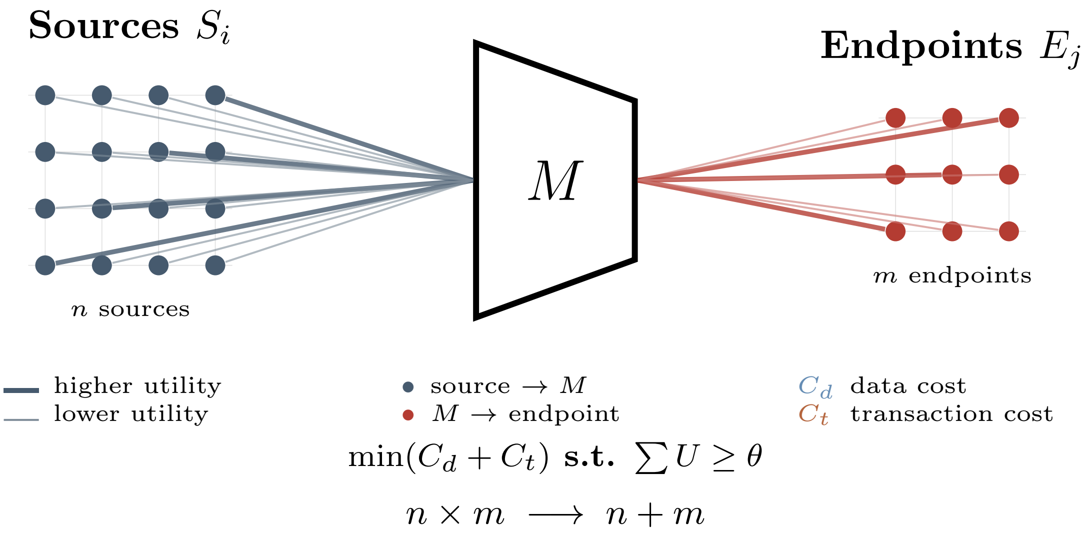
Brickroad

The Data Multiplexer for the Agent Economy

A Thesis on Efficient Data Routing Infrastructure

Brickroad | brickroad.network



The data multiplexer M acts as a universal adapter between sources S_i and endpoints E_j, routing flows that maximize utility U while minimizing total cost $C = C_d + C_t$. Edge thickness indicates utility; color indicates direction (blue: source→M, red: M→endpoint).

Executive Summary. Machine learning systems are fundamentally the composition of data and transformations. As these systems migrate from research to core economic infrastructure, shifts across scale, market structure, and consumer endpoints create both friction and opportunity. Today, friction materializes as transaction costs; such as search, integration or contracting; forcing $n \times m$ bilateral negotiations between sources and endpoints. However, advances in utility prediction, metadata standards, and LLM-powered interfaces make a different architecture feasible. We introduce the **Brickroad multiplexer** to harness the opportunities and reduce friction in data flow: a universal adapter where sources and endpoints each connect once, reducing integrations to $n + m$. By decomposing cost into *data cost* C_d (the value of bits) and *transaction cost* C_t, the multiplexer minimizes both, optimizing $\min(C_d + C_t)$ subject to utility thresholds. The result: data procurement shifts from months to milliseconds, the long tail of specialized data becomes accessible, and price reflects utility rather than negotiating power.

1 Structural Shifts in Data Systems

Machine learning ("ML") is fundamentally the composition of *data* and *transformations*. A dataset enters a transform and emerges as a new representation: a weight, a prediction, a model. The cycle perpetuates ad-infinitum:

$$D \xrightarrow{f_1} D' \xrightarrow{f_2} D'' \xrightarrow{f_3} \cdots$$

ML pipelines, from raw data through preprocessing, training, fine-tuning, and inference, are compositions of such mappings. However, the infrastructure that has been optimized to support these pipelines on research workloads and closed production stacks for over a decade is coming under strains in a world of networked ML systems proliferated by agent interfaces that come with scale, high connection density and production-grade latency requirements.

1.1 Infrastructure Shifts

For decades, data composition happened at research community scale: curated datasets, careful experiments, static and concentrated benchmarks. As ML systems propagate into the real-world economic value chain at a new scale as agents, increasingly operating in the long tail of real-world value creation, infrastructure requirements are shifting.

Scale creates friction and opportunities. Data streams now cross organizational boundaries, mixing heterogeneous sources that were never anticipated to interoperate. A single endpoint may need to compose text corpora, image repositories, and proprietary APIs into a unified data product. The space of available sources has grown from dozens to millions, each with distinct schemas, access patterns, and provenance. This heterogeneity is a coordination problem, but it also means that, for those who can solve the integration challenge, compositions previously impossible (cross-domain, cross-modal, cross-organization) are now within reach.

Endpoint diversity drives new markets. Consumers have multiplied from researchers running experiments to agents, engineers, pipelines, and applications that query data programmatically for mission- and business critical applications. Each endpoint cell in this lattice has different requirements: freshness constraints, license restrictions, budget limits, task-specific utility needs. This diversity fragments the market, but it also creates demand for specialized data products that could not find buyers in the research era.

Composability demands standardization. Static, monolithic datasets give way to modular, dynamic data products. Sources must interoperate without bespoke integration. The challenge is no longer acquiring a single dataset but composing flows from many sources to many endpoints, doing so continuously as both lattices evolve. This pressure is accelerating the adoption of metadata standards that make composition tractable.

API-first access enables dynamic routing. Credentials, rate limits, and programmatic access replace bulk downloads. Data is consumed as a flow, not owned as an artifact. This creates opportunity for infrastructure that routes flows dynamically, matching sources to endpoints in real-time based on current needs rather than static contracts.

1.2 Economic Shifts

Data is becoming an economic asset, and with that comes the machinery of markets [5].

The role of transaction costs. The friction in acquiring data is less about the payload itself than about the external *transaction* variables. Particularly in an opaque, closed market they span legal review, licensing and cost negotiation, integration engineering, quality evaluation. This overhead can make small transactions uneconomical, concentrating the market among large players who can amortize overhead [2, 10]. Without a multiplexer, each cell in the source lattice in Figure 1 must wire to each cell in the endpoint lattice: $n \times m$ transactions, each carrying full friction. But this is a structural problem, not a fundamental one: infrastructure that standardizes these transactions will collapse the overhead.

The viability frontier. Transaction costs create an economic boundary, a *viability frontier*, that determines which data sources are worth acquiring. When C_t is high, only large, known, and well-marketed or established datasets cross this threshold: the expected utility must exceed the transaction overhead before a deal makes sense. This concentrates the market around the “head” of the distribution: a handful of canonical datasets that are widely used, the old “benchmark to research” paradigm. The “long tail” of specialized, niche, and emerging sources needed for agentic value creation in-context remains locked behind transaction barriers, even when their utility for specific tasks would be substantial. Lowering C_t moves this frontier outward: sources that were previously uneconomical become viable. A rare disease imaging dataset with only 5,000 samples, a specialized sensor corpus from a single research lab, a curated collection of domain-specific annotations—all become accessible when the fixed cost of acquisition drops from tens of thousands of dollars to near-zero.

Estimating utility. Advances in task-data matching mean we can now predict the marginal utility $U(S, E, \text{task})$ of a dataset for a specific task *before* acquisition [6, 7]. It shifts procurement from marketing and broker-driven to utility-driven: endpoints can evaluate sources by measured contribution rather than vendor claims. And intermediary-free preview-before-purchase becomes possible, changing the economics of data acquisition.

1.3 Agentic Shifts

The profile of data consumers is changing from humans to machines.

Agents query dynamically. Autonomous systems compose data on-the-fly based on task requirements, selecting sources in real-time rather than curating corpora in advance. An agent cell needs to reach any source cell instantly, without pre-negotiated integrations. LLMs now enable natural-language interfaces to data procurement, where an agent can describe what it needs in plain text, and the system can translate that to structured queries across heterogeneous catalogs.

Pipelines require self-healing. Production-ready data feeds must adapt when upstream sources change: new schemas, deprecated endpoints, shifting quality and evaluation requirements. Modern ML tooling makes it possible to automatically detect schema drift, quality degradation, and distribution shift, enabling pipelines that adapt rather than fail. The infrastructure must support *liquid APIs*, interfaces that adapt as schemas evolve, endpoints change, and requirements shift. Unlike static integrations that break when upstream sources change format, liquid APIs absorb change: schema migrations are handled by the adapter layer, deprecated fields are mapped to replacements, and new capabilities are exposed without client-side updates. LLMs can now perform schema translation dynamically, making this adaptability practical where it was previously intractable.

Applications demand verification. Real-time data routing requires quality guarantees, provenance tracking, and continuous evaluation. When data feeds production models, errors propagate to user-facing decisions. The flow from source to endpoint must be auditable and trustworthy, with clear lineage from origin through every transformation.

These shifts create a friction-filled landscape where transaction costs block access to niche data sources, static infrastructure cannot serve dynamic endpoints, and heterogeneous sources resist composition. Each source cell wiring to each endpoint cell creates $n \times m$ integration points, each carrying legal, technical, and economic friction. But the same shifts that create this friction also provide the tools to build an adapter that collapses this complexity.

2 The Data Multiplexer

The **Brickroad multiplexer** infrastructure routes the right data to the right endpoint at minimal cost. Like a network multiplexer that routes signals from many inputs to optimal outputs, the Brickroad multiplexer consolidates data demand, discovery, evaluation, and purchasing into a single

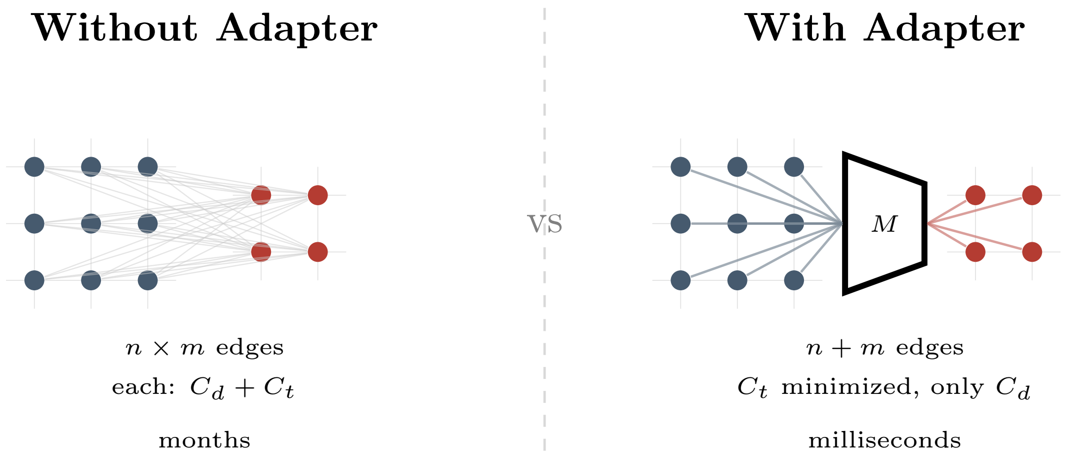


Figure 1: Without the adapter (left), each source cell wires to each endpoint cell: $n \times m$ connections, each carrying data cost C_d plus transaction cost C_t. With the adapter (right), cells connect once to the multiplexer: $n + m$ integrations, transaction cost minimized, data cost flows based on actual consumption.

interface. We formalize this as the **Data Multiplexer Protocol** (DMP), a standardized interface for source registration, endpoint queries, and transaction execution.

2.1 The Adapter

Consider a market with n sources and m endpoints: the source lattice $\{S_1, S_2, \ldots, S_n\}$ and the endpoint lattice $\{E_1, E_2, \ldots, E_m\}$, where i indexes sources and j indexes endpoints throughout. Without an adapter, each endpoint must configure its own connections to each source: $n \times m$ integrations, each carrying cost. The total friction scales multiplicatively.

The multiplexer M acts as a universal adapter between these lattices. Each source connects once to M; each endpoint connects once to M. The integration count collapses from $n \times m$ to $n + m$ for network endpoints. One adapter serves all participating cells. Each source-endpoint flow carries two distinct costs:

$$C_{d,i} = \text{data cost: the intrinsic value of the bits from source } S_i \tag{1}$$

$$C_{t,ij} = \text{transaction cost: overhead of connecting source } S_i \text{ to endpoint } E_j \tag{2}$$

Total cost is $C = C_d + C_t$. Without the multiplexer, both costs scale with each bilateral transaction. With the multiplexer, C_t is absorbed by the adapter layer and amortized across all transactions, while C_d remains tied to actual data value. Transaction costs never reach zero; there is always some overhead in routing, authentication, and coordination, but the multiplexer minimizes them by replacing $n \times m$ bilateral negotiations with standardized protocols. The optimization target shifts from minimizing negotiations to minimizing data spend for a given utility.

Each flow also has associated utility $U(S_i, E_j, \text{task})$, the value of source S_i for endpoint E_j on a given task. We write U_{ij} as shorthand when the task is clear from context. The multiplexer optimizes:

$$M : (\text{Query}, \text{Context}) \to \{(S_i, E_j, U_{ij}, C_{d,i}, C_{t,ij})\} \tag{3}$$

selecting source-endpoint pairs that minimize total cost subject to a utility threshold θ:

$$\min \sum \left(C_{d,i} + C_{t,ij}\right) \quad \text{subject to} \quad \sum U_{ij} \geq \theta \tag{4}$$

The multiplexer provides a *universal interface*: a standardized way for any source cell to expose data and any endpoint cell to consume it. Heterogeneous schemas, access patterns, and licensing terms are absorbed by the adapter layer, presenting a uniform surface to both sides of the lattice.

Without the multiplexer, each endpoint negotiates separately with each source. Both C_d and C_t scale multiplicatively. A typical enterprise data procurement cycle takes 3-6 months from initial discovery to production integration, with legal review alone consuming 4-8 weeks [5]. Utility is unknown until after acquisition. Niche and specialized sources remain inaccessible behind friction barriers. With the multiplexer, sources and endpoints connect once to a shared adapter. Transaction cost C_t is minimized through standardized contracts, pre-negotiated licenses, and automated integration. Data cost C_d scales only with actual data consumed. Procurement drops to API-call latency (milliseconds for cached sources, seconds for new acquisitions). Utility is estimated before acquisition through sample-based evaluation. The long tail becomes accessible because per-transaction friction drops by orders of magnitude.

2.2 Capabilities

The multiplexer is built on three technical capabilities that enable it to serve as this universal adapter - each addressing one of the structural shifts identified earlier: *compose* for scale, *estimate* for economy, and *automate* for agents.

2.2.1 Compose

Data sources are heterogeneous such as text, images, tables, audio, code or time series with varying schemas, formats, and metadata conventions. The multiplexer enables composition through standardized metadata that describes schemas, semantics, and provenance. Sources annotated with this metadata become discoverable across repositories and combinable without per-source integration. Provenance tracks lineage as data flows through transforms, enabling audit and attribution. Standards like Croissant (a metadata format for ML datasets) [1] provide a foundation; Brickroad extends this with tooling for schema mapping, quality signals, and dynamic discovery.

Listing 1: Compose: combining heterogeneous sources

```
# Discover sources matching domain and format requirements
sources = M.discover({
    "domain": "medical_imaging",
    "formats": ["dicom", "nifti", "png"]
})
# Returns: sources with standardized schema mappings

# Compose multiple sources into unified dataset
composed = M.compose([
    sources["chest_xray_nih"],      # 112k images, DICOM
    sources["covid_ct_scans"],       # 20k images, NIfTI
    sources["radiology_reports"]     # 500k reports, JSON
])
# Returns: unified dataset with cross-modal alignment
# Schema conflicts resolved via Croissant metadata mappings
```

2.2.2 Estimate

Rather than relying on marketing claims, the multiplexer employs task-data matching to predict utility before acquisition [6]. Sandbox evaluation protocols assess how a source would contribute to a specific task. Scaling laws enable forecasting gains from subsamples [7, 4]. The result is utilityranked recommendations: sources sorted by measured contribution to the endpoint's task, enabling budget-optimal selection and preview before commitment.

Listing 2: Estimate: predict utility before acquisition

```
# Search for candidate sources
candidates = M.search({"task": "chest_pathology_detection"})

# Estimate utility on sample before full acquisition
estimates = M.estimate(
    candidates=candidates,
    task_spec={"model": "resnet50", "metric": "auroc"},
    sample_budget=1000  # evaluate on 1k-sample preview
)
# Returns: {source_id: predicted_auroc, confidence_interval}
# "chest_xray_nih": 0.87 +/- 0.02
# "covid_ct_scans": 0.72 +/- 0.03
# Enables: select top-k sources by predicted utility
```

2.2.3 Automate

Agentic workflows integrate discovery, preview, evaluation, and checkout into a unified experience. Protocol integration brings data procurement directly into developer environments, from natural-language request to licensed data in seconds:

Listing 3: Automate: end-to-end procurement

```
# Full procurement workflow in one call
query = {
    "task": "sentiment_classification",
    "modality": "text",
    "budget": 1000,  # max data_cost in USD
    "license": "commercial",
    "min_samples": 50000
}
result = M.route(query)
# Returns: optimal source bundle with utility estimates
# {
#   sources: ["amazon_reviews", "yelp_dataset"],
#   total_utility: 0.89,
#   data_cost: 847,
#   tx_cost: 12  # minimized but nonzero
# }
```

3 The Future We Are Building For

The Brickroad multiplexer enables an economy where transaction costs are harmonized through standardized interfaces, transparent pricing, and pre-baked licensing that reduce the fixed costs currently blocking participation. Data flows at minimal friction from heterogeneous sources to diverse endpoints, routed by utility rather than by who can afford the lawyers. Price reflects utility through dynamic, task-dependent valuation that replaces opaque, one-size-fits-all deals. Feedback loops become sustainable as compensation flows back to data generators proportional to the utility their data creates [3, 9], ensuring the ecosystem regenerates rather than depletes.

3.1 Towards Learning Systems That Price Value Creation

Traditional learning theory largely treats data as given: fixed datasets with known properties. The central questions concern generalization, sample complexity, and convergence. Active learning and data selection have studied *which* samples to acquire. When data is acquired from a *market* with multiple sources and varying prices, a new learning signal emerges rooted in the fundamentals of economic value creation.

The Brickroad multiplexer enables a learning theory that prices in economic value creation. The marginal utility of a data source can be estimated before acquisition through sample-based evaluation and scaling law extrapolation [6, 7]. This transforms learning curves into economic curves: the diminishing returns of additional data, well-studied in statistical learning, now carry dollar values. Each point on the curve represents not just accuracy but cost.

This enables *optimal stopping*: acquire data until marginal cost exceeds marginal utility. Define the marginal value of source S_i as $\Delta U_i = U(\{S_1, \ldots, S_i\}) - U(\{S_1, \ldots, S_{i-1}\})$, the incremental utility from adding S_i to the existing data. The fair market value of S_i is bounded by this marginal utility, adjusted for scarcity:

$$\mathrm{FMV}(S_i) \approx \Delta U_i \cdot \phi(\mathrm{scarcity}_i) \tag{5}$$

where ϕ captures the premium for unique or hard-to-replicate sources. In a functioning market, price should track this value. The multiplexer provides the infrastructure to compute these signals at scale, enabling rational data acquisition where budgets are allocated to maximize utility per dollar spent. Fair market value aggregates information about downstream task performance across the entire market. This is a distributed evaluation that most single buyers could not afford. When data is priced by utility, price itself becomes a signal: high prices reveal scarce, high-impact sources; price movements track shifts in what the market needs.

3.2 Expanding the Space of Feasible Tasks

High transaction costs make entire *tasks* infeasible. Consider a task requiring composition of 10 niche datasets, each with $50k in transaction overhead. The total transaction cost is $500k before any data is acquired. This prices out academic research on specialized topics, startups exploring new domains, and agents that need novel data combinations.

Let $\mathcal{F}(B)$ denote the set of feasible tasks given budget B:

$$\mathcal{F}(B) = \{\tau : \sum_{i \in \tau} C_{t,i} < B\} \tag{6}$$

where $C_{t,i}$ is the transaction cost for source i. As transaction costs approach their minimum, this set expands dramatically.

3.3 Value Creation at the Information Frontier

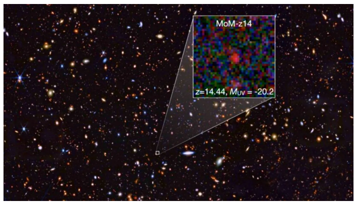


Figure 2: MoM-z14, the farthest known galaxy as of January 2026 [8].

In May 2025, JWST captured MoM-z14 (Figure 2): light emitted 280 million years after the Big Bang and priced, in practice, by the engineering and coordination required to observe it. That is the first economic lesson of the frontier: *information is not discovered for free.* It is produced under a series of constraints–capital, time, bandwidth, and institutional friction–until a boundary is reached where the next bit costs meaningfully more than the last and it becomes unfeasible to produce.

In information economies, the same principle holds. Widely available data is a commodity: replicable, substitutable, and therefore priced near its marginal cost of reproduction. Its marginal contribution to economic performance tends toward zero as the market saturates; competition drives rents out of the "center." What remains defensible – what sustains surplus – is information that is (i) *decision-relevant* for a particular task and/or (ii) *scarce* in the economic sense: costly to obtain, costly to verify, costly to integrate, or costly to replicate.

The Brickroad multiplexer is aimed at tackling this boundary condition. Its job is not to manufacture scarcity; it is to reduce the deadweight loss that prevents scarce, high-signal data from clearing to the endpoints that value it most. In microstructure terms, the Brickroad multiplexer lowers the bid–ask spread created by search, contracting, evaluation, and integration overhead: it compresses *transaction cost* without collapsing *data cost.* When the fixed costs fall, the long tail becomes liquid enough to trade: small datasets can clear; episodic feeds can clear; specialized sources can clear. The frontier becomes accessible not by lowering standards, but by making verification cheaper than doubt.

Brickroad | `brickroad.network` | Building the data multiplexer for the AI economy.

References

[1] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. Croissant: A metadata format for ml-ready datasets. In *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning (DEEM '24)*. Association for Computing Machinery, 2024.

[2] Ronald H Coase. The nature of the firm. *Economica*, 4(16):386–405, 1937.

[3] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[4] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[5] Ruoxi Jia, Luis Oala, Wenjie Xiong, Suqin Ge, Jiachen T. Wang, Feiyang Kang, and Dawn Song. A sustainable AI economy needs data deals that work for generators. *NeurIPS Position Paper*, 2025.

[6] Feiyang Kang, Hoang Xu, Harit Vishwakarma, Ruoxi Jia, et al. Performance scaling via optimal transport: Enabling data selection from partially revealed sources. *arXiv preprint arXiv:2307.02460*, 2023.

[7] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[8] Rohan P. Naidu et al. A cosmic miracle: A remarkably luminous galaxy at $z_{\text{spec}} = 14.44$ confirmed with JWST. *arXiv preprint arXiv:2505.11263*, 2025.

[9] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[10] Oliver E Williamson. *The Economic Institutions of Capitalism*. Free Press, 1985.