From Server-centric to Client-centric Data Integration over Decentralized Knowledge Graphs with Personal Query Engines

Ruben Taelman

Department of Electronics and Information Systems, Ghent University – imec

Abstract

Knowledge Graph technologies provide us with the ability to interlink data across organizational boundaries, which lead to implicit *Decentralized Knowledge Graphs (DKGs)*. This interlinking ability makes DKGs a natural foundation for public and private data, as all data can be stored and managed by the people or authorities owning this data. When one wants to build applications or services on top of DKGs, this decentralized nature leads to major technical challenges regarding data integration. Today, most of these challenges are being tackled in a *server-centric* manner, where decentralized data is integrated and *centralized* within servers that run somewhere in the cloud, far from the control of the people that actually own the data. Consequentially, this server-centricity lies at the root of many privacy concerns surrounding personal data. In this article, I argue that the ability to store data in a decentralized manner is insufficient for solving issues surrounding privacy and trustworthiness. Additionally, we need to rethink how and where we *integrate* this data, and perform computations over it. Concretely, this article focuses on a paradigm shift from *server-centric* to *client-centric* data integration, by providing users with reusable and *personal query engines*. These are client-side engines that are in the driving seat for integrating decentralized data, as a basis for offering users with full algorithmic control over how their data is processed, in a transparent manner, which is private by design. This article discusses three open research challenges that are fundamental to *client-centric query processing* for driving data integration within decentralized applications. Namely, the challenges of heterogeneity, personalization, and performance. While server-centric data integration maximizes performance by sacrificing user control, transparency, and privacy, a client-centric approach does not require this sacrifice, by shifting focus to the client. This offers a new paradigm for application domains where decentralization is an inherent property, such as healthcare, social media, and personal genomics.

Keywords

Decentralized Knowledge Graphs, Query Processing, Personalization

1. Introduction: *How can decentralized applications integrate data?*

The Web has been a driving force behind societal advancement, transforming areas such as our economy, healthcare, social media, and more. Despite its massive benefits, **most data on the Web are centralized**. This centralization leads to data 1) being in control of a handful multinationals, 2) being spread over a small number of large *data silos*, and 3) silos being mostly application-domain-specific, focusing on data relating to social media, personal genomics [1], and healthcare. Unfortunately, this **centralized nature of data silos has lead to problems** such as election influence [2, 3, 4], emotional manipulation [5], censorship [6, 7], and personal data breaches [8].

Due to growing awareness [9] and privacy-concerning legislation (GDPR, CCPA, AI Act, . . .), there is an increasing push to **enable people to control their data**. Various **decentralized environments** [10, 11, 12, 13, 14, 15] aim to break down centralized data silos. As a consequence of decentralization, 1) people gain control over how and where their data is stored, and who can use (parts of) their data, 2) data is spread across many (personal) data sources that become decoupled from applications, and 3) data sources are heterogeneous and application-domain-agnostic. While advancements are being made on decentralized storage [10, 11, 12, 13, 14, 15], identity [16], and usage control [17, 18, 19], the **integration**

Posters, Demos, Blue Sky, and Tutorials at SEMANTiCS 2026, Sep 2026, Ghent, Belgium

ruben.taelman@ugent.be (R. Taelman)
https://www.rubensworks.net/ (R. Taelman)
0000-0001-5118-256X (R. Taelman)

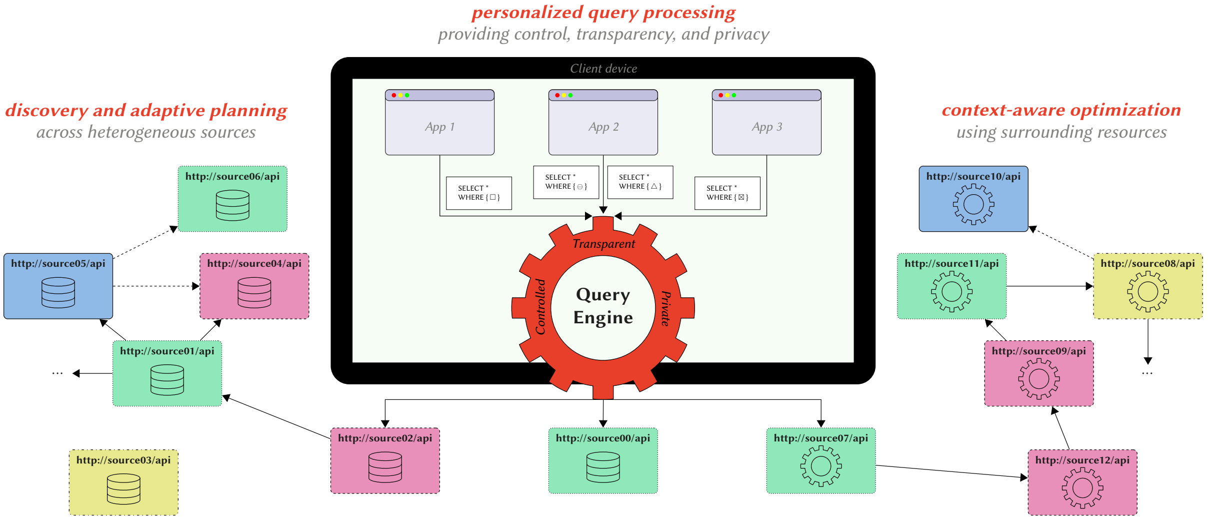

Figure 1: By focusing on client-centric data integration, Decentralized Knowledge Graphs that are spread across multiple data sources can be integrated within **personal query engines**. Such a query engine runs close to the user, and is reusable by multiple user-specific applications through a query-based abstraction layer. These engines must have the necessary algorithms to 1) discover decentralized data sources, 2) offer full control, transparency, and privacy to the user, 3) and be sufficiently fast by also exploiting surrounding resources.

of decentralized data is not well understood yet. As we evolve towards such a decentralized future, we urgently require a fundamental understanding of decentralized data integration.

Today, the integration of data mostly happens in a *server-centric manner*, where data is extracted from various data sources, possibly transformed, and loaded into a central server. While applications have become highly optimized for this paradigm to meet our human attention spans [20], our technology is incompatible with the goals and needs of decentralized applications due to 1) processing not being under the user's control, 2) systems often acting as a black-box, and 3) the difficulty of preserving privacy when handling personal data. Instead of attempting to solve these problems that are inherent to the server-centric paradigm, I propose a **mindset shift towards *client-centric data integration***, where data storage locations and data processing locations become separated, and where the user's device 1) is in control of how the data flows, 2) works in a white-box manner, and 3) is private by design. With decentralized applications becoming decoupled from both where the data is stored *and* integrated, applications can run closer to the client-side, with more user control, transparency, and privacy. This shift towards client-side computation is timely due to aforementioned **legislative requirements**, and **our increasingly more powerful client devices and rapid networking advancements** (e.g. 5G, Wi-Fi) that reduce latency and boost bandwidth.

Today, working with decentralized data is highly challenging for application developers compared to working with centralized data, because 1) data is spread over a large number of heterogeneous sources, 2) users and legislation require control and transparency, and 3) data processing over these remote data sources must be fast enough to meet human attention spans for interactive applications. To not require every decentralized application to tackle these same challenges, there is a need to **hide such complexities behind *declarative queries***, with *reusable* query algorithms enabling decentralized data integration, while being in control of users and still achieving acceptable levels of performance. As shown in Figure 1, these query algorithms can be implemented in *personal query engines*, which are in full control of users, and offer decentralized data integration capabilities for the applications they use. Such personal query engines also offer a convenient interface for AI Agents to interact with decentralized data, which enables non-experts to interact with it as well [21].

After discussing related work in the next section, the main scientific challenges of client-centric data integration will be discussed. Finally, we conclude by discussing the possible wider impact of this paradigm shift.

2. Related Work: *Current technology focuses on server-driven data integration*

Knowledge Graphs organize decentralized data To support *data format and storage heterogeneity* in a decentralized environment, many decentralization initiatives [10, 11, 15] use Knowledge Graph (KG) technologies [22, 23] such as RDF and SPARQL. *RDF* [22] is a standard for representing KGs as a set of *triples* with universal semantics, allowing data interlinking across sources without centralized ontological agreements. *SPARQL* [23] is the standard RDF query language for gaining insights.

SPARQL federation is unsuitable for decentralization Since KGs can be interlinked across organizational boundaries [24, 25], this data can be virtually integrated through *SPARQL federated query processing (FQP)* [26, 27, 28, 29, 30, 31, 32, 33, 34]. In contrast to *polystores* [35, 36, 37, 38] that tackle the heterogeneity of storage engines through query-time translation, FQP delegates such translation to data publishers through Web interfaces. Despite performing well in controlled distributed environments, FQP algorithms are unsuitable for decentralized environments as they assume 1) homogeneous interfaces with limited scalability [39, 40, 41, 34, 42], 2) a low number of sources (±10) [43], and 3) prior knowledge of sources [44].

Tackling heterogeneity and discovery In recent years, work has been done towards tackling the *heterogeneity* of Web interfaces during FQP, and balancing client- and server-effort. Yet, **current algorithms [45, 46, 47, 48] still assume prior knowledge about the interface's querying capabilities** [49], which is unrealistic in decentralized environments. To tackle the issues involving the number and prior knowledge of sources, work has been done in the field of *Link Traversal Query Processing (LTQP)* [50, 51, 52, 53, 54, 55, 56, 57, 57], which discovers sources *during* query execution, by shifting towards *exploiting structural properties* in decentralized environments with many sources (±100.000) [58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]. However, **performance is still low for complex queries** due to 1) not all interface capabilities being exploited [68], 2) large numbers of links being followed [64, 57], and 3) ineffective query plans [58, 61]. To tackle these problems, **there is a need to integrate the LTQP and FQP paradigms to combine their respective strengths**, which has been an open problem for 7+ years [69, 70].

Personalized query processing is highly centralized Content-driven applications for topics such as movies, music, and social media are often personalized using Recommender Systems [71, 72], which suggest items based on user's navigation history and preferences. They rely on *application-dictated* models [73, 72] that lack transparency [74] and have limited configurability [75]. As these have led to privacy and manipulation scandals [2, 3, 4, 5, 6, 7, 8], **there is a need to move from *application-driven* algorithms to *user-driven* algorithms [76]**. While such *algorithmic freedom* could be given to users by using declarative queries as abstraction layer, existing **personalized query algorithms [77, 78, 79, 60] are insufficient for decentralized environments** as they lack 1) preference models for cross-application interoperability, 2) trust definitions across data sources, and 3) operational preferences [80].

Optimizing decentralized SPARQL query execution is not well understood First, trade-offs between I/O, battery usage, and CPU time for query planning [81, 82, 83] have not been explored yet in the context of SPARQL, and existing **resource adaptation techniques do not yet consider the dynamicities experienced in personal client devices** [80, 84], such as changing environments for mobile devices or performance fluctuations due to competing processes. Second, materialized views [85, 86, 87, 88, 89] are commonly used in centralized settings to reduce execution time of query execution at the cost of increased storage. Despite their benefits, **we lack understanding of how materialized views behave in a decentralized environment**. Third, collaborative SPARQL querying could improve performance, but either assumes public data [26, 90, 91, 92], or performs full privacy-preservation [93, 94] which is too slow for interactive use [95]. While decentralized environments consist of public *and* private data, we lack understanding of **combining public and private collaborative query paradigms**.

3. Scientific challenges of client-centric data integration

Enabling data integration for decentralized applications calls for a client-driven query processing approach, for which I foresee three high-level research challenges: 1) A **fundamental understanding of client-driven query processing in a large heterogeneous environment** is required. 2) This then opens the door towards empowering users in gaining **personalized control over client-driven query processing**. 3) There is a need to make this form of processing useful in practice through **optimization to user-relevant metrics** [20]. Hereafter, these challenges are explained in more detail, with conceptual solution directions in the form of a research roadmap.

3.1. Challenge 1: Client-driven query processing over heterogeneous data sources

To query over data sources that have varying interfaces and are not known upfront, there are three fundamental research objectives to achieve discovery, query planning, and link pruning (Figure 2).

Objective 1a: **Hypermedia-based federation over heterogeneous sources** To enable querying without making prior assumptions on interface capabilities, we require query planning and evaluation algorithms to federate over such interfaces through *autonomous usage of hypermedia controls*.
Objective 1b: **Adaptive query planning over dynamically discovered sources** To optimize querying over sources that are *not known beforehand*, we require authority-aware query planning algorithms that adaptively *assign sources discovered during execution* to subqueries.
Objective 1c: **Join- and authority-aware link pruning** To achieve more selective link traversal across data sources, we require an authority-aware link pruning approach that takes into account *how triple patterns are joined* instead of considering them in isolation.

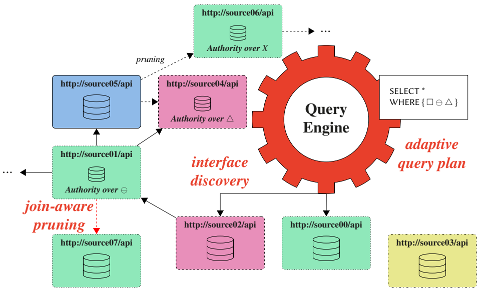

Figure 2: Adaptive planning and join-aware pruning over heterogeneous sources discovered during execution.

Objective 1a will enable engines to execute queries over query interfaces without relying on prior assumptions on the expressivity and capabilities of these interfaces. To determine query decomposition over interfaces, we can introduce **declarative hypermedia-based [96] query shapes** as an extension of *RDF data shapes* [97]. Using discovered query shapes, we can apply *query containment* [66, 88] to assess during query planning how to use interfaces. Evaluation must compare query execution time against hypermedia overhead, which is likely to be negligeable compared to HTTP-induced latencies.

Objective 1b will use the full expressivity of query interfaces that are *discovered during execution*. As opposed to considering triple-pattern-level source assignment over document-oriented interfaces during LTQP [52], we can incorporate the *exclusive groups* principle [27] from FQP to **enable source assignment to larger subqueries**. Since exclusive groups are incompatible with source discovery, we can introduce **authority-based source selection** allowing publishers to signal *authority over shapes* of data. Instead of relying on heuristics [98], we must adapt the query plan *during* execution [57, 99, 100, 101, 102, 61]. Evaluation should measure lower total and incremental [103] execution times and bandwidth usage with higher server costs.

Objective 1c will reduce the number of links during cross-source traversal. Since the number of links can explode when traversing sources [65], we can use *authority shapes* (Objective 1b) as a *link pruning* *approach* [60]. This **pruning will rely on early-completing join operations** after being established as authoritative subqueries, which otherwise only complete when the link queue is fully drained. Links determined from this subquery through reachability criteria [52] over triples that did not contribute to the joined results can be pruned. Evaluation should measure reductions in HTTP requests, intermediary results, total and incremental [103] execution times, without reductions in completeness.

3.2. Challenge 2: Personalized client-driven query processing

To give users personalized control over their query engine, there are three fundamental research objectives to manage and use decentralized preferences, and achieve user-oriented optimizations.

Objective 2a: **Decentralized user preferences framework** To give users fine-grained control on how their data will be processed, we need to model explicit preferences (such as interests, trust, and operational settings), and implicit preferences based on user history. Techniques are required to track and manage them in a decentralized setting for cross-application interoperability.

Objective 2b: **Query processing driven by decentralized preferences** To give users fine-grained control and insights into how decentralized data are retrieved, combined, and processed, we need query algorithms that are driven by combinable user-oriented preferences.

Objective 2c: **User-oriented inter-query optimization** To lower execution time and bandwidth usage for sequences of query executions under equal user preferences, we need a client-side caching strategy based on personal usage patterns, which will also anticipate future data needs.

Objective 2a will model decentralized user preference representation. Inspired by recommender system models [104], we can introduce an **abstract framework and vocabulary [105] that captures explicit and implicit user preferences** that are *scattered* and *composable* across multiple data sources. Besides traditional explicit preferences such as topical interests [75], this may include **trust relationships** [60], and **operational settings** like preferring trustworthy results instead of fast results. Implicit preferences can be represented as **user history through application-agnostic provenance** [106].

Objective 2b will enable query algorithms to be driven by user preferences and providing insights. First, user-related preferences that are discovered during LTQP [52] can define a **preference-driven cost model** that drives focused traversal [107] through link pruning [60] and prioritization [57]. Second, to provide insights we can **incorporate *how- and where-provenance* [108] into LTQP** by extending the *spm-semirings* [109] model towards discovered sources during traversal. Evaluation must measure subjective trustworthiness of personalization and insights through user studies, and their overhead on execution time is measured in terms of preference size and complexity.

Objective 2c will enable engines to execute sequences of queries faster, as user-specific queries are similar due to recurring patterns [110, 111]. As opposed to existing user-agnostic caching techniques for LTQP [112], we can design a **user-pattern-aware caching and link prefetching strategy** using existing user-patterns [110, 111]. Furthermore, **query plans for subsequent queries will be determined from partial cardinalities** obtained from the cache rather than only relying on heuristics [98]. Evaluation should evaluate the trade-off between cache size, execution time, and completeness [112].

3.3. Challenge 3: Context-aware optimization of client-driven query processing

To optimize processing with discovered capabilities, there are three fundamental objectives (Figure **3**).

Objective 3a: **Resource-aware query planning** To optimize the utility of local and remote server resources, we require algorithms that construct resource-aware query plans.
Objective 3b: **Federation-aware materialized views** To minimize computational effort across clients handling similar public data, we need federation-aware materialized views that cover heterogeneous sources, and use them during federated query execution.
Objective 3c: **Privacy-aware outsourcing of query processing** To minimize the computational effort of single client-side engines handling protected data, we need privacy-aware query decomposition to outsource computation to one or multiple parties.

Objective 3a will enable engines to optimize execution with surrounding resources. Since client performance is context-dependent, we can introduce a **resource-aware cost model for SPARQL query planning** that –besides speed– incorporates metrics such as battery usage [82], CPU usage [83], latency, and bandwidth [113]. Metrics related to remote sources will be discovered through declarative server announcements [105]. As metrics can vary across time for long-running queries or mobile clients [80], the query plan can be adapted *during* execution [57, 99, 100, 101, 102] based on this cost model. Evaluation must compare adaptation of the cost model to different scenarios, such as varying battery levels and fluctuating network stability.

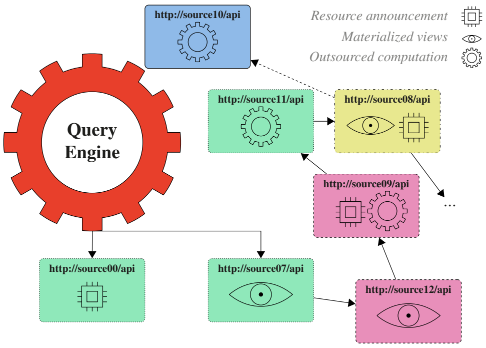

Figure **3**: Query engine exploits discovered interface capabilities, including resource announcements, materialized views, and outsourced computation.

Objective 3b will enable engines to reuse executions by other engines over similar public data. First, we can **extend the model of materialized RDF views to incorporate *multiple heterogeneous* data sources** that can vary per view and query, as opposed to assuming one source [86]. Based on this, we can introduce ***federated* query rewriting algorithms** [89, 114], by extending SPARQL query containment [88] towards federation and heterogeneity. Evaluation should measure the trade-off between view storage size and query execution time.

Objective 3c will enable query execution to be outsourced with privacy preservation. In contrast to full query outsourcing via computationally-expensive MPC techniques [95, 93, 94], we can introduce **privacy-aware query decomposition across heterogeneous sources**. Using MPC for SPARQL [115], we can balance computations between the client, privacy-preserving parties, and privacy-agnostic parties. To support heterogeneous sources beyond SPARQL endpoints [116], we can annotate query shapes (Objective 1a) with *usage control descriptions* [18, 17], for which we can design a novel **privacy-aware query containment algorithm [66, 88]**. Evaluation should aim to measure reductions in execution time, MPC computation size, and bandwidth usage.

4. Impact and conclusions

Privacy concerns in data-oriented applications are establishing a new relationship between users and service providers. By shifting from *server-centric* to *client-centric* data integration via personal query engines, users can be in full control over not only how their data is *stored*, but how it's *processed*. Besides the immediate technical challenges described above that need to be tackled first, such a paradigm-shift will also open up the following research directions in the field of (SPARQL) query processing:

1. **Objective 1: Performant querying *without* prior knowledge of sources**: Research mostly focuses

on the FQP paradigm due to its superiour performance over LTQP. Tackling the open problem [70, 69] of combining both paradigms, will result in a hybrid paradigm with on-the-fly source discovery *and* high performance, which will trigger new query optimization research opportunities.
2. **Objective 2: Preference-specific optimization**: Establishing a preference-driven querying paradigm for decentralized environments opens doors to preference-specific query optimization research, such as specific query planning techniques for achieving both high trust and low bandwidth usage.
3. **Objective 3: Shift from singular performance optimization to context-aware optimization**: while SPARQL query optimization research today mostly optimizes for total execution time, we can broaden this view towards measuring trade-offs between resource utility and privacy-preservation.

Long-term, such a paradigm shift will impact our society long-term through:

1. **Increased decentralization of data**: With data being decoupled from applications, the need for silo-oriented data centralization is reduced. This leads to data being usable *across* applications, which enables businesses to focus more on application development instead of data collection, and aligns with the European *"once-only"* principle [117, 118] on avoiding storing data multiple times.
2. **Lower cost of decentralized application development**: By decoupling data discovery and integration from applications, businesses can focus development on their interests and expertise.
3. **Higher trust from application users**: Giving users higher transparency and controllability data in how their data is processed, can restore trust [9] in the Web and its applications following recent scandals [2, 3, 4, 5, 6, 7, 8]. Further, it will open business opportunities in areas that benefit from personalization, such as social media [11, 12, 13], mixed reality devices, wearables, brain-computer interfacing.

Declaration on Generative AI

No generative AI was used to produce this article.

Acknowledgements

Ruben Taelman is a postdoctoral researcher at the Research Foundation – Flanders (FWO) (1202124N).

References

[1] E. Crum, R. Taelman, B. Buelens, G. Ertaylan, Genomic sequence data sharing for clinical practice: A scoping review, Computers in Biology and Medicine 200 (2026) 111335. URL: https://www.sciencedirect.com/science/article/pii/S0010482525016890. doi:`https://doi.org/10.1016/j.compbiomed.2025.111335`.

[2] How filter bubbles distort reality: Everything you need to know, 2019. URL: https://fs.blog/filter-bubbles/.

[3] Congress grills facebook ceo over data misuse – as it happened, 2018. URL: https://www.theguardian.com/technology/live/2018/apr/10/mark-zuckerberg-testimony-live-congress-facebook-cambridge-analytica.

[4] R. Epstein, R. E. Robertson, The search engine manipulation effect (seme) and its possible impact on the outcomes of elections, Proceedings of the national academy of sciences 112 (2015) E4512–E4521.

[5] Facebook emotion study breached ethical guidelines, researchers say, 2014. URL: https://www.theguardian.com/technology/2014/jun/30/facebook-emotion-study-breached-ethical-guidelines-researchers-say.

[6] J. D. Miller, Against big tech viewpoint discrimination, 2019. URL: https://quillette.com/2019/06/06/against-big-tech-viewpoint-discrimination/.

[7] A. Picchi, Twitter files: What they are and why they matter, 2022. URL: https://www.npr.org/2025/03/24/nx-s1-5338622/23andme-bankruptcy-genetic-data-privacy.

[8] J. Hernandez, 23andme is filing for bankruptcy. here's what it means for your genetic data, 2024. URL: https://www.cbsnews.com/news/twitter-files-matt-taibbi-bari-weiss-michael-shellenberger-elon-musk/.

[9] Solid monitor, 2023. URL: https://solidlab.be/wp-content/uploads/2025/01/SM_2024_online_version.pdf.

[10] E. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, T. Berners-Lee, A demonstration of the Solid platform for social web applications, in: Proceedings of the 25th International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 223–226.

[11] M. Zignani, S. Gaito, G. P. Rossi, Follow the "Mastodon": Structure and evolution of a decentralized online social network, in: Twelfth International AAAI Conference on Web and Social Media, 2018.

[12] M. Kleppmann, P. Frazee, J. Gold, J. Graber, D. Holmgren, D. Ivy, J. Johnson, B. Newbold, J. Volpert, Bluesky and the at protocol: Usable decentralized social media, in: Proceedings of the ACM Conext-2024 Workshop on the Decentralization of the Internet, 2024, pp. 1–7.

[13] Y. Wei, G. Tyson, Exploring the nostr ecosystem: A study of decentralization and resilience, arXiv preprint arXiv:2402.05709 (2024).

[14] A. Braud, G. Fromentoux, B. Radier, O. Le Grand, The road to european digital sovereignty with gaia-x and idsa, IEEE network 35 (2021) 4–5.

[15] T. Kuhn, R. Taelman, V. Emonet, H. Antonatos, S. Soiland-Reyes, M. Dumontier, Semantic micro-contributions with decentralized nanopublication services, PeerJ Computer Science (2021). URL: https://peerj.com/articles/cs-387/. doi:`10.7717/peerj-cs.387`.

[16] M. Sporny, A. Guy, M. Sabadello, D. Reed, Decentralized Identifiers (DIDs) v1.0, Recommendation, W3C, 2022. URL: https://www.w3.org/TR/did-core/.

[17] A. Lazouski, F. Martinelli, P. Mori, Usage control in computer security: A survey, Computer Science Review 4 (2010) 81–99.

[18] R. Iannella, S. Villata, ODRL Information Model 2.2, Recommendation, W3C, 2018. URL: https://www.w3.org/TR/odrl-model/.

[19] M. Sporny, T. Thibodeau Jr, I. Herman, M. B. Jones, G. Cohen, Verifiable Credentials Data Model v2.0, Candidate Recommendation Draft, W3C, 2024. URL: https://www.w3.org/TR/vc-data-model-2.0/.

[20] J. Nielsen, Response times: the three important limits, Usability Engineering (1993).

[21] R. Taelman, J. De Smet, Comunica mcp sparql: Improving the accuracy of ai agents using sparql-based access to decentralized knowledge graphs, in: Proceedings of the 23rd Extended Semantic Web Conference: Posters and Demos, 2026. URL: https://rubensworks.github.io/article-eswc2026-demo-comunica-mcp/.

[22] O. Hartig, P.-A. Champin, G. Kellogg, A. Seaborne, RDF 1.2: Concepts and Abstract Syntax, Working Draft, W3C, 2024. URL: https://www.w3.org/TR/rdf12-concepts/.

[23] O. Hartig, A. Seaborne, R. Taelman, G. Williams, T. Pellissier Tanon, SPARQL 1.2 Query Language, Working Draft, W3C, 2024. URL: https://www.w3.org/TR/sparql12-query/.

[24] T. Berners-Lee, Linked data, Design Issues, 2009. URL: https://www.w3.org/DesignIssues/LinkedData.html.

[25] C. Bizer, T. Heath, T. Berners-Lee, Linked data: The story so far, in: Semantic services, interoperability and web applications: emerging concepts, IGI Global, 2011, pp. 205–227.

[26] Z. Gu, F. Corcoglioniti, D. Lanti, A. Mosca, G. Xiao, J. Xiong, D. Calvanese, A systematic overview of data federation systems, Semantic Web 15 (2024) 107–165.

[27] A. Schwarte, P. Haase, K. Hose, R. Schenkel, M. Schmidt, Fedx: Optimization techniques for federated query processing on linked data, in: International semantic web conference, Springer, 2011, pp. 601–616.

[28] M. Saleem, A.-C. N. Ngomo, Hibiscus: Hypergraph-based source selection for SPARQL endpoint federation, in: European semantic web conference, Springer, 2014, pp. 176–191.

[29] O. Görlitz, S. Staab, Splendid: SPARQL endpoint federation exploiting void descriptions, in:

Proceedings of the Second International Conference on Consuming Linked Data-Volume 782, CEUR-WS. org, 2011, pp. 13–24.

[30] M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo, E. Ruckhaus, Anapsid: an adaptive query processing engine for sparql endpoints, in: The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10, Springer, 2011, pp. 18–34.

[31] A. Charalambidis, A. Troumpoukis, S. Konstantopoulos, Semagrow: Optimizing federated sparql queries, in: Proceedings of the 11th international conference on semantic systems, 2015, pp. 121–128.

[32] M. Saleem, A. Potocki, T. Soru, O. Hartig, A.-C. N. Ngomo, Costfed: Cost-based query optimization for sparql endpoint federation, Procedia Computer Science 137 (2018) 163–174.

[33] J. Aimonier-Davat, B. Nédelec, M.-H. Dang, P. Molli, H. Skaf-Molli, Fedup: Querying large-scale federations of sparql endpoints, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 2315–2324.

[34] R. Verborgh, M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, P. Colpaert, Triple pattern fragments: a low-cost knowledge graph interface for the web, Journal of Web Semantics 37 (2016) 184–206.

[35] M. J. Carey, L. M. Haas, P. M. Schwarz, M. Arya, W. F. Cody, R. Fagin, M. Flickner, A. W. Luniewski, W. Niblack, D. Petkovic, et al., Towards heterogeneous multimedia information systems: The garlic approach, in: Proceedings RIDE-DOM'95. Fifth International Workshop on Research Issues in Data Engineering-Distributed Object Management, IEEE, 1995, pp. 124–131.

[36] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, J. Widom, The tsimmis project: Integration of heterogenous information sources (1994).

[37] J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, S. Zdonik, The bigdawg polystore system, ACM Sigmod Record 44 (2015) 11–16.

[38] M. Karpathiotakis, I. Alagiannis, A. Ailamaki, Fast queries over heterogeneous data through engine customization, Proceedings of the VLDB Endowment 9 (2016) 972–983.

[39] C. Buil-Aranda, A. Hogan, J. Umbrich, P.-Y. Vandenbussche, Sparql web-querying infrastructure: ready for action?, in: The Semantic Web–ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II 12, Springer, 2013, pp. 277–293.

[40] P.-Y. Vandenbussche, J. Umbrich, L. Matteis, A. Hogan, C. Buil-Aranda, Sparqles: Monitoring public sparql endpoints, Semantic web 8 (2017) 1049–1065.

[41] J. Hanski, E. Crum, R. Taelman, Observations on automated client-side query federation over wikidata SPARQL endpoints, in: 5th Wikidata Workshop for the scientific Wikidata community, 2025. Accepted for publication.

[42] R. Taelman, Did ai crawlers kill sparql federation?, 2026. URL: https://www.rubensworks.net/blog/2026/04/13/did-ai-clawlers-kill-sparql-federation/.

[43] M.-H. Dang, J. Aimonier-Davat, P. Molli, O. Hartig, H. Skaf-Molli, Y. Le Crom, Fedshop: A benchmark for testing the scalability of sparql federation engines, in: International Semantic Web Conference, Springer, 2023, pp. 285–301.

[44] A. Seaborne, R. Taelman, G. Williams, T. Pellissier Tanon, SPARQL 1.2 Protocol, Working Draft, W3C, 2024. URL: https://www.w3.org/TR/sparql12-protocol/.

[45] R. Taelman, J. Van Herwegen, M. Vander Sande, R. Verborgh, Comunica: a modular SPARQL query engine for the web, in: International Semantic Web Conference, Springer, 2018, pp. 239–255. URL: https://comunica.github.io/Article-ISWC2018-Resource/.

[46] L. Heling, M. Acosta, Federated sparql query processing over heterogeneous linked data fragments, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1047–1057.

[47] S. Cheng, O. Hartig, Fedqpl: A language for logical query plans over heterogeneous federations of RDF data sources, in: Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services, 2020, pp. 436–445.

[48] G. Montoya, C. Aebeloe, K. Hose, Towards efficient query processing over heterogeneous RDF

interfaces, in: 2nd Workshop on Decentralizing the Semantic Web, DeSemWeb 2018, CEUR Workshop Proceedings, 2018.

[49] R. Taelman, R. Verborgh, Declaratively describing responses of hypermedia-driven Web APIs, in: Proceedings of the 9th International Conference on Knowledge Capture, 2017. URL: https://linkeddatafragments.github.io/Article-Declarative-Hypermedia-Responses/.

[50] O. Hartig, An overview on execution strategies for linked data queries, Datenbank-Spektrum 13 (2013) 89–99.

[51] O. Hartig, SPARQL for a Web of Linked Data: semantics and computability, in: Extended Semantic Web Conference, Springer, 2012, pp. 8–23.

[52] O. Hartig, J.-C. Freytag, Foundations of traversal based query execution over Linked Data, in: Proceedings of the 23rd ACM conference on Hypertext and social media, ACM, 2012, pp. 43–52.

[53] D. P. Miranker, R. K. Depena, H. Jung, J. F. Sequeda, C. Reyna, Diamond: A SPARQL query engine, for linked data based on the Rete match, in: Workshop on Artificial Intelligence meets the Web of Data, 2012.

[54] J. Umbrich, A. Polleres, A. Hogan, S. Decker, Link traversal querying for a diverse Web of Data, Semantic Web 6 (2015) 585–624.

[55] G. Ladwig, T. Tran, Linked Data query processing strategies, in: ISWC, Springer, 2010, pp. 453–469.

[56] O. Hartig, Squin: a traversal based query execution system for the web of linked data, in: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, ACM, 2013, pp. 1081–1084.

[57] O. Hartig, M. T. Özsu, Walking without a map: Ranking-based traversal for querying linked data, in: International Semantic Web Conference, Springer, 2016, pp. 305–324.

[58] R. Taelman, R. Verborgh, Link traversal query processing over decentralized environments with structural assumptions, in: Proceedings of the 22nd International Semantic Web Conference, 2023. URL: https://comunica.github.io/Article-ISWC2023-SolidQuery/.

[59] R. Eschauzier, R. Taelman, R. Verborgh, Revisiting link prioritization for efficient traversal in structured decentralized environments, in: Proceedings of the 24th International Semantic Web Conference, 2025. URL: https://tinyurl.com/muau6nwt.

[60] B. Bogaerts, B. Ketsman, Y. Zeboudj, H. Aamer, R. Taelman, R. Verborgh, Link traversal with distributed subweb specifications, in: Rules and Reasoning: 5th International Joint Conference, RuleML+RR 2021, Leuven, Belgium, September 8 – September 15, 2021, Proceedings, 2021. URL: https://www.bartbogaerts.eu/articles/2021/005-RuleML-GuidedLink-SubwebSpec/SubwebSpecifications.pdf.

[61] J. Hanski, R. Taelman, R. Verborgh, Link traversal over decentralised environments using restart-based query planning, in: Proceedings of the 25th International Conference on Web Engineering, 2025. URL: https://tinyurl.com/4wy8smau.

[62] J. De Smet, R. Taelman, Sgf: Sparql updates over decentralized knowledge graphs without access path dependencies, in: Proceedings of the 22nd Extended Semantic Web Conference: Posters and Demos, 2025. URL: https://sgf-demo-eswc-2025.jitsedesmet.be/.

[63] B.-E. Tam, R. Taelman, J. A. Rojas Meléndez, P. Colpaert, Optimizing traversal queries of sensor data using a rule-based reachability approach, in: Proceedings of the 23rd International Semantic Web Conference: Posters and Demos, 2024. URL: https://arxiv.org/pdf/2408.17157.

[64] R. Eschauzier, R. Taelman, R. Verborgh, The r3 metric: Measuring performance of link prioritization during traversal-based query processing, in: Proceedings of the 16th Alberto Mendelzon International Workshop on Foundations of Data Management, 2024. URL: https://rubeneschauzier.github.io/traversal-metric-workshop-paper/.

[65] R. Eschauzier, R. Taelman, R. Verborgh, How does the link queue evolve during traversal-based query processing?, in: Proceedings of the 7th International Workshop on Storing, Querying and Benchmarking Knowledge Graphs, 2023. URL: https://rubeneschauzier.github.io/QuWeDa2023-Link-Queue-Analysis-Camera-Ready/QuWeDa_2023_Link_Queue_Analysis_Final.pdf.

[66] B.-E. Tam, R. Taelman, P. Colpaert, R. Verborgh, Opportunities for shape-based optimization of link traversal queries, in: Proceedings of the 16th Alberto Mendelzon International Workshop on Foundations of Data Management, 2024. URL: https://arxiv.org/pdf/2407.00998v2.

[67] J. Hanski, R. Taelman, R. Verborgh, Observations on bloom filters for traversal-based query execution over solid pods, in: Proceedings of the 21st Extended Semantic Web Conference: Posters and Demos, 2024. URL: https://www.rubensworks.net/raw/publications/2024/hanski_bloom_solid_2024.pdf.

[68] R. Taelman, Towards applications on the decentralized web using hypermedia-driven query engines, ACM SIGWEB Newsletter (2024). URL: https://tinyurl.com/ytmkb35y.

[69] O. Hartig, K. Hose, J. Sequeda, Linked data management, in: Encyclopedia of Big Data Technologies, Springer, 2018, pp. 1–7.

[70] R. Eschauzier, R. Taelman, Towards hybrid link traversal: Challenges and research directions for heterogeneous dataspaces, in: Proceedings of the 4th International Workshop on Semantics in Dataspaces, 2026. URL: https://rubeneschauzier.github.io/position-paper-hybrid-link-traversal-sds-workshop/position_paper_hybrid_querying_sds_2026_camera_ready.pdf.

[71] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-based systems 46 (2013) 109–132.

[72] C. Kalyvas, T. Tzouramanis, A survey of skyline query processing, arXiv preprint arXiv:1704.01788 (2017).

[73] I. Portugal, P. Alencar, D. Cowan, The use of machine learning algorithms in recommender systems: A systematic review, Expert Systems with Applications 97 (2018) 205–227.

[74] Y. Ge, S. Liu, Z. Fu, J. Tan, Z. Li, S. Xu, Y. Li, Y. Xian, Y. Zhang, A survey on trustworthy recommender systems, ACM Transactions on Recommender Systems 3 (2024) 1–68.

[75] D. Jannach, S. Naveed, M. Jugovac, User control in recommender systems: Overview and interaction challenges, in: E-Commerce and Web Technologies: 17th International Conference, EC-Web 2016, Porto, Portugal, September 5-8, 2016, Revised Selected Papers 17, Springer, 2017, pp. 21–33.

[76] J. Graber, Algorithmic choice, 2023. URL: https://bsky.social/about/blog/3-30-2023-algorithmic-choice.

[77] G. Koutrika, Y. Ioannidis, Personalization of queries in database systems, in: Proceedings. 20th International Conference on Data Engineering, IEEE, 2004, pp. 597–608.

[78] D. Yoo, Hybrid query processing for personalized information retrieval on the semantic web, Knowledge-Based Systems 27 (2012) 211–218.

[79] O. Hartig, Querying trust in RDF data with tsparql, in: The Semantic Web: Research and Applications: 6th European Semantic Web Conference, ESWC 2009 Heraklion, Crete, Greece, May 31–June 4, 2009 Proceedings 6, Springer, 2009, pp. 5–20.

[80] H. Grine, T. Delot, S. Lecomte, Adaptive query processing in mobile environment, in: Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing, 2005, pp. 1–8.

[81] A. B. Waluyo, B. Srinivasan, D. Taniar, Research in mobile database query optimization and processing, Mobile Information Systems 1 (2005) 225–252.

[82] B. Guo, J. Yu, B. Liao, D. Yang, L. Lu, A green framework for dbms based on energy-aware query optimization and energy-efficient query processing, Journal of Network and Computer Applications 84 (2017) 118–130.

[83] A. Baumstark, M. A. Jibril, K.-U. Sattler, Adaptive query compilation in graph databases, Distributed and Parallel Databases 41 (2023) 359–386.

[84] Y. Li, L. Wang, S. Wang, Y. Sun, Z. Peng, A resource-aware deep cost model for big data query processing, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 885–897.

[85] S. Chaudhuri, R. Krishnamurthy, S. Potamianos, K. Shim, Optimizing queries with materialized views, in: Proceedings of the Eleventh International Conference on Data Engineering, IEEE, 1995,

pp. 190–200.
[86] R. Castillo, C. Rothe, U. Leser, RDFmatview: Indexing RDF data using materialized sparql queries, in: International Workshop on SSWS, 2010.
[87] D. Ibragimov, K. Hose, et al., Optimizing aggregate sparql queries using materialized RDF views, in: ISWC 2016, Proceedings, Part I 15, Springer, 2016.
[88] M. Wudage Chekol, J. Euzenat, P. Genevès, N. Layaïda, Evaluating and benchmarking SPARQL query containment solvers, in: International Semantic Web Conference, Springer, 2013, pp. 408–423.
[89] W. Le, S. Duan, A. Kementsietsidis, F. Li, M. Wang, Rewriting queries on sparql views, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 655–664.
[90] A. Grall, H. Skaf-Molli, P. Molli, M. Perrin, Collaborative sparql query processing for decentralized semantic data, in: International Conference on Database and Expert Systems Applications, Springer, 2020, pp. 320–335.
[91] Y. He, M. Li, Y. Liu, Collaborative query processing among heterogeneous sensor networks, in: Proceedings of the 1st ACM international workshop on Heterogeneous sensor and actor networks, 2008, pp. 25–30.
[92] K. Hose, A. Roth, A. Zeitz, K.-U. Sattler, F. Naumann, A research agenda for query processing in large-scale peer data management systems, Information Systems 33 (2008) 597–610.
[93] D. Evans, V. Kolesnikov, M. Rosulek, et al., A pragmatic introduction to secure multi-party computation, Foundations and Trends® in Privacy and Security 2 (2018) 70–246.
[94] Y. Yang, X. Huang, X. Liu, H. Cheng, J. Weng, X. Luo, V. Chang, A comprehensive survey on secure outsourced computation and its applications, IEEE Access 7 (2019) 159426–159465.
[95] M. Sepehri, S. Cimato, E. Damiani, Privacy-preserving query processing by multi-party computation, The Computer Journal 58 (2015) 2195–2212.
[96] R. T. Fielding, Architectural styles and the design of network-based software architectures, University of California, Irvine, 2000.
[97] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), Recommendation, W3C, 2023. URL: https://www.w3.org/TR/shacl/.
[98] O. Hartig, Zero-knowledge query planning for an iterator implementation of link traversal based query execution, in: Extended Semantic Web Conference, Springer, 2011, pp. 154–169.
[99] A. Deshpande, Z. Ives, V. Raman, Adaptive query processing, Foundations and Trends® in Databases (2007).
[100] N. Kabra, D. J. DeWitt, Efficient mid-query re-optimization of sub-optimal query execution plans, in: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, 1998.
[101] K. Munagala, U. Srivastava, J. Widom, Optimization of continuous queries with shared expensive filters, in: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, 2007, pp. 215–224.
[102] R. Avnur, J. M. Hellerstein, Eddies: Continuously adaptive query processing, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 261–272.
[103] M. Acosta, M.-E. Vidal, Y. Sure-Vetter, Diefficiency metrics: Measuring the continuous efficiency of query processing approaches, in: International Semantic Web Conference, Springer, 2017, pp. 3–19.
[104] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, S. Luo, A review of content-based and context-based recommendation systems, International Journal of Emerging Technologies in Learning (iJET) 16 (2021) 274–306.
[105] D. Brickley, R. Guha, RDF Schema 1.1, Recommendation, W3C, 2014. URL: https://www.w3.org/TR/rdf11-schema/.
[106] Y. Gil, S. Miles, PROV Model Primer, Recommendation, W3C, 2013. URL: https://www.w3.org/TR/prov-primer/.
[107] S. Chakrabarti, M. Van den Berg, B. Dom, Focused crawling: a new approach to topic-specific web resource discovery, Computer networks 31 (1999) 1623–1640.

[108] P. Buneman, S. Khanna, T. Wang-Chiew, Why and where: A characterization of data provenance, in: Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8, Springer, 2001, pp. 316–330.

[109] F. Geerts, T. Unger, G. Karvounarakis, I. Fundulaki, V. Christophides, Algebraic structures for capturing the provenance of sparql queries, Journal of the ACM (JACM) 63 (2016) 1–63.

[110] E. Adar, J. Teevan, S. T. Dumais, Large scale analysis of web revisitation patterns, in: Proceedings of the SIGCHI conference on Human Factors in Computing Systems, 2008, pp. 1197–1206.

[111] E. Ferrara, A large-scale community structure analysis in facebook, EPJ Data Science 1 (2012) 1–30.

[112] O. Hartig, How caching improves efficiency and result completeness for querying linked data., in: LDOW, 2011.

[113] D. Xu, C. Wu, P.-C. Yew, On mitigating memory bandwidth contention through bandwidth-aware scheduling, in: Proceedings of the 19th international conference on Parallel architectures and compilation techniques, 2010, pp. 237–248.

[114] J. De Smet, R. Taelman, Traqula: Providing a foundation for the evolving sparql ecosystem through modular query parsing, transformation, and generation, in: Proceedings of the 23rd Extended Semantic Web Conference, 2026. URL: https://traqula-resource.jitsedesmet.be/.

[115] R. Ciucanu, P. Lafourcade, A secure framework for graph outsourcing and sparql evaluation, in: IFIP Annual Conference on Data and Applications Security and Privacy, Springer, 2020, pp. 347–366.

[116] K. M. Endris, Z. Almhithawi, I. Lytra, M.-E. Vidal, S. Auer, Bouncer: Privacy-aware query processing over federations of RDF datasets, in: International Conference on Database and Expert Systems Applications, Springer, 2018, pp. 69–84.

[117] Regulation (EU) 2018/1724 of the European Parliament and of the Council of 2 October 2018 establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services and amending Regulation (EU) No 1024/2012, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02018R1724-20250119, 2018. Consolidated version as of 19 January 2025.

[118] R. Buyle, R. Taelman, K. Mostaert, G. Joris, E. Mannens, R. Verborgh, T. Berners-Lee, Streamlining governmental processes by putting citizens in control of their personal data (2019). URL: https://tinyurl.com/25bbk6vd.