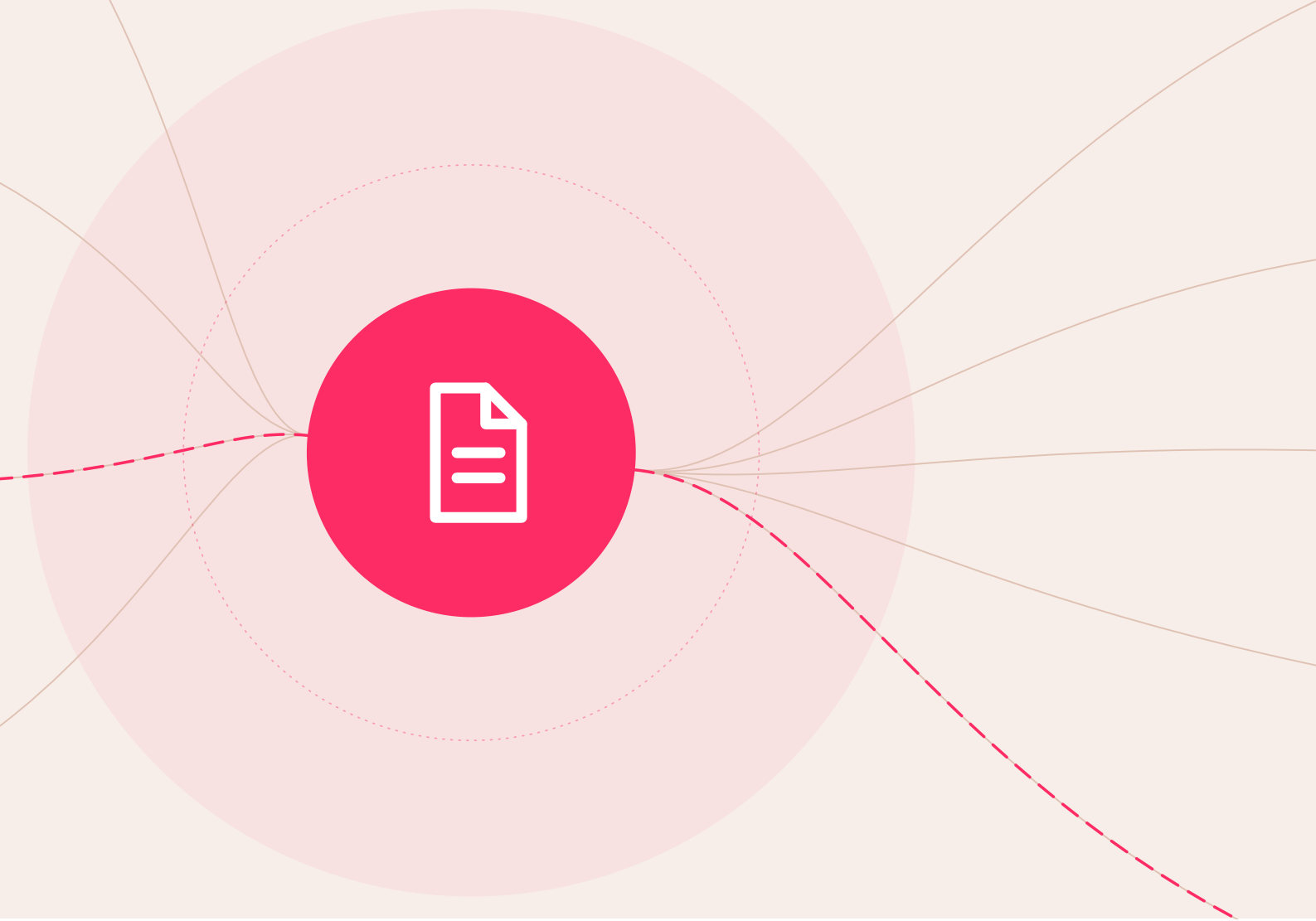
WHITE PAPER



Generic Integration Package

How Hopp connects to the system you migrate to

This paper describes the parts of Hopp's generic implementation that apply to any target system when you migrate application data.

Hopp is a generic solution for mapping, running and tracking complex data migrations. Its integrations include the specific integration to the target system, whether that is Oracle EBS, SAP or any other system the project needs to migrate data to.

Contact us to find out which Hopp integrations are already available, or how to build one for your target system.

01

Hopp and your systems

Hopp is a generic tool for complex data migrations. As a generic tool, it has to be applied to the specific situation: the legacy source system you migrate from, and the new target system you migrate to.

Hopp is built around business objects. They structure the business logic, and they are also the unit carried through the migration itself. You map in terms of business objects, and you track the data and events through the migration by business object as well.

Any generic tool takes some work to integrate with a specific environment, and Hopp is no exception. We have prepared it by building standards for every key element needed, which makes it easy to apply to any source and target system.

Several parts let Hopp integrate. Together, we call them Hopp Integration.

Hopp Integration connects Hopp to a specific target system when you migrate data from a legacy system, or from an older version of the same target system.

Out of the box, Hopp Integration has a blank target map, in which the target system and its data requirements are specified. It also has the integration points and message loops that let Hopp deliver data to the target system and receive information back from it.

Hopp can be implemented in several ways, but Hopp Integration is our recommended, standard approach.

We have already built a number of these integrations, such as Hopp EBS, for migrating data to Oracle EBS.

02

Mapping and data flow in Hopp

The Hourglass video explains how Hopp structures its maps. Here are a few points on how Hopp handles mapping and data flow, to position the integration.

In short: we map from the target system back to the source system, but run the migration from the source to the target.

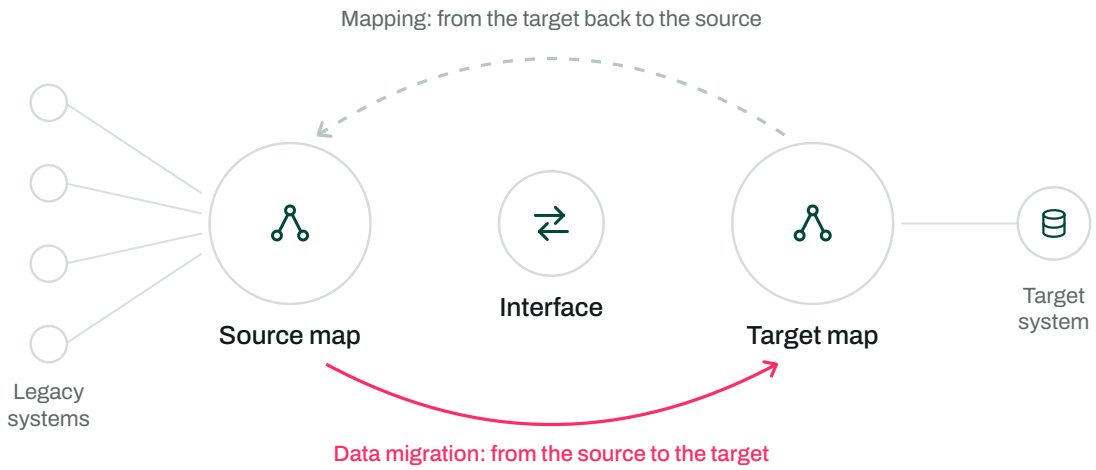


Figure 1. Hopp maps from the target back to the source, and runs the migration from the source to the target.

Some parts of Hopp Integration have a role in mapping, some in the data flow, and some in both.

First, the mapping, in more detail:

- The metadata describing the data to be delivered to the target system is imported into the target map.
- The target map defines the target interface: the data it must receive. It then defines the rules, transformations and validations that map the interface data to the metadata.
- The metadata describing the data available in the source system is imported into the source map, along with the target interface, which describes the data the target map expects.
- The source map defines how to extract the data from the source system, and the rules, transformations and validations that produce the data the target interface expects.

Then, a broad overview of the data flow:

- Once you have mapped the source data to the requirements of the target system, you can start moving data in Hopp.
- When you run the migration in Hopp, it processes all the mapped data, applies the rules and validations, and creates a data set that can be inserted into the target system.
- The final step receives the data produced by Hopp and delivers it to the target system.

It is not quite that simple. The migration may need to call the target system for parameters, or for data used in validations or as input to rules. It may also need to receive (error) messages from the target system while the data is delivered. Hopp handles all these options in a standard, predefined way.

As we go through the parts, we point out each part's role in mapping, because it is important to keep in mind.

03

The parts of Hopp Integration

Hopp Integration is made of several parts, which can be applied in several ways. That flexibility lets it work with any system, on any platform, database or technology.

Figure 2 shows what each part does, and where it sits.

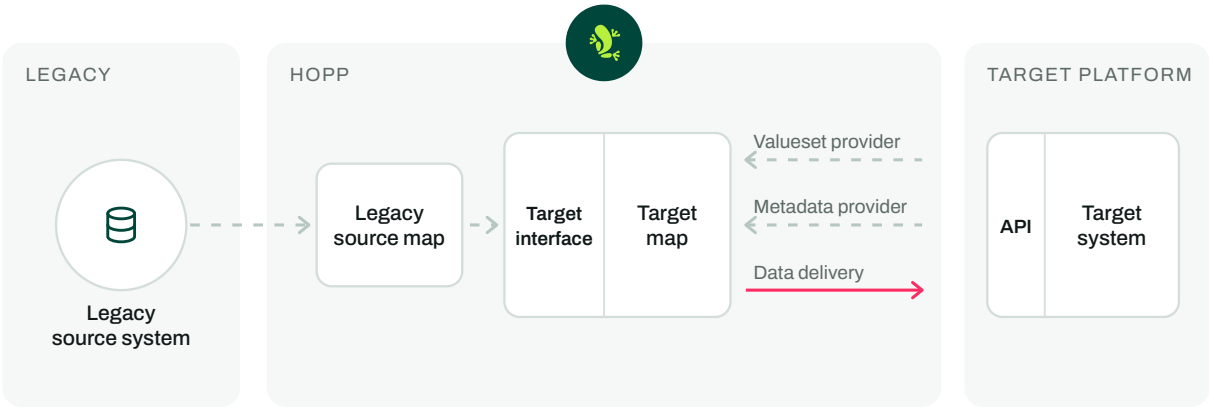


Figure 2. The parts of a Hopp integration. The target platform provides valuesets and metadata to Hopp; data delivery places the migrated data in the target system.

We start with the target system, on the right, and work our way across to the source system, on the left.

Target system and API

In the end, the target system receives the migrated data produced by Hopp. When the data is delivered, it may be necessary to retrieve error or information messages from the target system, to show them in Hopp and take corrective action.

It also helps to retrieve the metadata describing the data to be delivered automatically. This matters especially in the common case where the target system changes during the migration project.

Finally, Hopp needs to extract data from the target system to fill its valuesets, which it uses for validation and completion.

Valueset provider

The valueset provider connects to a well-defined Hopp interface, through which Hopp calls it. Its task is to extract data from the target system to fill the valuesets defined in the target map. Hopp then uses the valuesets to validate and complete the data flowing through the migration.

Examples of valuesets:

- A list of the product codes known in the target system. Hopp uses it to validate the product codes the migration generates.
- A list of the interest steps required for account products in a bank. Hopp uses it to insert missing interest steps for an account product when the steps do not exist in the source system.

Metadata provider

The metadata provider also connects to a well-defined interface, through which Hopp calls it to import the metadata describing the target system. That is useful at the start, and throughout the project, to record every change to the target system.

Data delivery

The migrated data Hopp generates for a business object is an XML document. It holds all the migrated data for the business object and its whole hierarchy of child business objects.

Data delivery receives these XML documents and places the data correctly in the target system, in the way the target system prescribes: through an authorized API, by direct insertion, or by another method. It must handle every aspect of redelivery:

- process only redelivered business objects that have not already been placed in the target system;
- clean up selected (or all) business objects, to place corrected versions of them in the target system.

The data delivery interface also lets the delivery mechanism send information or error messages back to Hopp, where they are flagged and handled in a structured way before the next attempt. The delivery mechanism can be files, databases or anything else, on any platform. Hopp formats and restructures the target object's data, held in XML, into the format the target system requires.

This coherent flow improves the quality of the whole run: it resolves the dependencies, and finds issues before or when the data is inserted into the target system. Any issues are returned to Hopp, flagged, and dealt with in a structured way before the next attempt.

Target map and target interface

Through the metadata provider, the target map imports the metadata describing the migrated data that data delivery will place in the target system.

The target map declares the data it must receive to generate that migrated data, and exposes this requirement as the target interface.

The target map removes internal references, and data that can be derived from other data. So the target interface exposes only the data that cannot be derived, and must therefore be received.

The target map can also run a range of validations at runtime, to give the target data generated by the migration the highest possible quality.

The target map is closely tied to the target system, and can be reused in every migration to the same target system. The legacy source map sees only the target interface, which keeps the two maps independent and reusable.

Legacy source map

In the legacy source map, you specify exactly what data you request from the legacy source system. Often you take what you can get, too much or too little; you do not know until you link it to the data the target map needs.

The legacy source map matches the source data descriptions (metadata) with the data requirements the target map exposes through the target interface.

So for each Hopp migration, the maps must be created and an integration set up, for mapping and data to flow from the source to the target system. If the Hopp Integration you need does not exist yet, we set it up as part of the project.

04

What runs where?

Hopp is installed on a Windows platform with SQL Server. Hopp Integration spans the Microsoft platform and the target system's platform: the target map and the target interface on the Microsoft side; the valueset provider, data delivery and the API package on the target system's side.

> **Want to know more?**
>
> See the integrations Hopp already has at hopp.tech/data-migration-software/integrations, or write to us at sales@hopp.tech about yours.



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