

# Data Sync Pro

## #Batch: Bulk Data Processing

---

Replace Apex batches, enrichments, migrations, syncs, archives, and more with a declarative engine — processing millions of records up to 20× faster, natively in Salesforce.



# The Batch Job Management Challenges

1

## Slow, Manual Job Creation

Building batch jobs from scratch demands extensive custom code—consuming weeks of developer time for each new process and making repeatable delivery nearly impossible.

2

## Poor Batch Performance

Standard batch processing is slow and resource-heavy—large volumes take hours or days, tying up system resources and delaying downstream processes.

3

## No Native Job Chaining

Salesforce lacks built-in support for running batches in sequential order—teams resort to fragile workarounds that consume scheduled job slots and risk hitting governor limits.

4

## Opaque Job Scheduling

Scheduling batch Apex often requires cron expressions, with no easy way to view which job runs which process, pause, reschedule, or get a holistic view of all scheduled jobs.

5

## Limited Transformations

Reshaping, enriching, or validating data during batch processing demands custom Apex for every scenario—expensive to build, hard to maintain, and impossible to standardize.

6

## No Granular Logging

Most batch frameworks offer minimal visibility—no record-level detail, no easy way to isolate failures, and no built-in audit trail for troubleshooting, re-execution, or compliance.

# How Data Sync Pro Solves Batch Job Management

1

## Rapid Job Creation

Build batch jobs in a fraction of the time—95% faster than traditional approaches. Creating a batch job is as simple as creating records—no custom code required.

2

## 20× Faster Execution

Run batch jobs 20× faster with an optimized execution engine that maximizes throughput while staying within governor limits—dramatically cutting processing time.

3

## Job Chaining

Chain batch jobs to run in sequential order—eliminating fragile workarounds and freeing up scheduled job slots to avoid hitting governor limits.

4

## Definitive Job Scheduling

Schedule, pause, reschedule, change users, and monitor all batch jobs through simple CRUD on Schedule records—no cron expressions, full visibility across all jobs.

5

## Rich Function Library

Leverage 170+ built-in functions—covering logics, strings, dates, math, lookups, regex, roll-ups, masking, and more—and extend with Apex for tailored transformations.

6

## Record-Level Logging

Each batch has its own log with detailed record-level information—enabling fast troubleshooting, targeted re-execution, data validation, and full compliance auditability.

# The DSP Architecture Advantage

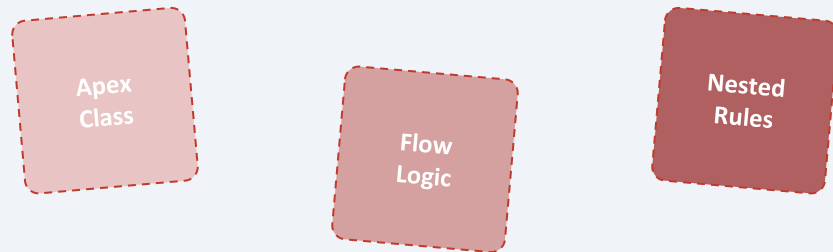
Enforced modularity that eliminates technical debt and scales with the enterprise

## Traditional Approach

- ✗ Apex and Flows create nested, tangled logic within each process
- ✗ No enforced boundaries—one change can break downstream processes
- ✗ Technical debt compounds over time, making the system fragile and hard to scale
- ✗ Each solution is unique—no consistent structure across automations

## The DSP Way

- ✓ Every Executable has clear **Input → Transformation → Output**
- ✓ Record-based rules keep each module well-bounded and self-contained
- ✓ Enforced modularity eliminates technical debt and scales with the enterprise
- ✓ Every Executable follows the same consistent process—no unique snowflakes



↑ *Technical debt compounds over time*



# DSP Batch Features

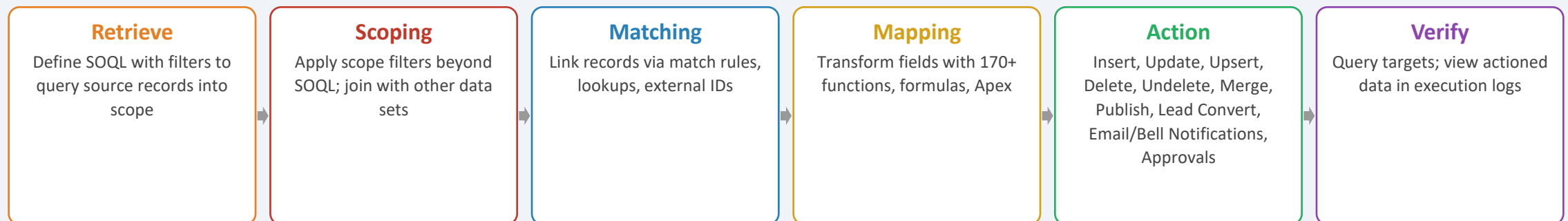
## Core Capabilities

- **Rapid Configuration** — 95% less effort: just create records with simple, declarative rules—each process is a modular, self-contained unit with no custom code and no hidden dependencies.
- **20× Faster Execution** — Optimized engine maximizes throughput within governor limits.
- **Chained Batch Orchestration** — Sequenced execution, enforced dependencies, fewer scheduled jobs.
- **Definitive Job Scheduling** — Manage schedules as simple records: create, pause, reschedule, reassign, and monitor all jobs with full visibility.
- **Rich Function Library** — 170+ built-in transformations + extensible Apex.
- **Complete Execution Transparency** — Per-batch logs with record-level detail for troubleshooting, re-execution, validation, and compliance.

## Additional Features

- **Bulk API**
- **Cross-Org Connections**
- **Incremental Retrieve**
- **Delta Update**
- **Execute on Demand**
- **Selective Execution**
- **Notification on Completion**
- **Serial Mode**
- **Auto-Retry When Failed**
- **Preview Transformation Before Execution**
- **Source Writeback**

**Batch Execution Engine** — Every batch job follows a powerful six-stage process



# Common Use Cases

1

## Apex Batch Replacement

Run large-scale Inserts, Updates, Upserts, Deletes, Merges, Lead Conversions, Notifications, and Approvals.

2

## Transformations & Cleansing

Apply formulas, enrich records, normalize data, and deduplicate at scale.

3

## Cross-Org Migration & Sync

Move and consolidate data between Salesforce orgs without exporting externally.

4

## Compliance & Continuity

Archive history, back up data, restore records, and track changes with Big Objects.

5

## Testing & Deployment

Seed sandboxes, mask sensitive data, deploy record-based configs, and stress-test governor limits.

6

## Schedule & Chaining Management

Manage job schedules and chain batches into orchestrated workflows—beyond Salesforce's 100-job limit.

# Security & Trust

*Runs entirely within your Salesforce org — governed by Salesforce's native security model*

## Local Data Processing



All processing runs in Apex inside the installed org. No external systems or third-party services involved.

## Native Security Model



All DSP rules are stored as Executable records, governed by standard object-, field-, and record-level security.

## Managed Cross-Org



Connections to other Salesforce orgs use Named Credentials, fully managed by your admins.



**SOC 2 TYPE II AUDITED** — Independently audited security, availability, and confidentiality controls.

# Why #Batch

100%

## Salesforce Native

No external systems, no data leakage, full compliance

95%

## Less Build Effort

Compared to other tools

20×

## Faster Batch Processing

Optimized execution engine

→0

## Tech Debt

Drive tech debt toward zero with declarative, modular rules

## Zero Risk, Full Compliance

All data is processed inside your Salesforce org—no external systems, no data exposure. AppExchange security reviewed and SOC 2 Type II attested.

## Accelerate Delivery

Build each Executable in minutes—even for the most complicated data models with deep relationships and complex sequencing.

## Peak Efficiency

Dramatically improve processing throughput, reduce resource consumption, and keep your org healthy—also reducing oversight from the team and enabling smoother project execution.

## Modularity Enforced

Every Executable is a bounded record with clear input, logic, and output. Tech debt stops compounding — by design, not discipline.