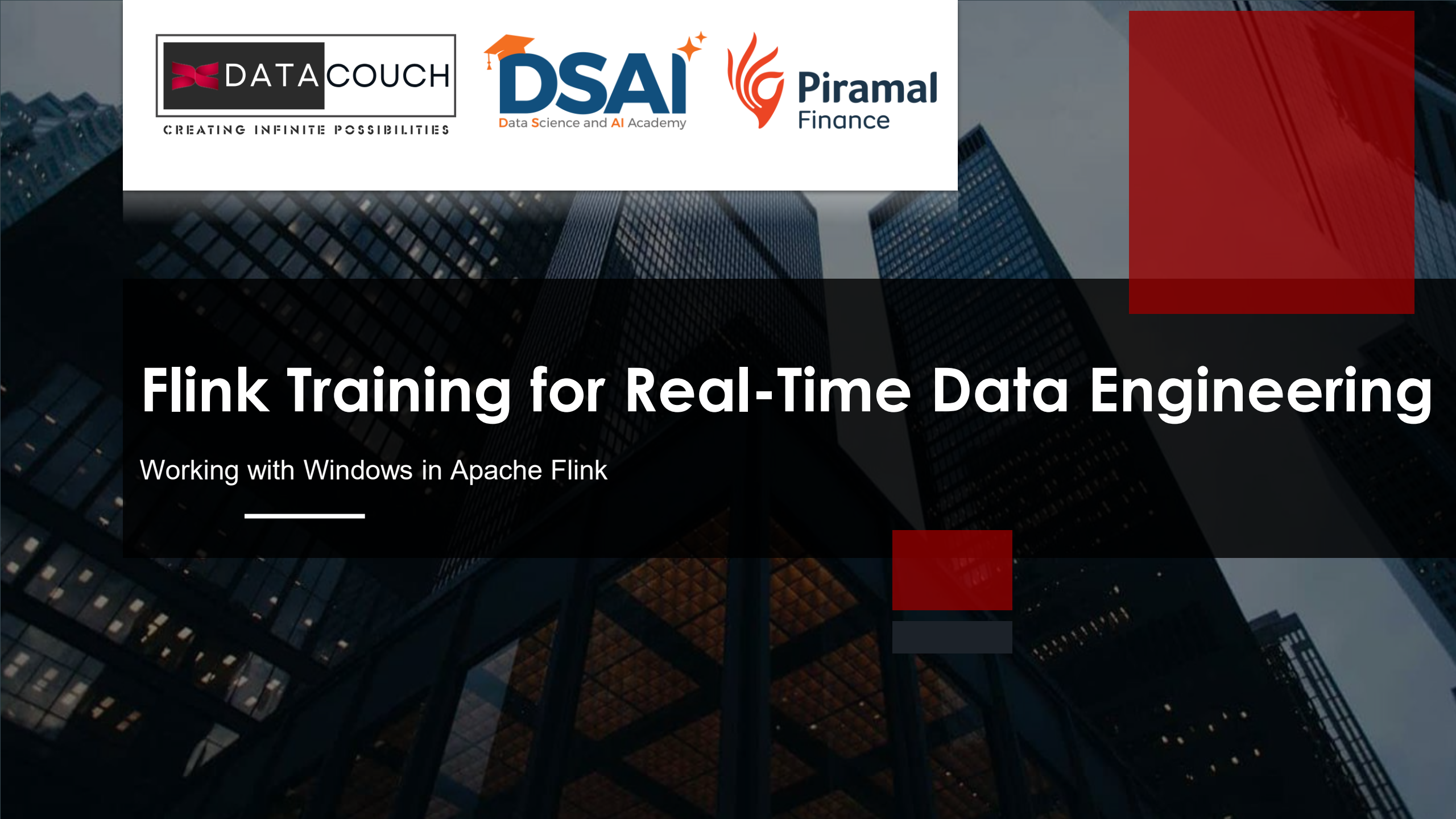
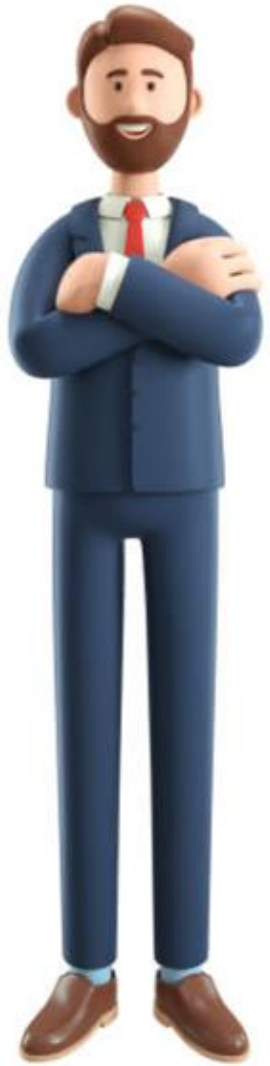

DATACOUCH
CREATING INFINITE POSSIBILITIES
DSAI
Data Science and AI Academy
Piramal
Finance
Flink Training for Real-Time Data Engineering
Working with Windows in Apache Flink



AGENDA

- Why "state" is critical in real-time applications (e.g., aggregations, fraud detection).
- Understanding Flink's state backends (e.g., RocksDB).
- Ensuring reliability with Checkpoints and Savepoints for no-data-loss guarantees.

Why State Is Critical in Real-Time Stream Processing

Memory Across Events

Real-time applications like fraud detection and session tracking must remember past events to make accurate decisions about current ones.

Continuous Intelligence

Counting unique users or detecting suspicious patterns requires maintaining evolving state across millions of streaming events.

Beyond Single Events

Without state, operations become stateless and unable to track trends, limiting analytics to isolated event views.

State-Enabled Use Cases

- Fraud detection patterns
- User session analytics
- Real-time aggregations
- Trend monitoring

The Power of Stateful Stream Processing

Real-time applications like fraud detection and aggregations rely on remembering past events — this is "state".

Apache Flink treats state as a first-class citizen, transforming how we process continuous data streams.

Fault Tolerance

Consistent checkpoints and savepoints ensure your applications recover gracefully from failures without losing critical data.

Exactly-Once Processing

Guarantees no data loss or duplication, maintaining data integrity even during system failures or network issues.

Terabyte-Scale State

Efficient handling of massive, complex state through incremental snapshots and optimised storage mechanisms.

Fault Tolerance & Exactly-Once Guarantees

01

Asynchronous Checkpointing

Flink periodically snapshots operator state and source offsets, storing them durably in distributed file systems.

02

Seamless Recovery

On failure, Flink restores the latest checkpoint and replays events to resume processing exactly where it left off.

03

Zero Data Loss

Critical applications maintain consistent, accurate state despite machine or network failures—no duplication or loss.

State Backend Power

RocksDB enables scalable, fault-tolerant state storage, supporting large, complex state with minimal performance impact for business-critical streaming applications.

What is State in Apache Flink?

Memory of Past Events

Stateful stream processing remembers past events to influence future computations, enabling complex analytics and windowing operations.

Local Storage & Snapshots

State is stored locally during processing and periodically snapshotted for fault tolerance, ensuring no data loss during failures.

Backend Determines Storage

State backends determine how and where this critical state information is stored and checkpointed across the cluster.

Benefits and Trade-offs of RocksDB Backend

Benefits

Massive Scale

Handles very large state sizes that extend far beyond JVM heap memory limits.

Efficient Checkpoints

Incremental checkpointing significantly improves performance and reduces system overhead.

Safe Object Reuse

Enables safe object reuse due to proper serialisation, preventing data corruption issues.

Trade-offs

CPU & IO Overhead

Higher CPU and IO overhead due to continuous serialisation and deserialisation processes.

Throughput Impact

Lower maximum throughput compared to heap-based backends due to disk access patterns.

Size Limitations

RocksDB JNI imposes a maximum limit of 2GB for individual keys and values.

Summary: RocksDB Powers Scalable Stateful Streaming

Unlocks Large-Scale Management

RocksDBStateBackend enables large-scale, fault-tolerant state management that transcends JVM memory limitations entirely.

Balanced Performance Trade-off

Expertly balances durability and scalability requirements with acceptable performance costs for most production workloads.

Production-Grade Essential

Absolutely essential for production-grade Flink applications that handle massive state volumes and require reliable operation.

Choose Wisely

Selecting the optimal backend depends entirely on your specific state size, latency requirements, and throughput performance needs.

Why No Data Loss Matters

Data loss can devastate businesses: lost revenue, damaged reputation, and compliance failures create ripple effects across entire organisations.

Real-time systems like fraud detection, IoT monitoring, and AI training demand continuous uptime and absolute accuracy to function effectively.

Checkpoints and savepoints are key tools to guarantee data integrity and fault tolerance in modern distributed systems.

What Are Checkpoints?

The System's "Save Button"

Periodic Snapshots

Automatic captures of application state and processed data positions at regular intervals

Recovery Enabled

Enable fast recovery from unexpected failures without losing processing progress

System Managed

Automatically handled by the system for lightweight, efficient recovery operations

Example: Apache Flink's checkpoints ensure exactly-once processing semantics in streaming applications

Savepoints: Planned, User-Controlled Snapshots

Manual Creation

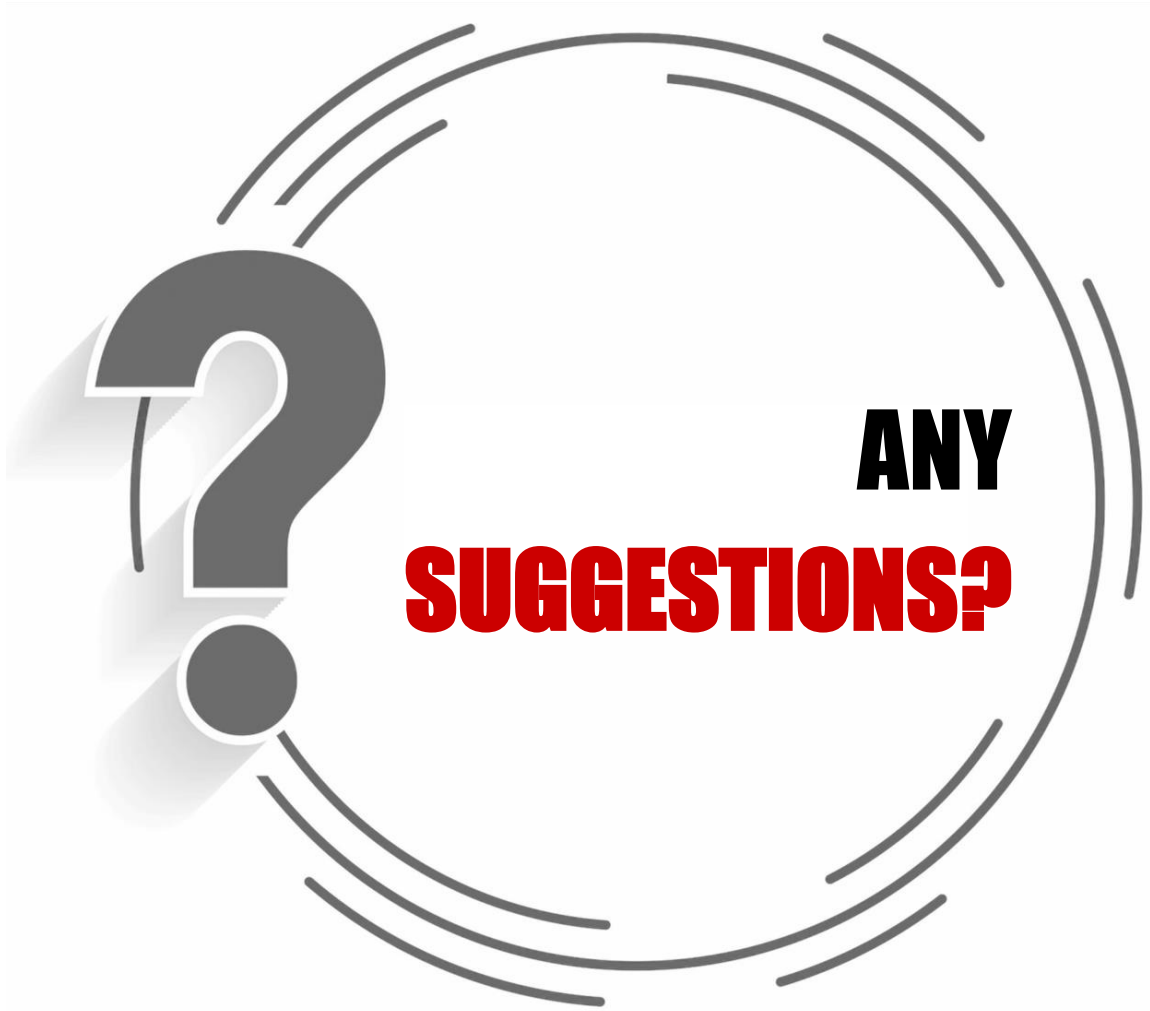
Created manually for operational tasks like upgrades, scaling, or maintenance windows

Flexible & Portable

More portable than checkpoints, supporting schema evolution and job configuration changes

Persistent Storage

Persist beyond job termination until explicitly deleted by users or administrators



ANY SUGGESTIONS?

DATACOUCH
CREATING INFINITE POSSIBILITIES

DSAI
Data Science and AI Academy

Piramal Finance


THANK YOU