IEEE-CNSV
Consultants' Network of Silicon Valley

The Evolution of Data Warehouses to Lakehouses: The Influence of AI/ML

Suresh Bysani

12th November 2024

IEEE-CNSV
Consultants' Network of Silicon Valley

Suresh Bysani

Director of Engineering,
Core SAAS AND AI Infrastructure at Eightfold.ai - responsible everything data and infra

Agenda

IEEE-CNSV
Consultants' Network of Silicon Valley

1. Introduction and Welcome

2. Evolution of OLAP

3. Advancements in OLAP

Break

4. AI Agentic Architecture

5. Demo (Agents with OLAP)

6. Summary

OLTP VS OLAP

IEEE-CNSV
Consultants' Network of Silicon Valley

OLTP (Online Transactional Processing)

OLTP systems focus on handling individual transactions, such as customer orders, inventory updates, and financial transactions, in real-time.

OLAP (Online Analytical Processing)

OLAP systems are designed for complex data analysis, enabling users to perform multidimensional queries, data aggregation, and trend analysis.

Why Companies Need Both

Companies require both OLTP and OLAP systems to effectively manage their business operations. OLTP handles day-to-day transactions, while OLAP provides insights for strategic decision-making.

AI Use Cases Powered by OLAP

OLAP systems provide the robust data foundation required for many AI and machine learning use cases, such as predictive analytics, customer segmentation, and demand forecasting.

By understanding the fundamental differences between OLTP and OLAP, companies can leverage both systems to drive efficient operations and informed decision-making, ultimately supporting their AI and data-driven initiatives.

OLTP Systems

Postgres

My SQL

IEEE-CNSV
Consultants' Network of Silicon Valley

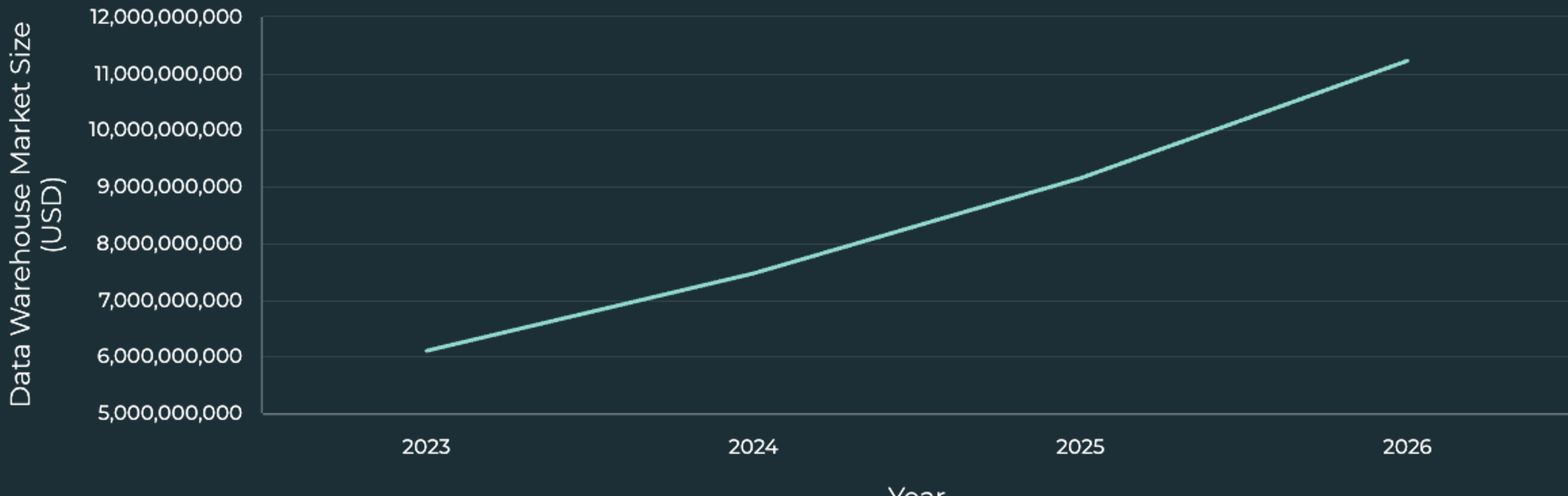
OLAP Systems

teradata.

Teradata

Redshift

Increasing Analytics Spend



The data warehouse market is projected to grow rapidly at a CAGR of 22.5% from 2024-2032.

Why do companies need OLAP

IEEE-CNSV
Consultants' Network of Silicon Valley

Multidimensional Data Analysis

OLAP systems allow companies to analyze data from multiple perspectives, such as time, product, region, and customer, providing deeper insights into business performance.

Trend Identification and Forecasting

OLAP systems help companies identify patterns, trends, and anomalies in their data, allowing them to make more informed decisions and better predict future performance.

Rapid Reporting and Dashboards

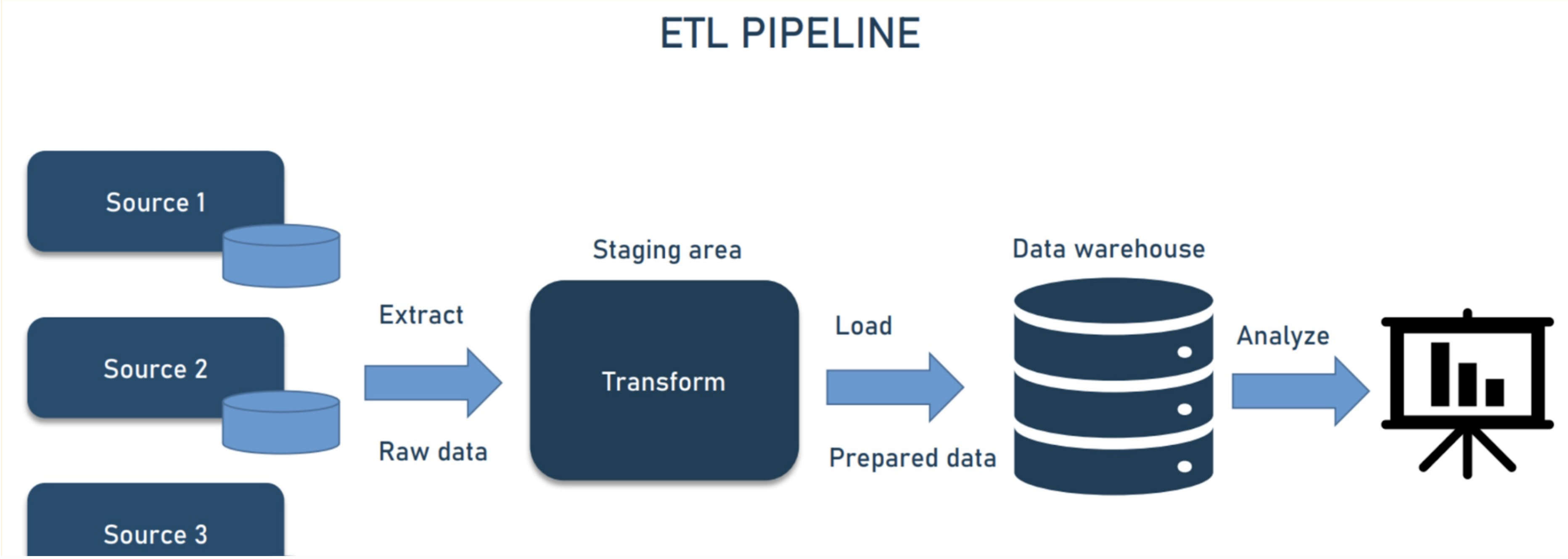
OLAP systems enable companies to quickly generate customized reports and interactive dashboards, empowering decision-makers with real-time data-driven insights.

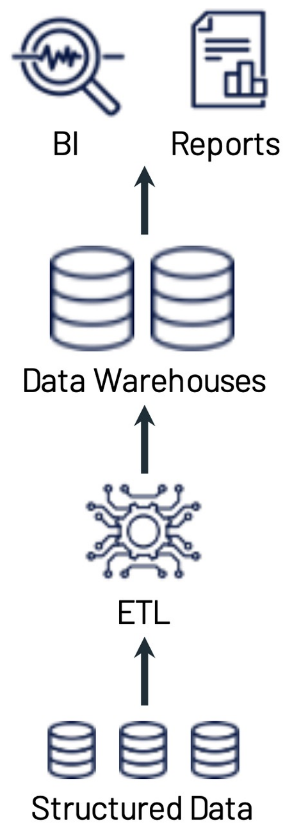
Improved Business Agility

OLAP systems provide companies with the flexibility to adapt to changing business requirements, enabling them to quickly access and analyze data from various sources.

OLAP systems are a powerful tool for companies to gain deeper, more comprehensive insights into their business operations, enabling them to make more informed, data-driven decisions and stay competitive in their respective markets.

IEEE-CNSV
Consultants' Network of Silicon Valley





(a) First-generation platforms.

Limitations of First Generation

IEEE-CNSV
Consultants' Network of Silicon Valley

Limited Data Handling Capabilities

First-generation data warehouses were designed primarily for Business Intelligence (BI) purposes, with a focus on structured data and limited ability to handle unstructured data and real-time analysis required for AI use cases.

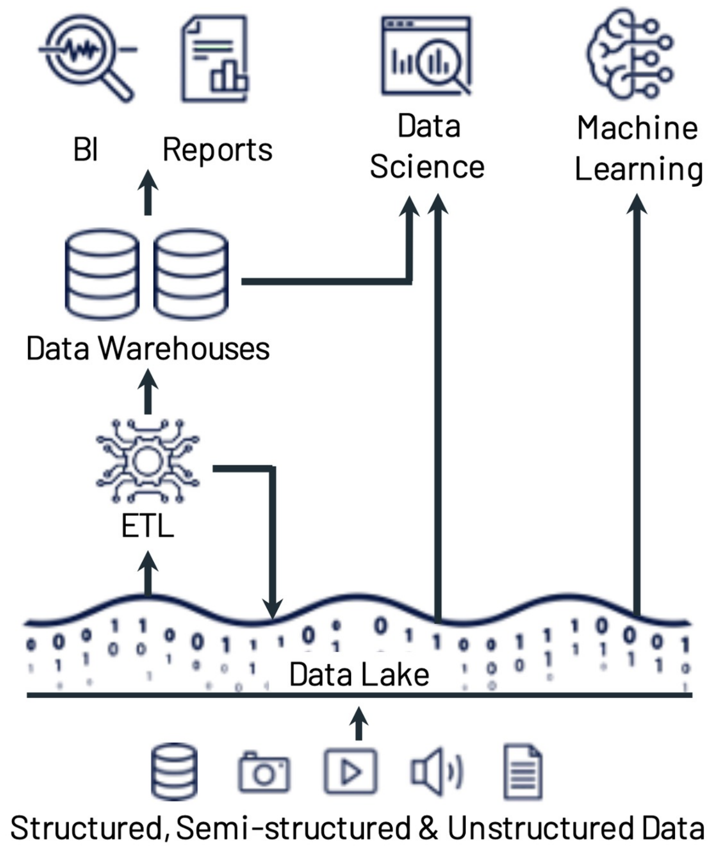
Rigid Data Model

The fixed schema and dimensional data model of first-generation data warehouses can't accommodate the flexible and evolving data structures often required for AI models, which need to process diverse data sources and types.

Slow Performance

First-generation data warehouses are optimized for batch processing and reporting, which may not provide the low-latency and high-throughput performance needed for real-time AI applications that require rapid data ingestion and analysis.

In summary, the design and architecture of first-generation data warehouses are optimized for BI use cases, but lack the flexibility, performance, and scalability required for modern AI applications, which demand the ability to handle diverse, real-time data at scale.



(b) Current two-tier architectures.

Data lake

Data Lakes and the Hadoop Movement

The second generation data analytics platforms offloaded raw data into data lakes - low-cost storage systems with a file API that hold data in generic and open file formats like Apache Parquet and ORC.

Cloud Data Lakes and Data Warehouses

Cloud data lakes like S3, ADLS and GCS replaced HDFS, offering superior durability, geo-replication, and extremely low-cost archival storage. This two-tier data lake and data warehouse architecture is now dominant in the industry.

Complexity and Challenges

While the cloud data lake and warehouse architecture is cheap due to separate storage and compute, it is highly complex for users. Data is ETLed into lakes and then ELTed into warehouses, creating delays and new failure modes. Advanced analytics like machine learning are not well-suited for these architectures.

The current data architectures, while cost-effective, suffer from increased complexity, delays, and challenges in supporting advanced analytics use cases.

Why datalakes are not good enough

Data Lakes for AI

Data lakes are well-suited for AI and machine learning workloads due to their ability to handle large, diverse, and unstructured data sets. The flexible schema and real-time processing capabilities make them ideal for AI-driven insights.

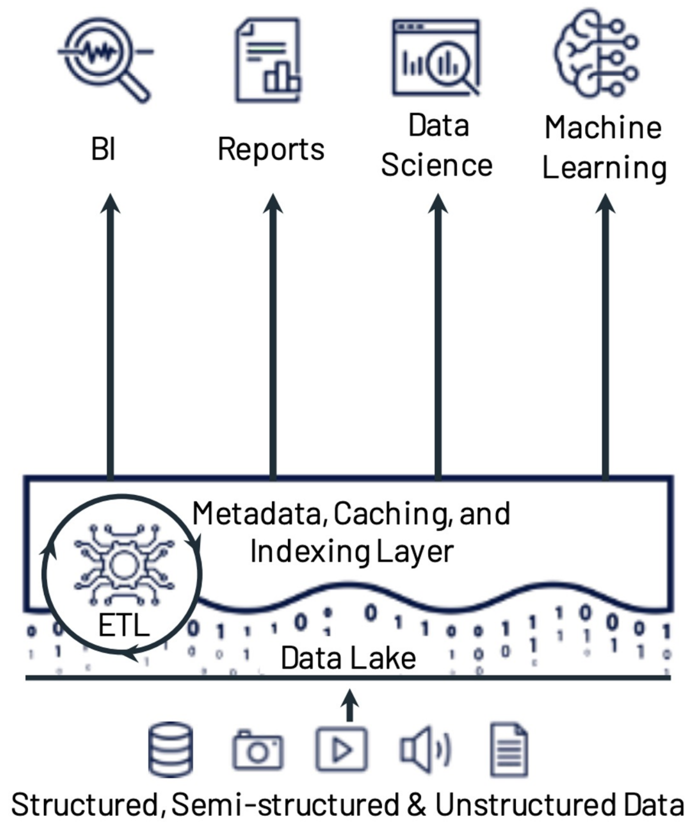
Limitations for BI

While data lakes can provide a centralized repository for data, they often lack the structured, schema-driven approach required for effective business intelligence (BI) reporting and analytics. BI tools typically perform better with data stored in a data warehouse.

Need for ETL to Warehouses

To achieve optimal BI performance, data from the data lake often needs to be extracted, transformed, and loaded (ETL) into a data warehouse. The data warehouse provides a more organized, schema-driven structure that BI tools can leverage for faster, more reliable reporting and analysis.

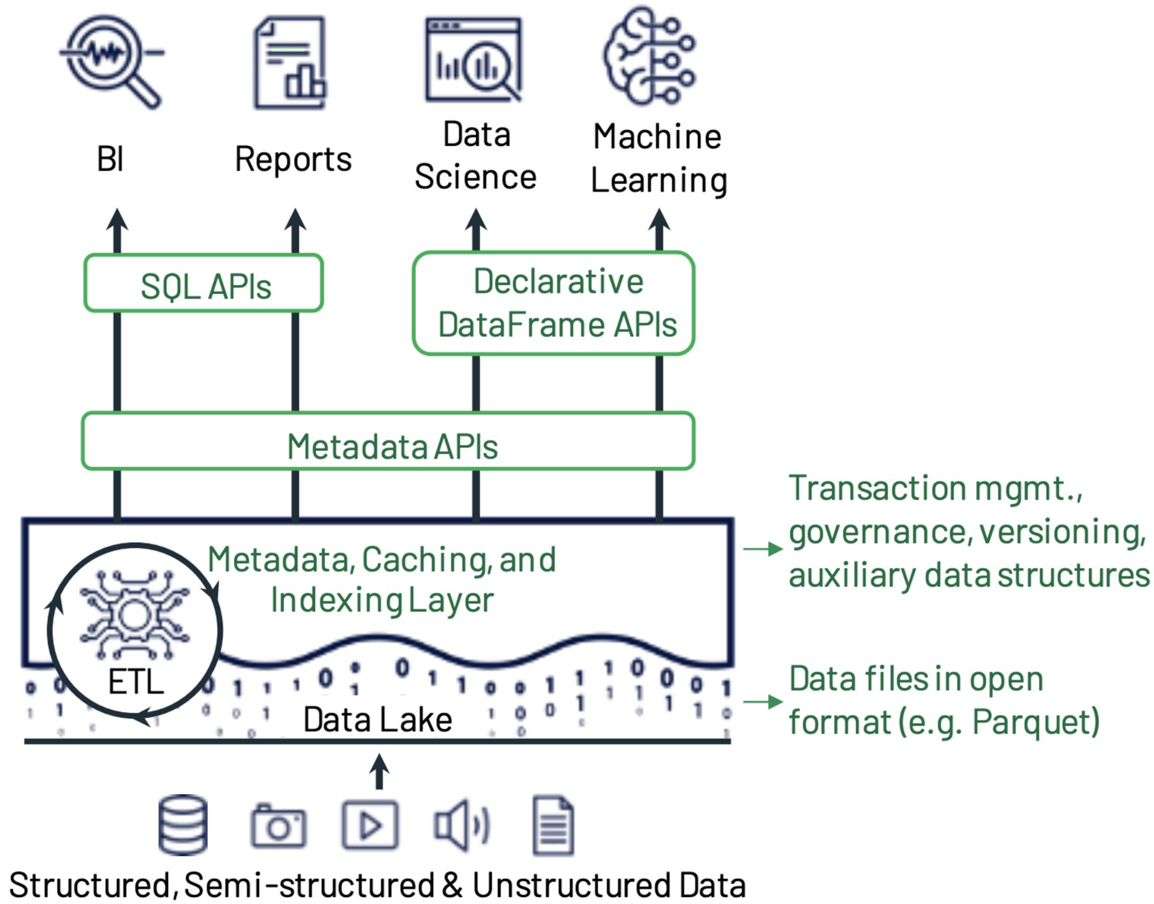
In summary, while data lakes are well-suited for AI and machine learning workloads, they often require additional ETL processes to a data warehouse to achieve optimal performance for business intelligence and reporting purposes.



(c) Lakehouse platforms.

IEEE-CNSV
Consultants' Network of Silicon Valley

IEEE-CNSV
Consultants' Network of Silicon Valley



Lakehouses

IEEE-CNSV
Consultants' Network of Silicon Valley

Lakehouse Architecture

Lakehouse architecture integrates the data warehouse and data lake concepts, providing a unified platform for analytics and machine learning.

Metadata Layer Significance

The metadata layer is the crux of the lakehouse architecture, as it manages the data schema, partitioning, and other metadata to enable efficient querying and processing.

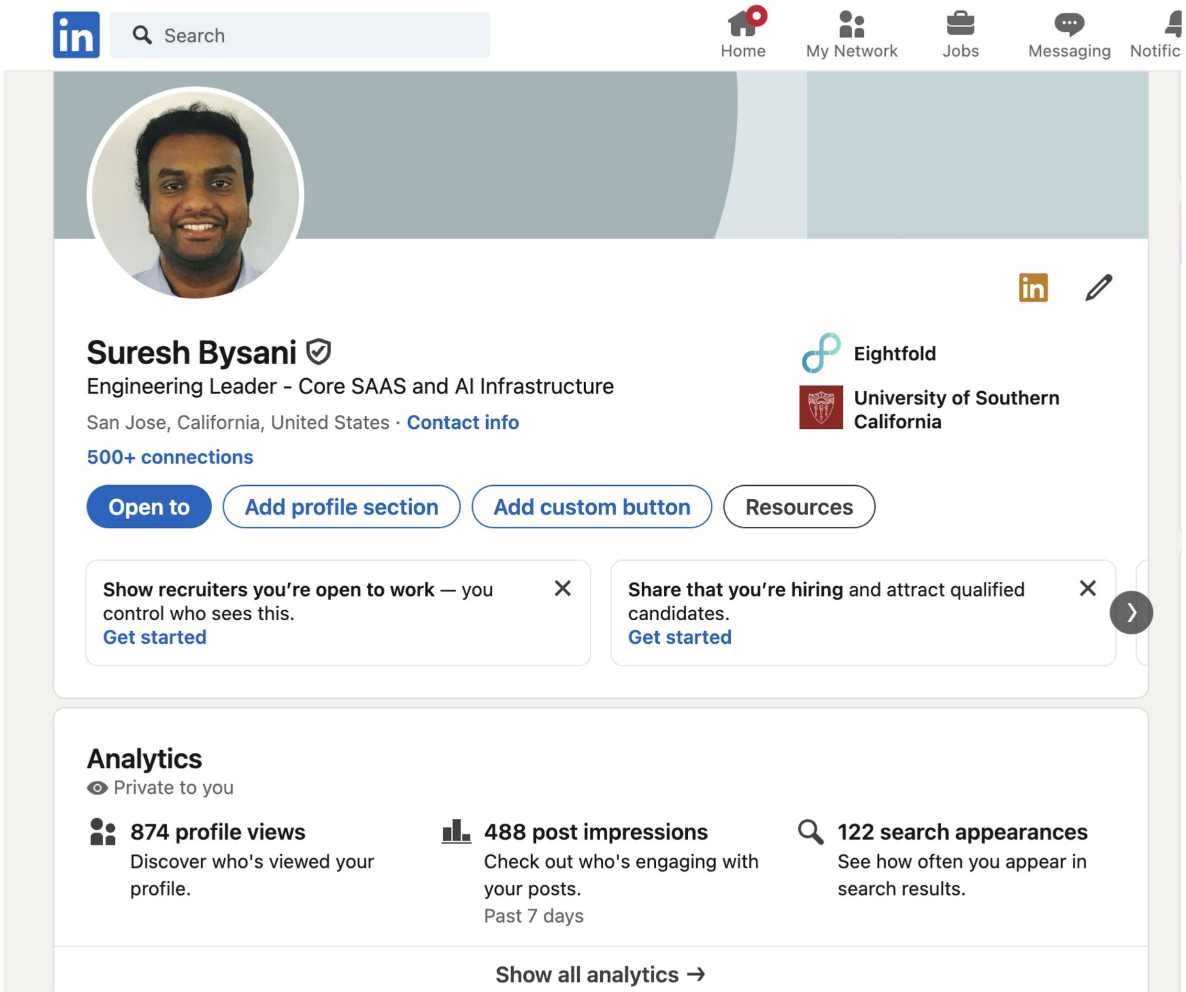
File Formats

Common file formats used in lakehouse architectures include Apache Iceberg, Hudi, and Delta Lake, which provide features like ACID transactions, time travel, and schema evolution.

The lakehouse architecture, with its emphasis on the metadata layer and use of advanced file formats, offers a powerful and flexible data platform for modern analytical and machine learning workloads.

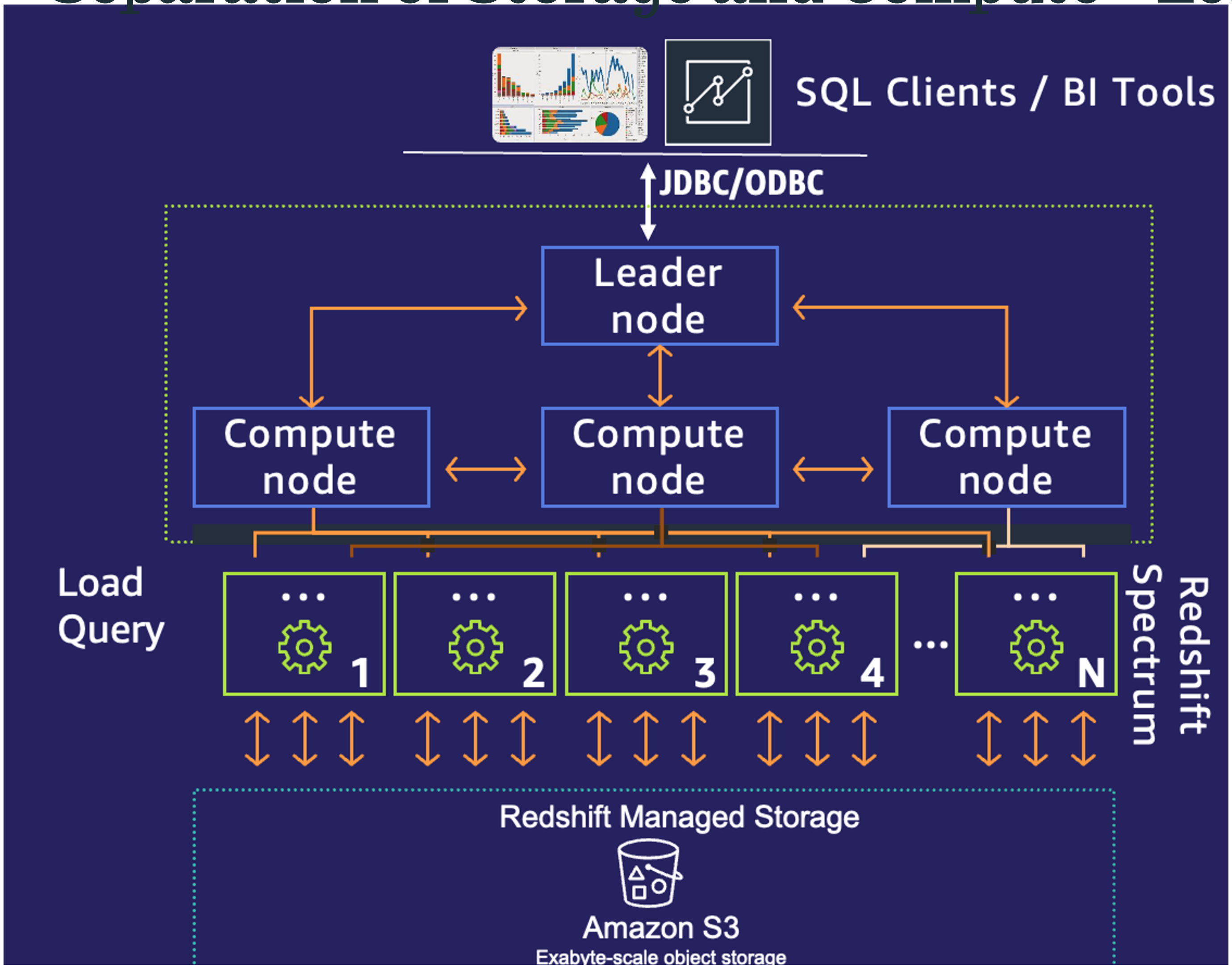
In Product Analytics

IEEE-CNSV
Consultants' Network of Silicon Valley



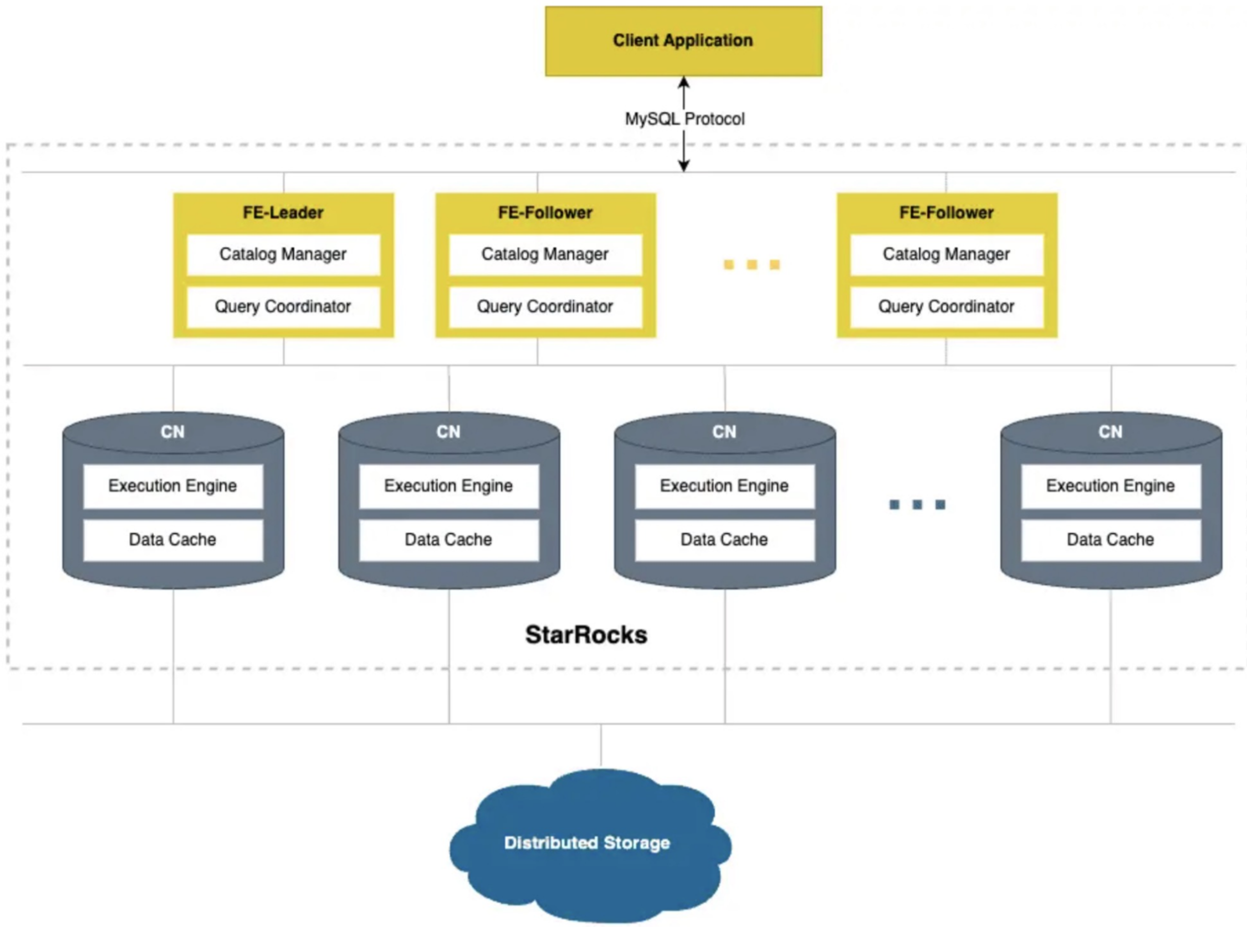
Separation of Storage and Compute - Leader

IEEE-CNSV
Consultants' Network of Silicon Valley



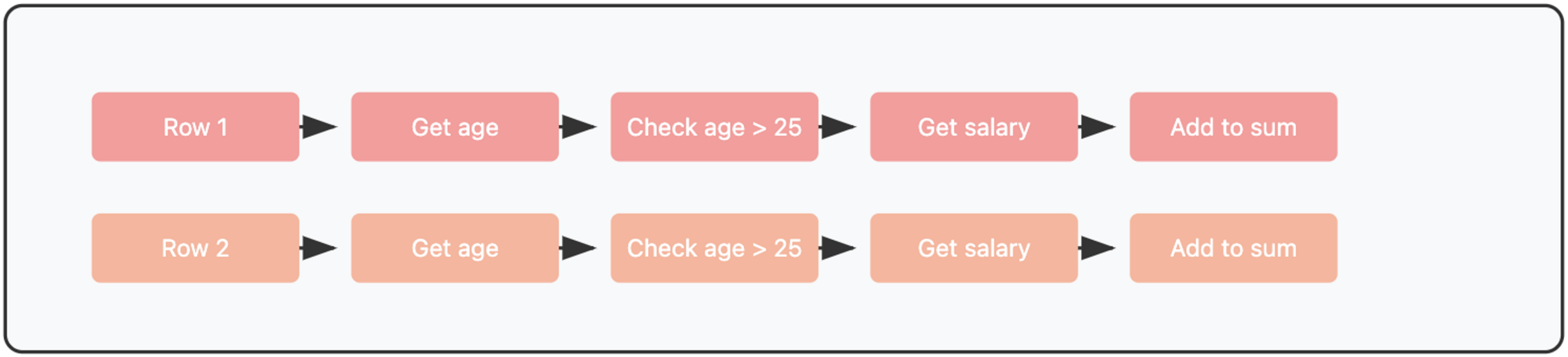
Additional EBS Cache

IEEE-CNSV
Consultants' Network of Silicon Valley



IEEE-CNSV
Consultants' Network of Silicon Valley

Basic SQL Execution (Row-by-Row Interpretation)



Problems:

✕ Function Call Overhead | ✕ Interpretation Cost | ✕ Poor CPU Cache Usage | ✕ One Value at a Time

SQL Query Compilation Example

Original SQL Query

```
SELECT SUM(salary)
FROM employees WHERE age > 25
```

Generated Machine Code

```
double sum = 0;
while (age_ptr < end_ptr) {
if (*age_ptr > 25) { sum += *salary_ptr; }
age_ptr++; salary_ptr++;
}
```

No Interpretation

✓ Direct age comparison
✓ No function overhead
✓ Native CPU instructions

CPU Optimization

✓ Predict age > 25 branch
✓ Pipeline addition
✓ Optimize pointer increments

Memory Access

✓ Direct array access
✓ Sequential scanning
✓ Efficient prefetching

Limitation: Processes one age/salary pair at a time (Can be improved with Vectorization)

Vectorized Processing with SIMD

SELECT SUM(salary) FROM employees WHERE age > 25

Process 8 values at once using SIMD

↓

Column Data in Memory

[23, 28, 35, 19, 45, 22, 31, 27]
Age Vector (8 values)

[45k, 55k, 75k, 40k, 85k, 42k, 65k, 52k]
Salary Vector (8 values)

↓

SIMD Operations (Single Instruction, Multiple Data)

```
1. __m256i ages = _mm256_load_si256(age_ptr); // Load 8 ages
2. __m256i mask = _mm256_cmpgt_epi32(ages, 25); // Compare 8 values
3. __m256d salaries = _mm256_maskload_pd(salary_ptr, mask); // Load matching
4. sum = _mm256_add_pd(sum, salaries); // Add 8 values at once
```

↓

Vectorization Benefits

✓ Process 8 values per instruction | ✓ Better Cache Usage | ✓ Fewer CPU Cycles | ✓ ~8x Speedup

IEEE-CNSV
Consultants' Network of Silicon Valley

Vectorized Compilation

Combining SIMD Operations with Compiled Code

SELECT SUM(salary) FROM employees WHERE age > 25

Compile once, process 8 values per iteration

Compiled Vectorized Code

```
// Compiled once to native code with SIMD
void process_batch(int* age_ptr, double* salary_ptr, int count) {
__m256d sum = _mm256_setzero_pd(); // Vector accumulator
while (count >= 8) {
__m256i ages = _mm256_load_si256(age_ptr); // SIMD load
sum = _mm256_add_pd(sum, process_vector(ages)); // SIMD ops
}
```

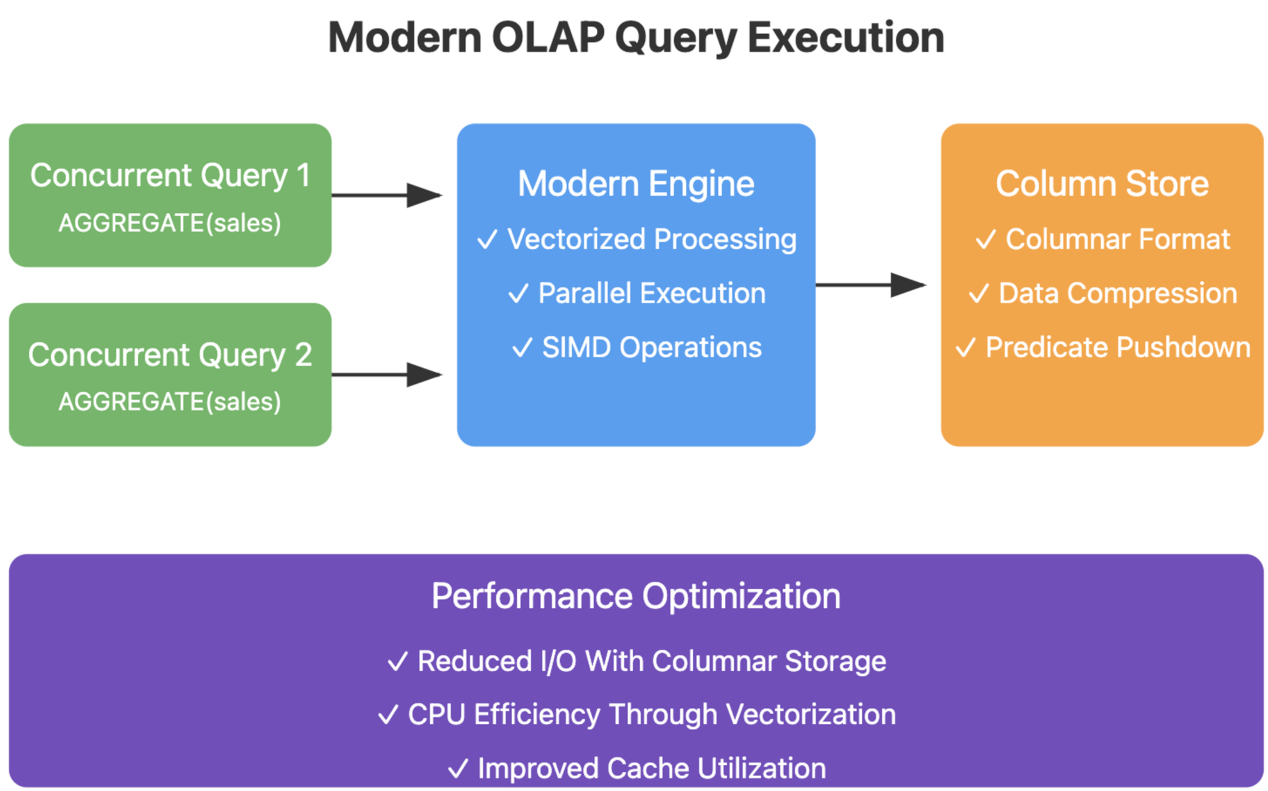
Compilation Benefits
- ✓ Direct Machine Code
- ✓ No Function Overhead
- ✓ CPU Optimization

SIMD Benefits
- ✓ 8 Values Per Cycle
- ✓ Parallel Processing
- ✓ Hardware Acceleration

Combined Power
- ✓ Best Performance
- ✓ Optimal Cache Usage
- ✓ Maximum Throughput

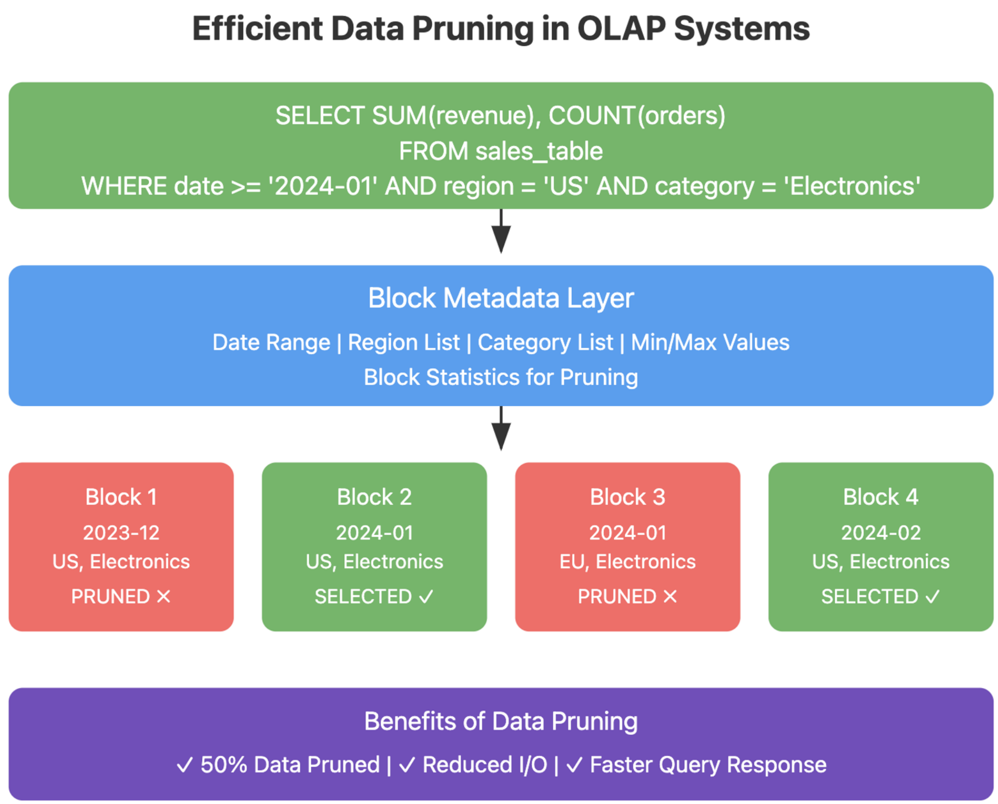
Combines compilation efficiency with SIMD parallel processing for maximum performance

Shared IO


Modern OLAP Query Execution
Concurrent Query 1
AGGREGATE(sales)
Concurrent Query 2
AGGREGATE(sales)
Modern Engine
✓ Vectorized Processing
✓ Parallel Execution
✓ SIMD Operations
Column Store
✓ Columnar Format
✓ Data Compression
✓ Predicate Pushdown
Performance Optimization
✓ Reduced I/O With Columnar Storage
✓ CPU Efficiency Through Vectorization
✓ Improved Cache Utilization

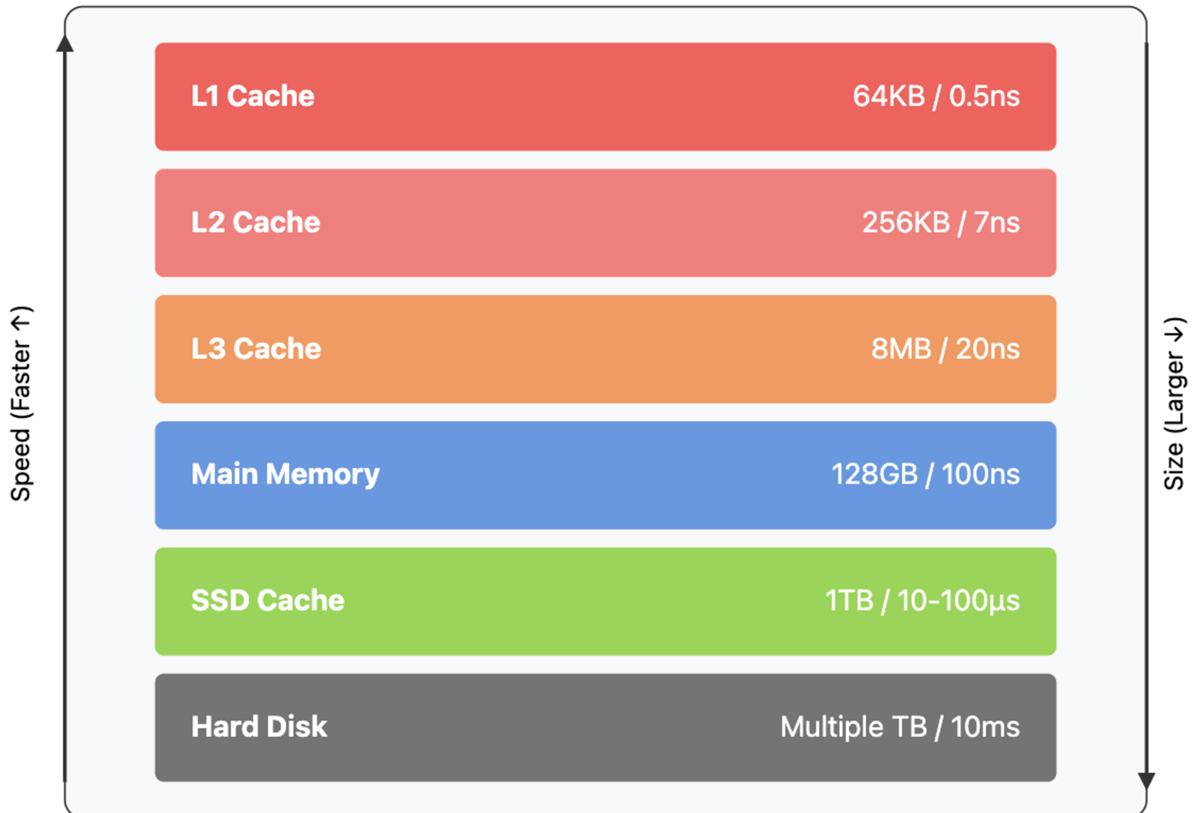
Data Pruning

IEEE-CNSV
Consultants' Network of Silicon Valley



IEEE-CNSV
Consultants' Network of Silicon Valley

Cache Hierarchy



* Times shown are approximate latency for single operation

Evolving OLAP Systems and Analytics Techniques

IEEE-CNSV
Consultants' Network of Silicon Valley

- **OLAP System Evolution**
  OLAP systems have evolved from traditional data warehouses to data lakes and lake houses, providing more flexibility and scalability for data storage and analysis.
- **Unified Analytics Platform**
  The need for a single system to power all use cases, from customer-facing analytics to conversational analytics, has become increasingly important.
- **Customer-Facing Analytics**
  Customer-facing analytics has become very popular, allowing businesses to gain valuable insights and make data-driven decisions.
- **Increased Concurrency Needs**
  The rise of conversational analytics has led to increased concurrency needs, requiring more efficient techniques to handle the increased workload.
- **Techniques Explored**
  We looked into three techniques to address the challenges: avoiding single leader (queues), data pruning, and reduced I/O and vectorized execution.

IEEE-CNSV
Consultants' Network of Silicon Valley

Time for a Break

AI Agents: Next Generation of Autonomous Systems

IEEE-CNSV
Consultants' Network of Silicon Valley

WHAT ARE AI AGENTS?

Autonomous software systems that can perceive their environment, make decisions, and take actions to achieve specific goals with minimal human intervention.

KEY CAPABILITIES

Perception
Understanding context and environmental data

Reasoning
Logical decision-making and problem solving

Action
Executing tasks and adapting to feedback

REAL-WORLD APPLICATIONS

- Personal Assistants
- Code Generation
- Data Analysis
- Process Automation

Note: AI agents represent a fundamental shift in how we interact with and leverage artificial intelligence systems.

Investment in Agentic Applications Utilizing LLMs

MARKET GROWTH

$6.4B → $36.1B
2024 to 2030 | CAGR 33.2%

ENTERPRISE ADOPTION

58% Experimental
23% Commercial

INVESTMENT TRENDS

2023 Total Funding
$11.6B

OpenAI Leading Investment
$13B

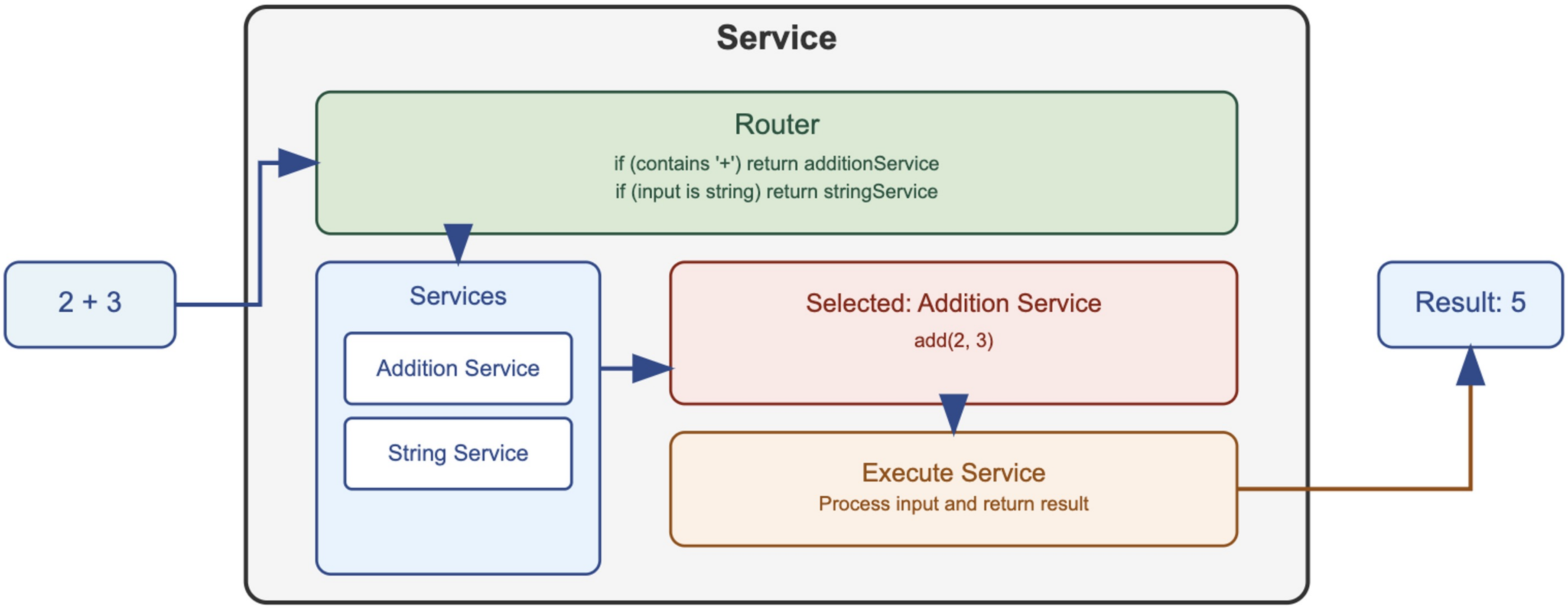
KEY INDUSTRIES

- Finance
- Healthcare
- Customer Service

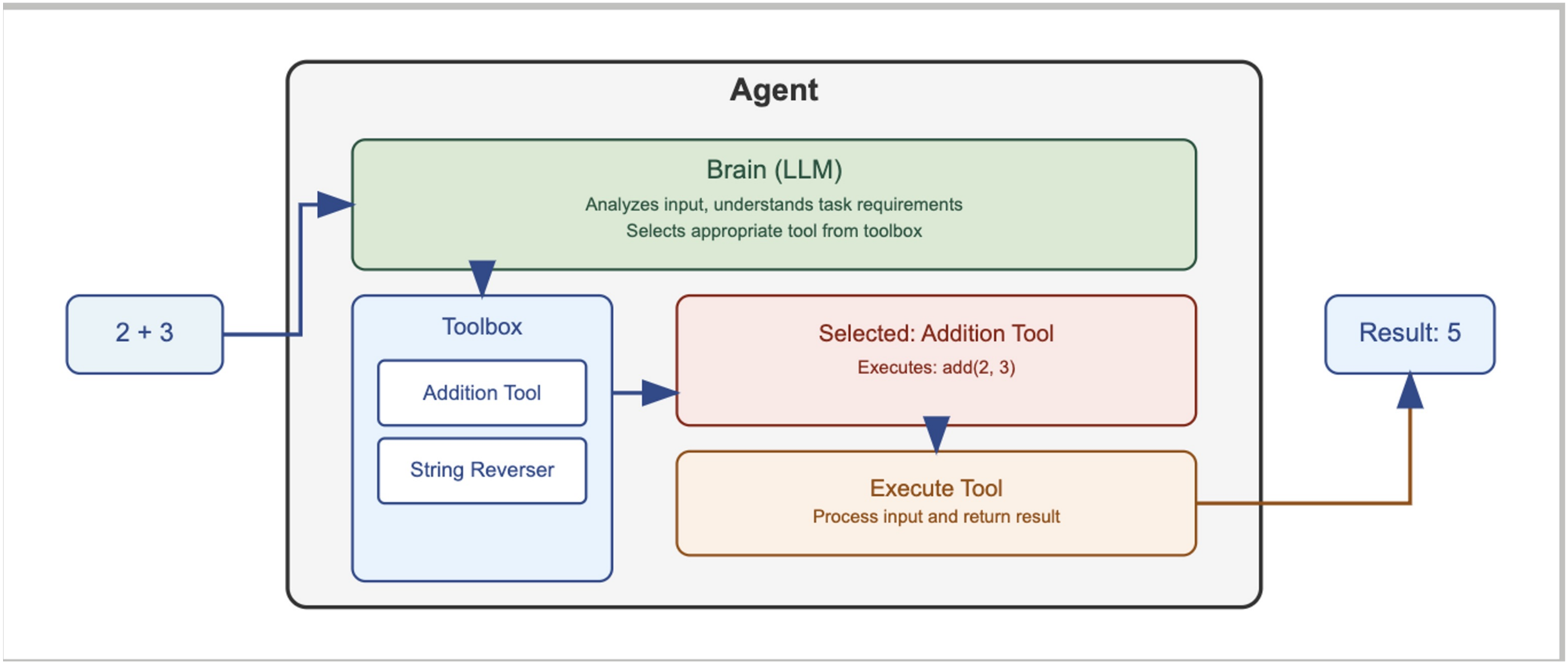
Note: The increasing investment and adoption rates underscore a significant interest in developing agentic applications leveraging LLMs.

Traditional Microservice

IEEE-CNSV
Consultants' Network of Silicon Valley

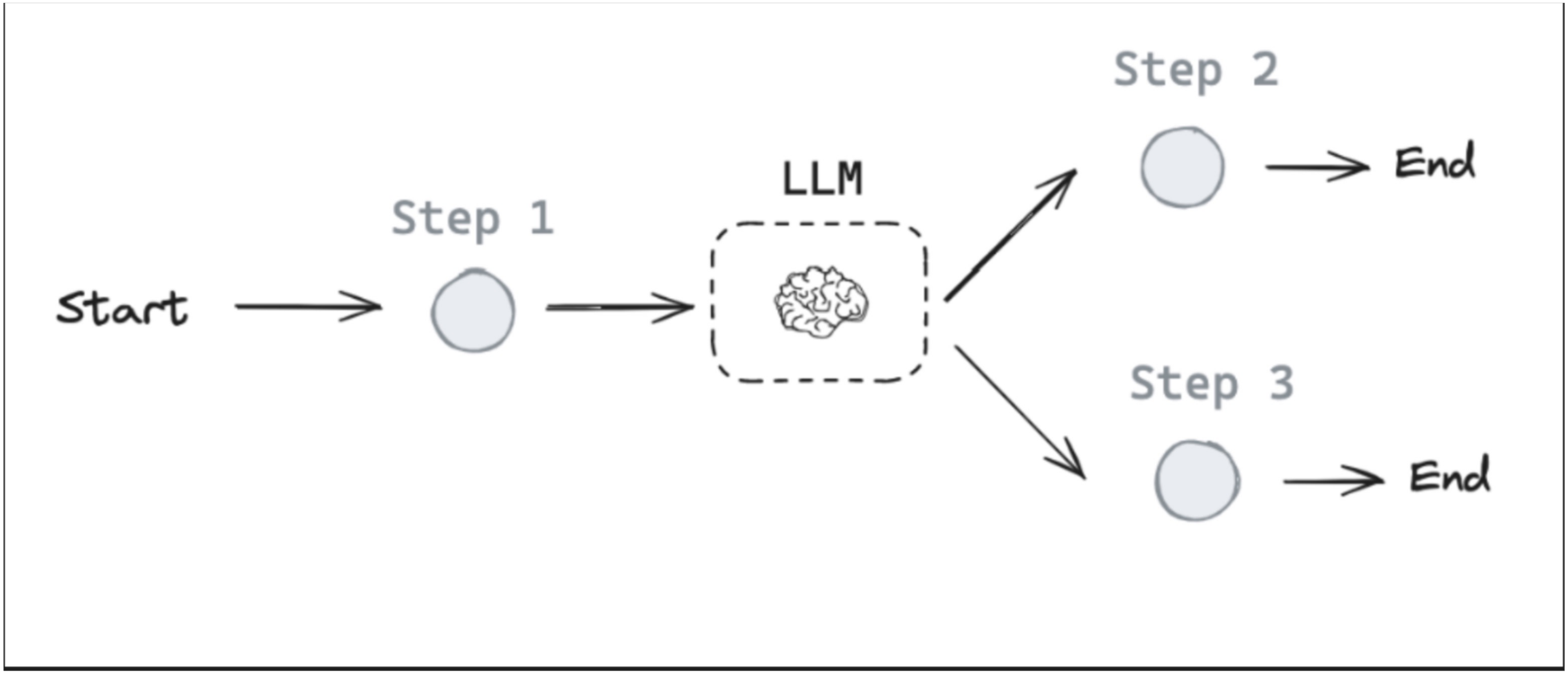


AI Agents



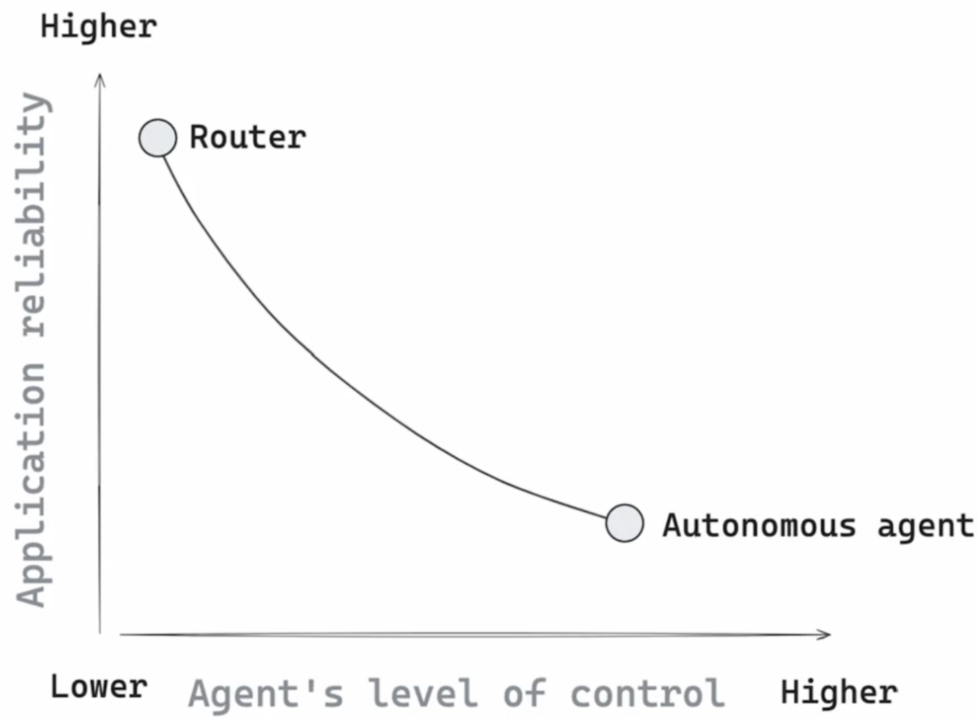
Agent
Brain (LLM)
Analyzes input, understands task requirements
Selects appropriate tool from toolbox
2 + 3
Toolbox
Addition Tool
String Reverser
Selected: Addition Tool
Executes: add(2, 3)
Execute Tool
Process input and return result
Result: 5

AI Agents



IEEE-CNSV
Consultants' Network of Silicon Valley

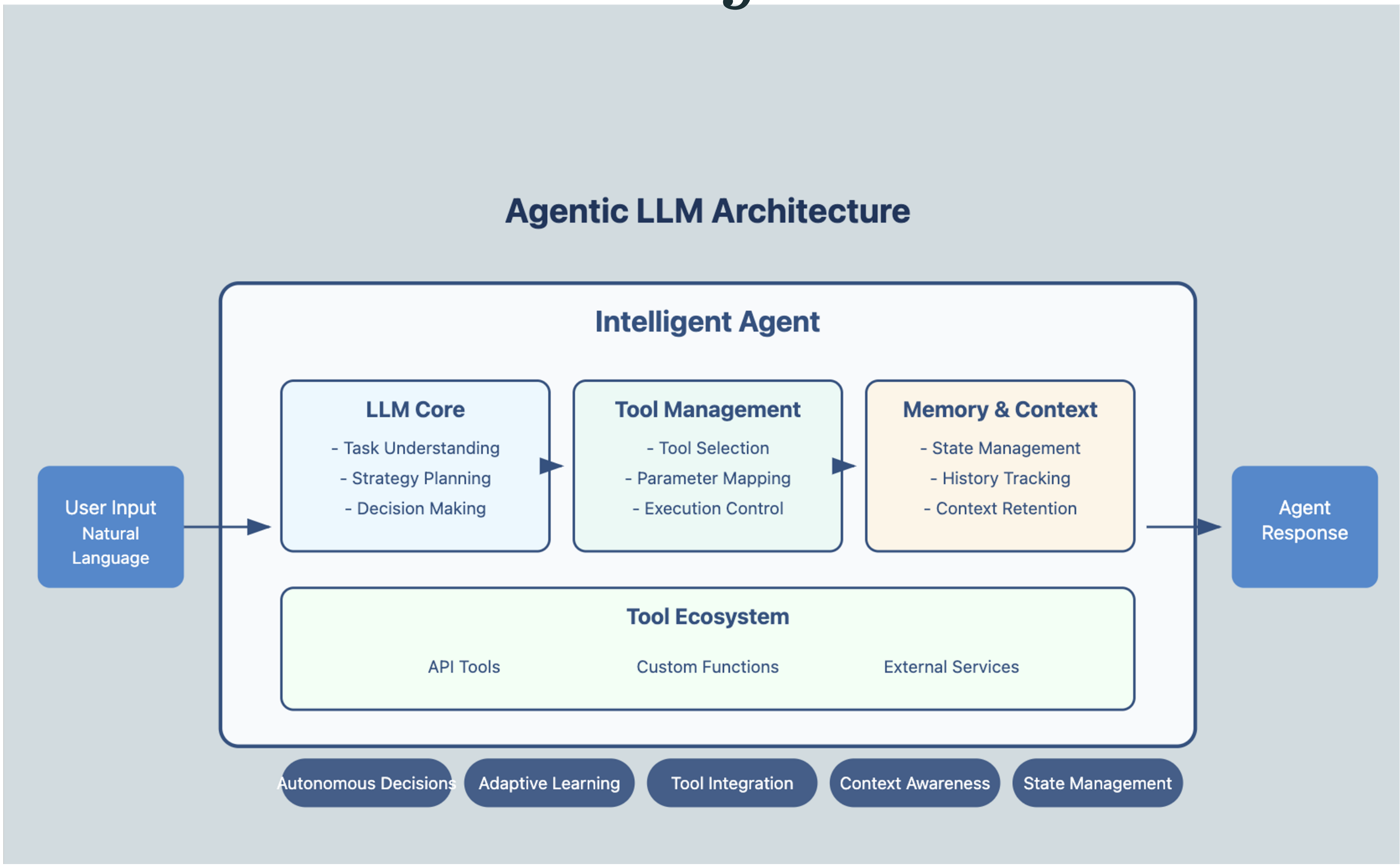
But, practical challenges.



Demo

IEEE-CNSV
Consultants' Network of Silicon Valley

AI Agents



Agentic LLM Architecture
Intelligent Agent
User Input
Natural
Language
LLM Core
- Task Understanding
- Strategy Planning
- Decision Making
Tool Management
- Tool Selection
- Parameter Mapping
- Execution Control
Memory & Context
- State Management
- History Tracking
- Context Retention
Agent
Response
Tool Ecosystem
API Tools
Custom Functions
External Services
Autonomous Decisions
Adaptive Learning
Tool Integration
Context Awareness
State Management

IEEE-CNSV
Consultants' Network of Silicon Valley

LangChain Core Abstractions

Models
- LLMs
- Chat Models
- Text Embedding Models

Prompts
- Prompt Templates
- Few-Shot Examples
- Output Parsers

Memory
- Conversation Buffer
- Vector Stores
- Summary Memory

Chains
- Sequential Chains
- Transform Chains
- Router Chains

Agents
- ReAct Agents
- Plan-and-Execute
- OpenAI Functions

Composability | Modularity | Extensibility | State Management

IEEE-CNSV
Consultants' Network of Silicon Valley

LangGraph Core Abstractions

Models
- State Graphs
- Graph Functions
- Node Types

Graph Elements
- Nodes
- Edges
- Graph Configs

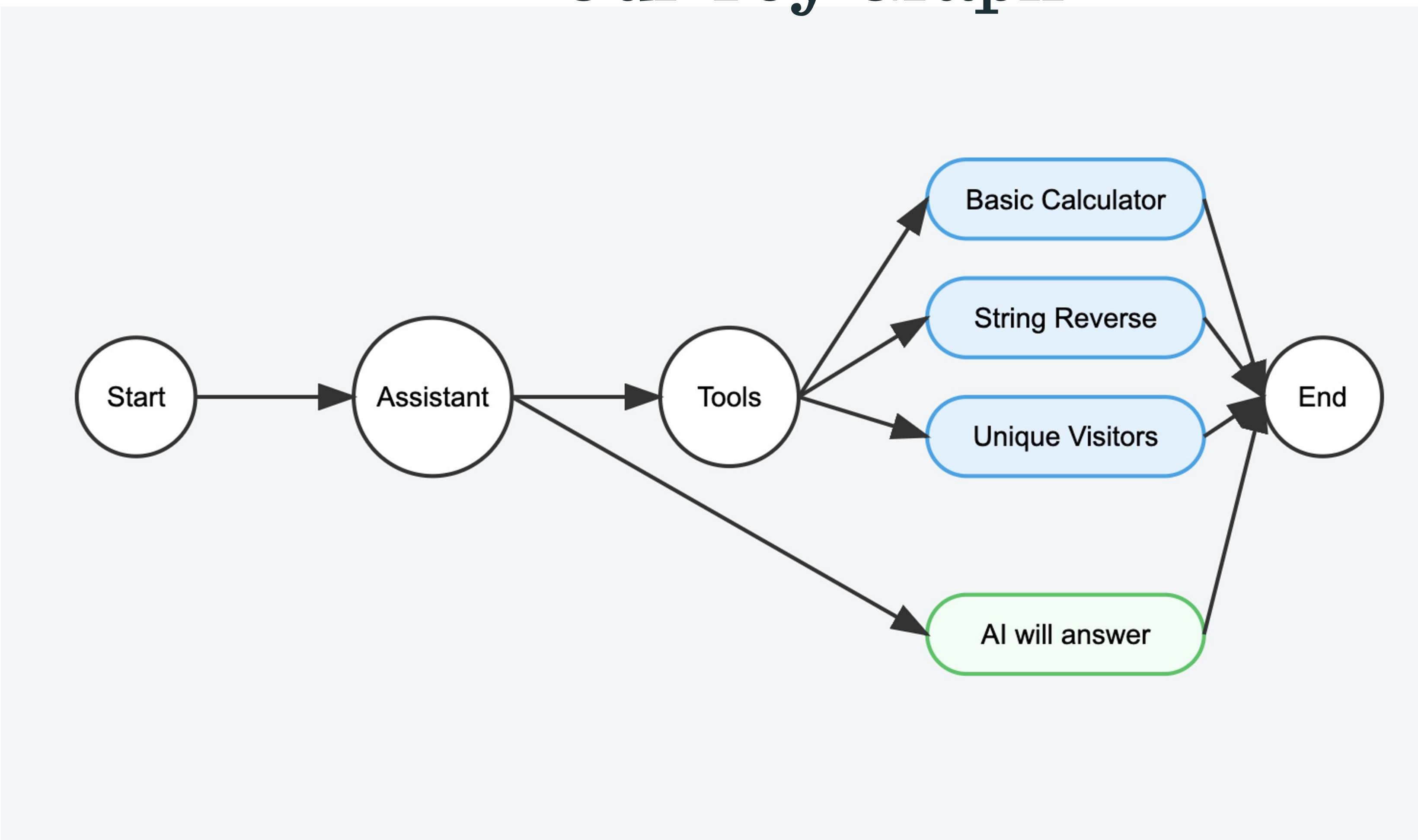
States
- State Management
- State Transitions
- State Updates

Execution
- Graph Traversal
- Conditional Routing
- Cycle Detection

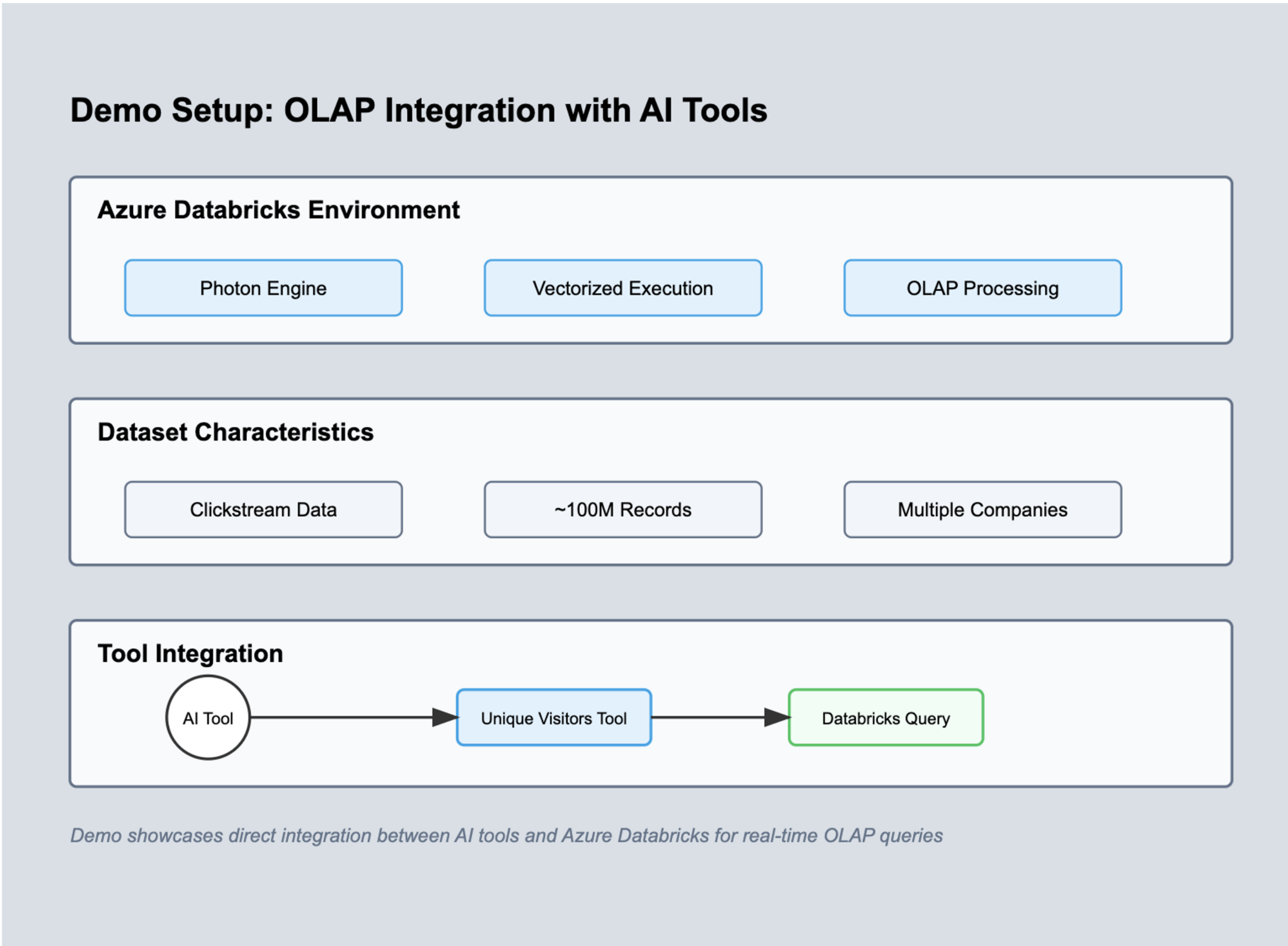
Flow Control
- Branching Logic
- Error Handling
- Async Processing

Graph Construction | Validation | Execution | Monitoring

Our Toy Graph



Demo



IEEE-CNSV
Consultants' Network of Silicon Valley

IEEE-CNSV
Consultants' Network of Silicon Valley

Summary Slide: OLAP Systems and Modern Analytics

OLAP System Evolution

Discussed the evolution of OLAP systems and the increased spend on these systems.

Increased Use of Analytics

Identified the increased use of in-product analytics and conversational analytics.

Concurrency as a Priority

Highlighted that solving concurrency is a top priority for OLAP systems.

Performance Techniques

Explored techniques to make OLAP systems faster.

AI Agents and Agentic Architecture

Looked at AI agents and agentic architecture as a modern industry paradigm.

Toy Agents with and without LangChain/LangGraph

Demonstrated building toy agents with and without LangChain/LangGraph.

Conversational Analytics Use Case

Presented a use case of conversational analytics to put the concepts in perspective.

IEEE-CNSV
Consultants' Network of Silicon Valley

Thank You