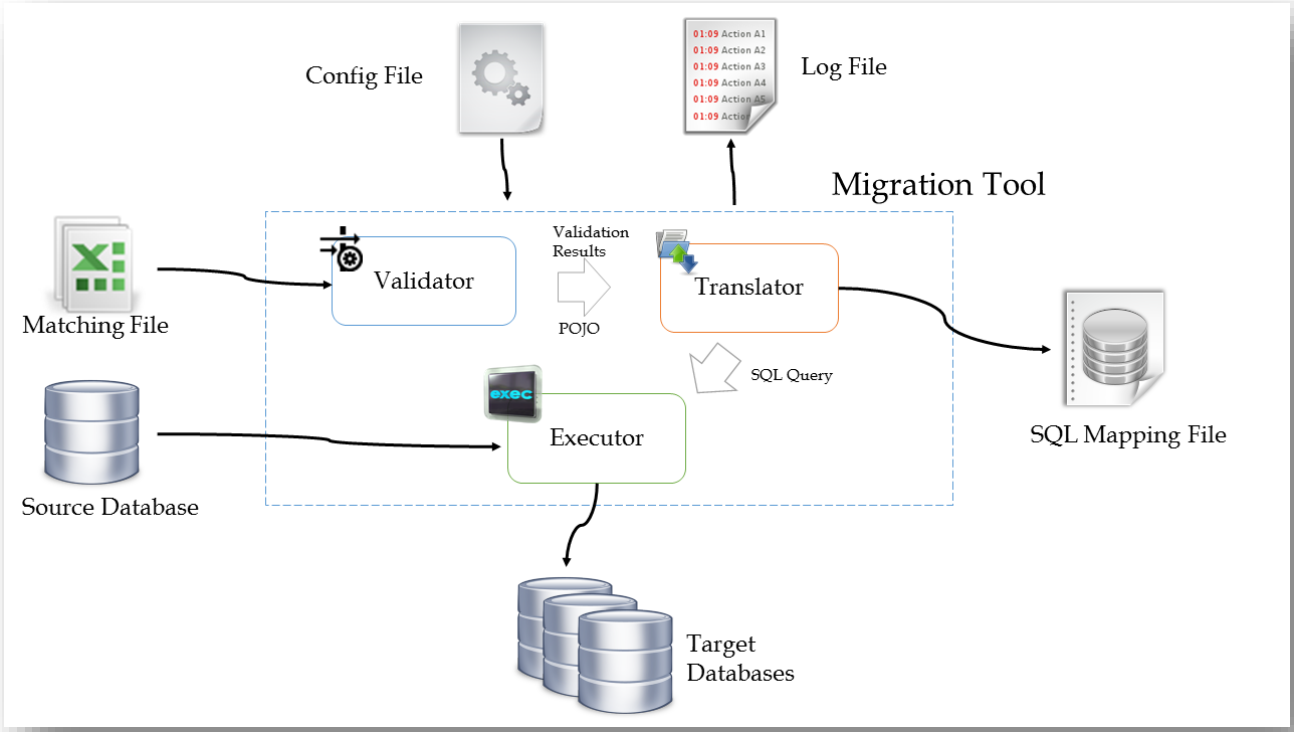
OpenMRS vs SESP Migration Tool Architecture Description (Briefing)

THE MATCHING APPROACH

The matching approach is LEFT-TO-RIGHT or TARGET-TO-SOURCE having OpenMRS in the LEFT (target) side and SESP in the RIGHT (source) side of the matching.

THE ARCHITECTURE

The main architectural idea is to have a tool that takes a matching file as input, validates the matches, translates them into SQL SELECT, INSERT and/or UPDATE statements (mapping), and execute the resulting mapping.



The matches are structured in a XSL input file that contains all the necessary information that the migration tool needs in order to make mapping decisions and generate the actual mapping, e.g.

(1) Where to get/set value?
(2) Where to get the FK's or referenced values?
(3) Does the value need to be converted, truncated, etc?
(4) What is the corresponding value of a particular parameterized value?

The bootstrap configurations are passed as input to the migration tool. It contains the info about the source and target database(s) (drive name, db name, host, port, and password).

The migration tool is a custom developed software that performs in three phases, namely: validation, translation and execution. The tool mainly validates all the matches at once, generate the mapping and execute the mapping as it is generated, all is done using automatic processes.

The validation tells whether or not the matching is correct, prior to the translation process, based on the info provided by the matching file, e.g.

(1) Does the TARGET column datatype matches with the SOURCE column datatype?
(2) Does the TARGET column size matches with the SOURCE column size?
(3) Can the TARGET column receive a NULL value, in case the SOURCE column value is NULL?

The validation is reported in form of *info, error* or *warning* that can be either used by the user in order to adjust or fix the matching, or by the tool in order to make matching decisions. In case of severity status in the report, the tool prevents the flow from going into the translation phase until the severity is fixed by the user.

During the validation phase, the tool encapsulates the matches in Java POJO objects structured based on composite design pattern. The composed objects are further used by the translation process.

The migration tool create the mapping and execute the mapping queries on-the-fly into the TARGET database and optionally generates an SQL output file for further use.

During the validation, translation and execution phases the translation tool report all the actions in a log file that can be used for auditing purposes.

MOTIVATIONS AND ADVANTAGES

This architecture is motivated by the following main advantages:

(1) The migration logic is placed in a versatile XSL file, it can be changed without having to re-implement the migration tool.
(2) The tool is reusable, it can be used to migrate from different SOURCE databases into OpenMRS.
(3) The tool is generic, it can be used in different migration contexts that evolve OpenMRS as the TARGET database.