Learning-Based Creation of Data Mesh Architectures

The modern cloud data mesh is painful to create, manage, and use

- Modern data processing workloads are multi-faceted: real-time transactions and analytics, continuous ingestion, stream processing, exploratory queries in a data lake, etc.
- “One size does not fit all” has led to a plethora of specialized cloud services for each kind of workload
- A paradox of choice: Too much complexity for end users
- The holy grail: One system with state-of-the-art performance for all workloads while still leveraging existing specialized systems

BRAD abstracts away a mesh of specialized cloud database engines as “one system”

- Users issue SQL queries to a single endpoint, underneath which there can be many systems (e.g., Aurora, PostgreSQL, Redshift, etc.).
- Using a learned cost model, the planner creates a multi-engine query plan that optimally uses the underlying systems.
- Forecaster predicts the workload based on observed queries.
- Based on forecasting, policy engine scales each system to handle the workload at the lowest possible cost.
- The policy engine also optimizes data placement and movement to speed up queries and satisfy freshness guarantees.
- For incremental adoption, each organization can specify which decisions the policy engine is allowed to make.