HELM CONNECT
TECHNICAL ARCHITECTURE

Technical Architecture

General technology stack

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

- .NET 4.5
- ASP.NET Web API
- SQL Server 2012 (or SQL Server 2012 Local DB on vessel)

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

- Knockout JS
- jQuery
- LESS

Rather than 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 vessel functionality is provided by a database synchronization process and a mirrored stack installed on the vessel computer.

Cloud Hosting

The Helm CONNECT server is hosted in 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.

Integrations

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

Requirements

- The data center server runs on Windows Server 2012 and SQL Server 2012 (any edition).
- The vessel server runs on Windows 7, Windows 8.1, or Windows 10.
- Browser requirement for the client is Google Chrome (recommended) or IE11.
- Any computer running the vessel or data center server must have .NET 4.5.1 or greater.
- Vessels must have a regular TCP/IP internet connection. It does not have to be continuously available.

Security

AUTHENTICATION

Every call to the API must include an authentication token. Tokens are obtained by logging in with an email address and password. Passwords are never stored in plain text, instead they are stored as PBKDF2 hashes that are verified on login.

PERMISSIONS

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

ENCRYPTION

Connections to the shore server from outside the data center use SSL encryption. The vessel server does not use SSL to avoid the maintenance overhead of managing SSL certificates on the vessel. Local traffic between the browser and the web server running on the same computer or on the same LAN is unencrypted. However, when data is transferred between the vessel and the upstream server it does use SSL. It is assumed that the vessel'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 communication between the load balancer and web servers is not encrypted.

Data Center Environment

TOPOLOGY

A Helm CONNECT server deployment consists of several components:

- **Primary SQL Server database:** Stores critical, high value data.
- **Web servers:** Stateless HTTP(S) handlers.
- **Worker processes:** Perform background tasks such as data transfer processing, sending emails, etc.

The application is split into these components so that they can independently scale in a data center environment, especially in the case of a cloud network like AWS.

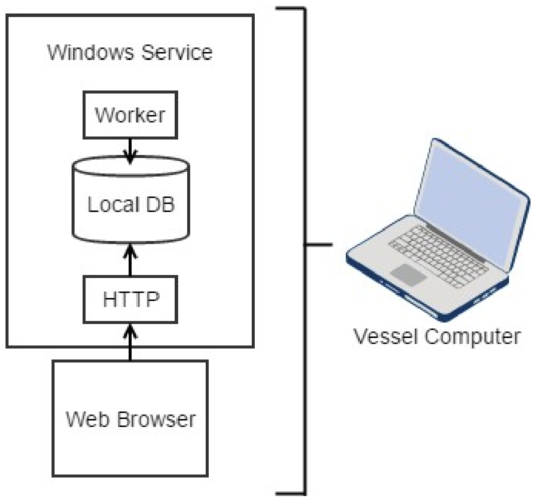
Vessel Environment

To allow for usage of Helm CONNECT on the vessel without a continuous internet connection, it is possible to 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:

- Use SQL Server Local DB (Can be packaged with Helm CONNECT installer or manually installed ahead of time)
- Always use one web server and no load balancer
- SSL 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 which 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 network diagram for a vessel install looks like the following diagram:



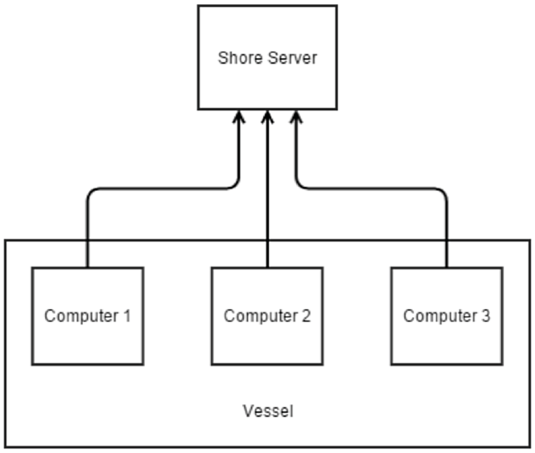
Data Transfer

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

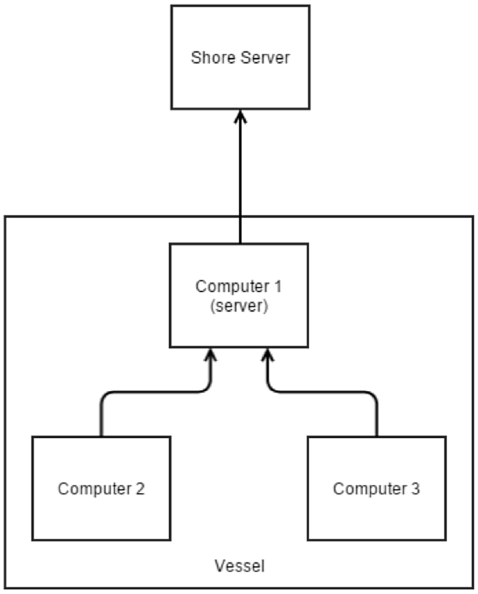
Multiple Computers on Vessel

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

- Option 1: Directly run the Helm CONNECT vessel installer on each computer. This will work fine, except that every computer will directly synchronize with shore, increasing bandwidth usage. Additionally, computers within the same vessel will not be able to see each other's data until it has been sent to the shore data center, and then back down to the other computers on that vessel. If the vessel's internet connection is down then they will not synchronize.



- Option 2: Designate one computer as the server and configure the internal network so that it is accessible from every computer. The Helm CONNECT installer will only have to be run 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 requires that the server stays online and that the computers are connected to the same LAN, but it has the benefit of smaller bandwidth usage and immediate data consistency between computers on the same vessel, even when there is no internet connection.


Shore Server
Computer 1
(server)
Computer 2
Computer 3
Vessel