

- **Iterative, verifiable modeling** Formal models need to evolve incrementally without introducing contradictions or drift. Future research could explore interfaces where LLMs propose edits that are checked by reasoners or validated against typed assertions.
- **Toolchain integration** Bridging the gap between LLMs and formal reasoning systems remains a challenge. Improved APIs, modeling environments, and reasoning-aware LLM prompts could help operationalize neurosymbolic modeling pipelines.
- **Round-trip modeling and explanation** One of the most exciting possibilities is the idea of a round-trip loop between symbolic models and LLMs: models extracted from LLMs are refined by humans and used in turn to improve LLM outputs or explanations. Supporting this requires mechanisms for traceability, grounding, and goal-sensitive reasoning.

5.4.4.3 Broader Implications

- **From world models to world views** If LLMs can generate multiple, diverging models from the same text (as our experiments suggest), this opens up the possibility of comparing different stakeholder perspectives – e.g., between a regulator, a provider, and an affected user. This introduces new opportunities for accountability, argumentation, and normative reasoning.
- **Explainability as emergent behavior** Formal models on their own do not constitute explanations – but they provide the scaffolding for tailored, stakeholder-specific narratives. Future work could explore how formal world models and LLM-based natural language generation can jointly support explainable AI in high-stakes domains like regulation.

References

- 1 European Union. Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024, 2024.
- 2 Cogan Shimizu and Pascal Hitzler. Accelerating knowledge graph and ontology engineering with large language models. *Journal of Web Semantics*, 85:100862, 2025.
- 3 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv*, arXiv:2408.03314, 2024.

5.5 Knowledge Graph Ecosystems

Sandra Geisler (RWTH Aachen, DE)

James A. Hendler (Rensselaer Polytechnic Institute – Troy, US)

Philipp D. Rohde (TIB – Hannover, DE)

Aisling Third (The Open University – Milton Keynes, GB)

Maria-Esther Vidal (TIB – Hannover, DE)

License © Creative Commons BY 4.0 International license

© Sandra Geisler, James A. Hendler, Philipp D. Rohde, Aisling Third, and Maria-Esther Vidal

5.5.1 Introduction

In this working group, we based our initial discussions on the concepts of Knowledge Graph Ecosystems (KGEs) inspired by the paper by Geisler et al.[1] In the paper, knowledge graph ecosystems describe the context and crucial aspects impacting the creation, updates, and usage of knowledge graphs by a sextuple. This includes the data sources populating the KG,

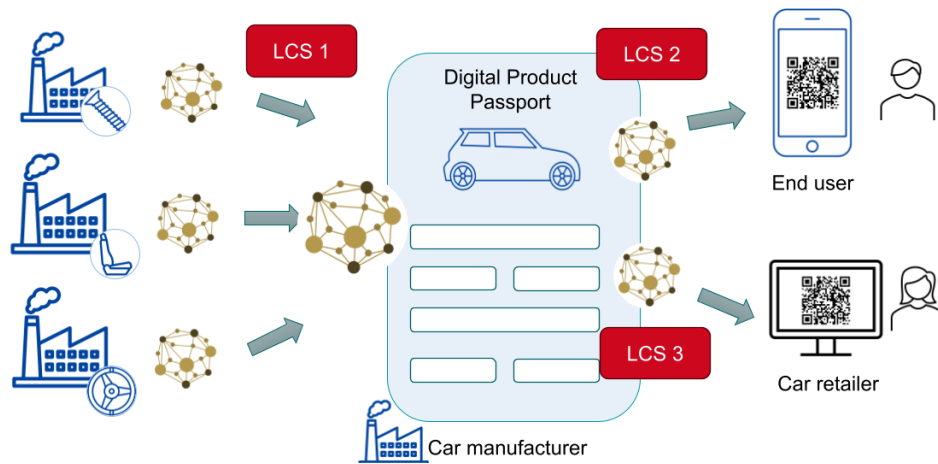
the ontologies to enrich and structure the data in the KG, mappings between the ontology and the data, constraints, which enable consistency and quality checks, the KG which is “living” in the KGE and subject to changes and usage, and finally a log, tracing the operations executed on the KG and corresponding events. Further, the paper delineates life cycles and life cycle steps which structure and order the operations on the KG. A life cycle step can be, e.g., the creation or the update of a KG, or a query or analysis on the data of the KG. Formally, it is defined also as a sextuple with a service executing the step, actors and roles involved in the step, as well as requirements, constraints, and needs which impact the life cycle step. While this work is a crucial step towards the formalization and therefore automatization and verifiability of operations on and around KGs, it only targets individual KGs. However, today many applications require sharing data, i.e., the querying and analysis of multiple distributed KGs across organizations, domains, and even countries. Hence, we need to not only take into account contexts and life cycles of single KGEs, but also networks or federations of KGEs.

In exploring the concepts of federated and decentralized Knowledge Graph Ecosystems (KGEs), this work inherently ties into the foundational themes of trust, accountability, and autonomy discussed in [2]. Note that each of these themes are very broad concepts with significant variation in their concrete application; for a fully flexible model of KGEs, it would be a mistake to attempt to narrow trust, accountability, and autonomy down to very specific or technical definitions only covering a subset of their broader meanings. We rather leave the framework open to accommodate different concrete understandings of these terms in different scenarios or use cases. Trust in federated KGEs is established through shared ontologies and robust data verification processes, similar to the way blockchain’s transparency and immutability enhance trust by ensuring the traceability and provenance of data. This aligns with decentralized KGEs where trust mechanisms, like reputation systems, are vital for ensuring knowledge integrity. Accountability is addressed through verifiable contributions and collaborative data management in federated networks, paralleling the mechanisms in [2], such as compliance-checking and provenance tracking that uphold data integrity through blockchain technology. Autonomy is a defining feature of decentralized KGEs, where stakeholders maintain control over their data and operations, consistent with the use of decentralized infrastructures in [2], which empower individuals through self-sovereign identities and personalized data management. By integrating these technologies, both papers emphasize the potential to advance KGEs in ways that are reliable, user-centric, and conducive to collaborative innovation.

In the following, we will present the examples for such networks and federations of KGs discussed in the seminar. Based on these examples, we will delineate the different types of KGEs we derived from the examples and discuss challenges and requirements for the different types. These challenges and requirements led to a reformulation and extension of the definitions for KGEs and the corresponding life cycles, which we will sketch subsequently. Finally, we will give an outlook on future work sparked by the results of the seminar.

5.5.1.1 Use Case 1: Digital Product Passports

Transparency of product constitution, production processes, and supply chains, especially in the light of sustainability, energy consumption, and circular economies, motivates the need for a Digital Product Passport (DPP). A DPP serves as a comprehensive digital record that comprises data about a product’s components and life cycle, including its materials, manufacturing processes, usage, and recycling potential. DPPs are promising transparency and traceability, ensuring that all stakeholders – from manufacturers to consumers – can



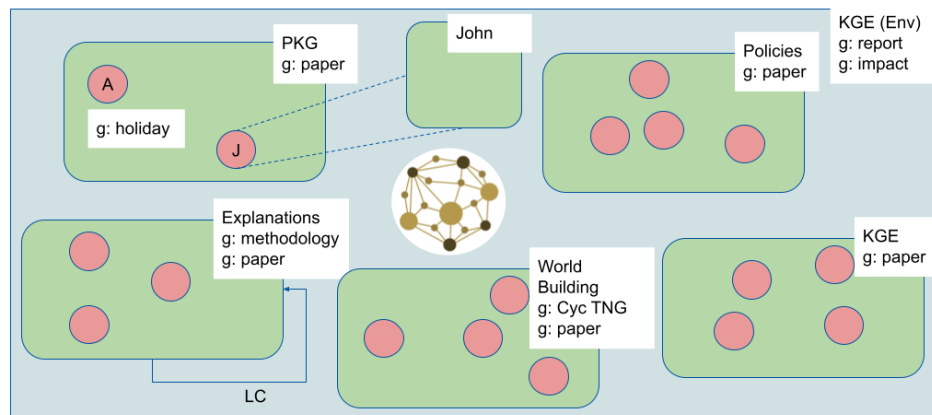
■ **Figure 3** A KGE of a Digital Product Passport.

make informed decisions while fostering accountability. Additionally, a DPP can facilitate compliance with regulatory requirements, improve waste management, and support efficient product recalls or repairs. However, the data for the DPP needs to be retrieved from the involved stakeholders. Suppliers, logistics, as well as manufacturers, can provide information regarding a product or parts of it. Furthermore, usage behavior, repairs, and other events related to the product user and impacting the product quality and integrity can also be part of a DPP.

Figure 3 shows the example of a DPP infrastructure for a car. Suppliers of screws, seats, or steering wheels all may maintain data about the part they are providing. It is assumed that product data for the DPP is stored in KGEs located at the suppliers. To enable a DPP, the suppliers extract and provide a part of this data to the car manufacturer who in our example is assumed to maintain the DPP of the car, i.e., integrate the data from the suppliers into a bigger KGE. We assume further that different stakeholders may have different views on the KGE of the product maintained by the manufacturer. Thereby, subsets in the form of views can be extracted from the KG of the DPP and are provided to consuming applications, such as mobile applications for end users or web applications for retailers. In this scenario, the stakeholders follow a common goal, namely to provide a DPP. In terms of trust, the car manufacturer trusts that the suppliers provide the DPP information of their part, i.e., they are accountable for the provision of correct, accurate, and complete data. They do not have a high autonomy as they are dependent on the car manufacturer as their customer and follow its rules.

5.5.1.2 Use Case 2: A Dagstuhl Seminar

Opposed to the example of a DPP, in a Dagstuhl Seminar multiple stakeholders (the participants and organizers) attend with one or more individual goals for the overall seminar, but do not necessarily share a common goal. They all have their own “internal KGE” and may update their knowledge in interaction with other participants. Further, they build smaller working groups, where the group follows common group goals and at the same time can contribute to the overall goal of the seminar. Hence, there are multiple KGEs (one for each participant, one for a group, and one for the seminar), which have their own life cycles, “interact” with each other and exchange knowledge between them. Between the participants,



■ **Figure 4** The KGEs of a Dagstuhl Seminar.

trust needs to be established as a basis to update their KGEs in the interaction. There is a very high autonomy as no participant is dependent on another and in consequence also no accountability.

5.5.2 Types of KGE Networks

We define a KGE network as a set of KGEs, which are interconnected by relationships.

We have considered each of the examples above in terms of how they demonstrate different assumptions about the three pillars of trust, accountability, and autonomy, and how they relate to each other. By configuring KGE networks with these assumptions, we can see that network phenomena such as *federation*, decentralization, and so on, emerge. In the DPP example, low autonomy of stakeholders, represented by the common specification set by the car manufacturer, and trust ensured via contractual accountability, looks very like a *federated* scenario; a Dagstuhl Seminar's high autonomy of participants, with low accountability and high trust, resembles more a *decentralized* scenario.

A relationship is established between two KGEs to enable knowledge exchange between them. Based on the observations from the above two and also other examples, we distinguish the following two types of KGE networks:

1. **Federated KGE Networks:** In a federated KGE network all stakeholders have a KGE and share a common goal towards which they exchange knowledge.
2. **Decentralized KGE Networks:** In this type of networks stakeholders also have a KGE, but may have one or more individual goals. Additionally, they can also share common goals with one or more other stakeholders.

These definitions are in terms of entities formally defined in the KGE framework of [1]: goals, knowledge graphs, etc. We show below how these can nonetheless provide models of these trust, accountability, and autonomy scenarios. This provides an insight into how to model scenarios involving the foundational pillars of [2] without overcommitting to narrow definitions of them, and we argue that this flexibility coupled with the representation formalism of KGEs is essential for AI ecosystems which work for such fundamentally human concepts.

5.5.3 Requirements

Understanding the requirements inherent in different use cases of Knowledge Graph Ecosystems (KGEs) provides insight into the diversity of challenges and solutions necessary to support various applications. This section describes the requirements for the two primary types of KGE networks – Federated and Decentralized – each characterized by unique aspects of collaboration, autonomy, trust, and lifecycle management.

5.5.3.1 Federated KGE Networks

The case of Digital Product Passports (DPP) serves as an archetype for federated KGEs where stakeholders share a collective objective and are inherently dependent on each other's data integrity and contributions. The requirements for such setups include:

1. **Shared Semantics and Ontologies:** Stakeholders must align on a common ontology to ensure interoperability across KGEs. This includes shared vocabularies and data models that describe products consistently.
2. **Data Integration and Consistency:** The integration of data from multiple suppliers entails rigorous consistency checks and validation mechanisms to maintain data integrity across the ecosystem.
3. **Accountability and Verifiable Contributions:** Federated systems necessitate mechanisms for traceable contributions to ensure that each actor can be held accountable for the data they provide, thus fostering trust in the overall system.
4. **Role-Based Access and Governance:** Clearly defined (e.g., suppliers, manufacturers, regulators) must be supported, with correspondingly specific access rights and responsibilities. Governance models ensure appropriate access and usage of the ecosystem by different stakeholders.
5. **Lifecycle Synchronization:** Operations across constituent KGEs must be coordinated, especially for updates and validations, to maintain a consistent state and ensure collaborative workflows proceed smoothly.

5.5.3.2 Decentralized KGE Networks

In contrast, decentralized KGE networks as exemplified by collaborative scenarios like scientific research or Dagstuhl Seminars, emphasize autonomy and less rigid interactions. The requirements for decentralized structures include:

1. **Alignment Mechanisms:** Decentralized networks require lightweight ontology alignment methods to facilitate knowledge exchange without enforcing uniformity, thus respecting participants' autonomy.
2. **Trust-Based Interactions:** Establishing trust through provenance and reputation mechanisms is critical in decentralized setups where actors operate independently but collaborate based on shared interests or goals.
3. **Asynchronous and Independent Lifecycles:** Supports for asynchronous updates and independent lifecycle management are needed, allowing participants to operate under varied goals and timelines without disrupting mutual interactions.
4. **Autonomy Preserving Policies:** Each participant should have the autonomy to apply local policies and operate independently, with knowledge exchange only occurring in mutually beneficial circumstances.
5. **Conflict Resolution and Negotiation Frameworks:** Mechanisms to resolve conflicts or inconsistencies are essential where differing goals or data interpretations may arise from the diverse participants.

5.5.3.3 Challenges and Opportunities

The interplay between these requirements in both federated and decentralized networks highlights several overarching challenges:

1. **Interoperability:** Striking a balance between shared understanding and local autonomy requires robust semantic alignment techniques applicable across diverse contexts.
2. **Trust and Accountability:** Crafting ecosystems where trust and accountability are clear yet flexible enough to accommodate decentralized decision-making and governance remains a pivotal challenge.
3. **Efficiency and Scalability:** Efficient data management, processing, and query execution across distributed KGEs need to support scalability while preserving data quality and provenance.

Addressing these requirements not only involves technical innovation in areas like ontology matching, knowledge integration, and lifecycle management but also necessitates careful consideration of legal, ethical, and organizational factors to foster successful KGE deployments. As such, continued research in synchronizing lifecycles, implementing robust alignment frameworks, and enhancing-based governance will pave the way for more adaptable and scalable KGE networks.

5.5.4 Extension of the KGE Concept

The original concept of a Knowledge Graph Ecosystem (KGE), as introduced by Geisler et al. [1], models the fundamental operational components required to manage the creation, evolution, and analysis of a knowledge graph. Formally, a KGE is defined as a sextuple: $KGE = (D, O, M, DC, KG, L)$, where:

- D is a set of **data sources**, each with schema $\theta(ds)$ and instances $\alpha(ds)$.
- O is the **ontology**, a logical theory describing the domain using a structured vocabulary.
- M is a set of **mappings** linking D to O through semantic assertions.
- DC is a set of **domain constraints** ensuring data consistency and quality.
- KG is the resulting **knowledge graph**, built from D using M under O .
- L is a **log** of lifecycle steps, tracking changes and ensuring traceability.

Limitations. While this formalization provides a solid foundation for the structured management of a single evolving knowledge graph, it assumes:

1. A single, self-contained knowledge graph (KG) under centralized control.
2. A fixed and isolated lifecycle of operations for one ecosystem context.
3. No explicit support for collaboration or interactions across multiple independent KGEs.

However, real-world applications increasingly rely on *networks of interconnected KGEs* operated by diverse stakeholders. These scenarios require knowledge to be collaboratively constructed, exchanged, and reused across institutional, geographical, or organizational boundaries. Consider the previously defined use cases:

- **Digital Product Passports (DPPs):** A manufacturer aggregates data from multiple suppliers, each maintaining their own KGE to describe components or materials. The manufacturer composes a global KGE that integrates these subgraphs. All actors are accountable for their data and contribute toward a shared objective.
- **Collaborative Scientific Research:** Researchers maintain local KGEs to organize data and hypotheses. As part of a collaborative research initiative, they align concepts and exchange selected data views, but retain autonomy in their lifecycle and modeling.

To enable such scenarios, we extend the original definition of KGE by introducing a recursive structure where ecosystems can consist of multiple interrelated KGEs – each with its own lifecycle, goals, and context.

Extended Definition. A Knowledge Graph Ecosystem is now defined as:

$KGE = (KGM, \{KGE_1, \dots, KGE_n\}, LC, G, C)$, where:

- $KGM = (D, O, M, DC, KG, L)$ is the underlying **knowledge graph model**.
- $\{KGE_1, \dots, KGE_n\}$ is a (possibly empty) set of **nested or interacting KGEs**.
- LC is a **lifecycle**, formally modeled as a partially ordered structure of lifecycle steps, each representing operations or analyses on the KG.
- G is a set of **goals** pursued by the ecosystem.
- C is the **context**, capturing domain-specific and regulatory constraints or assumptions.

This extended definition allows the modeling of composite or hierarchical KGEs, enabling both *federated* and *decentralized* structures.

Federated KGEs. A KGE is **federated** if it aggregates multiple KGEs that share:

- **Common goals and context:** $G \subseteq \bigcap_{i=1}^n G_i$ and $C \subseteq \bigcap_{i=1}^n C_i$.
- **Shared concepts:** $O \models \bigcup_{i=1}^n O_i$ and $KG \subseteq \bigcup_{i=1}^n KG_i$.
- **Shared data:** $D \subseteq \bigcup_{i=1}^n D_i$.

This setup supports alignment and accountability among actors collaborating toward a unified goal, such as regulatory compliance or industry standards.

Use Case 1 (DPP) is an example of a federated KGE, where suppliers and manufacturers contribute to a shared infrastructure, follow a common goal, and are held accountable for the correctness of their data.

Decentralized KGEs. A KGE is **decentralized** if it includes autonomous KGEs that:

- Maintain **individual goals and contexts**, i.e., $G \cap \bigcup G_i \neq \emptyset$ or $C \cap \bigcup C_i \neq \emptyset$.
- Communicate via **alignments**: for each pair (i, j) , there exists an ontology O_{ij} such that $O_i \models_a O_{ij}$ and $O_j \models_a O_{ij}$, where $\models_a$ denotes entailment up to alignment.
- Possibly share some **data elements**.

Such configurations reflect looser, trust-based networks – e.g., research collaborations or international knowledge exchanges – where autonomy is preserved, and accountability is local. **Use Case 2 (Dagstuhl Seminar)** exemplifies this structure, with researchers individually maintaining their KGEs while participating in collaborative groups with overlapping knowledge and trust-based data sharing.

Summary. Extending the KGE model from single knowledge graphs to federated and decentralized ecosystems enables:

- Formal modeling of complex, distributed knowledge infrastructures.
- Explicit lifecycle tracking across ecosystem components.
- Representation of autonomy, trust, and alignment across diverse actors.

These extensions are essential to support knowledge-centric systems that operate across organizational, geographic, and technical boundaries.

5.5.5 Challenges and Future Work

Extending the KGE concept to support federated and decentralized ecosystems introduces new research challenges that go beyond those of traditional or standalone knowledge graphs. These challenges stem from the need to enable collaboration, preserve autonomy, ensure accountability, and foster trust across heterogeneous, interlinked systems.

1. Semantic Alignment in Heterogeneous Ecosystems. While classical KGEs already face challenges in integrating heterogeneous data, these are further exacerbated in federated and decentralized settings. Each participating KGE may use distinct ontologies, vocabularies, and constraints, making alignment a prerequisite for knowledge exchange. Federated settings assume shared semantics; decentralized ones require ontology alignments that preserve local autonomy while supporting partial interoperability. Ontology matching, cross-KGE mappings, and alignment reasoning are required to maintain interoperability and trust.

2. Trust and Accountability in Distributed Architectures. As KGEs become interlinked, accountability must be clearly defined across organizational and jurisdictional boundaries. Federated KGEs require all participants to commit to shared goals and contexts, making it possible to define collective accountability [2]. Decentralized KGEs, by contrast, emphasize autonomy – participants operate independently but must still be trusted sources. Establishing trust involves formalizing provenance, defining verifiable contributions, and enabling transparency in lifecycle actions. Ecosystem-wide logging mechanisms and policy-compliant data usage protocols are needed to support accountability and traceable decisions.

3. Modeling Autonomy and Interactions. In decentralized KGEs, each participant may pursue different goals or operate under distinct regulatory or ethical frameworks. Maintaining autonomy requires the ability to define local policies, ontologies, and lifecycle models – while still enabling interaction through lightweight semantics and alignments. Ensuring that knowledge exchange does not infringe on local autonomy calls for formal contracts, soft alignments, and partial knowledge views. Such mechanisms allow for trust-based collaboration while upholding the principle of self-determination [2].

4. Lifecycle Across KGEs. Federated and decentralized KGEs introduce the challenge of coordinating lifecycles across independently evolving components. In federated settings, lifecycle steps (e.g., updates, validations) must be synchronized to ensure consistent outcomes. In decentralized settings, looser coordination must support asynchronous updates and versioning. Future work should explore distributed lifecycle management models, including temporal consistency, event-driven propagation, and local override mechanisms.

5. KGE Validation and Provenance. Validation across KGEs requires understanding how constraints defined in one ecosystem apply to others. This is particularly challenging when KGE components evolve independently or when only partial views are exchanged. Provenance models must capture not only source alignment but also how knowledge was transformed or aligned across systems. Explainability – especially for outputs from KG-based AI systems – requires tracing decisions back to source KGEs and validating their integrity.

6. Explainability and Repeatability in KG-Based AI. As KGEs serve as the backbone for AI systems, the ability to explain inferences and reproduce results is crucial for building trust. This requires detailed provenance, clear alignment semantics, and interpretable reasoning paths – especially when hybrid approaches (e.g., involving LLMs) are used. Repeatability in decentralized settings must account for potential disappearance or evolution of external data sources. Mitigation mechanisms, such as fallback reasoning strategies or versioned snapshots, are essential to uphold the trustworthiness and reliability of AI outcomes.

7. Defining Role-Based Trust and Governance Models. With multiple stakeholders, each with distinct roles (e.g., data provider, knowledge builder, auditor, consumer), KGEs must support differentiated views, permissions, and responsibilities. Role-specific governance models and access policies must ensure that users interact with the ecosystem in ways that are transparent, justified, and traceable. This includes defining what actions are permissible, who is accountable for data changes, and how conflicts or inconsistencies are resolved.

8. Modeling Roles, Personas, and Task-Specific Interactions. KGEs serve a diverse set of stakeholders – including data stewards, domain experts, developers, auditors, and end users – each with distinct objectives, capabilities, and responsibilities. The extension to federated and decentralized KGEs further amplifies this diversity, requiring systems to explicitly model and support *role-specific interactions*. These roles influence how users contribute to, consume from, or govern the KGE. For example, a data provider must ensure schema and content quality, while a consumer needs trustable and explainable insights. Supporting personas involves:

- Designing user interfaces and APIs tailored to task-specific workflows.
- Implementing role-based access control and provenance-based accountability.
- Supporting transparency through lifecycle-aware logs and explainable outputs aligned with each persona’s mental model.

Future work must focus on formalizing persona definitions, mapping tasks to lifecycle stages, and capturing the responsibilities of each actor. These models are essential to foster trust and usability in multi-actor KGEs, and to ensure that autonomy and accountability are respected across stakeholder boundaries.

Summary. Federated and decentralized extensions of KGEs offer a powerful abstraction for supporting complex, multi-actor knowledge infrastructures. However, realizing these systems at scale requires addressing a new set of challenges around interoperability, lifecycle synchronization, role-specific trust, and accountability. Future research must focus on:

- *Trust*: Establishing mechanisms for verifiable contributions, alignment justifications, and explainable inferences.
- *Accountability*: Formalizing roles, logs, and lifecycle provenance to assign responsibility and enable redress.
- *Autonomy*: Supporting local lifecycles, policies, and ontologies while enabling coherent interactions across KGEs.

These directions are essential to align the evolution of KGEs with principles outlined in the proposed EU AI Act ¹¹, including transparency, reliability, and respect for autonomy.

References

- 1 Sandra Geisler, Cinzia Cappiello, Irene Celino, David Chaves-Fraga, Anastasia Dimou, Ana Iglesias-Molina, Maurizio Lenzerini, Anisa Rula, Dylan Van Assche, Sascha Welten, et al. From genesis to maturity: Managing knowledge graph ecosystems through life cycles. *Proceedings of the VLDB Endowment*, 2025.
- 2 Luis-Daniel Ibáñez, John Domingue, Sabrina Kirrane, Oshani Seneviratne, Aisling Third, and Maria-Esther Vidal. Trust, accountability, and autonomy in knowledge graph-based AI for self-determination. *TGDK*, 1(1):9:1–9:32, 2023.

¹¹<https://artificialintelligenceact.eu/the-act/>