

CowBase - A Library for Dairy Farm Data Handling and Curation in Python

M.J. Gote¹, I. Adriaens^{1,2}, M. Ceccarelli³, L. D'Anvers¹, D. Meuwissen¹, B. Aernouts¹

¹KU Leuven, Biosystems, Kleinhoefstraat 4, 2440, Geel, Belgium

²Ghent University, Data Analysis and Mathematical Modelling, Coupure Links 653, 9000 Ghent, Belgium

³University of Bologna, Department of Agri-Food Science and Technology, Viale Fanin 50, 40127, Bologna, Italy



Introduction

More **high-frequency** and **cow-specific** data become available with the increased implementation of Automatic Milking Systems (AMS) and sensor technology on dairy farms. **Collecting, processing, and analyzing** these **on-farm sensor data** are required for **intensive production and health monitoring**. Often, with significant differences between data availability and standardization, a **time-consuming and error-prone pre-processing** makes data and software sharing difficult. Therefore, we present **CowBase**, an **open-source Python software package** that **streamlines and standardizes data management** for dairy farms with AMS.

Objectives

- (A) **Extracting data** – extraction of raw data from the on-farm AMS database or a backup thereof
- (B) **Data pre-processing and storage** – application of standardization and creation of a standardized CowBase database
- (C) **Application for data analysis** – a collection of functions, including a class to handle the data connection with the database and previously published functions/models

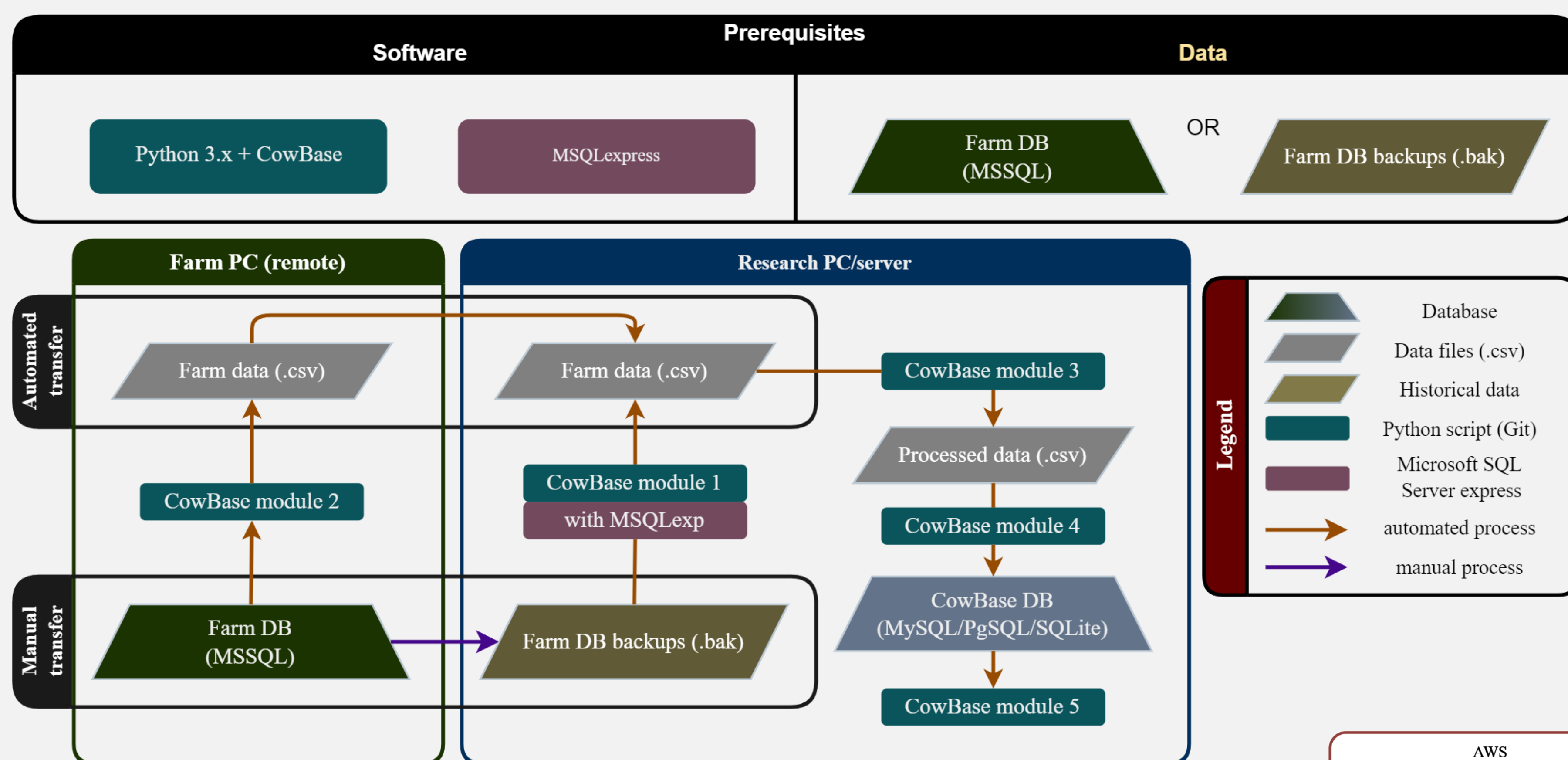


Fig. 1: Prerequisites and workflow for using CowBase. Modules 1 and 2 are used to extract data, and modules 3 and 4 are used for pre-processing and data storage. Module 5 offers a set of general functions that can be used for data analysis.

B. Data pre-processing and storage

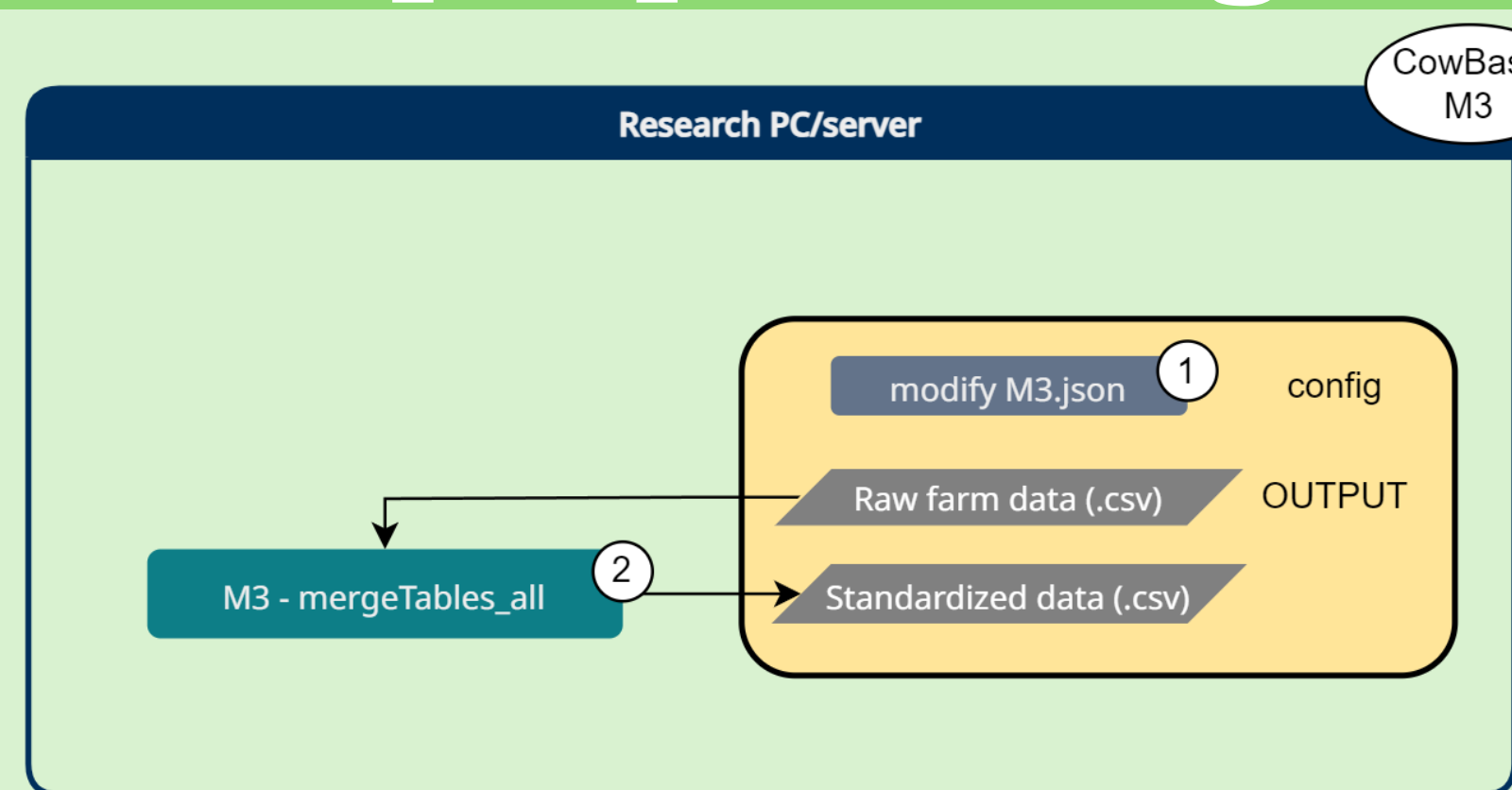


Fig. 4: Overview of module 3 – data pre-processing and standardization

- CowBase supports server-based (MySQL, PostgreSQL) and file-based (SQLite) database instances
- ER-diagram of CowBase is shown in Fig. 6

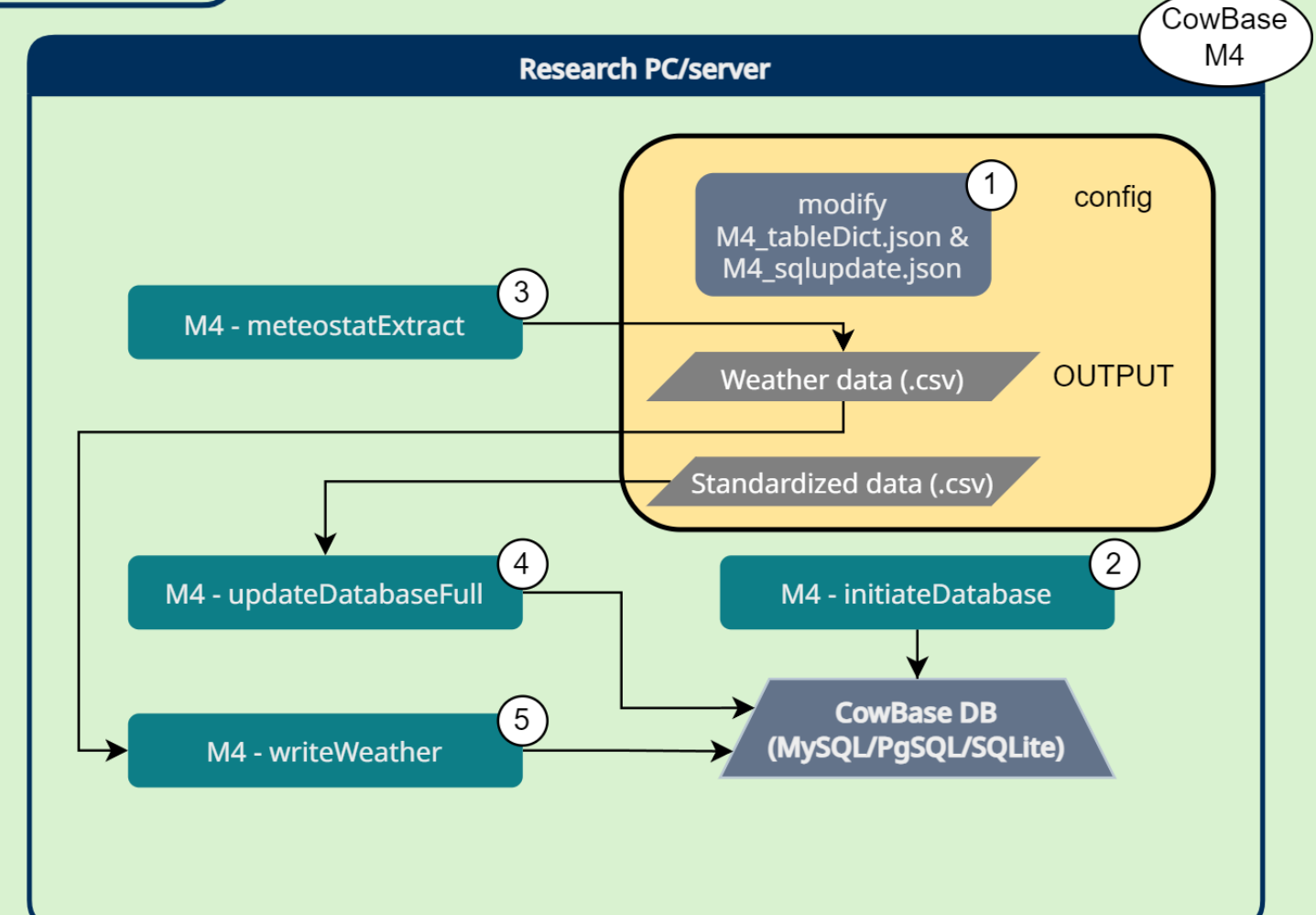


Fig. 5: Overview of module 4 – generating the CowBase database

Summary

Summary

- CowBase allows for fast and efficient **data mining and standardization**
- The well-structured **CowBase database** allows for **easy and fast data retrieval** and assures high reproducibility of research results
- With **standardization** applied, **sharing data and collaborating in research projects** is strongly supported
- Tested and validated on **70 European farms** with data dating back to 2005

A. Extracting data

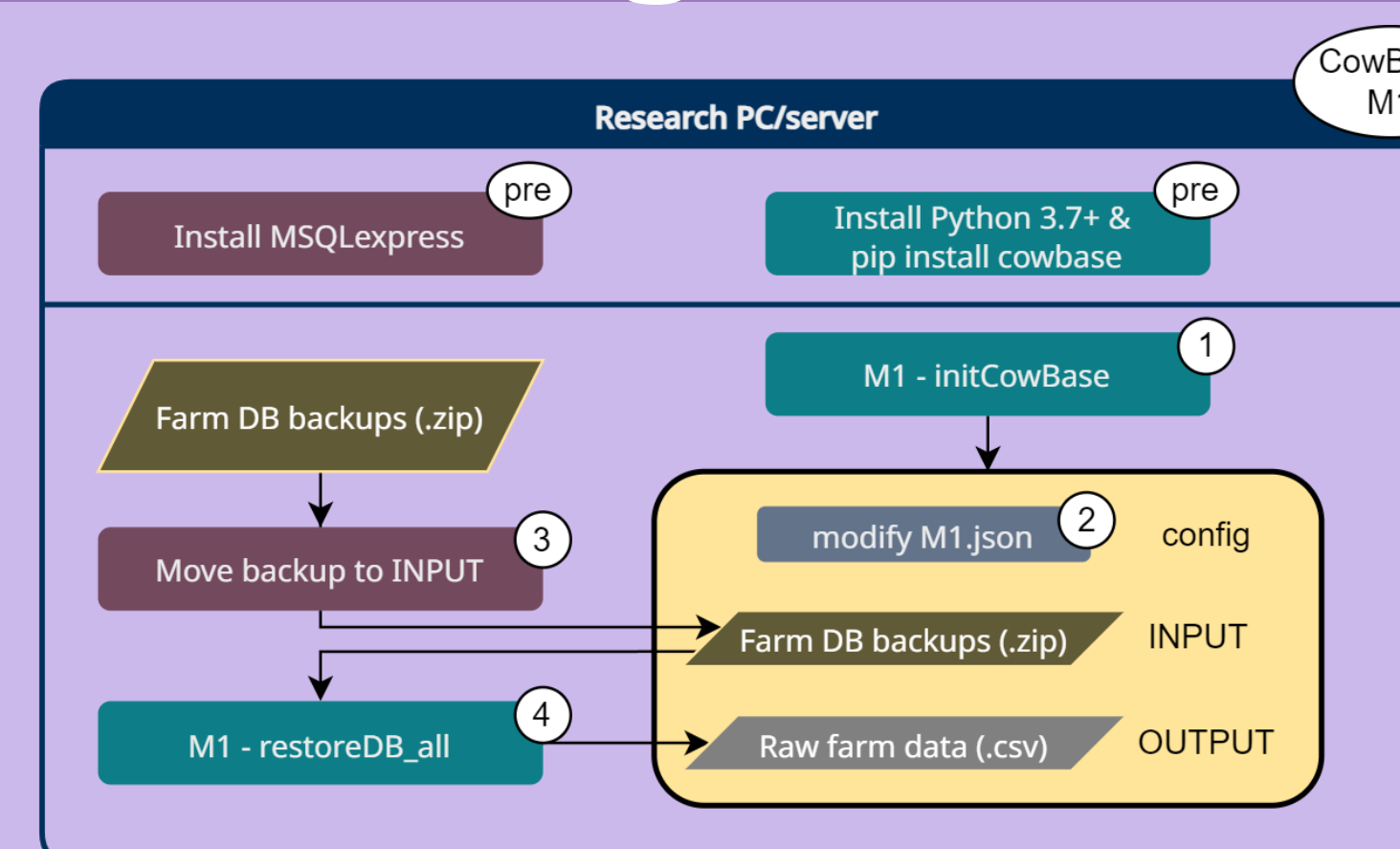


Fig. 2: Overview of module 1 – data extraction from AMS database backups

- CowBase works with both **DeLaval** and **Lely AMS**
- A **JSON dictionary** allows to specify tables & columns for extraction
- Extracted **raw data** is saved and can be accessed

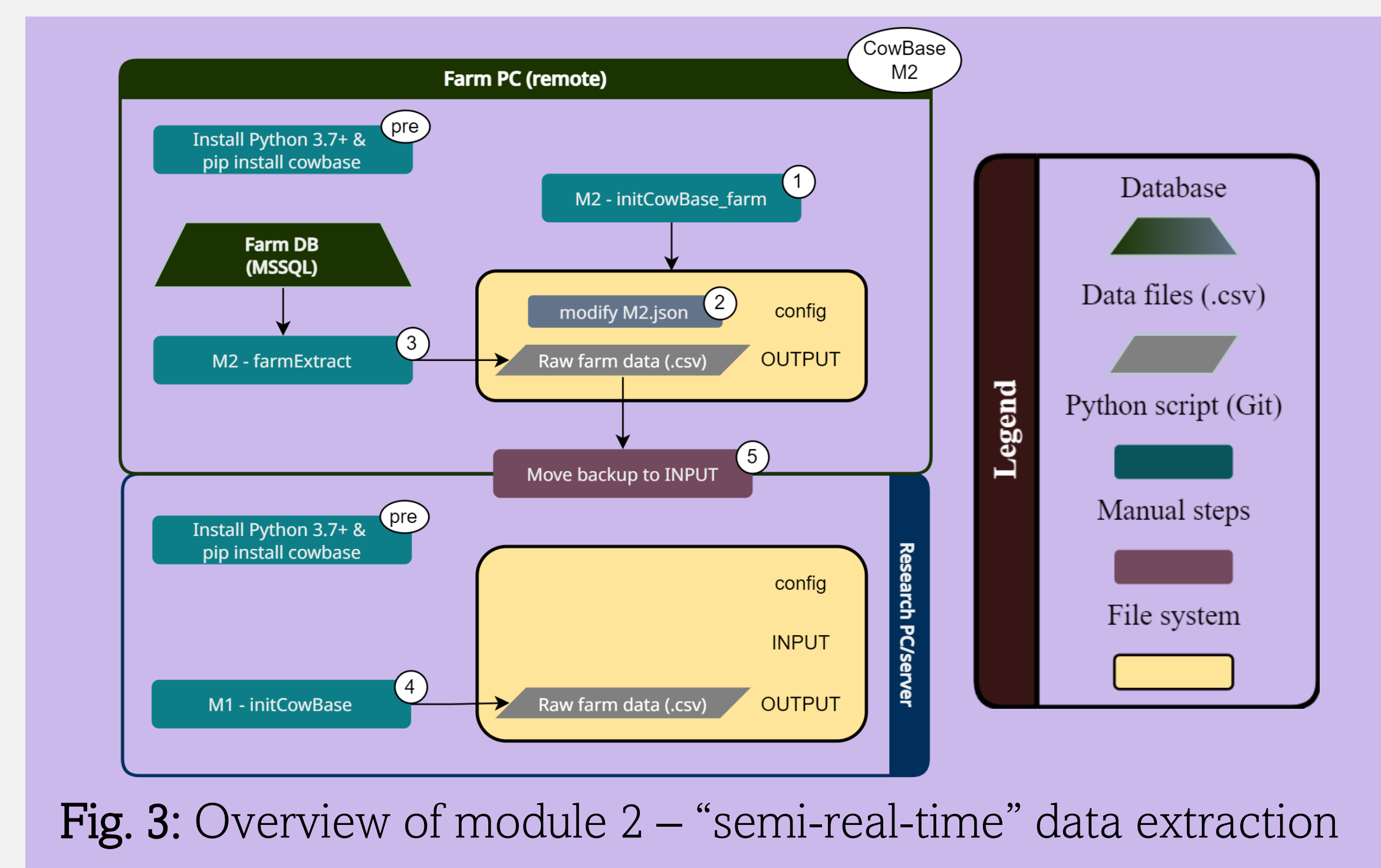


Fig. 3: Overview of module 2 – “semi-real-time” data extraction

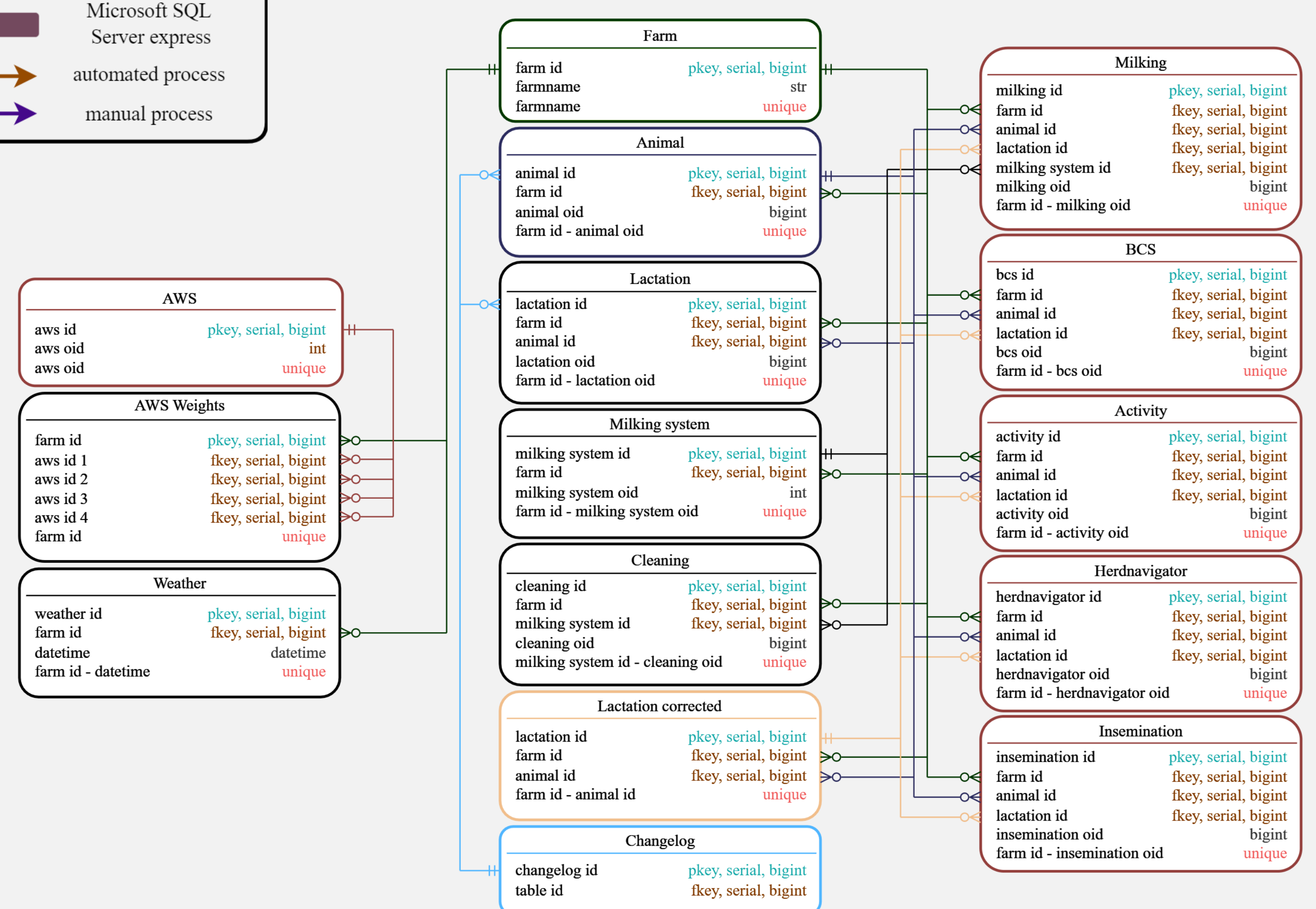


Fig. 6: Entity-relationship diagram of the CowBase database

C. Application for data analysis

Module 5: Data retrieval and analysis functions

- Connection function to query data from CowBase database in Python
- Previously developed models/functions:
 - “dailyMY”
 - “itw”, “pert”, “qreg”

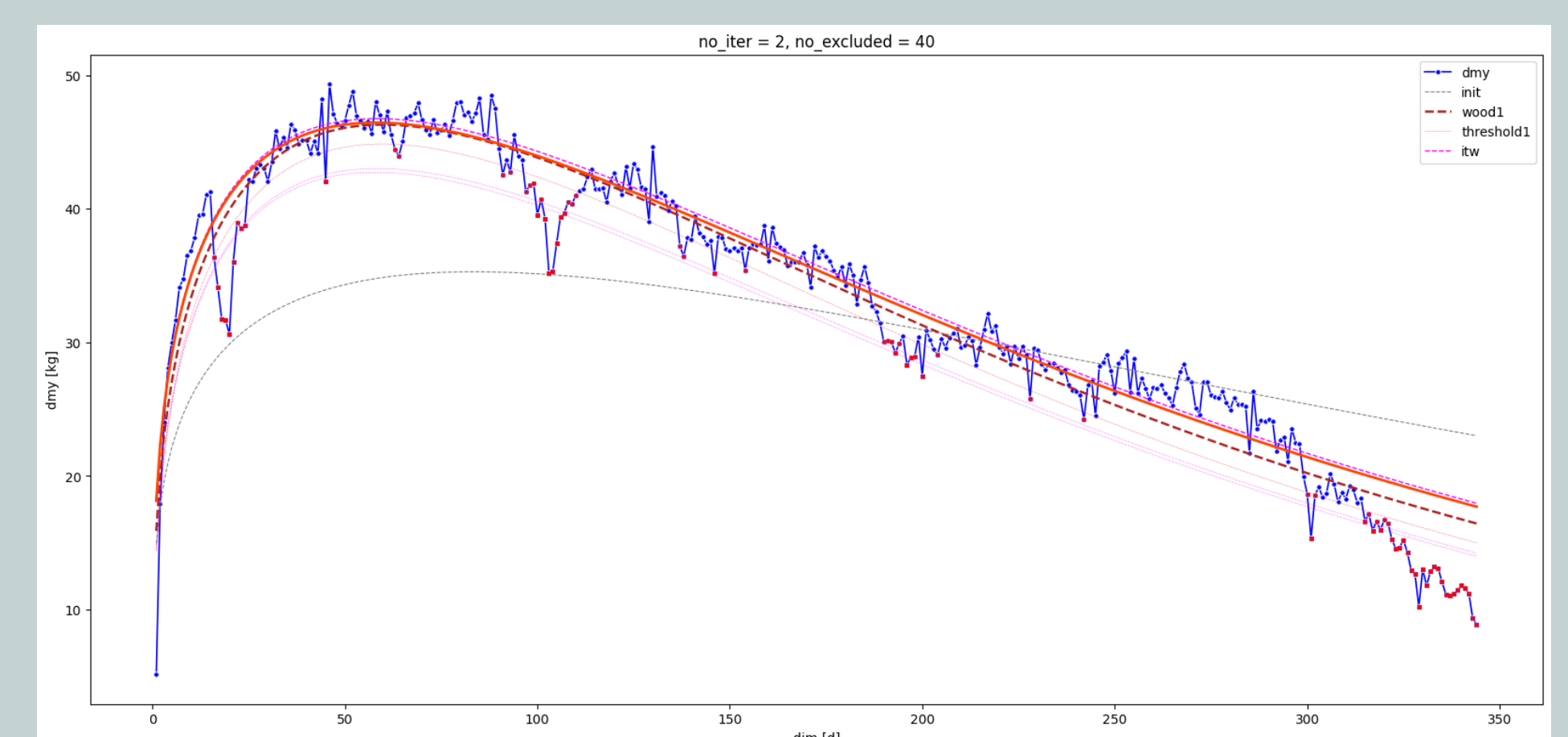


Fig. 7: Iterative wood model – “itw” - (Adriaens et al. 2021) applied to daily milk yield data



<https://gitlab.kuleuven.be/livestock-technology/cowbase>
or
pip install cowbase

