

Boston, Massachusetts, USA

*Prepared by Ariel Hu, Technology Systems Lead*

Current System: Used Technologies

Depending on the location, there may be several different technology combinations in place. Each location’s system description document will give you at least a high level overview of the technologies, but there are other things that you should probably consider.

1. Any custom code written for ChERP will be written in C++ and compiled as a DLL which is then referenced in the ChERP configuration files.
2. The ChERP database is in Oracle, but since we expect a new database to be constructed, we are open to shopping for the best solution.
3. Some of the other software packages in use throughout the company have APIs for custom code as well. Some of these APIs employ simple JavaScript solutions, others Python, and I think one might be PHP. ChERP is the only one that requires C++ for custom modules.
4. ChERP requires XML for all data imports. I believe most of the other packages can accept either XML or JSON.
5. Hardware in use varies widely by location. We just need to know what kind of hardware we will need for the new system once it’s designed.
6. Each of the additional solutions coupled with ChERP has its own independent data repository. This means that all data related to the functions of these solutions is stored in separate cloud locations, and much of the data is duplicated. We do not have access to any of these additional data repositories for customization. When changes are made to a set of data from any of the components, the others must be checked against the change and updated if necessary.
7. Overall, most system processes are quite slow due to the multiple integrations that are required to perform even basic tasks within the system.

ChERP Database

The database used by ChERP is Oracle-based. Its cost is included in the ChERP licensing fees. Figure 2 below shows the general database structure for ChERP. The figure does not show the original table names that ChERP comes with, but instead shows the names we changed everything to in order to meet our needs. This is the version in use here in the Boston office. The other offices will be similar, but not identical, as they have done their own customizations. For the purposes of your audit, I have advised the other locations not to include a database diagram so that you can work from ours as your primary model. Hopefully that will eliminate a lot of confusion.

A screenshot of a computer

Description automatically generated

Figure 2: Boston ChERP Database Structure (keys only)

I should note that one of the things we’d like to see in our new system is the ability to distinguish between the different types of office sites. For example, we have 6 regional administrative offices, and within each region there may be multiple area (territory) offices that manage the warehouses for the areas (territories), and then there are parcel offices, those that handle parcel pickups and deliveries for smaller sub-territories within each territory. These are not affiliated with a warehouse. There’s nothing in the database the lets us log or track parcels separately from regular warehouse shipments, so record keeping related to parcels is terrible. We definitely need a better solution.

Also, right now our suppliers are set up as regional, but this is not always the case. Sometimes they are territorial, or even sub-territorial. The same is true for our carriers. We even have our own internal carrier resources (trucks, etc) that are not part of the current database, but should be.