Runtime-optimized analytics

Anastasia Ailamaki

EPFL and RAW Labs SA

DATA-INTENSIVE APPLICATIONS AND SYSTEMS

An incessantly evolving landscape

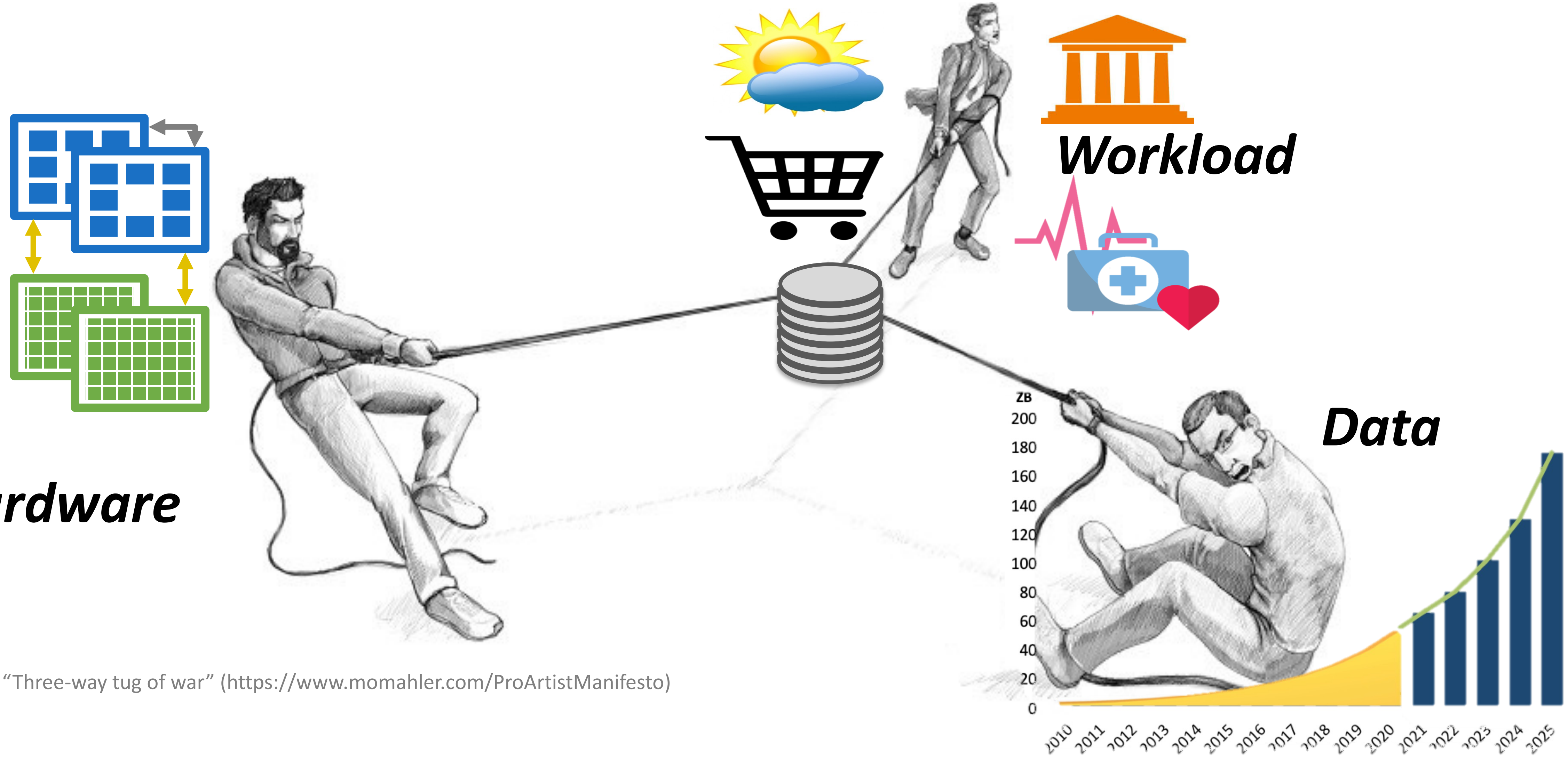
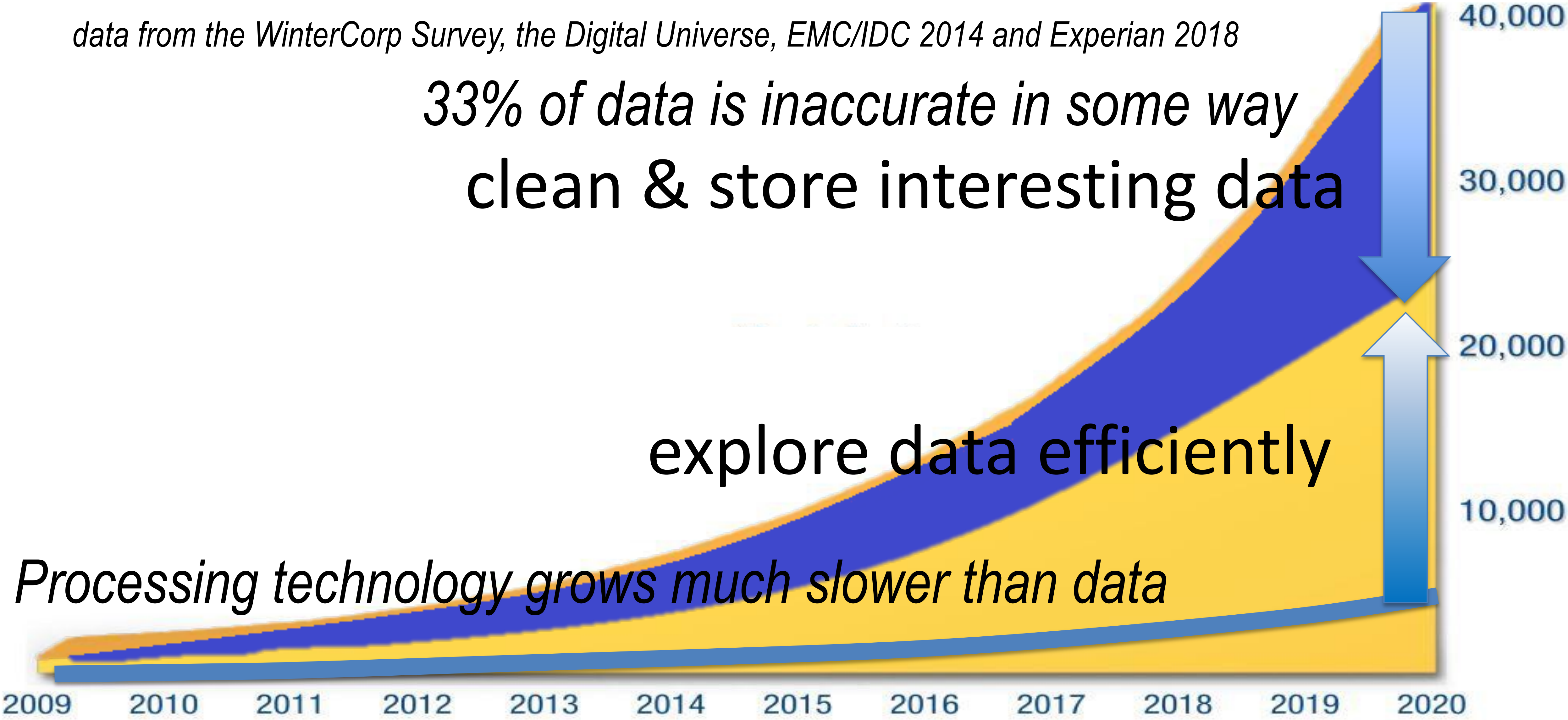


Image: “Three-way tug of war” (https://www.momahler.com/ProArtistManifesto)

Data management faces its most critical challenges

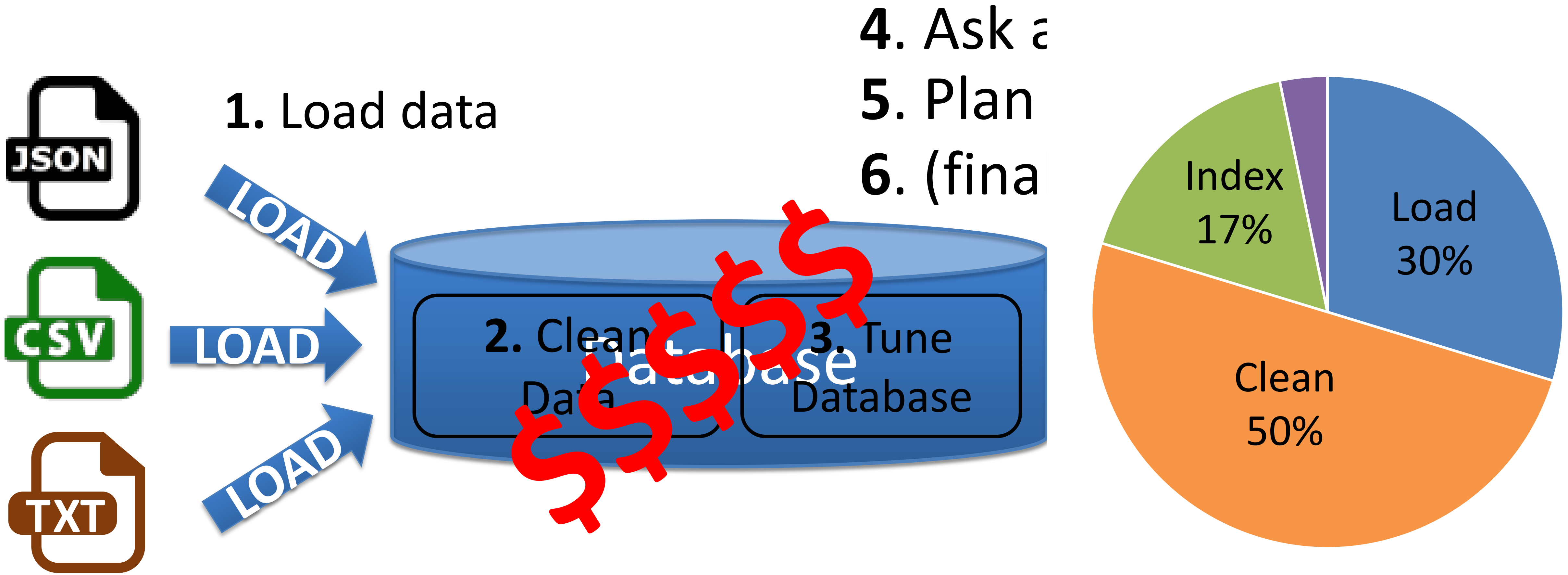
Game changer I: DATA

data from the WinterCorp Survey, the Digital Universe, EMC/IDC 2014 and Experian 2018



Paradox: 50-fold growth *impedes* discovery

Data preparation is expensive



Size, veracity, variety

biological disease signatures

coupling

clinical measurements with ***validated biomarkers***

Example: Alzheimer's disease

Clinical - Phenotype	Proteomic Biomarkers	Genomic Biomarkers	…arkers

…hippocampus atrophy/amyloid

Challenge:

Real-time integration of heterogeneous data

mem…

clinical+genetic+imaging data → signature

Patients (CSV)

id	Protein: AACT	Age	Phenotype	…
1	1.4	45	Trauma	…
2	2	55	Chronic Symptoms	…
3	0.2	56	…	…

Brain_GrayMatter (Binary)

	0	1	…	n
0	0.45	0.75	…	0.1
1	0.33	0.3	…	0.38
…	…	…	…	…
m	0.12	0	…	0.47

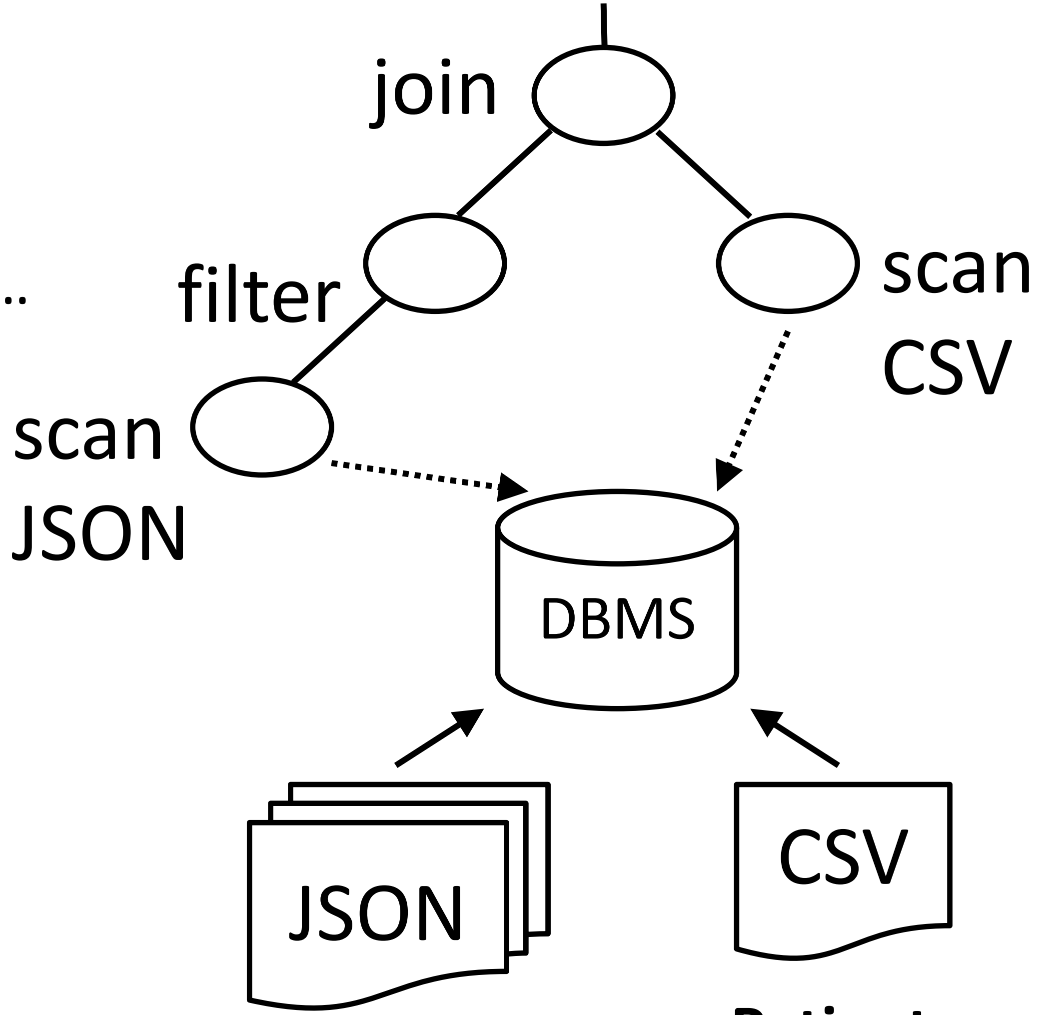
signature:
age > 50
AND
amygdala.Vol > 0.3
AND
AACT < 1

BrainRegions (JSON)

```
[{"id": 1,
 "amygdala": {"X":15,"Y":20, “Vol”: 0.5},
 "hippocampus": {"X":17, "Y":10, “Vol”:0.2}},
{"id": 2, ...},
{"id": 3, ...}]
```

Using a traditional database

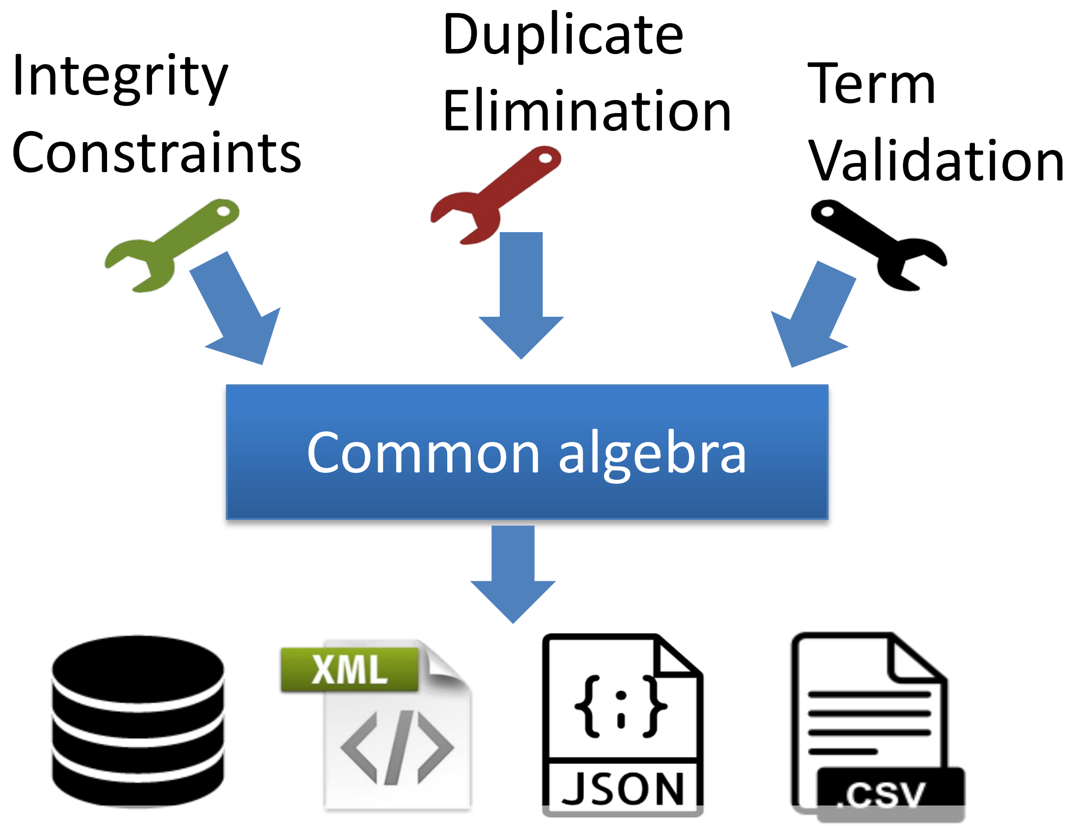
SELECT Phenotype, AACT, …
FROM BrainRegions, Patients
WHERE BrainRegions.id== Patients.id AND
BrainRegions. amygdala.Vol > 0.3 AND …

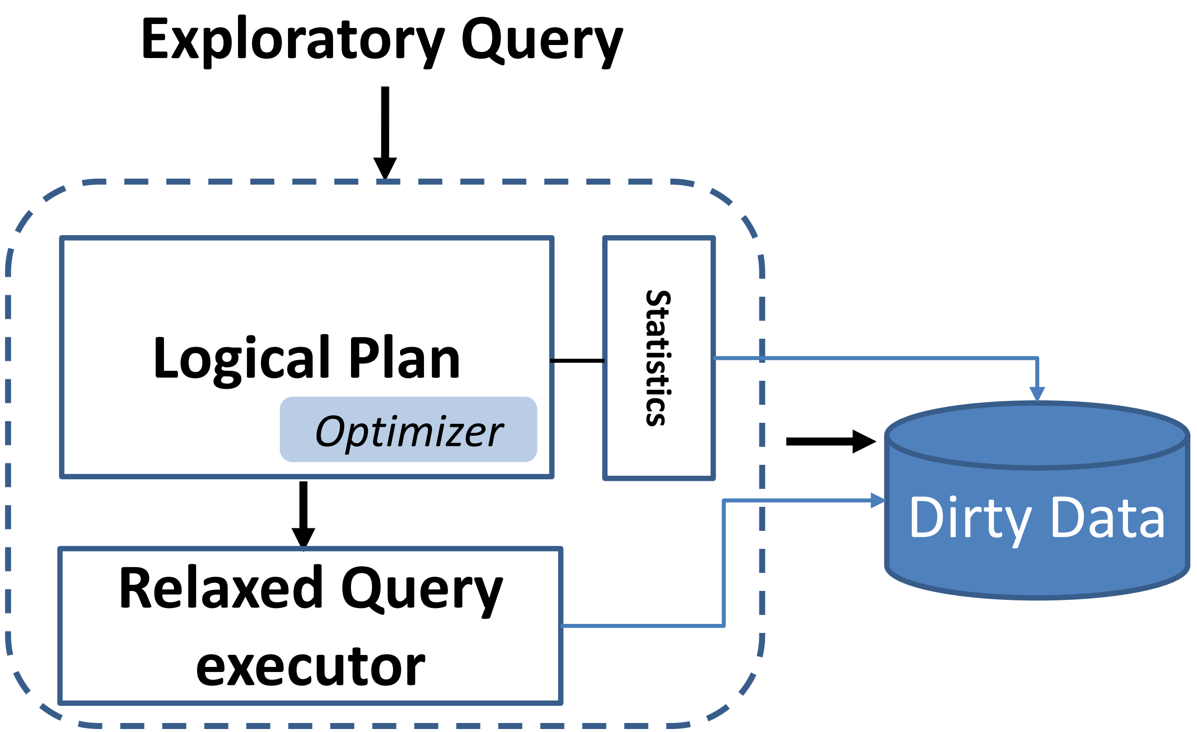


Data must first be cleaned and restructured

Efficient data veracity

Correct ALL errors on ALL data: costly and unnecessary!

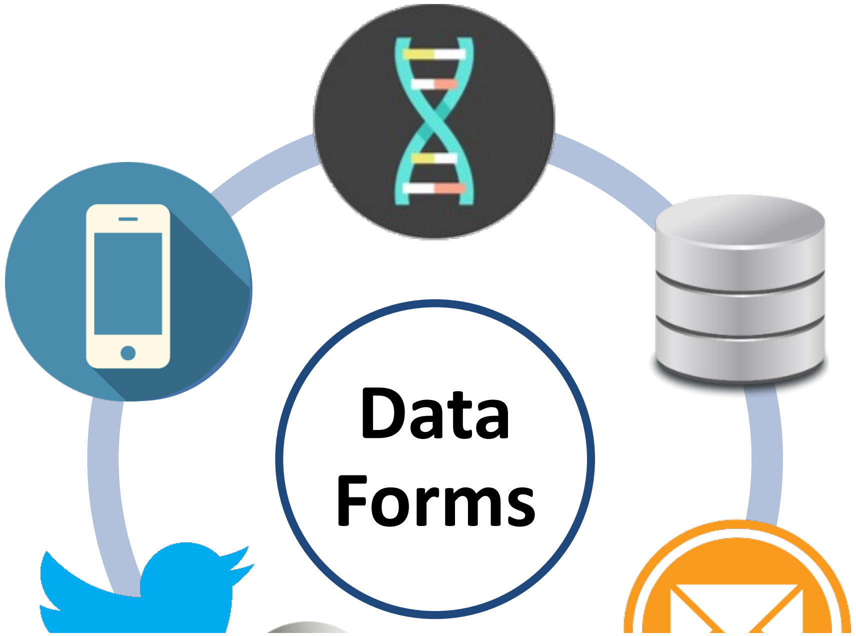




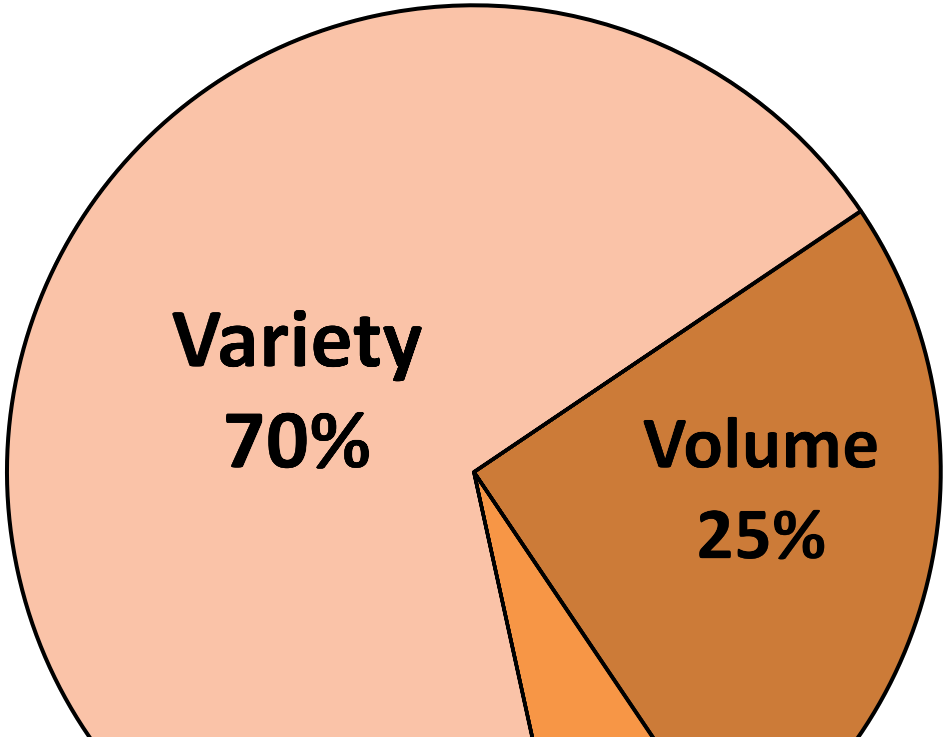
Clean data *adaptively* during analysis

The hidden foe: Data Variety

71% of data scientists:
Analysis more difficult due to variety, not volume [Paradigm4]



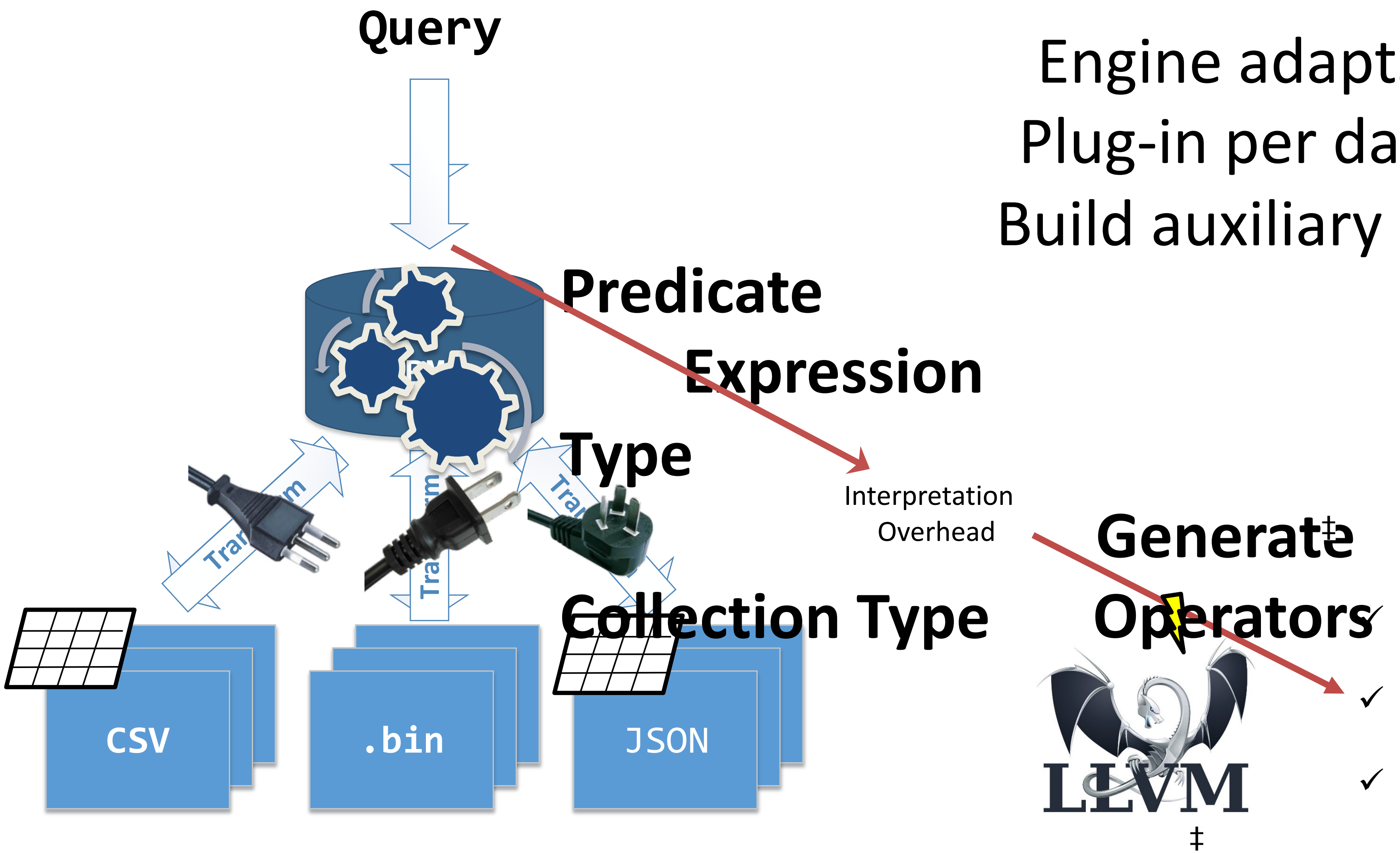
Variety, Volume, Velocity Importance [NVP Survey]



But… impossible to create a data system for all data and applications!

From LotsOfCode to NoCode

Engine adapts to data
Plug-in per data source
Build auxiliary structures



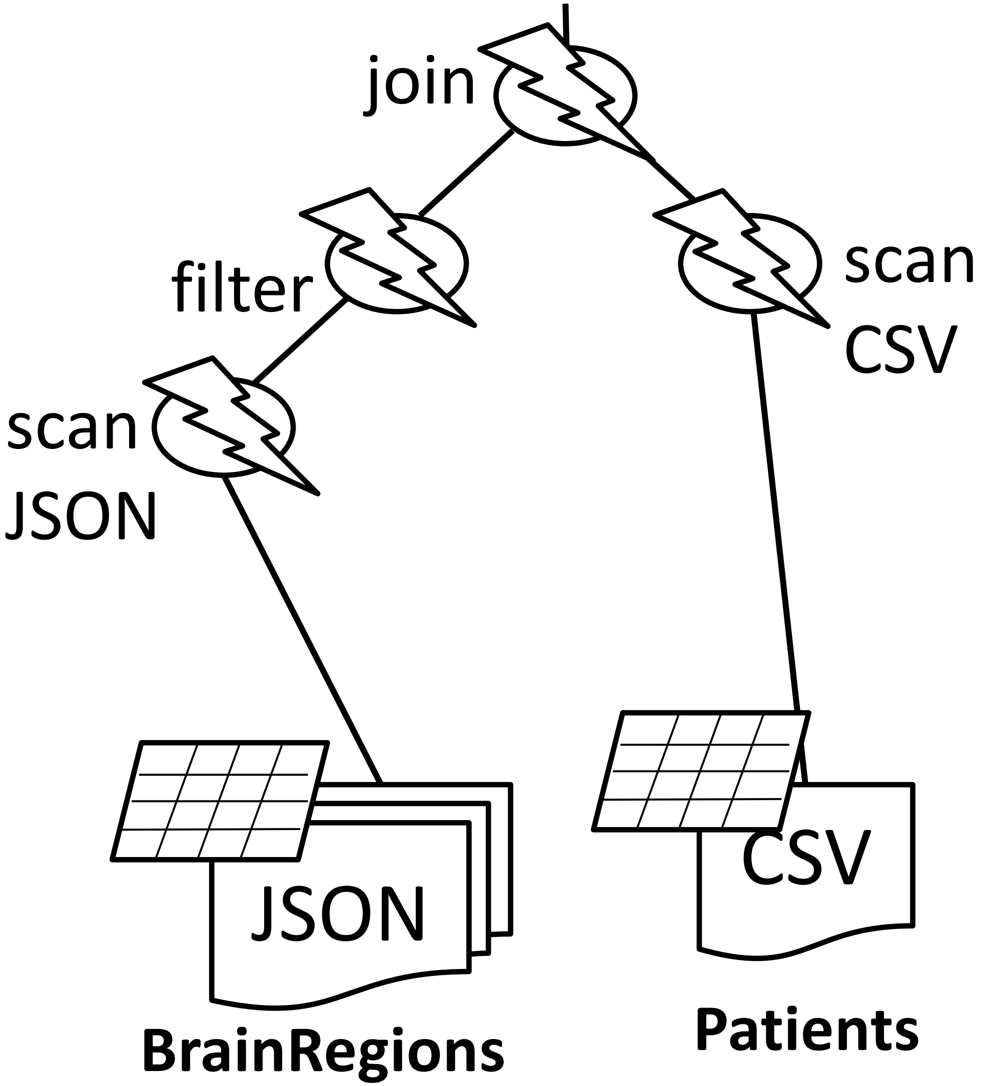
- ✓ Reduce branches
- ✓ Minimize function calls
- ✓ Pipelining

Codegen operators, continuously adapting engine

Harmonize useful data during execution

SELECT Phenotype, AACT, …
FROM BrainRegions, Patients
WHERE BrainRegions.id== Patients.id AND
BrainRegions. amygdala.Vol > 0.3 AND …

- Code-generate the access paths
- Code-generate the query
- Keep useful info in caches

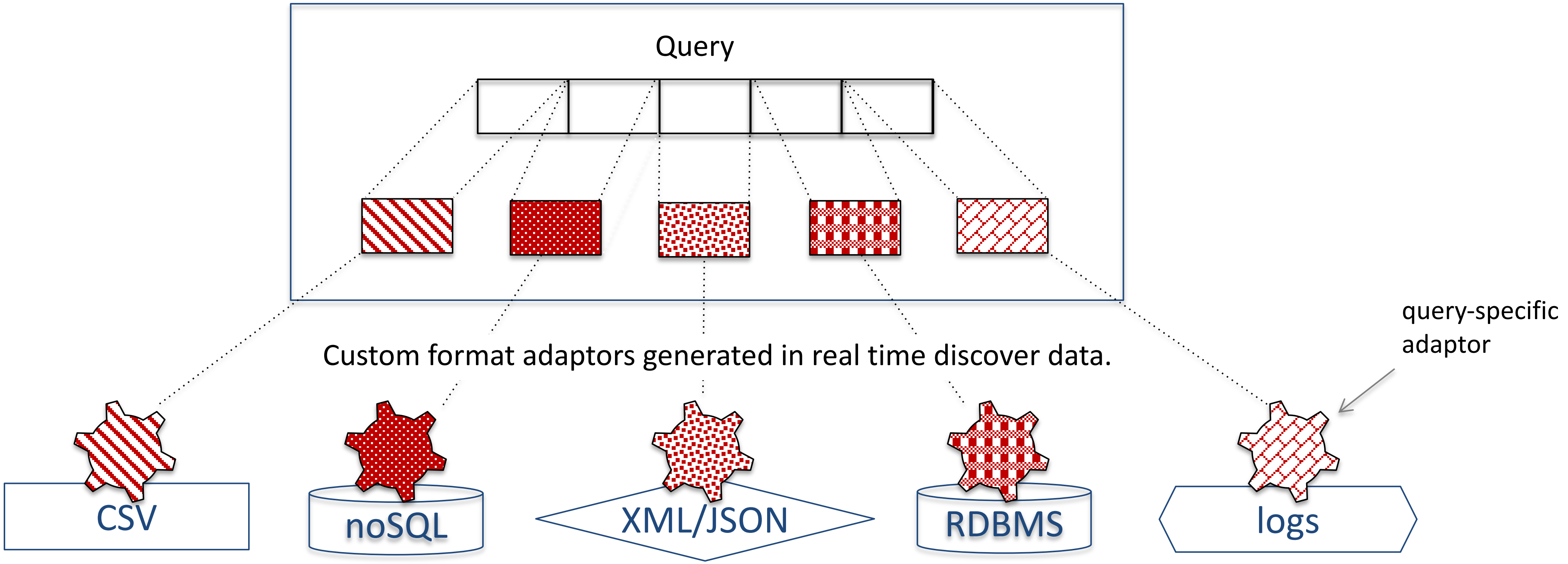


Data-to-query time = 0

RAW is a *single* engine for all data

RAW
Just ask.

RAW Query is automatically split up for each data source.



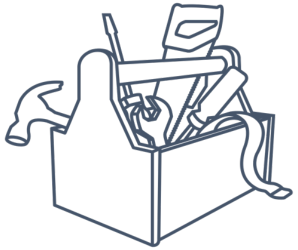
Data is integrated transparently and on-demand.

Users think of all of their data as a unified database, without preparation

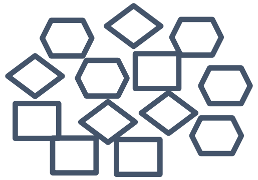
RAW is *fast*

RAW
Just ask.

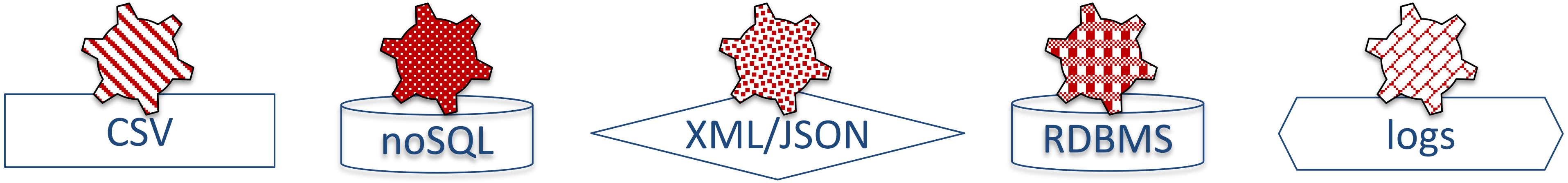
Just-built useful tools



RAW memory



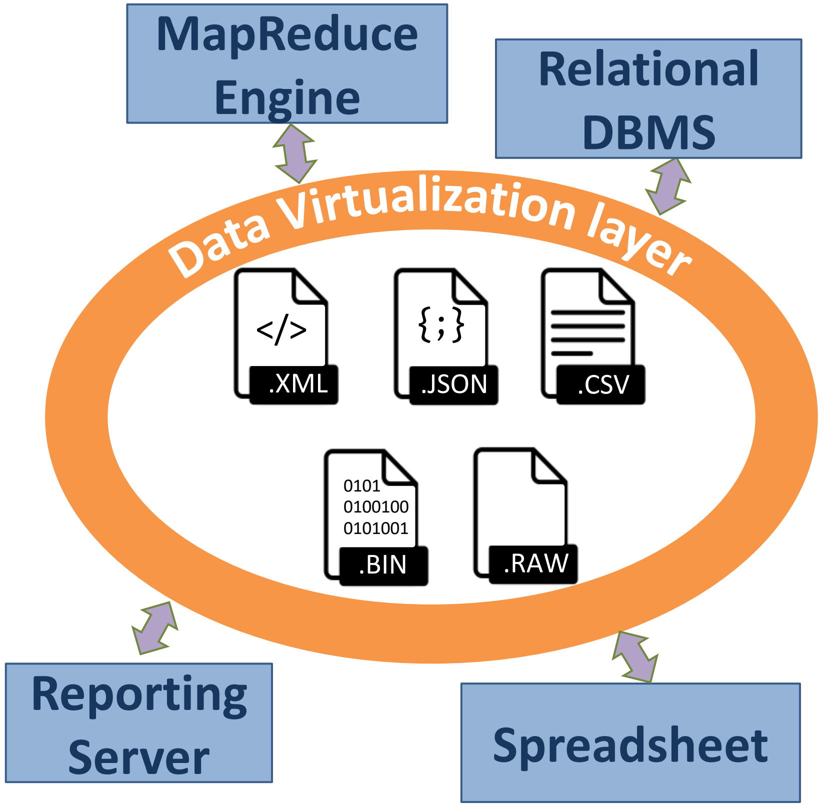
Frequently used data



As queries run, RAW remembers accessed data and generated code.
It builds and maintains a virtual datalake with only the useful data.

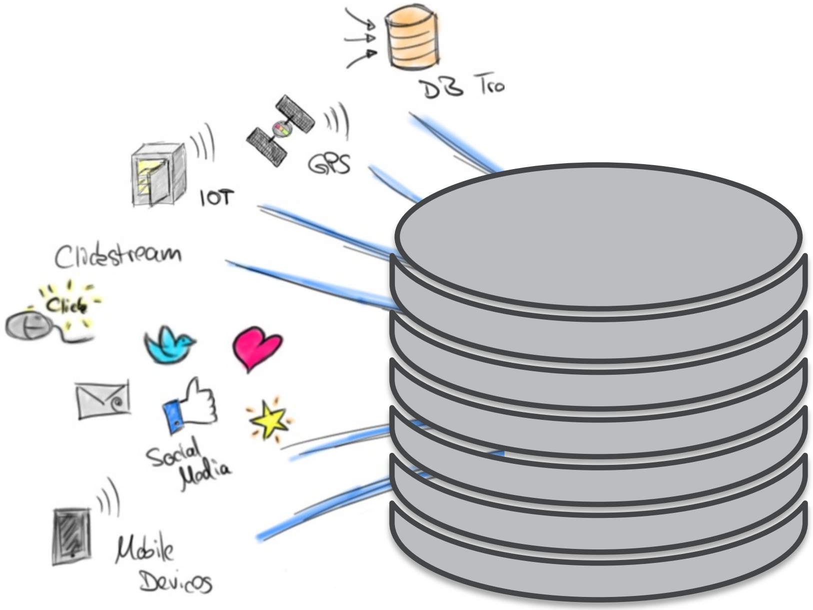
Hide data heterogeneity

- Many different data formats
- Modularity through virtualization
- Eliminate modularity overhead
  - JIT code generation



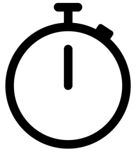
Self-generated engine *harmonizes* data

Game changer II: APPLICATIONS



Fresh Data

ETL:
Extract-Transform-Load



TRANSACTIONS

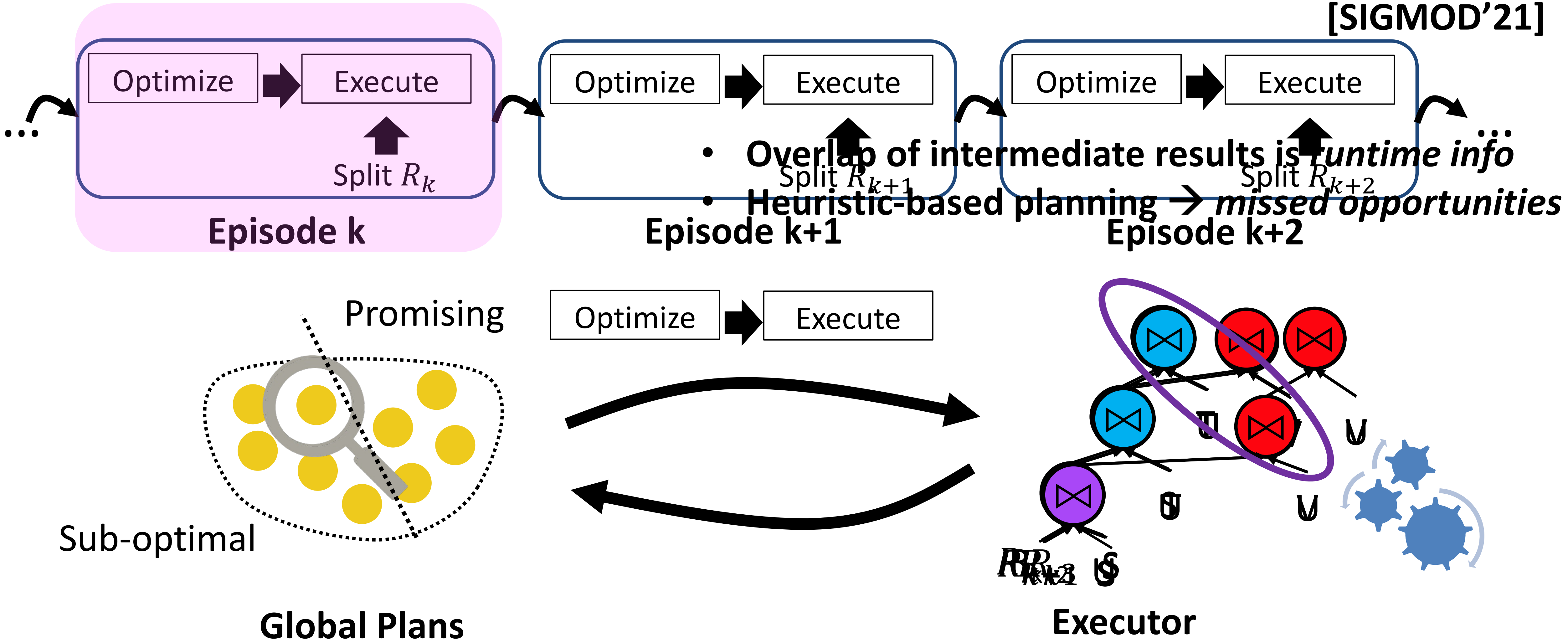
ANALYTICS

Ever-increasing number of concurrent queries

Data freshness bounded by ETL latency

Workload-conscious sharing

[SIGMOD'21]



Trial-and-error finds more and better sharing decisions

Hybrid Transactional and Analytical Processing

Transactions: task-parallel 

– High rate of short-lived processes

– Mostly “point accesses” (high data access locality)

Analytics: data-parallel 

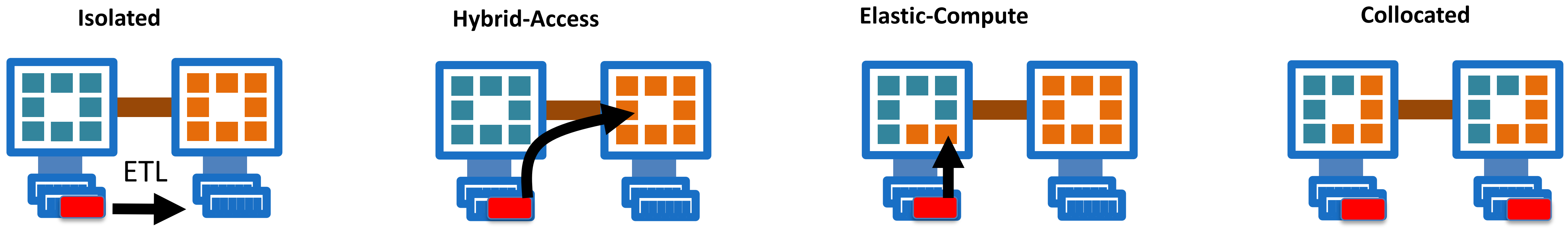
– Few, but long-running queries

Strong consistency is an invariant

Workload Isolation or Fresh Data?

TRANSACTIONS
ANALYTICS
Fresh Data

Collocated workloads fight for resources

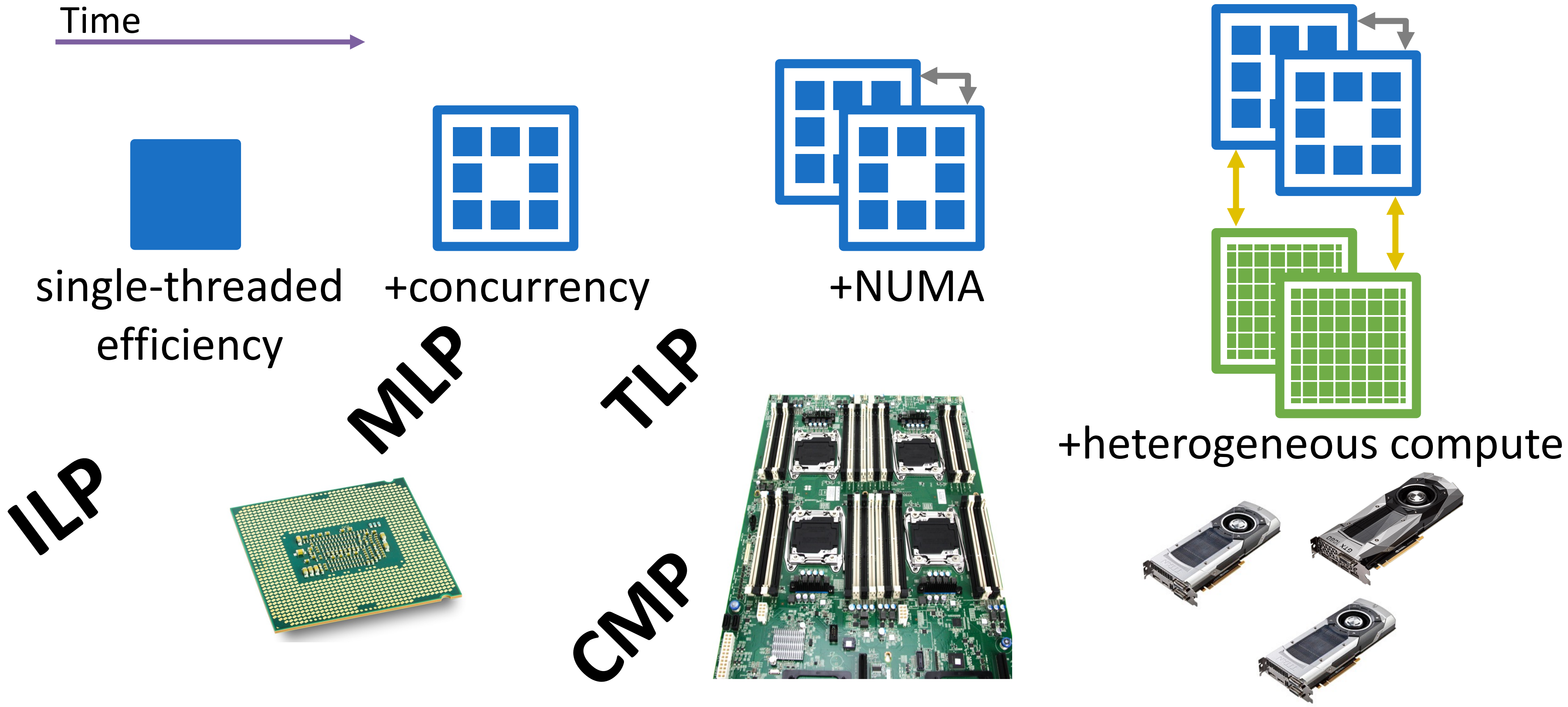


Interference → better data freshness

No interference → better performance

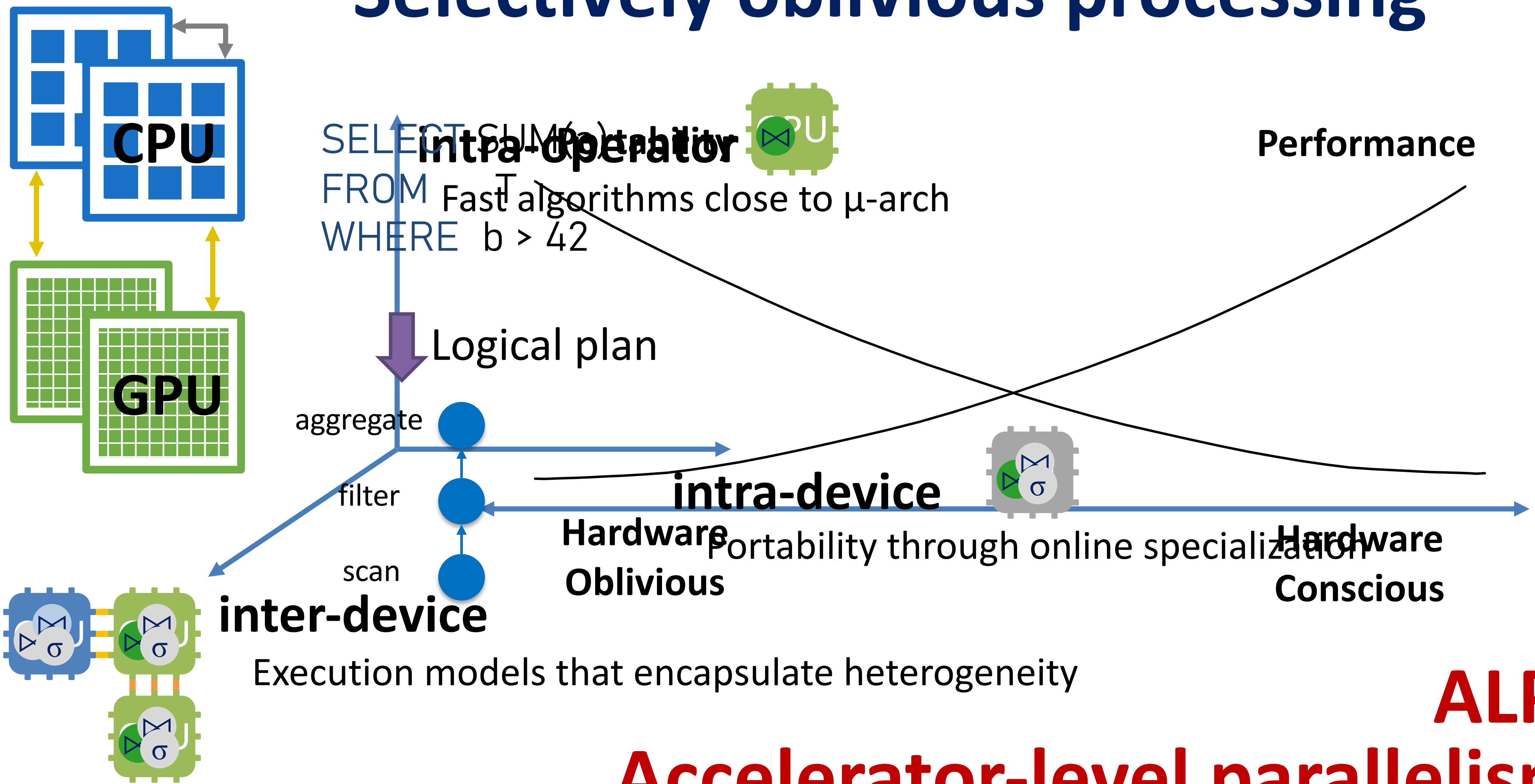
Viable hybrid alternatives

Game changer III: HARDWARE



Hardware conscious… or oblivious?

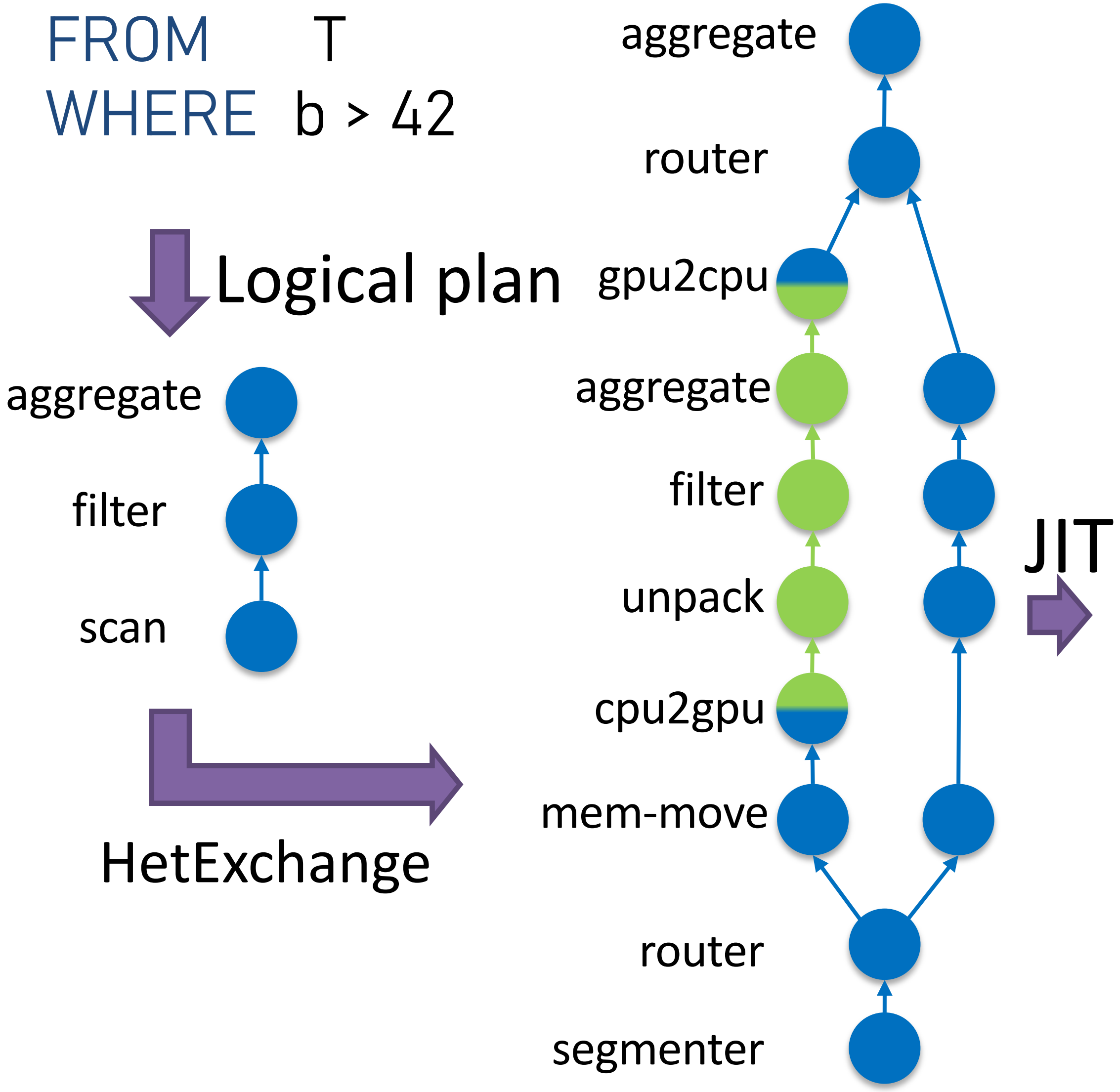
Selectively oblivious processing



ALP:
Accelerator-level parallelism

HetExchange: Heterogeneity-aware plans

SELECT SUM(a)
FROM T
WHERE b > 42

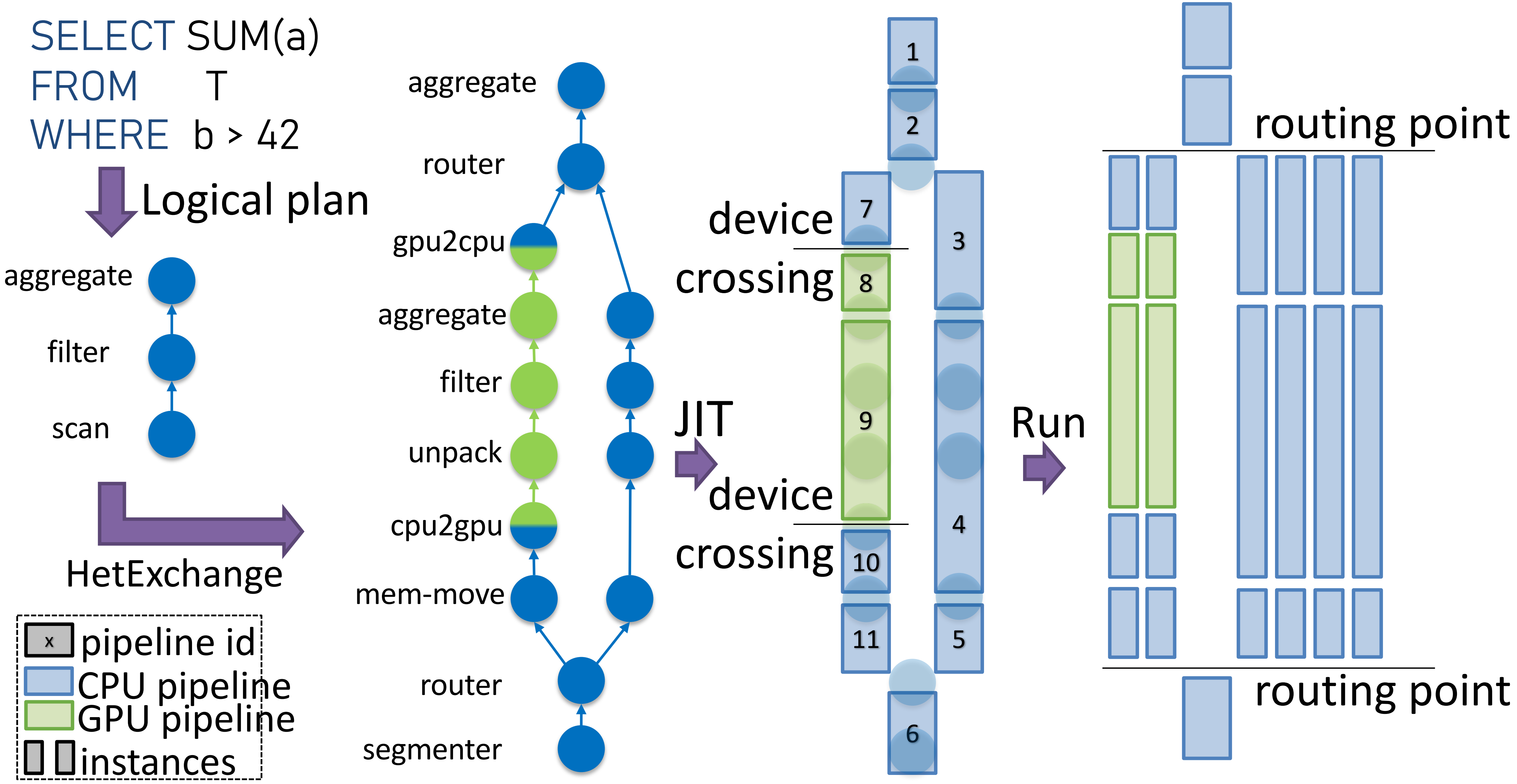


Decouple data- from control-flow

Operators encapsulate trait conversions

HetExchange in a JITed engine

SELECT SUM(a)
FROM T
WHERE b > 42

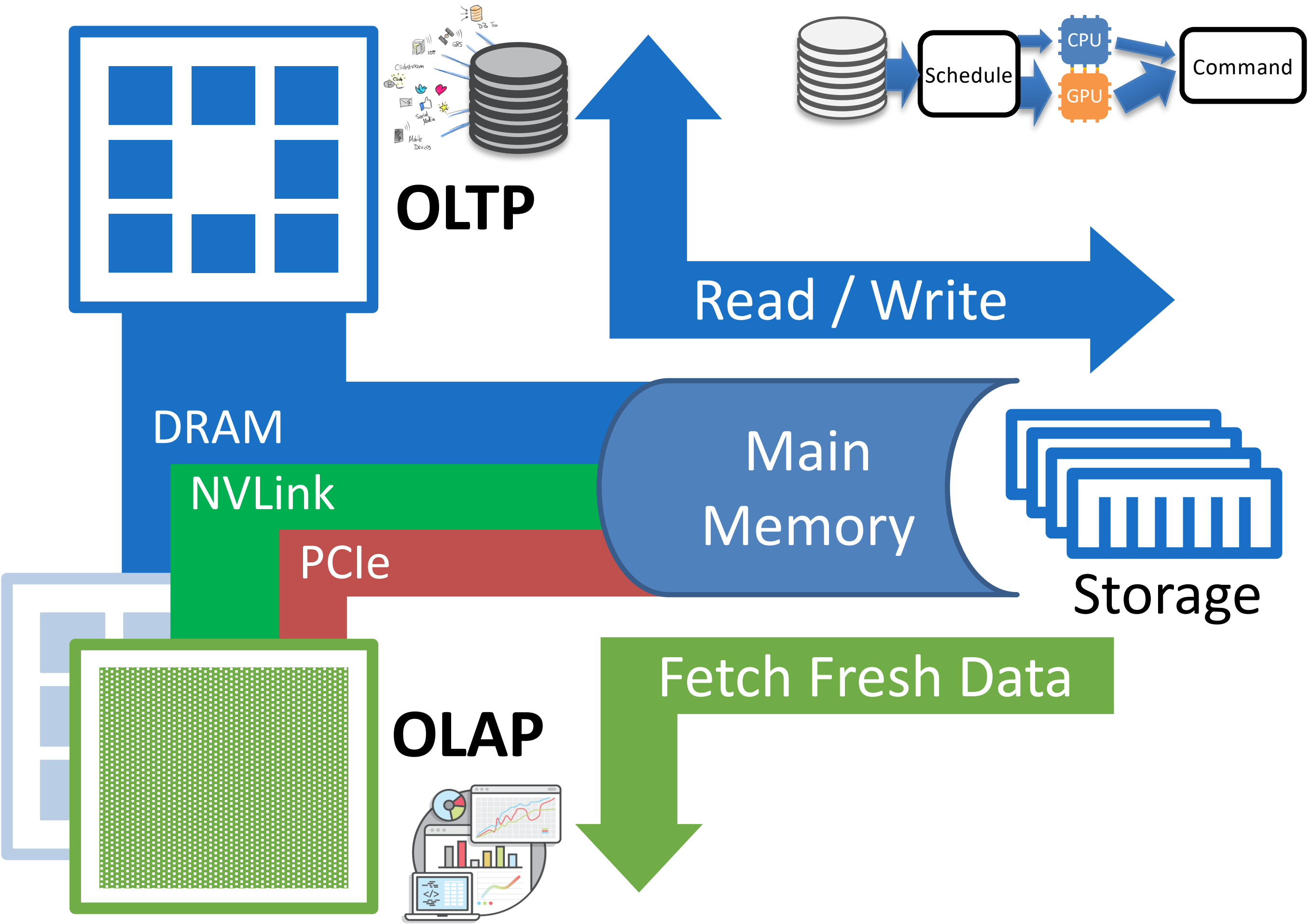


HTAP on heterogeneous hardware

Transactions store fresh data on CPU Memory

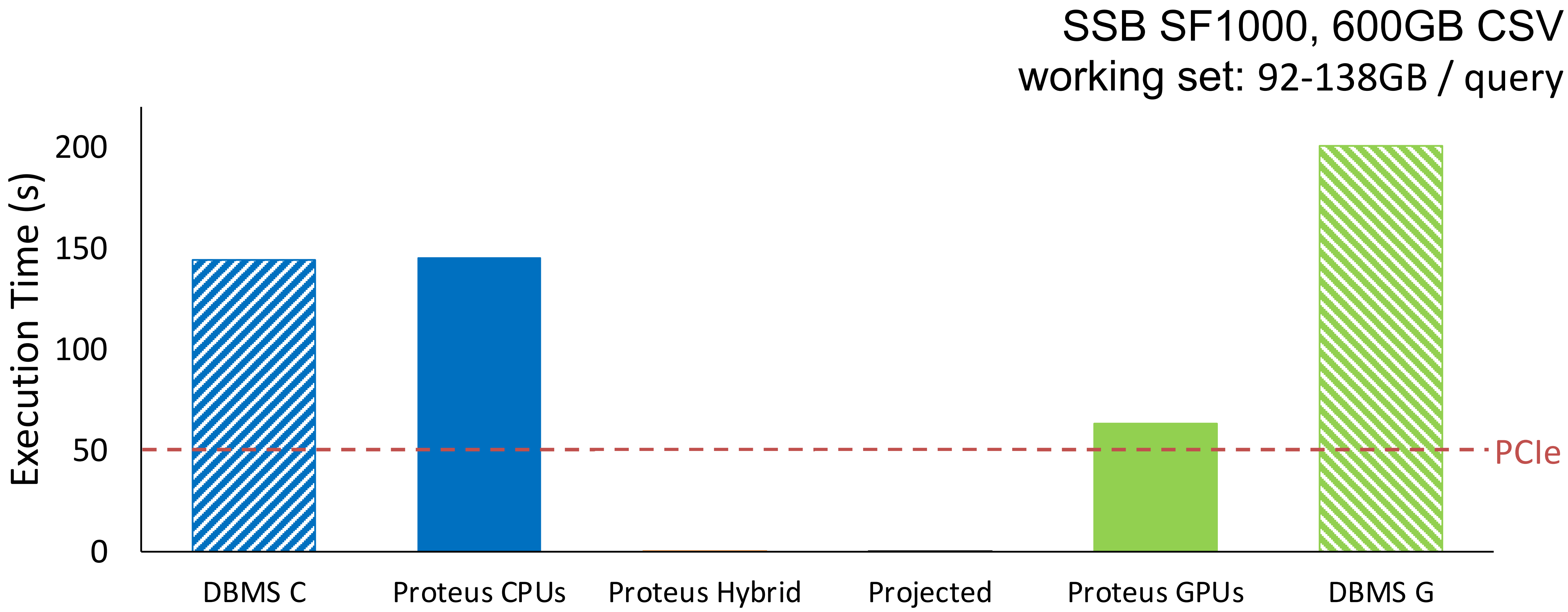
Data access protected by **concurrency control**

Analytics access fresh data through interconnect



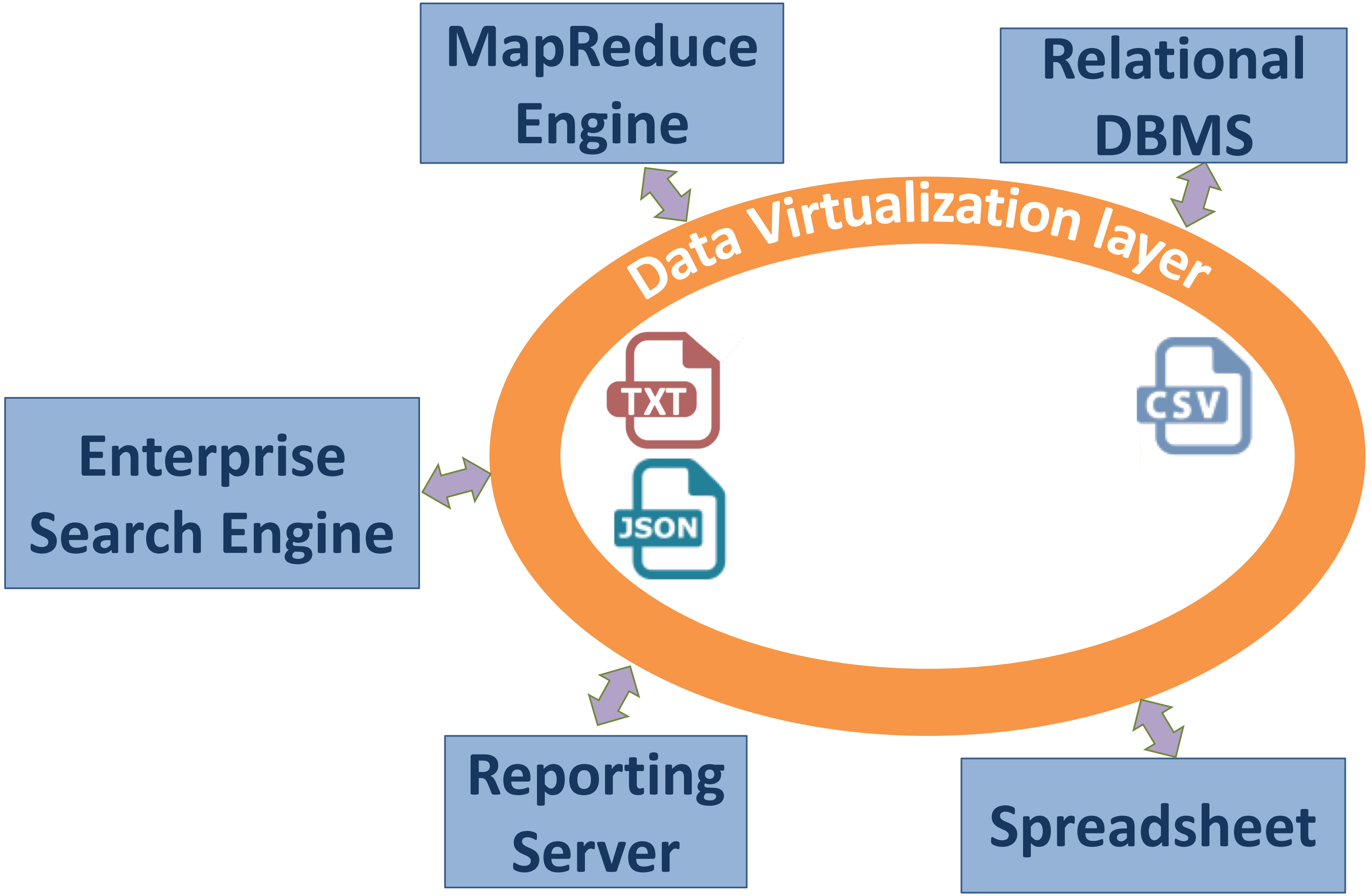
Real-time adaptive workload scheduling

Performance on CPU-resident data



Hybrid throughput = 88.5% (CPU-only + GPU-only), on average

Heterogeneity is *invisible*



Applications are portable and efficient

Increasing workload complexity

Commercial AI/ML

Diverse modern data problems

- IOT, OCR, ML, NLP, Medical, Mathematics etc…

Augmented analytics

DBMS catch-up for popular functionality

- Human effort and big delays
- Oblivious to out-of-DBMS workflows

Vast resource of libraries

- Authored by domain experts, used by everybody
- Loose library-to-data-sources integration and optimization

Conversational analytics and NLP

Combination of IoT and analytics

Need for systems that can “learn” new functionality

Network looks like a single machine

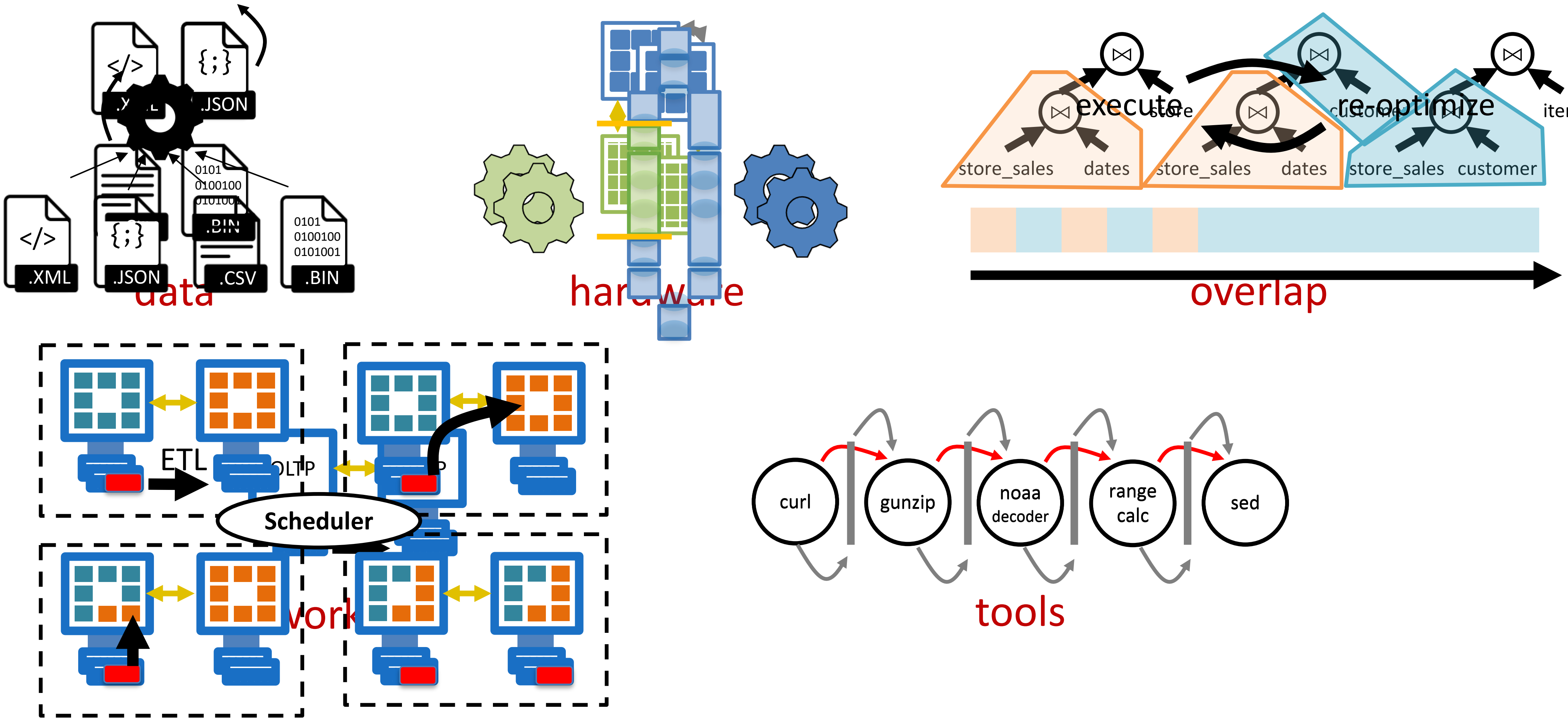
Similar intra-/inter-server interconnect bandwidth

Local memories and NUMA effects across devices

CPU-GPU: Capacity-Throughput

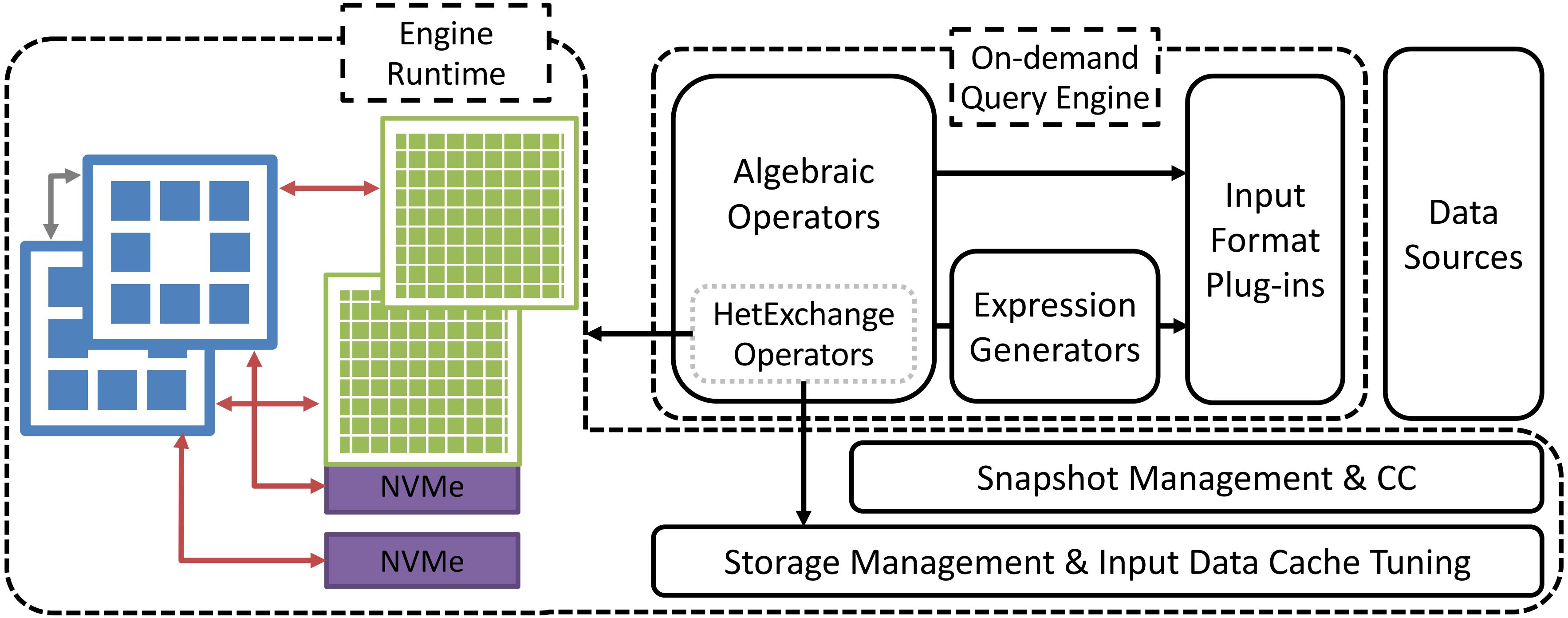
Efficient use of heterogeneous interconnected devices

Data pipelines are unpredictable



Taming heterogeneity through adaptivity

Proteus: Runtime-optimized analytics



Software is only as portable as its least adaptive component

Intelligent Real-time Systems

Incorporate change into native design.
Anticipate change and react, learning from errors.