The Pi provides a sandbox environment for immersive, hands-on training, and perhaps more importantly, can act as a simple infrastructure solution to support an OMOP CDM instance that may be sufficient for certain real-world use cases, especially those in low-resource settings.

### Methods:

- We selected a Raspberry Pi 4 (model B) with the following specs:
  - 8 GB memory
  - 1.5 GHz quad-core ARM CPU
  - A 128 GB Integral microSD card (V30, Class 10, 100 MB/s)
  - 32-bit Raspberry Pi OS (kernel version 5.15, Debian version 11 bullseye, 4 April 2022)
  - Docker Engine 20.10.14 (23 March 2022)
- Deployment proceeds in the following order:
  1. initialization of the OMOP CDM database with fake data
  2. execution and review of Achilles and DDO quality control processes
  3. initialization of the WebAPI application database
  4. initialization of the WebAPI service
  5. initialization of the Atlas web application
  6. execution of the ‘glue’ container to initialize Atlas and WebAPI
  7. initialization of R Studio and subsequent validation and estimation models
- In practice, all services are installed on a microSD memory card; such a card could be pre-configured and shipped to data sites for plug-and-play integration into the OHDSI framework

### Results & Discussion:

- Achieving a working (and usable) set of applications on the Pi was nontrivial, and posed the following challenges:
  - Optimization of Docker builds
  - Heavy/concurrent computational loads
  - Database tuning and resource optimization
- The deployment process (after initializing a Postgres database with 10,000 synthetic patients and ~1M distinct events) is semi-automated and can be completed in less than 2 hours on such a Pi, with approximately one hour required to execute the quality checks, and one hour to execute the glue container and prepare the databases for interactions with WebAPI
- We orchestrate the deployment in this work using docker-compose (v2.4.1), but it can also be staged using Ansible playbooks, or via workflow managers like Apache Airflow
- Usability of the various web applications in this fully contained scenario (i.e. the Pi hosts the OMOP CDM database and all associated tooling) is moderate at best.
- Both deployment time and usability improve drastically, however, in the scenario where the Pi communicates with an external OMOP database (e.g. an AWS RDS instance, or a server on the local network).
- Note here that most of the computational load required for operating the ensemble of services is carried by the database

### Conclusions:

- The work presented above represents a preliminary effort to deploy an OMOP CDM instance and associated tooling in a compact and lightweight manner
- While modest, we hope that the output from this work can serve as both a training tool and as inspiration for projects looking to expand access to the OHDSI community and adoption of the OMOP CDM

---

Authors: Jared Houghtalinga and Lars Halvorsen
a edenceHealth NV (Kontich, BE)