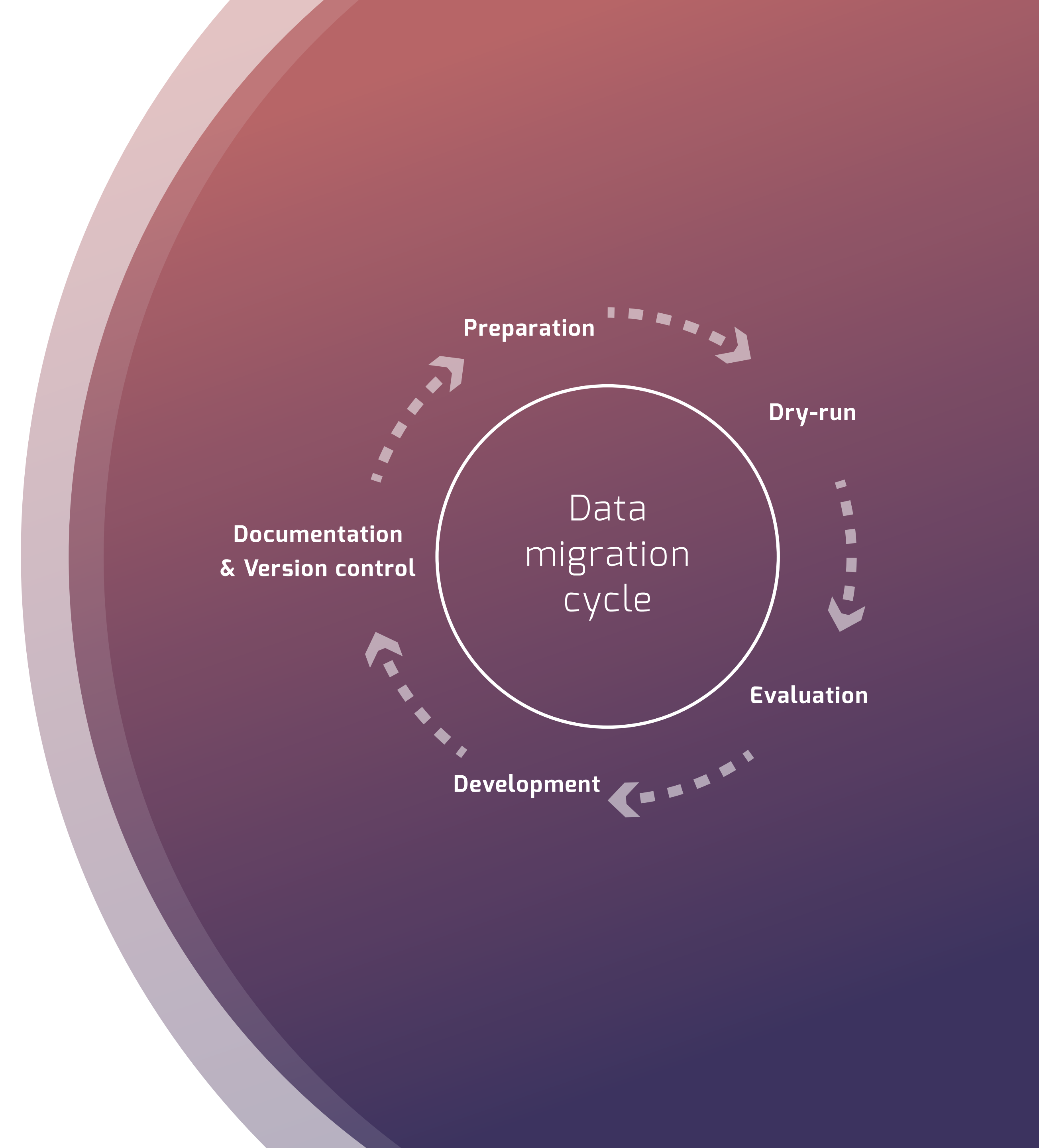


Your survival guide for
DATA MIGRATION
projects

Some experts believe that data migration is simple and straightforward: nothing more than extracting data from old databases, transforming them to the new structure and format, and loading the result into new systems. This is what the ETL tools are for.

Other experts, maintaining longer track record in the jungles and swamps of large and complex data migration projects, had to learn that data migration is rather a cyclic than a straightforward, one-time exercise: to compress the whole process to the desired length – fitting into the available time window of a migration weekend – you need to develop and refine the transformation routines, the environment set up, the validation procedures as well as many other factors.

You need to repeat this cycle which consists of Preparation, Dry-run, Evaluation, Development and Documentation & Version control until you reach the expected results.





A powerful ETL tool can help a lot in the hard days of data migration projects. However even these products can sometimes fall short of your needs. Why?

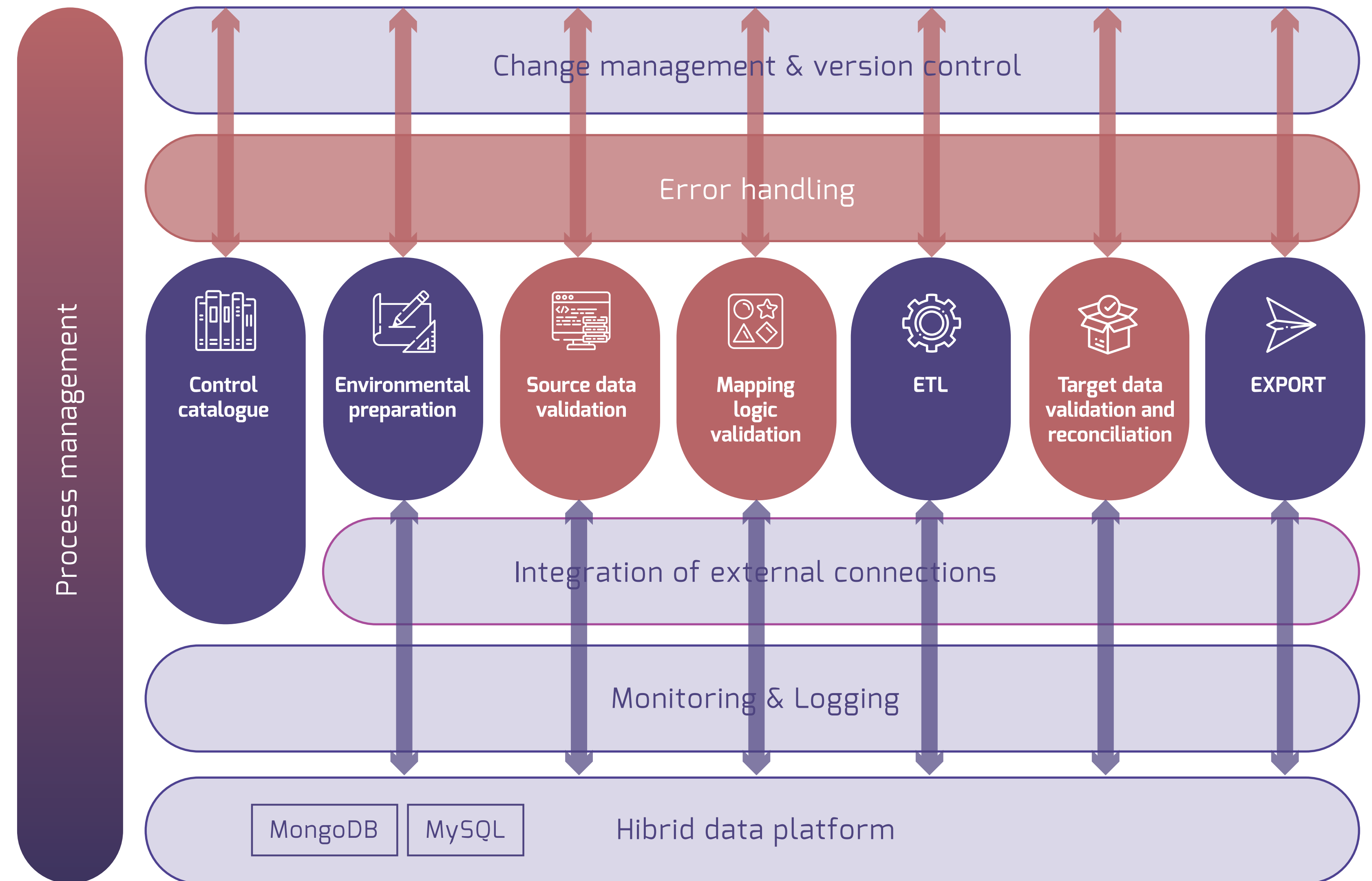
Because ETL tools are focusing just on the core of data migration projects: data extraction, transformation and loading. But what about automated environment handling, automation of repetitive programming work, concealed but powerful version control, pre-checking and early warning mechanisms preventing day-long dry-runs resulting major failures at the end?

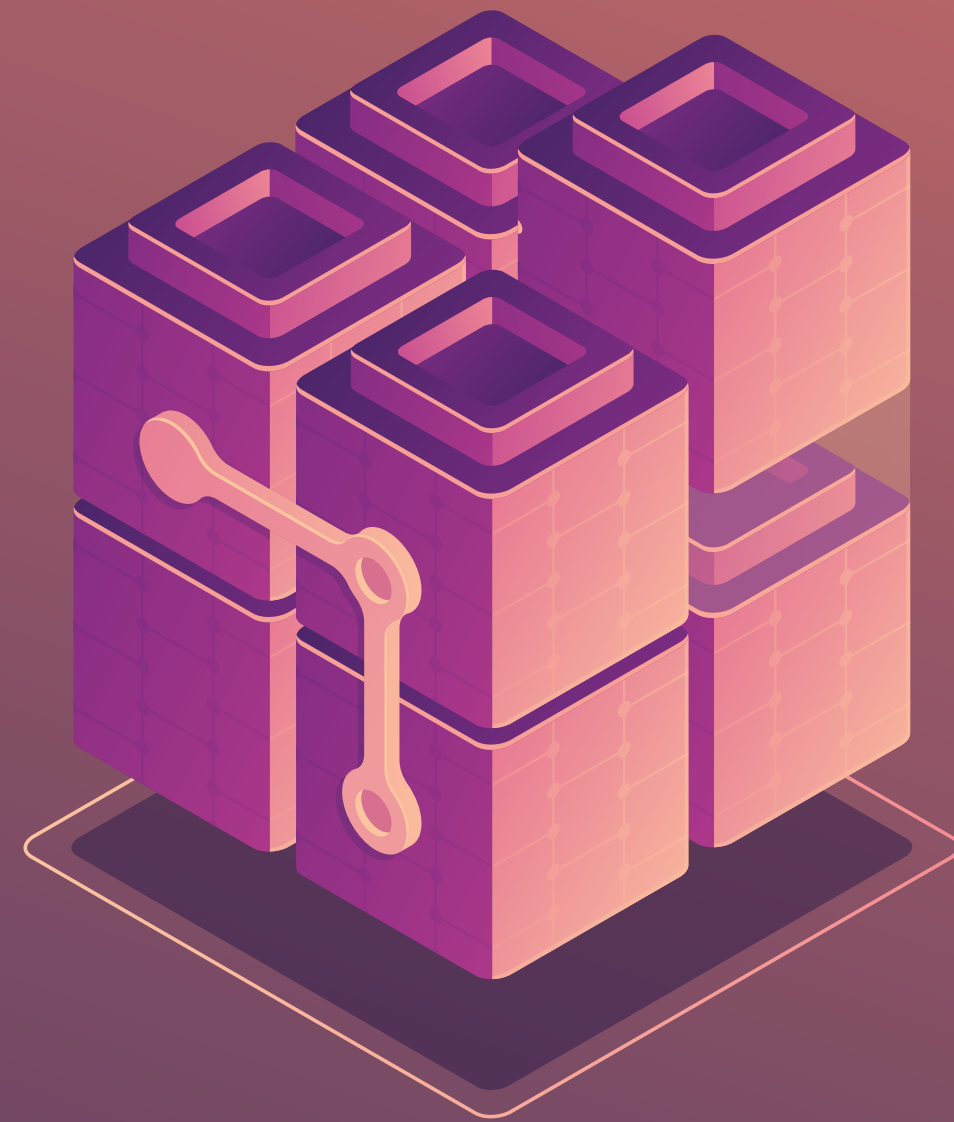
Furthermore, usage of ETL tools mostly requires programming knowledge and database expertise. You often have to give tools into the hands of business users and business analysts to maintain remarkable business domain knowledge necessary for the success.

You also need to make sure that the whole process is transparent for the management, whose members are not interested in the technical aspects. Additionally, you would like to use a framework that supports the teamwork of technical staff by supporting issue tracking, debugging, bottleneck identification and does the boring administration work and allowing you to focus on the most interesting part of the project like planning, mapping and programming.

This is exactly what **Data Migration Assistant** was designed for.

The following figure shows the architecture of this tool:





You can see that the ETL functionality itself is actually a core part of Data Migration Assistant's architecture, but it definitely provides a wider variety of services.

Let's see how Data Migration Assistant can help you during the different stages of the data migration process!

Preparation phase

Automated building of migration-environments:

Common pitfalls

Not documented and not communicated (intentional or undetected) database structure modifications (input or output side)

Trying to execute migration cycles on incorrect data structures

Data Migration Assistant solution

During each run: discarding existing database structures and regenerating the whole environment (structures and interfaces) based on catalogue (meta) information

Extracting legacy data:

Common pitfalls

Starting a migration cycle with incomplete or incorrect data

False alarms due to data-inconsistency

Data Migration Assistant solution

Pre-validation of data structure (expected record counts, etc.) before the dry-run

Dry-run phase

Execution of data migration scripts:

| Common pitfalls | Data Migration Assistant solution |
|---|--|
| Running inappropriate (older or newer) versions of mapping logics/scripts | Integrated version control of migration scripts |
| Inappropriate initiation of procedures (i.e. ignorance of dependencies) | Meta data controlled, scheduled, automatic execution |
| Incomplete run (i.e. skipped items) | |

Monitoring and documenting the migration cycle:

| Common pitfalls | Data Migration Assistant solution |
|--|---|
| No real-time information on the exact status of the execution of a migration cycle | Real-time monitoring of processes and results on a live dashboard; immediate intervention options |
| No exact information on errors and no effective support to localize them | Automated and version-controlled documentation of the whole migration cycle (including migration logic, execution statistics, errors and results) |
| No effective support to investigate errors (e.g. What are the exact differences compared to the previous run?) | Comprehensive logging to enable bottle-neck detection |
| No effective support to identify bottleneck (e.g. due to missing indexes) | Benchmarks to evaluate previous execution statistics |

Evaluation phase

Automated dry-run supervision and post-dry-run checking:

Common pitfalls

No automated quantitative check of results of dry-runs. (e.g. a common error is when a dry-run is executed on an inadequate data set due to the limited coverage of developer tests)

Lengthy analysis of detailed error lists, correcting irrelevant failures.

Gaps when divergent changes of mapping logic and validations happen; this may induce lengthy and iterative analysis with extended correction periods.

Data Migration Assistant solution

Indication of deviation from the expected record number or execution time.

Two-level classification of errors, drill down visualization of errors in the source code, error rate statistics [%].

Version control of validation logic. (e.g. running a migration cycle with predefined script versions, highlighting differences between versions)



Development phase

Development support:

Common pitfalls

Running a full and comprehensive factory test in the development environment may disable other activities like development or testing due to the lack of segmentation.

Lengthy runs and complex process dependencies.

Each business domain is in various stages of completion.

A new version of the migration mapping may be tested on an outdated data structure.

Each version is not historically. (e.g. in a version control system)

In case of changes in the migration data model, all the source codes have to be scanned and adjusted manually if necessary.

Repetitive tasks require massive human resources and often results in a high error rate.

Data Migration Assistant solution

Allows to run the migration of domains separately. (on smaller data sets simultaneously)

Smaller and less complex data sets with easier to understand dependencies.

Automatic data updates in case of structure changes regarding source data.

In case of a successful run, the modified source code is automatically checked-in to a version control system.

In case of changes in data model, the auxiliary functions are modified / adjusted automatically. (e.g. data loading and exporting components etc.)

Step by step execution with well-defined restoration points, decreasing development and testing efforts by automating repetitive tasks.

Documentation & Version control phase

Comprehensive documentation:

Common pitfalls

The exact setup and results of the migration are not accurately documented. This makes the analyzing of errors difficult and resource intensive.

Data Migration Assistant solution

All documentation is generated and stored automatically in a version control system. (e.g. execution time, actual source code of scripts used for mapping and validation, all results and errors)

Comprehensive version control:

Common pitfalls

Errors caused by the use of incorrect versions of legacy and migration data model, mapping and validation scripts are stored and presented in a non-historic way.

Data Migration Assistant solution

All the necessary information can be downloaded from the version control system. Every action or change is logged.

Fully reproducible migration procedure:

Common pitfalls

Complete fallback to previous setups is often not feasible. (due to overlapped development and analysis phases)

Data Migration Assistant solution

The entire migration procedure is fully reproducible based on the actual data structures stored in the version control system.



The Entegrity™ Toolkit

Entegrity™ is a collection of high-performance database management applications. It empowers you to save time and the integrity of data throughout its whole lifecycle.

visit our website