Helm CONNECT

TECHNICAL ARCHITECTURE

Powerfully built

General technology stack

Helm CONNECT is a web application built on Microsoft technologies. The server uses the following frameworks:

- .NET Framework 4.8
- ASP.NET Web API
- PostgreSQL 13+ (or SQL Server 2017 Local DB on assets)

The client is built to run on a web browser and uses the following frameworks:

- Vue.js
- Knockout JS
- Less.js

Instead of a traditional page load life cycle, Helm CONNECT uses a single page that updates dynamically by executing direct API calls to the server. Except for initial loading, the client always fetches raw JSON data from the server instead of preformatted HTML.

Offline asset functionality is provided by a database synchronization process and a mirrored stack installed on the asset computer.

Integrations

All external integrations (such as import/export tools, alternative user interfaces, etc.) work by interacting with the Helm CONNECT API rather than by connecting directly to the database. The entire API is composed of RESTful HTTP endpoints making it easy to use from almost any programming language or framework, including web browsers.

Cloud Hosting

The Helm CONNECT server is hosted on Amazon Web Services (AWS) using the North Virginia region. The database takes advantage of Amazon's Multi-AZ functionality which means that multiple availability zones within that region are utilized for redundancy.

Requirements

- The data center server runs on Windows Server 2019 and PostgreSQL 13+.
- The asset server runs on Windows 11 or Windows 10 (64-bit).
- Any computer running the asset or data center server must have .NET Framework 4.8 installed.
- Our recommended client browser is the latest released version of Google Chrome. We also support the latest released version of Microsoft Edge.
- For data transfer between your assets and the shore, we recommend that your assets have, at minimum, a stable 1 Megabit/second TCP/IP connection to the internet. Most major cellular connections (3G, 4G, etc.) and most satellite connections provide this.

Security

AUTHENTICATION

Every call to the API must include an authentication token, which is obtained by logging in with an email address and password. Passwords are stored as salted PBKDF2 hashes that are verified on log in.

PERMISSIONS

Super users can create roles, assign features to roles, and then assign roles to users. Users will automatically get the API permissions necessary to execute the features they have been granted. The user interface will also remove functionality that has not been granted to the user.

ENCRYPTION

We use TLS encryption for connections to the shore server from outside the data center. However, to avoid the maintenance overhead of managing TLS certificates on your assets, we don't use TLS for asset servers. Although we don't encrypt local traffic between the browser and the web server running on the same computer or on the same LAN, we use TLS for data transfer between the asset and the upstream server because we assume that the asset's internal network is secure but that anything transmitting over the internet is not secure and requires encryption. We also assume that the internal network in the data center is secure, so we don't encrypt communication between the load balancer and web servers.

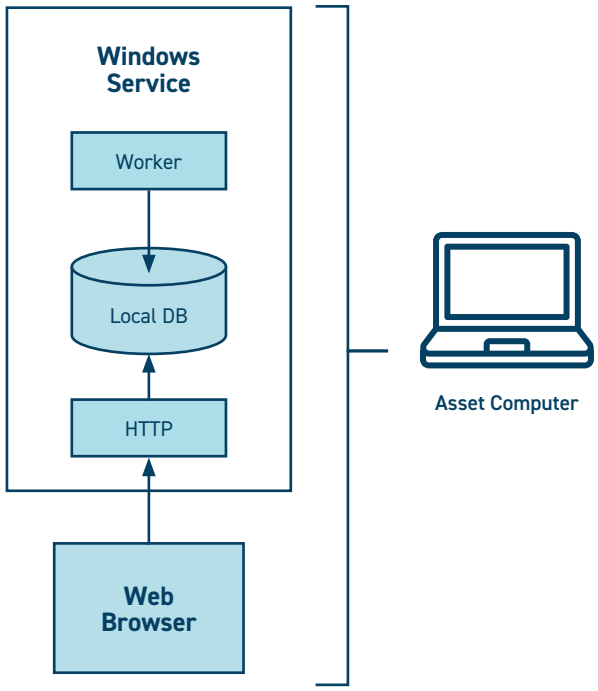
Asset Environment

To use Helm CONNECT on an asset without continuous internet connection, you can install Helm CONNECT as a desktop application that synchronizes with the primary server. This is simply a specialized version of the regular server with the following configuration differences:

- Uses SQL Server Local DB (can be packaged with Helm CONNECT installer or installed manually ahead of time)
- Always uses one web server and no load balancer
- TLS is disabled

These simplifications mean that we can install a single Windows service that controls the web server, databases, and background tasks in a very lightweight fashion. The installer will create this Windows service and put an icon on the desktop that launches your browser to the URL of the local web server. Even if you close the browser, the Windows service will continue running and synchronizing data with the primary data center when an internet connection is available.

The following figure shows the network diagram for an asset installation.



Data Transfer

The worker component of the asset Windows service will check periodically for an internet connection and attempt to send and receive updates from the upstream data center. The database on the asset has the same structure as the one in the data center, but it requires only a subset of the data. The asset transfers all its changes to the shore, but the shore only sends a subset of its data to the asset. The functionality on the asset is limited intentionally so that it will receive only the data necessary to carry out its functions.

Multiple Computers on a Single Asset

Helm CONNECT can be run simultaneously on multiple computers on the same asset. There are two different ways to accomplish this:

- **Option 1:** Run the Helm CONNECT installer on each asset computer individually. With this option, each asset computer will synchronize directly with shore which may increase bandwidth usage and means that the computers won't be able to see each other's data until it has been transferred first to shore then back down to the other computers on the asset. If the asset isn't connected to the internet, the asset computers won't be able to synchronize with each other.
- **Option 2:** Designate one asset computer as the server and configure the internal network so that it's accessible from the other asset computers and mobile devices, such as tablets. You will run the Helm CONNECT installer only on the server computer and other devices can access it by navigating to the host name or IP address of that computer in their browser. This option requires the server computer to stay online and the other computers or devices to connect to the same LAN, but it has the benefit of lower bandwidth usage and immediate data synchronization between the asset computers, even when there's no internet connection.