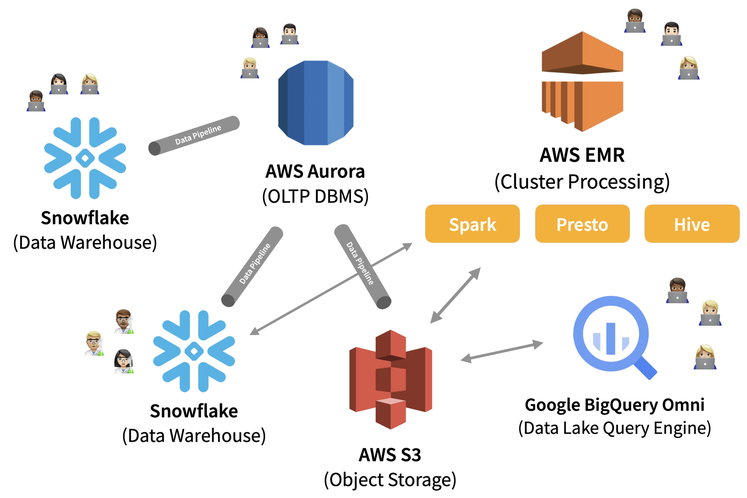
Check Out the Big Brain on BRAD:
Simplifying Cloud Data Processing with Learned Automated Data Meshes

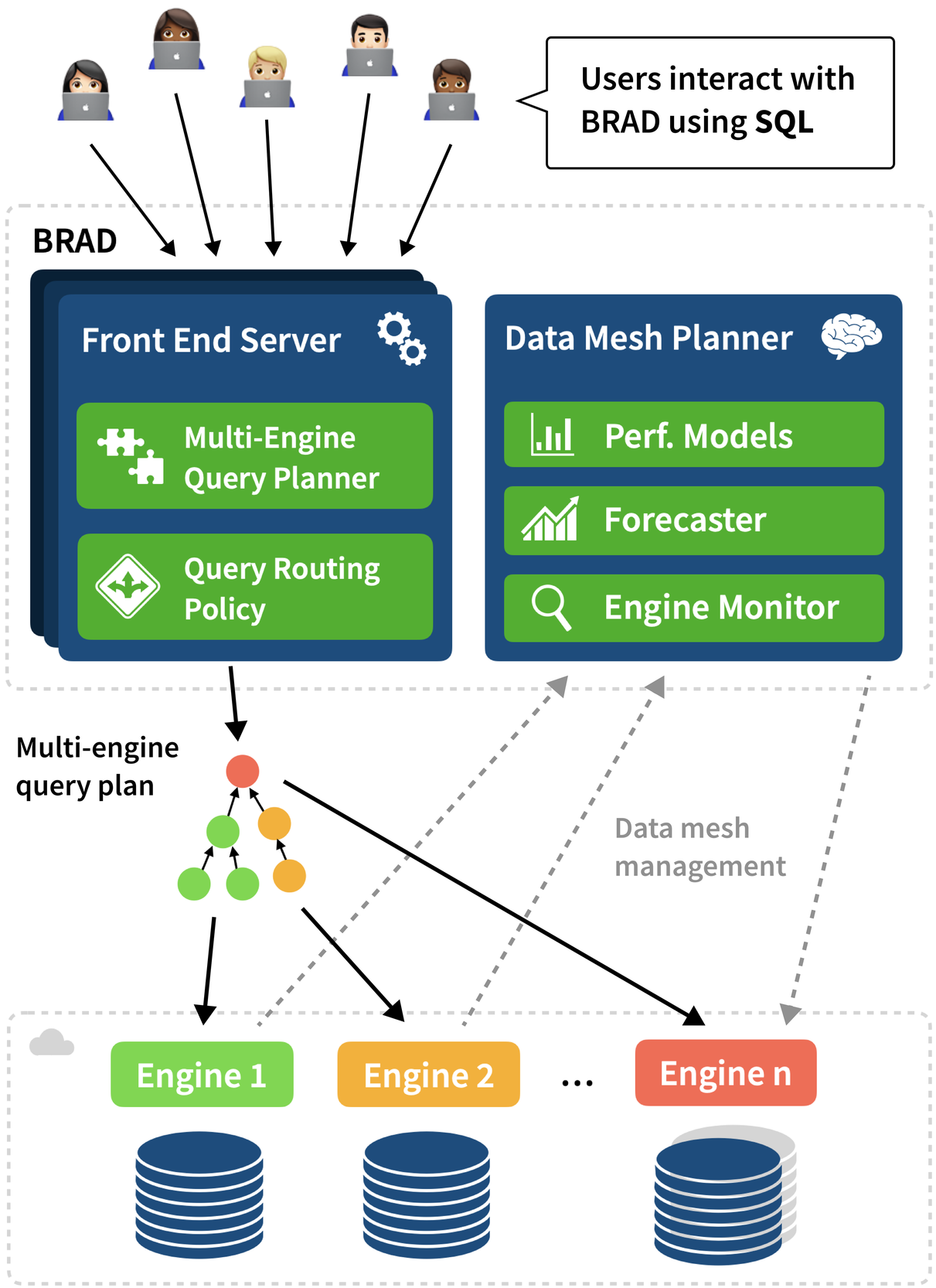
Tim Kraska*, Tianyu Li*, Samuel Madden*, Markos Markakis*, Amadou Ngom*, Ziniu Wu*, Geoffrey X. Yu*

BRAD is a cloud-native multi-engine database system where "one interface fits most"

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

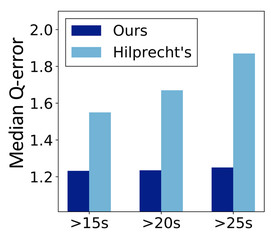
- "One size does not fit all" led to a **plethora of specialized cloud services** for data workloads
- **High administrative burden:** select the "best" engines, provision resources, evolve the mesh
- **The holy grail:** One system with state-of-the-art performance **for all workloads** while **leveraging existing specialized systems**

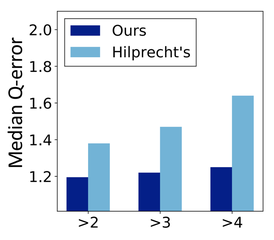




Learned query planning

- **Routing:** Select the best single engine for a query
- **Multi-engine query planning:** How to best split a plan across engines
- Powered by learned query **run time models** (cost models)
  - Cannot rely on physical plan features because BRAD might need predictions for an engine that is not yet running





BRAD's query run time model outperforms Hilprecht's transferable model [1] and generalizes across different queries

[1] Benjamin Hilprecht and Carsten Binnig. Zero-Shot Cost Models for Out-of-the-box Learned Cost Prediction. PVLDB, 15(11): 2361 - 2374, 2022

Joint mesh, query, cost optimization

- Jointly consider query routing, mesh design, data placement
- Optimize for cost under performance ceiling (or vice-versa)
- BRAD goes beyond cloud auto-scaling; workloads can be **moved across engines**
- Driven by an **ensemble** of models: (i) query run time on different hardware and system load, (ii) data scanned, (iii) data transfer times, (iv) cloud instance resize times

BRAD predicts query run time on different hardware configurations with **18%/22%** mean absolute percentage test error Aurora/Redshift

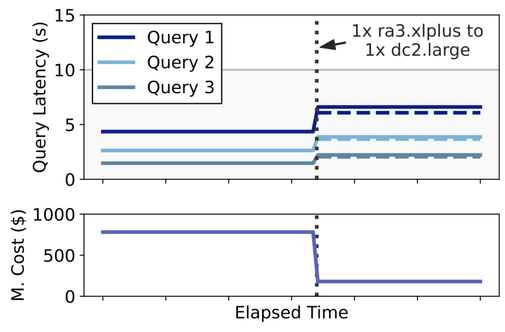
Over Different Instances

Over a Different Number of Nodes

Data consistency and auto ETLs/ELTs

Open Research Challenges

- Efficiently achieving strong consistency across engines?
- **Session-based freshness** guarantees; implemented using **tuple multi-versioning**, epoch-based **logical snapshots**
- Users describe transforms (ETLs/ELTs) using a declarative language
- BRAD auto-schedules transforms on the best system(s); using **special technologies** if able (e.g., AWS Glue, zero-ETL)



BRAD optimizes a data mesh containing Redshift, **saving 4x in cost** (change instance type while maintaining SLAs)

dsg.csail.mit.edu/projects/brad brad-project@csail.mit.edu tinyurl.com/brad-vision