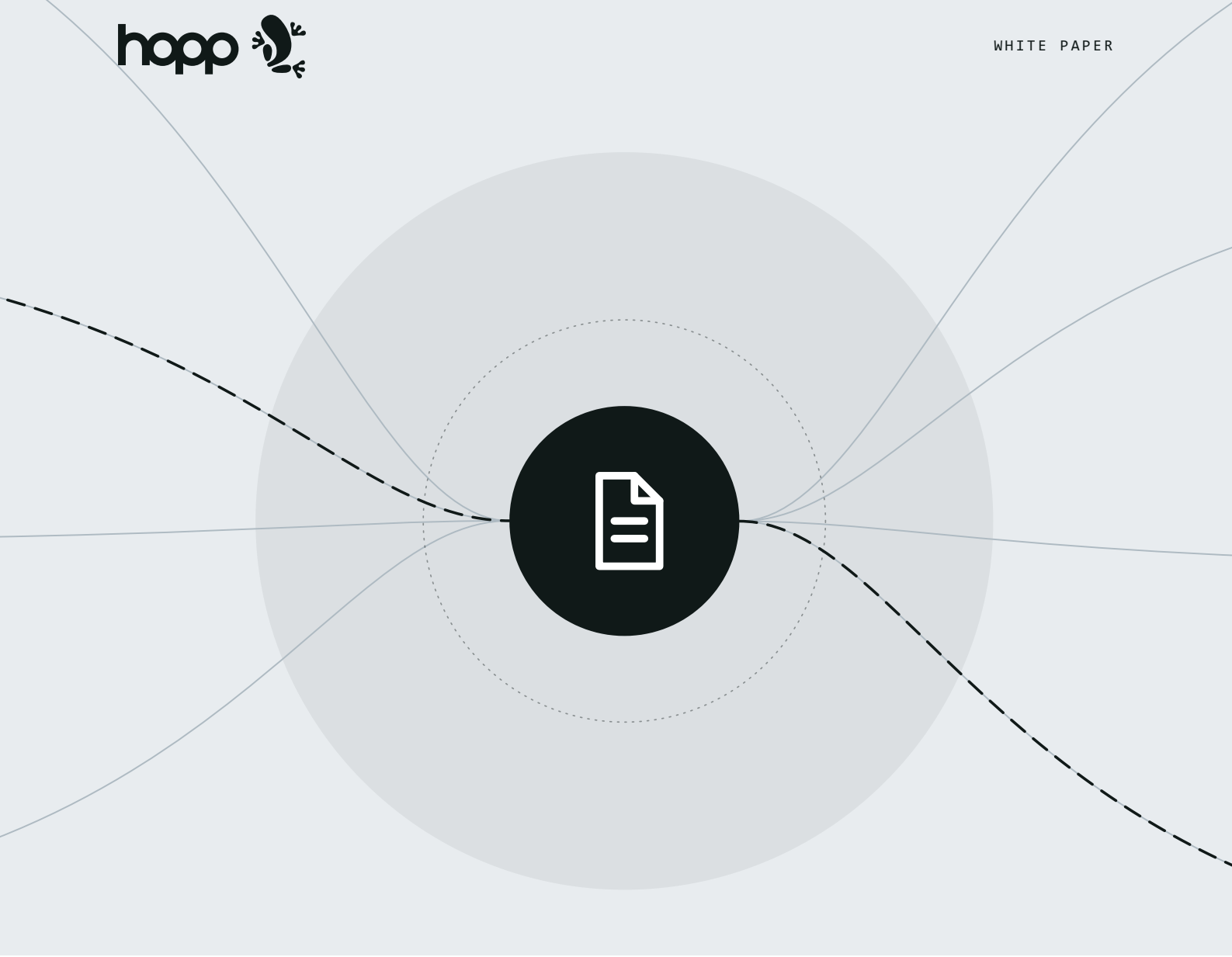
WHITE PAPER



Why Hopp and not ETL

For complex data migration

One challenge in any complex data migration is moving the data from an old system to a new one. The real challenge is transforming the business logic to match the new system.

This paper is about recognizing that what matters most is the connections and dependencies within the data.

We argue for a solution designed for complex data migrations: it saves the time and money of building the migration system yourself, and it builds the quality checks into the result.

01

The problem, and the question

What are the first questions you ask when you face a complex data migration?

- Do I have the right people?
- Do I have the data I need?
- Do I have the right methodology?
- And do I have the right tools?

All of these questions matter, and they are connected. We believe the right answer to each adds knowledge and structure to the project.

A complex data migration can be extensive and, above all, complex. The complexity is in the dependencies, relationships and business rules, for instance when you migrate from a large enterprise legacy system to a new ERP-like system.

In this paper, we explain why you should choose a proven tool that moves business logic in an agile, iterative process, not just tables in defined flows.

02

In summary

Some choose to invest time and effort in building a bespoke data migration solution. But common industry practice is to buy specialized tools for the purpose, rather than replicate something that already exists.

We know from experience across the industry that building a flexible, comprehensive and robust data migration solution takes significant time and resources. Key design choices must be made to get the agile, iterative functionality needed, and some of them are not possible with flow-based ETL tools such as Microsoft SQL Server Integration Services (SSIS).

Many underestimate the task, and end up with poor solutions or outright failures.

In short, we believe that anyone who builds a data migration solution on ETL will:

- build a first-generation solution, instead of buying a mature one;
- spend more time and resources than needed;
- end up with an inferior solution;
- end up with a bespoke setup that depends on key people;
- have no proven, transparent onboarding to offer new customers.

We recommend a dedicated data migration tool like Hopp, instead of starting with in-house development.

> **Hopp: key features**
>
> **An end-to-end solution for complex data migrations:** everything you need to complete one.
>
> **A graphical interface for the whole migration:** in Hopp Studio you map the data and define the business rules and validations; in Hopp Portal you follow the runs and track the events.
>
> **Business objects:** you work in terms of business logic, not tables.
>
> **Dependencies, validations and rules applied consistently:** mapping, rules and validations are applied the same way in the generated code, which keeps errors out of the process.

Before you decide on a path, look at existing solutions like Hopp, and consider what it would take to build something similar or better.

A project on an off-the-shelf solution starts with the migration itself, not with building the system to run it. So it delivers usable output and insight before an in-house design would even be finished.

03

Here are the whys

Where you would have to build a data migration solution on an ETL tool such as SSIS, Hopp is a proven data migration solution that already exists, built on standard Microsoft technology (SQL Server, .NET, C# and Visual Studio).

To build something yourself, you need ETL-skilled people with a deep understanding of data migration, to build and then maintain it. Hopp is already built, and is maintained and developed continuously, so those ETL skills are not needed.

Hopp needs knowledge of the business, the data and the systems involved. Apart from the few manual rules described below, no other technical skills are required. With an ETL-based solution, the emphasis is on technical ETL skills, and the core competence, knowledge of the systems and the data, comes second. That is not ideal.

Building a robust data migration solution is not trivial. Whatever the foundation, ETL or otherwise, a solution that stays stable, recognizable, robust and maintainable over years of repeated migrations into an ever-changing target system is a challenge.

Hopp is such a solution. It has matured over more than 20 years of critical, highly complex data migrations very like the ones you will face: repeated migrations into (and out of) the same target system, in competition with other providers.

The ability to migrate data safely into your system is an important factor when you negotiate with new customers.

So do not underestimate the task of building a robust and lasting data migration solution, and do not build one when you can buy one.

04

Core qualities of Hopp

Below are some of the key differences between Hopp and any solution built on an ETL tool such as SSIS. We focus on Hopp's particular strengths, which differ so much from a typical ETL tool that they are very hard to reproduce in an ETL-based solution.

The unit of migration: business objects

Any solution built on an ETL tool is table-centric. It flows records from tables, or other sources, to the target, enriching and validating them on the way with data from other tables. It works like a Ford assembly line.

Hopp does not migrate tables and records this way. It is based entirely on business objects, which are migrated as complete, self-contained units. Hopp extracts one business object at a time from the source, with the data from all its tables, and migrates it as one unit. To stay with the cars: unlike the assembly line, Hopp builds one car at a time, completely.

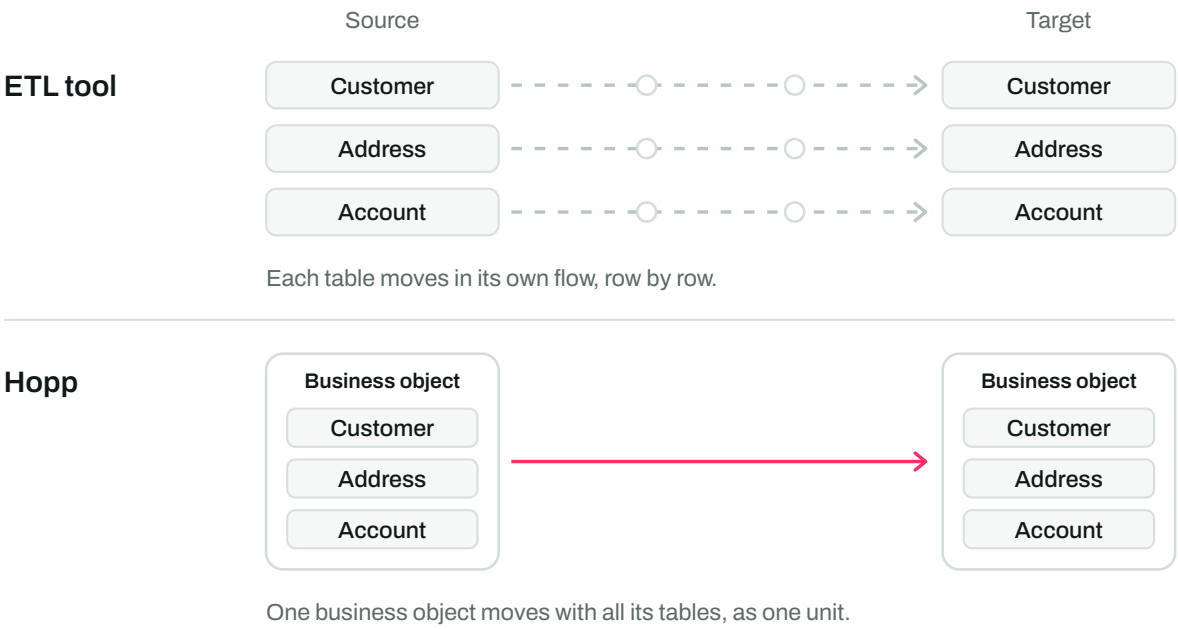


Figure 1. An ETL tool works like an assembly line, table by table. Hopp builds one car at a time: a business object, complete, which can be run again on its own.

Business objects are the foundation of Hopp's design, and they bring a long list of benefits. Here are some of them:

- **Hopp can migrate down to a single business object.** That is valuable throughout the project, and at a crucial go-live. You can work on a specific subset of business objects to resolve issues, instead of redoing the whole migration, as you must in a flow-based solution, where a single business object cannot be isolated.
- **Hopp resolves the dependencies between business objects** recursively, and migrates them in the right order automatically. That is valuable during the project, and especially at go-live: running a subset again also runs every business object that depends on it, and only those.
- **Hopp links all the data and events from the migration to business objects.** Hopp Portal can then show a complete view of each business object: every event raised, the data extracted from the source, and the data sent to the target. That supports the project from iterative development to user acceptance, and gives full documentation to archive after go-live.
- **Business objects abstract away the tables and fields of the target system.** Everyone involved, from the experts in the target system and the people defining the requirements to the migration team and the business users who test, shares one understanding and one vocabulary, based on the business objects.

Given the table- and flow-based nature of an ETL tool, it is very difficult, and probably impossible, to build units of migration like Hopp's business objects on one.

Specification and implementation

An ETL-based project needs developers skilled in that ETL tool. They do not necessarily have the business knowledge of the source and target systems.

When you build a new solution on ETL, you risk a problem that occurs in nearly every large data migration: the people who know the data and systems must specify to the ETL developers how to migrate the data. The developers do their best to implement the intent of the specifications. But the specifications grow very large, and over time the implementation drifts away from them, as the process iterates and the target system changes.

For a one-off migration, that may be acceptable. For ongoing migrations, such as those into your system, it often causes problems: the process slows down over time, and keeping up with future changes in the target system costs more.

Hopp works differently. In Hopp Studio, people with knowledge of the data and systems work together to specify the migration in full, and Hopp generates the implementation from that specification. The specification can include manual rules, written by hand alongside the generated code. They are the only part that needs development skills (C#).

Developing and maintaining a migration in Hopp means creating the business objects and field mappings in Studio, directly, by the people who know the systems and the data. The implementation is generated. Only a few C# developers are needed, for the relatively few manual rules: in our experience, one or two are enough even for large, complex migrations.

An ETL-based solution, by contrast, needs a large ETL development staff to build and maintain a large, varied collection of ETL flows.

It is the business objects that let Hopp generate the implementation this way. Given the nature of ETL, you are unlikely to build a similar solution in which the ETL flows are generated. Even if you did, you would also need to build an equivalent of Hopp Studio, to provide the input to generate them from.

Standards and uniformity

As the need for ETL professionals shows, a migration solution built on SSIS ends up with many ETL flows. ETL developers create them on the ETL design surface, where data connections and transformations are drawn on a canvas and joined by arrows for the flowing data. Most, if not all, of these flows grow large and complex over time.

There will be standards, but each flow is bound to be customized to some degree, as it is developed and maintained by hand. Even with plenty of ETL experts on hand, the solution will probably always depend most on the small group of developers who built it and know its details.

In Hopp, everything is uniform, and always done the same way. Every business object is mapped the same way in Studio, and the generated implementation migrates every business object the same way. Once you understand how one business object is mapped, you understand them all.

There is no magic wand that makes complex data migration simple. But Hopp goes a long way to make it standardized and accessible, far less dependent on key people, and easier for new people to join.

Handling errors

In an ETL flow, many transformations have an error output. Records that trigger an error in a transformation are sent to its error stream. A complex solution, with many flows of many transformations, has a very large number of error streams.

Presenting that error information consistently, in a meaningful and aggregated way, is a discipline in itself. So is tracing each error back to the exact point in the ETL flow where the problem arose.

In Hopp, specifying where events are raised, and what information from the migration goes with them, is part of mapping a business object in Studio. The events are raised consistently when the migration runs, and Hopp Portal presents them to all stakeholders.

Portal shows the events aggregated ("this event was raised by this many business objects"), and lets you drill down to the individual business objects that raised them. With Studio's cross-references, it is quick to find exactly where in the business object's mapping an event is raised, and where to correct it.

With careful design from the start, similar capabilities can be built in an ETL-based solution. Without business objects, though, it is doubtful that it would reach the level Hopp has out of the box.

Stakeholder involvement

An ETL-based solution risks being technical and hands-off. The migration becomes a black box, open only to the ETL specialists who know the many complex flows and their dependencies. Errors, progress and the quality of the result are hard for key stakeholders to see: management, and above all the responsible users at new customers in future migrations.

Hopp is built for stakeholder involvement. From the start, Studio's reports give structured information about the data requirements. During migration and testing, Hopp Portal gives a complete, logical and understandable overview of the whole migration, both aggregated and down to each data field and every error of a single business object.

Portal is open to management, to show progress and quality, and to the users who take part in the acceptance test of the migrated data. Its search finds any business object and shows exactly how it was migrated. Its workflow lets test users and the migration team communicate clearly, by commenting on the errors found and giving each one an owner and a status.

Besides its value in the project, stakeholder involvement is a strong point when you negotiate with future customers. It shows that you have a strong, transparent process for getting their data into your system safely.

> **Want to know more?**
>
> To see how Hopp compares with ETL on your own systems, write to us at sales@hopp.tech.



hopp.tech

sales@hopp.tech

Hopp Tech Ltd
86-90 Paul Street
London EC2A 4NE
United Kingdom

Hopp Tech ApS
Gl. Kongevej 11, 3rd floor
1610 Copenhagen
Denmark