

## EMT Device Manufacturer Uses Bluehatsoft's Services to Ensure End-to-End Functionality



### Client Overview

A start-up that designs ultrasound devices used by Emergency Medical Technicians approached Bluehatsoft to address the low performance of their devices.

### Architecture

The ultrasound device is a hand-held, battery-operated scanner that can securely pair with an app running on an iPad. The scanner can stream the result of an ongoing scan to the paired app running on the iPad. The app on the iPad has various options to securely pair with a scanner, to configure the scanner, as well as to render the stream in real-time.

### The Problem

The video frame rate was about 8-10 frames per second, which caused glitches in the stream and lowered the quality of the videos. Secure pairing was unstable.

### The Diagnosis

Bluehatsoft designed a diagnostic suite on a reference device to accurately replicate the failure scenarios. Using the instrumented software stack, it was diagnosed that the hardware's capability for collision avoidance and ability to use multiple antennas simultaneously were not exploited by the software stack. In addition, the hardware was configured to operate in the lower Wi-Fi spectrum (2.4 GHz), whereas it was fully capable of operating in the higher Wi-Fi spectrum (5 GHz). The issue was exacerbated by the presence of any Bluetooth device which also operated in the 2.4 GHz range.

### The Solution

Bluehatsoft re-designed the device driver software stack by employing a different architecture to grab completed frames from the FPGA memory while the next scan was in progress and transferring it in place by designing a glue layer that implemented zero-copy mechanism for interfacing with the network software stack. This approach cut down the frame grabbing interval to half of the previous one.

Bluehatsoft enabled the WiFi chips 802.11N capabilities, helping it to use more than one antenna, which improved the signal strength and connectivity to a very stable level, thus eliminating dropped connection counts.

Bluehatsoft implemented a method to dice the large video frame buffer into smaller chunks on the device and reassemble them into their original form in the software stack running on the rendering device (iPad). This eliminated re-sending of the large frame buffer in case of data corruption resulting from network interference, or any other reason, as any re-sending required the retransmission of only the missing/corrupted smaller chunks instead of the entire video frame buffer. This change resulted in significant improvement to the streaming/rendering and made it near real-time (NRT).

### The Result

With all design and architectural changes, Bluehatsoft could deliver about 35 frames per second, an improvement of 3.5-4 folds!