Brokering as a Decision Loop: Trust-Aware Multi-Agent Architectures for Data Marketplaces

Imane Hocine[1], Abdelaziz Kella[2], Asma Abboura[2], Maria Hanini[3], Oscar J. Castro-Lopez[1], Grégoire Danoy[1], and Jedrzej Musial[4]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
{imane.hocine, oscar.castro, gregoire.danoy}@uni.lu
[2] Hassiba Benbouali University, Chlef, Algeria
{a.kella, a.abboura}@univ-chlef.dz
[3] University of Sheffield, Sheffield, UK
m.hanini@sheffield.ac.uk
[4] Institute of Computing Science, Poznan University of Technology, Poznan, Poland
jedrzej.musial@put.poznan.pl

Abstract. Earth-observation data marketplaces increasingly depend on repeated service commitments made under uncertainty about provider behavior and delivery risk. Existing brokering workflows treat each request separately. Evidence from prior transactions is retained locally within bilateral relationships and is not used to inform subsequent decisions. This paper makes the case that brokering should be treated as a data management problem. Allocation, pricing, and trust assessment can be expressed as queries over accumulated marketplace knowledge rather than as ad hoc negotiations. We outline SPECTRAM, a multi-agent brokering architecture in which a logical orchestrator coordinates specialized agents over a shared knowledge graph. Agents do not invoke one another, and all interactions are mediated by the orchestrator's state and the shared graph. Service commitments rely on trust and service performance from prior transactions, while transaction outcomes are recorded and used for subsequent decisions. We emphasize agent roles, read/write separation, and orchestration as key design choices for trust-aware brokering. This short paper outlines a research direction for next-generation earth-observation marketplaces and identifies open challenges for MAS and data marketplaces. The proposed framing applies to service marketplaces where transaction history informs future commitments.

Keywords: Trust-aware brokering · Multi-agent systems (MAS) · Data marketplaces · Earth observation · Knowledge graphs

1 Introduction

The Earth Observation (EO) data and services market has grown to a scale where procurement depends on repeated service commitments made under uncertainty about provider behavior and delivery. Recent estimates place the global EO data market at roughly $5.9 billion in 2024, with projections exceeding $11 billion by

2033 [9,21]. Deciding which provider to assign, at what price, and with what confidence demands shared historical evidence.

Present-day platforms treat each request independently and trigger an isolated evaluation. Feasibility screening, provider ranking, and price negotiation all proceed without reference to prior outcomes. Marketplaces cannot therefore learn from historical provider performance and evidence is fragmented across platforms. Delivery outcomes and reliability assessments remain localized rather than a shared representation of provider behavior. For instance, NOAA's commercial data-buying initiatives and NASA's Commercial Smallsat Data Acquisition program assess vendor reliability within self-contained acquisition cycles, and performance evidence is contained in bilateral relationships [8,19]. As commercial constellations such as Planet acquire imagery at global scale [20], per-request manual assessment does not scale. The gap is architectural. Transaction outcomes exit the decision process when delivery completes, and are not captured as reusable evidence in a shared marketplace. Brokering therefore requires coordinating multiple decision steps including feasibility, ranking, pricing, and commitment, along with the maintenance of contextual knowledge that is updated after each delivery. Existing marketplace platforms offer limited support for closing this loop [2,32,33]. We argue that brokering should be reframed as an orchestrated multi-agent decision loop over shared marketplace state, where transaction outcomes are recorded as evidence and used to update trust that informs subsequent allocation and pricing.

We propose SPECTRAM (Space Provider-Exchange Coordinated through Trust, Reasoning, and Agentic Matching), a hybrid multi-agent architecture blueprint that instantiates this framing. A central orchestrator governs the decision loop, invoking five specialized agents over a shared knowledge graph. Agents do not communicate directly, and the orchestrator mediates data flows and controls state transitions. The pattern applies where service allocation recurs, provider performance generates observable feedback, and decisions decompose into modular stages with clear read/write separation over versioned knowledge. It assumes that transaction outcomes can be captured as structured evidence and that trust can be updated and reused across decisions. Trust models, pricing mechanisms, and learning techniques are replaceable components within the architecture. The pattern does not extend to one-off negotiations or domains lacking delivery feedback.

2 Related Work

Existing EO data platforms emphasize data discovery and access rather than coordinated, request-by-request brokering. Marketplaces such as the Copernicus Data Space Ecosystem [10], AWS Data Exchange [3], and UP42 [29] focus on catalog exploration and predefined pricing, leaving provider screening and suitability assessment to manual processes. Information asymmetry and pricing opacity in digital data-exchange ecosystems are well documented [17,33]. Similar traits appear in NOAA's data-buying initiatives and NASA's CSDA framework [8,19].

Research on trust mechanisms addresses scoring and verification but lacks end-to-end integration [13]. Blockchain-based frameworks improve transparency and tamper-resistance [28]. IoT and smart-contract-based systems add direct and indirect trust metrics, secure negotiation, and third-party validation [16,34]. Grid and cloud brokering use trust propagation and feedback-based scoring for allocation [4,23]. Across these approaches, trust remains a scoring or verification layer, not part of an evolving marketplace state that affects allocation and pricing. Studies on data markets and brokerage report persistent information asymmetry, pricing opacity, and dispersed governance [17,24]. Regulatory analyses point to the difficulty of auditing decisions and tracing accountability when evidence spans organizational boundaries [18]. Architectures that retain transaction outcomes as reusable evidence are largely absent.

In the multi-agent system (MAS) literature, coordination and workflow orchestration are well-studied [1,6,27]. Brokering and evolving trust are treated as separate problems. Coupling coordination and trust within a single process that operates over shared state is rarely tackled. Recent work on inter-agent trust focuses on protocol-level mechanisms such as reputation, proof, and stake [14]. Existing trust-aware brokering, whether resource matchmaking [4] or cloud service brokering [23], centres on scoring or matchmaking.

Our vision departs from existing work by casting brokering as an orchestrated multi-agent system in which allocation, pricing, and trust are explicit decision stages operating over accumulated evidence. SPECTRAM proposes an architectural blueprint that integrates preference-based feasibility filtering, reliability pricing, and evolving trust over a shared knowledge graph, with clear agent roles and a logical orchestrator mediating all interactions.

3 Architectural Challenges

Satellite tasking and data delivery operate under conditions that shift between transactions. Providers differ in coverage, capacity, reliability, and quality-of-service (QoS) commitments. Requests differ in scope, urgency, and budget. An integrated architecture faces five challenges. *(1) Brokering:* normalizing incompatible provider interfaces and pricing for dynamic composition across suppliers. *(2) Trust management:* accumulating past outcomes into explicit trust signals that persist beyond individual transactions [15,22]. *(3) Matching:* determining feasible, stable request–supplier assignments for downstream pricing [11,25]. *(4) Dynamic pricing:* adapting to demand and supply using auction or incentive-compatible mechanisms [32]. *(5) Transaction history:* storing outcomes as queryable records so that the system can learn from past interactions.

Coordinating these challenges under latency and consistency constraints surfaces tensions that shape the architecture. *Latency vs. evidence:* prioritizing execution can bypass trust updates. *Feasibility vs. optimization:* hard constraints (coverage, deadline, capacity) must hold before cost and reliability optimization. *Commitment vs. adaptation:* allocations are committed but trust updates as deliveries complete. The system must track which trust version each allocation

used. *Persistence vs. interference:* the shared graph must be queryable and updated without Feedback-agent writes delaying reads and other agents decisions.

SPECTRAM addresses these tensions through two design choices. Trust is treated as versioned state, updated by the feedback agent after delivery. Agents receive clear, non-overlapping roles such as feasibility screening (Graph Query), trust-based ranking (Preference and Shortlist), and pricing (Auction), without direct agent-to-agent calls.

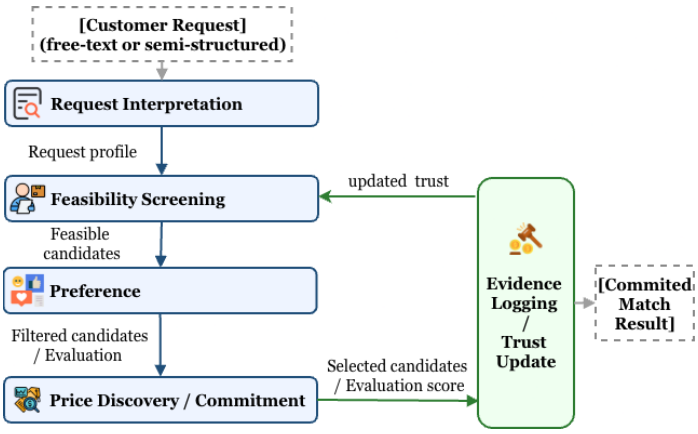

Fig. 1. SPECTRAM multi-agent workflow

4 SPECTRAM Architecture Overview

SPECTRAM realizes brokering as a multi-agent decision loop over a shared knowledge graph (Figure 1). Five agents execute distinct stages of the loop. A single orchestrator maintains workflow state and passes data to the next agent. No agent invokes another directly. The knowledge graph is the only shared persistent store. It is *read* during feasibility screening and preference ranking by the graph query agent and the preference-and-shortlist agent, and *written* exclusively by the feedback agent after delivery. This separation ensures that trust updates from completed transactions are visible to the next request's agents. The decomposition into five agents reflects the structure of the brokering problem itself. Parsing, feasibility screening, preference ranking, price discovery, and feedback recording address different concerns. A request first enters through the *parser agent*, which translates it into a structured *ServiceRequest*. This is passed to the *graph query agent*, which applies hard constraints (coverage, timing, capacity, trust) and retrieves the set of feasible candidate providers from the knowledge graph. From this set, the *preference and shortlist agent* weighs customer preferences against provider constraints and trust thresholds, producing a filtered candidate set along with matching scores and explicitly excluded

providers. The *auction and decision agent* then selects a winner and clearing price using only the information provided by the orchestrator. Post delivery, a *feedback agent* records delivery outcomes and updates provider trust in the knowledge graph, closing the loop.

The choice of centralized orchestration over peer-to-peer agent coordination is deliberate. In a marketplace where allocation decisions carry financial commitments and must be auditable, deterministic sequencing and a single point of state management reduce the risk of inconsistent decisions. Agents in SPECTRAM are task-specialized modules invoked by the orchestrator whose state can be checkpointed and externalized to a replicated store after each agent invocation. A standby coordinator can resume after failure without divergent replays and the orchestrator can be sharded by request key under load. A fully decentralized variant using a shared event log would trade audit transparency for autonomy. SPECTRAM is still a multi-agent architecture as decisions are decomposed into specialized agents with distinct semantics and clear interfaces, using symbolic orchestration with data-driven trust and optional LLM-backed components. Whether and where to introduce greater agent autonomy is an open design question.

5 Core Mechanisms

The architectural roles introduced above are instantiated through concrete mechanisms, chosen to ground the discussion and reflect our motivating use cases. They are not canonical choices, and alternative designs are possible.

i Orchestrator. The orchestrator governs agent execution and maintains workflow state. Because decisions are cyclical and contingent, graph-based control is suitable. A possible instantiation is LangGraph [26]. Its role is to invoke each agent in turn, support conditional routing and loopbacks , and pass data between steps via its state.

ii Parser agent. The parser converts free-text or semi-structured user requests into a structured *ServiceRequest* that captures spatial, temporal, quality, latency, and budget constraints. Implementations may rely on schema-driven extraction, LLM-assisted parsing, or hybrid techniques. The resulting normalized request profile is passed to the graph query agent. Parsed outputs are validated against a domain schema before entering the decision pipeline with rule-based fallback to handle critical fields when confidence is low. Fields not present in the schema are dropped, containing LLM hallucinations at the entry point.

iii Graph query agent. Feasibility screening is performed by querying a shared knowledge layer. Neo4j [12] is a viable candidate for a graph database. It can represent providers, offerings, requests, transactions, and feedback as structured entities. The graph is read during candidate retrieval and preference construction and written only after service delivery.

iv Preference and shortlist agent. This agent narrows the feasible candidate set by reconciling requester and provider preferences under feasibility and trust constraints. It employs a constrained matching-and-ranking procedure

inspired by Gale–Shapley deferred acceptance [11,25]. Both sides expose preferences through customer and provider utility functions.

Customer utility U_p over provider p combines QoS, latency, price, and trust T_p from the graph:

$$U_p = w_1 \cdot \mathrm{QoS}_p + w_2 \cdot \left(1 - \frac{\mathrm{latency}_p}{\mathrm{maxLat}}\right) + w_3 \cdot \left(1 - \frac{\mathrm{price}_p}{\mathrm{budget}}\right) + w_4 \cdot T_p, \tag{1}$$

where QoS_p captures expected service quality, $\mathrm{latency}_p$ is normalized by the maximum acceptable latency, price_p is evaluated relative to budget, and T_p is accumulated trust from prior transactions. The weights w_i encode request-specific priorities. Trust values are subject to context-dependent thresholds that exclude providers whose reliability falls below acceptable levels.

Provider utility V_c over request c:

$$V_c = v_1 \cdot R_c + v_2 \cdot (1 - \mathrm{SLA_risk}_c) + v_3 \cdot \frac{\mathrm{profit}_c}{\mathrm{maxProfit}} + v_4 \cdot \mathrm{capacity}_a, \tag{2}$$

where R_c is requester reputation, $\mathrm{SLA_risk}_c$ captures anticipated compliance risk, and $\mathrm{capacity}_a$ is the provider's remaining operational capacity.

Modified deferred acceptance. This procedure [11,25] reconciles preferences under capacity and risk constraints. It outputs a feasible candidate set, excluded providers (provider → reason), and matching scores.

v Auction and decision agent. Final selection among shortlisted providers is handled by an auction-based decision agent. A possible design employs a *Vickrey-inspired* second-price mechanism [30] combined with QoS-aware scoring [7,32]. The score used to pick the winner is:

$$\mathrm{Score}_p = \mathrm{Benefit}(p) - \alpha \cdot (\mathrm{BaseBid}_p + \mathrm{Penalty}_{\mathrm{QoS}}(p)), \tag{3}$$

where $\mathrm{BaseBid}_p$ denotes the provider's submitted bid and $\mathrm{Penalty}_{\mathrm{QoS}}(p)$ captures adjustments reflecting expected service quality. The parameter $\alpha \in [0, 1]$ regulates sensitivity to cost relative to benefit. Auctions run only over the shortlist. The mechanism does not necessarily satisfy classical Vickrey properties and is treated as a practical scoring rule.

vi Feedback agent. After service delivery, the feedback agent records outcomes and updates provider trust in the knowledge layer. A simple weighted rule [15,22] is:

$$T_p^{\mathrm{new}} = \beta\, T_p^{\mathrm{prev}} + (1 - \beta)\, F_p, \tag{4}$$

where T_p^{prev} is the previous trust score of provider p, $F_p \in [0, 1]$ is the latest feedback score, and β is a smoothing factor. Completed transactions produce graph evidence (e.g., `SET p.trust`, `MATCHED_WITH`, `RECEIVED_FEEDBACK`). More expressive models such as GNN-based propagation [31] or learned preference models [5] can be swapped in without altering the decision flow.

vii Coordination and concurrency. The graph is read by multiple agents (graph query, preference and shortlist) and written only by the feedback agent, coordination therefore matters when requests or trust updates overlap. Key design choices include *synchronization* (immediate, micro-batch, or asynchronous trust writes), *read semantics* for graph consumers (e.g. snapshot isolation), and *admission control* so heavy trust updates do not delay the orchestrator's decision steps. This exposes a trade-off between evidence freshness and throughput [33]. Each brokering round reads a consistent trust snapshot identified by a version timestamp while feedback writes create a new trust version without overwriting in-flight decisions.

6 Illustrative Scenario

This section walks through a compact end-to-end scenario to illustrate how the agents interact in SPECTRAM. The example is not a benchmark and values are synthetic. LangGraph and Neo4j are illustrative, not required for the architecture. Consider 3 customers, ESA, NASA, and JAXA, requesting high-resolution daily imagery over the Amazon, Sahara, and Arctic regions.

i Parser and graph query agents: A free-text request from ESA can be parsed into the following *ServiceRequest*:

```
{
  "region": "Amazon",
  "frequency": "daily",
  "resolution": "high",
  "latency": "24h",
  "duration": "30 days",
  "budget": 150
}
```

The graph query agent retrieves feasible providers by enforcing the specified constraints over the Neo4j knowledge graph:

```
MATCH (p:Provider)-[:OFFERS]->(prod:Product)
WHERE prod.resolution = 'high' AND prod.region = 'Amazon'
  AND prod.frequency = 'daily' AND prod.latency <= 24
RETURN p, prod;
```

This query yields five candidate providers (P1–P5) with partially overlapping coverage and differing QoS, reliability, and pricing.

ii Preference and shortlist agent: Candidates are evaluated using utility functions. Providers violating binding constraints (e.g., trust or latency thresholds) are removed before preference reconciliation. Table 1 summarizes utilities and outcomes. The constrained deferred-acceptance procedure produces a stable shortlist $\{\mathrm{P1}, \mathrm{P2}, \mathrm{P5}\}$. Providers appearing in multiple shortlists are treated as capacity-bound i.e. when capacity conflicts arise, the orchestrator may reroute to the next-ranked provider. **iii**

Table 1. Provider candidates and matching results (synthetic).

Prov.	Price	Lat.(h)	QoS	Trust	Utility	Status
P1	100	18	0.95	0.90	0.88	Shortlist
P2	80	24	0.90	0.85	0.85	**Selected**
P3	60	48	0.70	0.75	0.65	Excl. (lat.)
P4	110	22	0.80	0.78	0.72	Excl. (trust)
P5	95	20	0.85	0.83	0.80	Shortlist

Auction and decision agent: Price discovery proceeds within the filtered shortlist using a second-price Vickrey-inspired auction. Selection uses:

$$\mathrm{Score}_p = (\mathrm{QoS}_p \times 100) - \alpha \cdot \big(\mathrm{BaseBid}_p + \mathrm{Penalty}_{\mathrm{QoS}}(p)\big), \tag{5}$$

with latency-based penalty: 0 if $\mathrm{latency}_p \leq 24\mathrm{h}$, +5 if $24 < \mathrm{latency}_p \leq 36\mathrm{h}$, +15 otherwise. In this scenario, provider P2 achieves the highest score and is selected, paying the second-highest effective bid.

iv Feedback agent: The feedback agent records the outcome and updates P2's trust in Neo4j: `MATCH (p:Provider {id:"P2"}) SET p.trust = 0.865;`

P2 had prior trust $T^{\text{prev}} = 0.85$ (Table 1). With feedback $F = 0.9$ and $\beta = 0.7$, the trust update yields $T^{\text{new}} = 0.7 \times 0.85 + 0.3 \times 0.9 = 0.865$, which the Feedback agent writes to the graph. On the next request, the Graph query and Preference agents *read* this trust, so one transaction informs the next matching and auction.

7 Vision, Challenges, and Research Opportunities

SPECTRAM is a vision and early-stage blueprint for repeated service allocation over evolving trust, within and beyond EO marketplaces.

Agent autonomy. SPECTRAM's agents are orchestrator-invoked modules. They receive inputs, compute and return results without initiating actions or maintaining persistent goals. This favors auditability and deterministic control. Alternatives, such as feedback-triggered trust re-evaluation without the orchestrator, could improve responsiveness but weaken centralized control. How much autonomy each role should have and how trust in providers relates to trust in agent behavior remain open questions.

Trust and preference modeling. SPECTRAM treats trust as a shared state in a knowledge graph. Several questions follow. How should trust be represented in shared agent environments? Should trust decay be local, or propagate across the graph? Graph-based propagation could capture transitive reliability but introduces convergence and manipulation concerns. How should trust versioning interact with concurrent access? Multiple brokering loops executing simultaneously may read different trust versions moments apart. Richer interpretable models for context-dependent reliability and user priorities are needed. Pricing and allocation face similar tensions as they create feedback loops with trust and capacity whose joint dynamics are not yet understood.

Hybrid architectures. SPECTRAM combines a knowledge layer, an orchestration layer, and components where LLM-driven reasoning could enter. The parser agent is a natural LLM-integration point, but a malformed or hallucinated constraint at this stage propagates through the entire decision loop. Verifying LLM-generated intermediate representations within a multi-agent pipeline is an unsolved engineering problem. This extends to the decision rationale. Operators in regulated markets need counterfactual explanations of allocation decisions. Embedding specifications and governance rules in brokering workflows connects SPECTRAM to foundational MAS research.

Evaluation and generalization. No shared benchmarks exist for trust-aware multi-agent brokering. Evaluation could operate along three levels: component (parser accuracy, matching stability, trust-update correctness), system (allocation quality and SLA adherence over repeated rounds), architectural (end-to-end latency, recovery time after failure and the cost of splitting the work across agents). A practical first step is a simulation with synthetic providers and repeated customer requests evolving over multiple rounds to compare stateless brokering and trust-aware allocation in terms of QoS, cost, and delivery reliability. The same architectural questions arise in settings such as cloud service brokerage and API marketplaces where prior delivery outcomes affect future commitments, suggesting that SPECTRAM's framing extends beyond EO markets.

Acknowledgments

This work was funded by the Luxembourg National Research Fund (FNR), INTER programme, UltraBO Project (ref. INTER/ANR/22/17133848), and the Polish National Centre for Research and Development (NCBR) (ref. POLLUX-XI/15/Serenity/2023).

References

1. Adimulam, A., Gupta, R., Kumar, S.: The orchestration of multi-agent systems: Architectures, protocols, and enterprise adoption. arXiv preprint arXiv:2601.13671 (2026), https://arxiv.org/abs/2601.13671
2. Akerlof, G.A.: The market for "lemons": Quality uncertainty and the market mechanism. The Quarterly Journal of Economics **84**(3), 488–500 (1970)
3. Amazon Web Services: Aws data exchange (2024), https://aws.amazon.com/data-exchange/
4. Azzedin, F., Maheswaran, M., Mitra, A.: Trust brokering and its use for resource matchmaking in public-resource grids. Journal of Grid Computing **4**(3), 247–263 (2006)
5. Bajari, P., Chernozhukov, V., Hortaçsu, A., Suzuki, J., Xu, Y.: A choice modeling approach to recommender systems. Marketing Science **41**(5), 853–877 (2022)
6. Bandara, E., Gore, R., Foytik, P., Shetty, S., Mukkamala, R., Rahman, A., Liang, X., Bouk, S.H., Hass, A., Rajapakse, S., Ng Wee Keong, De Zoysa, Kasun, Withanage, A., Loganathan, N.: A practical guide for designing, developing, and deploying production-grade agentic ai workflows. arXiv preprint arXiv:2512.08769 (2025), https://arxiv.org/abs/2512.08769
7. Chakraborty, A., Mondal, A., Roy, A., Misra, S.: Dynamic trust enforcing pricing scheme for sensors-as-a-service in sensor-cloud infrastructure. IEEE Transactions on Services Computing **14**(5), 1345–1356 (2018)
8. U.S. Department of Commerce, N.O., Administration, A.: Guidance for noaa commercial data buys: A framework for noaa programs and offices conducting commercial data buys. Tech. rep., NOAA Observing Systems Council (NOSC) (12 2024), https://www.nesdis.noaa.gov/s3/2025-08/NOAA-Commercial-Data-Buys-Guidance-2024.pdf
9. DataHorizon Research: Earth observation big data service market (2024), https://datahorizzonresearch.com/earth-observation-big-data-service-market-44729, accessed: 2026-01-31
10. European Space Agency: Copernicus data space ecosystem (2024), https://dataspace.copernicus.eu/
11. Gale, D., Shapley, L.S.: College admissions and the stability of marriage. The American Mathematical Monthly **69**(1), 9–15 (1962)
12. Guia, J., Soares, V., Bernardino, J.: Neo4j: A graph database for knowledge management. Journal of Information Systems Engineering & Management **2**(3), 15 (2017)
13. Hou, F., Jansen, S.: A survey of the state-of-the-art approaches for evaluating trust in software ecosystems. Journal of Software: Evolution and Process (2024). https://doi.org/10.1002/smr.2695
14. Hu, B.A., Rong, H.: Inter-agent trust models: A comparative study of brief, claim, proof, stake, reputation and constraint in agentic web protocol design—a2a, ap2, erc-8004, and beyond. arXiv preprint arXiv:2511.03434 (2025), https://arxiv.org/abs/2511.03434
15. Jøsang, A., Ismail, R., Boyd, C.: Trust and reputation systems. In: Foundations of Security Analysis and Design IV, pp. 209–245. Springer (2007)
16. Latif, R., Yakubu, B.M., Saba, T.: Markettrust: blockchain-based trust evaluation model for siot-based smart marketplaces. Scientific Reports **13**(1), 11571 (2023)
17. Majumdar, R., Gurtoo, A., Maileckal, M.: Developing a data pricing framework for data exchange. Future Business Journal **11**(4) (2025)

18. May, A.: How regulators are responding to new challenges in data brokerage transparency. PersonZoom (2025), https://www.personzoom.com
19. NASA Earth Science Division: The commercial satellite data acquisition (csda) program — program overview (2025), https://science.nasa.gov/earth-science/csda/
20. NASA Earthdata: Csda | nasa earthdata — vendors and program overview (2025), https://www.earthdata.nasa.gov/about/csda
21. NovaSpace: Earth observation data and services market 2022 free extract (Dec 2022), https://nova.space/hub/wp-content/uploads/2022/12/EODSM_2022-free-extract.pdf, accessed: 2026-01-05
22. Ramadan, Q., Boukhers, Z., AlShaikh, M., Lange, C., Jürjens, J.: Data trading and monetization: Challenges and open research directions. arXiv preprint arXiv:2401.09199 (2024)
23. Ramya, P., Saraswathy, S., Sharmila, S., Sivakumar, S.: A trust-aware service brokering scheme for multiple clouds collaborative services. International Journal of Scientific Research in Science, Technology and Engineering **2**(2) (2016)
24. Reviglio, U.: The untamed and discreet role of data brokers in surveillance capitalism: a transnational and interdisciplinary overview. Internet Policy Review **11**(3), 1–27 (2022)
25. Roth, A.E.: What have we learned from market design? The Economists' Voice **5**(1) (2008)
26. Sapkota, R., Shrestha, R., Rijal, M., Karkee, M.: Langchain vs. langgraph vs. langsmith: Taxonomies of agentic ai toolchains for end-to-end orchestration. Authorea Preprints (2025)
27. Sun, L., Yang, Y., Duan, Q., Shi, Y., Lyu, C., Chang, Y.C., Lin, C.T., Shen, Y.: Multi-agent coordination across diverse applications: A survey. arXiv preprint arXiv:2502.14743 (2025), https://arxiv.org/abs/2502.14743
28. Tan, T.M., Saraniemi, S.: Trust in blockchain-enabled exchanges: Future directions in blockchain marketing. Journal of the Academy of marketing Science **51**(4), 914–939 (2023)
29. UP42: Up42: Geospatial developer platform (2024), https://up42.com/
30. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance **16**(1), 8–37 (1961)
31. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems **32**(1), 4–24 (2021)
32. Xu, H., Liu, W., Liu, X.: Blockchain-based trust auction for dynamic virtual machine provisioning and allocation in clouds. Wireless Communications and Mobile Computing **2021**(1), 6639107 (2021)
33. Zhang, G., Gupta, S., Vasilakopoulou, P., Pappas, I.O.: Pricing data in data marketplaces: A systematic literature review (2025)
34. Zhou, C., Yang, Z., Yue, S., Xuan, B., Wang, X.: Securing big data exchange: an integrated blockchain framework for full-lifecycle data trading with trust and dispute resolution. Electronics **12**(24), 4894 (2023)