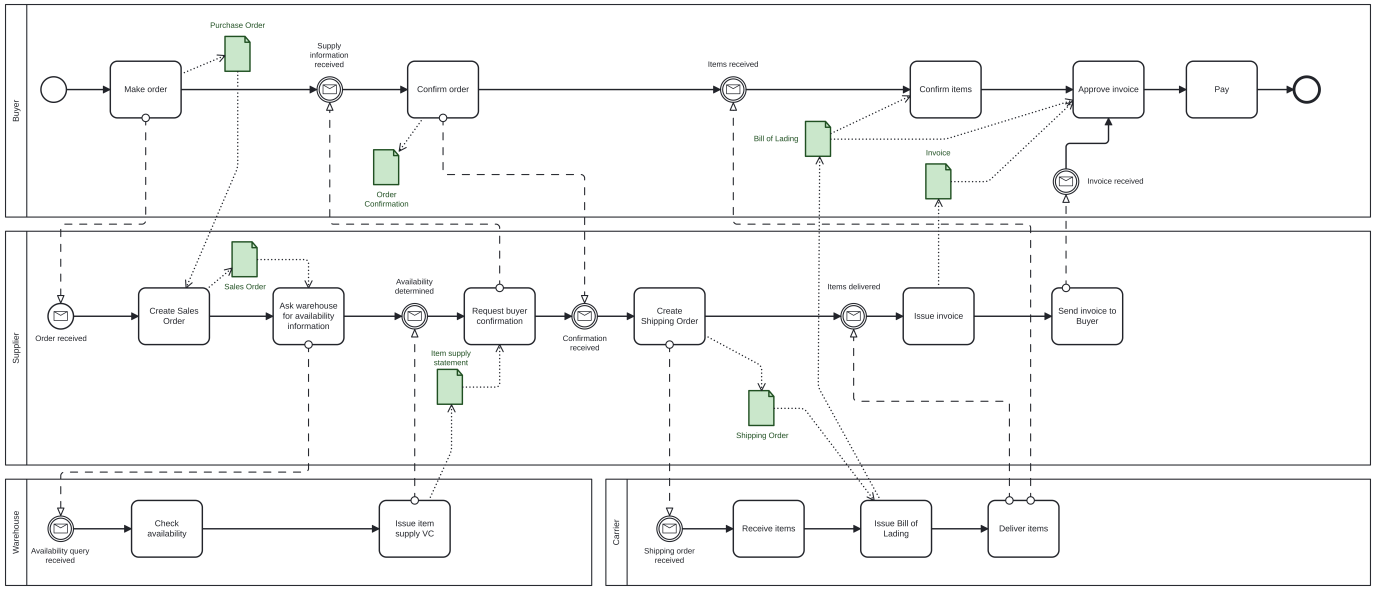

Figure 1. Example Supply Chain Process Model with Dependent Credentials

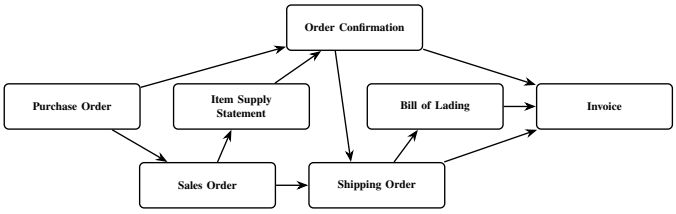

Figure 2. VC dependency graph from the running example in Figure 1

II. Related Work

To the best of our knowledge, there has been no prior work on analyzing the impact of VCs with false claims in chained VC systems or applying the tools of dependable computing in this context.

However, the problems with false claims are closely related to the overarching concept of *trust* in credential systems – a trustworthy party is unlikely to issue credentials with false claims because their reputation shows that they take measures to ensure their claims are valid and that they do not engage in dishonest behaviour. On the other hand, if any participant is observed to make erroneous claims frequently, their reputation becomes questionable, possibly prompting collaborating organizations to cross-check all VCs issued by that participant in all active credential systems.

The propagation of trust and distrust within trust networks – and often particularly social networks – has been the subject of several research papers [9]–[12]. For example, [9] presents a probabilistic trust propagation model based on belief propagation in social networks using factor graphs. The authors of [10] developed a trust propagation scheme framework that can predict the trust between any two entities in large e-commerce or recommendation systems based on a small number of expressed trust-/distrust relations.

While trust is a crucial aspect in the design of credential systems and certain elements of trust-focused analysis can be tackled by existing approaches, not all potential errors necessarily stem from trust faults. Erroneous claims can also be expected to enter the system for other reasons, such as software faults and other non-intentional causes. A well-designed system must sufficiently tolerate faults of any kind – the analysis perspective shown in this paper and the listed defence adaptations provide a first step towards achieving this dependability goal.

This paper proposes the application of EPA in a context which differs from its classical uses, i.e. qualitatively exploring error propagation paths in the architectural and component-level models of composed IT and embedded systems. [13] introduces a similar non-classical application of EPA for *requirement change propagation,* providing a way to assess the safety integrity level (SIL) impact of system requirement changes. The authors present multiple propagation types with varying levels of abstraction in terms of how much information regarding the requirement change is modelled.

III. Running Example

We now introduce a simplified logistics-based example that could be implemented with chained VCs in a credential system; we will refer to certain elements of this model later in the paper to illustrate concepts with an example.

Consider the supply chain business process model in Figure 1 based on [14] and a trade finance example from [4]. This model only shows the *happy path:* first, a Buyer creates a purchase order for the Supplier, who proceeds to create a sales order based on it for internal processing. Item availability (stock) is retrieved from a warehouse, and based on the results, an order confirmation is requested from the buyer. Once confirmed, the order's shipping begins; the Carrier receives

a shipping order from the Supplier and issues a bill of lading on receipt of the items to ship. After the items have been delivered, the Supplier issues an invoice to the buyer, who approves and pays it.

Many document artifacts in this process can be represented by W3C VCs to ensure data integrity and auditability, improve interoperability, and enable automation, while also eliminating the need for trusted central parties. These documents have been highlighted green in the model. Note how a false claim relatively early in the process – for example, within the item supply VC – can invalidate the rest of the process along with several other VCs issued after it.

In the rest of this paper, we discuss the impact analysis of erroneous claims, and, particularly, the application of EPA to analyze the 'system-level' effects of claim errors, followed by a brief overview of mechanisms and techniques adaptable from traditional dependable computing as defences against failures stemming from these errors.

IV. Impact Analysis of False Claims

Our key insight in this paper is that the impact of invalid claims (i.e., *lies* or otherwise erroneous claims) can be tackled with a methodology adapted from classic dependability engineering. *Dependability,* as a composite concept, can be broken down into *threats* that make the system fail, *attributes* that refine what makes a system dependable, and *means* to increase dependability [15]. We define threats as the classic fault-error-failure trinity for the context of credential systems in the following way.

Faults are what cause false, wrong, or invalid claims to potentially end up in VCs. These may be benign or malicious in nature; for example, a simple software fault can lead to a miscalculation and a subsequent claim based on the erroneous value. However, in many cases, an actor may be motivated to make a claim they know is not true purposely. A fault can be considered active if it causes a false claim to enter a VC. In our running example, the Carrier colluding with the Supplier against the Consumer can be considered a fault. A bug in the order handling software can also be considered a fault.

Errors form when, due to a fault, a VC is issued containing a false or invalid claim. Errors can propagate throughout the graph of chained VCs simply because an actor will often require the presentation of a specific VC to issue another VC (with their own set of claims). For example, the Warehouse issues a VC $\mathcal{C}_{\text{stock}}$ with a claim of having 50 tons of copper wiring ready to ship, when in reality, there are only 30 tons currently available. Later, the Customer issues a VC $\mathcal{C}_{\text{OC}}$ representing a purchase order confirmation with a commitment to pay based on the claim in $\mathcal{C}_{\text{stock}}$. It is clear that the Customer would not have confirmed the purchase were they not under the assumption that there are sufficient assets in stock for them. Therefore, in a sense, the error in $\mathcal{C}_{\text{stock}}$ propagates to $\mathcal{C}_{\text{OC}}$.

Failures occur when the falseness of a claim makes a system-level impact. In our example, the Carrier being unable to pick up 50 tons of copper wire ordered due to the false initial stock claim in $\mathcal{C}_{\text{stock}}$ breaks the entire trade process.

Error Propagation Analysis in Credential Systems

EPA is 'a systematic model-based approach to assess the impact of incidental or malicious faults in the dependability and security analysis of complex systems' [6]. In this paper, we refer to *qualitative* EPA and apply it explicitly to credential systems.

Figure 2 shows a dependency graph of the VCs of our running example business process (from Figure 1). The arrows represent a need to be able to verify claim dependencies before issuing a given credential. For example, the item Supplier will only create a *Shipping Order* for the Carrier provided there is already a valid *Sales Order* and the Customer has presented an *Order Confirmation* VC.

If some VCs are issued on the basis of prior claims, then any potentially false claim (an *error* in our terminology) can propagate throughout the dependent claims. In our example, a false *Item Supply Statement* could theoretically affect the *Order Confirmation* of the customer and, transitively, the *Shipping Order,* and the *Invoice.*

At this level of abstraction, one can only perform an elementary impact analysis of false claims that essentially boils down to a reachability check in the credential dependency graph. For a more accurate analysis, a more refined model is required that includes the exact claims within the credentials and describes the dependencies between the *claims* themselves. Furthermore, credentials within the graph may have different error propagation characteristics. For example, the *Shipping Order* credential may not propagate the error introduced in the *Item Supply Statement* any further if an external check reveals the inconsistency between the supply communicated to the Consumer and what is actually possible to ship.

The main goal of EPA in this application is twofold: first, it can provide insight into incorrectly or poorly designed credential chains at relatively early design time. Second, it can determine what *defences* are worth implementing to protect against failure, especially in case the credential dependency relations cannot be sufficiently improved directly. In section V, we present adaptations of several well-known dependability techniques to credential systems.

Several techniques can be employed to perform the concrete qualitative error propagation analysis in credential systems. In our experience, answer set programming (ASP) [16] is a powerful approach to encode the system model as well as the propagation rules and it can be used for both deductive (*What does a specific false claim affect?*) and inductive (*Which claims could affect the validity of a certain VC?*) reasoning.

V. Defending Against False Claims

We propose that some of the traditional dependability techniques and patterns of fault prevention, tolerance, removal, and forecasting [15] can be mapped to a credential system context. In the rest of this section, we briefly elaborate on how common approaches falling into these categories can be applied; Table I summarizes the methods explored.

Table I
DEPENDABILITY TECHNIQUES APPLIED TO CREDENTIAL SYSTEMS

	Dependability Technique	Credential System Adaptation
prevention	Common mode failure prevention	Source independence analysis
	Bulkhead pattern	Trust domain isolation
tolerance	N-version programming	Multi-authority attestation
	Standby spares	Backup verification paths
	Fail-safe behaviour	Conservative trust decisions
	Load shedding	Selective claim requirements
removal	Error detection	Claim consistency checking
	Circuit breaker pattern	Issuer trust circuit breaking
	Compensation	Trust recovery actions
	Data diversity	Freshness & temporal diversity
forecasting	Trust calibration	Adaptive trust thresholds
	Anomaly detection	Credential network antipattern recognition
	Read team simulation	Adversarial testing

None of the defences come for free – their costs show either in increased development and design efforts or additional 'runtime' overhead. Deciding which defences to apply should be based on a thorough error propagation analysis of the credential system in question and a cost-benefit analysis.

A. Fault Prevention

Fault prevention means preventing false claims from entering a credential system via being included in VCs.

Common Mode Failures → Source Independence Analysis. When employing redundancy, it is vital to ensure that redundant components – in our context, claim sources – do not have common failure modes. For example, multiple departments of the same organization – or, in our case, the Supplier and Carrier – might collude and attest to the same false information. These risks can be reduced by modelling and analyzing the relationships and dependencies between claim sources. Preventive measures such as temporal separation, authoritative diversity, and information hiding can be employed.

Bulkhead → Trust Domain Isolation. In the classic *bulkhead* architectural pattern, 'elements of an application are isolated into pools such that if one fails, the others will continue to function' [17]. In our context, the elements are VCs (more precisely, claims in VCs) and the pools can be formed by smaller, isolated credential subnetworks to control the propagation of false claims. For example, the 'internal' credentials associated with the delivery of an order need not be required outside the shipping part of a trade process.

B. Fault Tolerance

Despite the prevention efforts proposed above, false claims will inevitably slip through the safeguards and potentially become the basis for other claims and credentials. Designing a credential system to be fault-tolerant means being able to operate correctly even in the presence of some false claims in some VCs.

N-Version Programming → Multi-authority Attestation. In software dependability, N-version programming (NVP) is a technique where multiple versions of the same software component are developed based on the same specification and are then redundantly executed at runtime to tolerate independent faults in their implementation [18]. NVP works when the implementations are sufficiently diverse; i.e., were made by different teams, written in different programming languages, etc.
With claims, a close analogy is requiring diversity in the sources making the claims and the entities that verify their associated credentials. For example, instead of solely relying on the Supplier's claim regarding the available stock from a product, an auditor's attestation could also be required. The more diverse the sources' interests, the less likely they are to collude and make consistent false claims. Multi-authority attestation also prevents single points of failure (SPOFs) where a single compromised issuer can fabricate false claims affecting the entire credential system.

Standby Spares → Backup Verification Paths. Spares provide redundancy in the form of backup systems that can take over when a primary component fails. In the credentials context, a system can fall back to alternative sources or even verification methods in case the default fails or cannot be considered reliable anymore. For example, if the Carrier's IT systems are compromised, attackers may be able to fabricate VCs in their name, meaning their credentials can no longer be trusted. In a fault-tolerant design, there would be a backup source (e.g., a port operator) who can issue alternative credentials that can still securely support the trade process until the Carrier can be trusted again.

Fail-Safe Behaviour → Conservative Trust Decisions. Fail-safety means bringing the system to a safe state when failures occur [19]. While the question of *safety* is typically not relevant in credential systems, we can still consider undesirable states to avoid even in the presence of failures caused by false claims. Business-critical decisions should be made conservatively: when there is insufficient confidence in the truthfulness of a claim the decision is based on – for example, it contradicts another trusted claim –, it is best to err on the side of caution.

Load Shedding → Selective Claim Requirements. Instead of *load* in the traditional computing sense, we can consider the 'administrative' overhead of redundant VCs,

freshness requirements, and the other defence approaches proposed in this paper. When faster processing is paramount (e.g., a Consumer needs certain assets urgently), some less critical defences and verification steps can be dropped at the cost of lowering the trust levels.

C. Fault Removal

A compromised or no longer trusted participant will persistently hinder business processes where it is required as an issuer of VCs. Fault removal maps to detecting and eliminating false claims and their sources from credential networks.

Error Detection → Claim Consistency Checking. One of the most straightforward methods to detect false or incorrect claims is to perform basic consistency checks and corroborate them with external information. For example, in a bill of lading VC, a claim of having picked up 2000 pallets of OLED monitors combined with a statement that the total shipment weight is 20 kg should trigger a consistency issue alarm.

Circuit Breaker → Issuer Trust Circuit Breaking. The classic circuit breaker pattern introduces a wrapper component that stops attempting to call a failing service after a given threshold of retries in order to prevent cascading failures [20]. Considering credential systems, taking the pattern in a more general sense, a similar approach can be employed to isolate faults by monitoring the rate of false or invalid claims detected from an issuer and ceasing to accept new credentials from them, should this rate exceed some threshold.

Compensation → Trust Recovery Actions. Suppose one or more false claims have been detected to have been made by a given issuer. In that case, actions should be taken to minimize the future impact of the faulty issuer and determine if previously issued and still relevant VCs are also affected. Optimally, all recent and currently used VCs issued by the entity should be quarantined and validated, preferably via external sources. Furthermore, dependent VCs should be quarantined and revoked if necessary. When possible and desired, steps should be taken to re-establish trust with the affected issuer.

Data Diversity → Freshness & Temporal Diversity. Another way to detect and minimize the rate of false claims is to set requirements on the *freshness* of the VCs issued in a process. For example, a tax authority can request a more up-to-date (i.e., issued within the last month) statement regarding the assets held by an entity. The motivation is that it is harder for dishonest or malicious participants to maintain false narratives over time, especially due to the verifier being able to retrieve contradictory evidence in the meantime.

D. Fault Forecasting

Anticipating and preparing for potential false claims and transitive VC chain invalidations is an essential building block of dependable credential systems.

Trust Calibration → Adaptive Trust Thresholds. Learning from previous experiences and incidents, the dependability of trust decisions in a credential system can be improved by continuous adaptation. For example, if experience shows that Carriers are often liable to make false claims regarding shipment statuses, then the requirements for issuing further VCs based on these claims should be made more strict and possibly be extended by additional checks.

Anomalies → Credential Network Antipatterns. Provided credential systems are sufficiently monitored, AI- and machine-learning-based approaches can recognize and flag unusual, suspicious patterns in the interdependent VCs. This can help provide early warnings of potential failures.

Red Team Simulation → Adversarial Testing. For a credential system to be verifiably tolerant against false claims to some extent, it must be well-tested. Adversarial and fault-injection-based testing is particularly applicable to these systems: during testing, let a selected entity issue VCs with invalid claims deliberately and analyze the propagation of these faults and their system-wide effects. Such tests can be used to identify potential weaknesses in the system proactively.

VI. Conclusion

In this paper, we have presented a novel perspective on a key challenge with chained VC: the propagation and impact of false or erroneous claims within these systems. By adapting established dependability engineering concepts to the VCs, we have demonstrated how traditional fault-error-failure models can provide valuable insights into the risks and mitigation strategies for credential-based business processes.

We have outlined how EPA can be applied to credential dependency graphs to reason about dependability guarantees and determine what defences the system would benefit from best. We have also presented a comprehensive taxonomy of traditional dependability techniques adapted specifically to credential systems, spanning fault prevention, tolerance, removal, and forecasting. These concepts were illustrated by a theoretical supply chain case study included as a running example in our paper.

The possibility of mapping dependability patterns to credential systems reveals promising avenues for overcoming trust and robustness challenges in credential-based cross-organizational collaboration ecosystems.

Challenges that remain to be addressed in future work include the development of a detailed and formal taxonomy of faults, errors, and failures in this domain, as well as the definition of an error propagation rule library for the different types of VCs that can be present in these systems. Once 'executable' qualitative models are available, we can shift our focus towards quantifying probabilities, risks, and costs, in our models, which will provide insight even at later stages of the design process for credential systems.

This exploratory work paves the way to implementing concrete applications based on the rich toolset of dependable computing to enhance reliability and trustworthiness in VC systems.

References

[1] M. Sporny, D. Longley, D. Chadwick and I. Herman, 'Verifiable credentials data model v2.0,' W3C, W3C Proposed Recommendation, 2025, https://www.w3.org/TR/vc-data-model-2.0/.

[2] 'EBSI-VECTOR Project: Revolutionising digital identity in education and social security.' (7th May 2024), [Online]. Available: https://www.ebsi-vector.eu/wp-content/uploads/2024/05/EBSI-VECTOR-One-pager.pdf (visited on 31/05/2025).

[3] C. Hennings and S. Schwalm, 'TRACE4EU D1.5 interim report,' TRACE4EU Consortium, Tech. Rep., 2024. [Online]. Available: https://trace4eu.eu/wp-content/uploads/2024/08/TRACE4EU-D1.5-Interim-Report.pdf (visited on 31/05/2025).

[4] The United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT), *eDATA verifiable credentials for cross border trade*, Whitepaper, 2022. [Online]. Available: https://unece.org/sites/default/files/2023-10/WhitePaper_VerifiableCredentials-CrossBorderTrade.pdf (visited on 31/05/2025).

[5] European Commission, 'Regulation (EU) 2024/1183 of the European Parliament and of the Council of 11 april 2024 amending Regulation (EU) no 910/2014 as regards establishing the European Digital Identity Framework,' *Official Journal of the European Union*, vol. L 2024/1183, 2024.

[6] A. Földvári, G. Biczók, I. Kocsis, L. Gönczy and A. Pataricza, 'Impact assessment of it security breaches in cyber-physical systems: Short paper,' in *2021 10th Latin-American Symposium on Dependable Computing (LADC)*, 2021, pp. 1–4. DOI: 10.1109/LADC53747.2021.9672582.

[7] A. Pataricza, 'Systematic generation of dependability cases from functional models,' in *FORMS/FORMAT 2007 – 6th Symposium: Formal Methods for Automation and Safety in Railway and Automotive Systems*, 2007.

[8] G. Urbanics, L. Gönczy, B. Urbán, J. Hartwig and I. Kocsis, 'Combined error propagation analysis and runtime event detection in process-driven systems,' in *Software Engineering for Resilient Systems*, I. Majzik and M. Vieira, Eds., Cham: Springer, 2014, pp. 169–183. DOI: 10.1007/978-3-319-12241-0_13.

[9] R. Zhang and Y. Mao, 'Trust prediction via belief propagation,' *ACM Trans. Inf. Syst.*, vol. 32, no. 3, 2014, ISSN: 1046-8188. DOI: 10.1145/2629530.

[10] R. Guha, R. Kumar, P. Raghavan and A. Tomkins, 'Propagation of trust and distrust,' in *Proceedings of the 13th International Conference on World Wide Web*, ser. WWW '04, New York, NY, USA: Association for Computing Machinery, 2004, pp. 403–412, ISBN: 158113844X. DOI: 10.1145/988672.988727.

[11] C.-N. Ziegler and G. Lausen, 'Propagation models for trust and distrust in social networks,' *Information Systems Frontiers*, vol. 7, pp. 337–358, 4 2005, ISSN: 1572-9419. DOI: 10.1007/s10796-005-4807-3.

[12] A. Jøsang, S. Marsh and S. Pope, 'Exploring different types of trust propagation,' in *Trust Management*, K. Stølen, W. H. Winsborough, F. Martinelli and F. Massacci, Eds., Berlin, Heidelberg: Springer, 2006, pp. 179–192, ISBN: 978-3-540-34297-7. DOI: 10.1007/11755593_14.

[13] A. Pataricza, I. Kocsis, F. Brancati, L. Vinerbi and A. Bondavalli, 'Lightweight formal analysis of requirements,' in *Certifications of Critical Systems – The CECRIS Experience*, River Publishers, 2022, pp. 143–166.

[14] N. B. Erik Hofmann Urs Magnus Strewe, *Supply Chain Finance and Blockchain Technology, The Case of Reverse Securitisation* (SpringerBriefs in Finance). Cham: Springer, 2017. DOI: 10.1007/978-3-319-62371-9.

[15] A. Avizienis, J.-C. Laprie, B. Randell *et al.*, 'Fundamental concepts of dependability,' *Technical Report Series-University of Newcastle upon Tyne Computing Science*, 2001.

[16] V. Lifschitz, *Answer Set Programming*. Cham: Springer, 2019, vol. 3. DOI: https://doi.org/10.1007/978-3-030-24658-7.

[17] I. A. Buckley and E. B. Fernandez, 'Dependability patterns: A survey,' *Computers*, vol. 12, no. 10, 2023, ISSN: 2073-431X. DOI: 10.3390/computers12100214.

[18] A. Avizienis, 'The N-version approach to fault-tolerant software,' *IEEE Transactions on Software Engineering*, vol. SE-11, pp. 1491–1501, 1986. DOI: 10.1109/TSE.1985.231893.

[19] H. Kopetz and W. Steiner, 'Dependability,' in *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Cham: Springer, 2022, pp. 143–175, ISBN: 978-3-031-11992-7. DOI: 10.1007/978-3-031-11992-7_6.

[20] M. Nygard, *Release It!: Design and Deploy Production-ready Software* (Pragmatic programmers). Pragmatic Bookshelf, 2018, ISBN: 9781680502398.