Abstract
I created Python scripts to automate data centre asset management with the Infor EAM database.

The problem
The Data Centre Operations team is one of many groups at CERN that use Infor EAM for asset management. This is a database where all assets (e.g. servers, racks, power distribution units) in the data centre are recorded. It saves information that can track the whole life cycle of an asset from order to retirement.

The data centre (CDC) currently keeps track of almost 25 000 assets [1, assets in CDC departments] and we can only expect this number to rise during and after the long shutdown to keep up with the increasing computing needs of CERN [2]. We are also currently in the process of closing the Wigner site in Budapest. This will result in more than 3 000 assets changing location, both in real life and in the database [1, assets in CDC02].

Unfortunately, the graphical web interface of Infor EAM is rather cumbersome and slow. It requires navigating many sub-menus to access and modify information. It is particularly unsuited for altering information for several assets at once.

My task was to implement Python scripts that use the SOAP API of Infor EAM to automate access to the database without having to rely on the graphical web interface.

Previous work
Previously, the group used several PowerShell scripts to automate access to InforEAM, written mostly by Christian Boissat. However, PowerShell has some inherent limitations and frustrations. This has kept others from building on the existing scripts. And so a decision was made to migrate to Python, using the existing PowerShell code as a guide for the first few functions. This was supposed to be my project.

Shortly before my arrival at CERN, it was found out that Luca Gardi had already written some Python code that could retrieve some information from InforEAM using the Python package suds. This made the first few weeks of my work significantly more straightforward and became the foundation of my code.

Background - What is an asset?
Infor EAM is used to keep track of all kinds of assets: turbomolecular pumps, CO2 containers, printers, magnets, just to name a few. However, for the time being, we are only interested in data centre assets. These are identified by an asset ID which has the following structure:

\textbf{HCCSPDU}016-00000001

All assets that we are interested in will start with \textbf{HCCS}. This stands for \textbf{LHC Computer Server} [3]. This particular asset is a power distribution unit. This is what \textbf{PDU} stands for. Other classes of assets include \textbf{SYS} (system), \textbf{SAR} (system array) and \textbf{ENC} (enclosure).
The **016** stands for the type of PDU. There can be several PDUs of the same model. These would have the same part code. Parts themselves are objects in the database that can be manipulated. The rest of the asset ID is simply a counter to distinguish identical assets. In this case, the PDU is the first of its kind.

InforEAM saves lots of information for each asset, like its serial number, its location, its status or its manufacturer. Some of these fields can be set to any value, like the “description” field. Others cannot, like the “status code”. These can only be set to certain values at any time (e.g. “I” for “Installe et Maintenu” and “C” for “En magasin”) which can depend on the value of other fields.

**Result**
The final python script includes the following functions to manipulate assets:

- **Getters**
  - getAttribute, getPosition, getStatusCode, getParent...
- **Setters**
  - setAttribute, setCommissionDate, setDepartmentCode, setDescription...
- **Store transactions**
  - issueAssetFromStore and returnAssetToStore
- **Warranty related functions**
  - getWarrantyCoverage, createWarrantyCoverage, updateWarrantyCoverage
- **Creation related functions**
  - createAsset, createPart, createStoreBin...
- **Searches**
  - search, searchByBin, searchBySerialNumber, searchByDescription...
- **Withdrawal**
  - withdrawAsset

The code also includes some simple command line menus that guide users through the lifecycle of an asset.

For details on these and more functions, please refer to the well-commented code and the help file [4].

**Further work**
The existing code can track an asset from creation to destruction, however, there still are some interesting functions left to implement depending on the needs of the end users. In total, Infor EAM keeps information on over 1.3 million assets across CERN [1] and only a fraction of these belongs to the data centre. This automation project could potentially be interesting for other CERN groups as well.

[1] cmmsx.cern.ch


[3] naming.cern.ch/naming/ui/codes/LHC/CS

[4] docs.google.com/document/d/1_jlRlW6aH18ZPHeilV0SAtfwLtp8-YJ_OUdIfpznT1o/edit?usp=sharing