# Remote Audio and Video Sensing Using Custom a iOS Client and Windows Server Software

This paper proposes a baseline architecture and implementation of a remote AV sensing platform based on iOS devices (iPads and iPhones) and server components running on Microsoft Windows. The solution will provide the following functionality:

## iOS Client Components

* A custom iOS app with a minimalist user interface
  + A start button that registers the running app with the server. This registration creates a notification channel by which the server software can start/stop the AV feeds.
  + A settings page that collects the server endpoint URL.
  + A diagnostics/debug page showing network connectivity and server-related statistics
* Connectivity to server based on web service API library
  + The server connectivity code will reside in a separate class file so that it can be reused by alternate MonoTouch apps without any coupling to the demonstration-only client app

## Windows Components

* A WCF-based web service that acts as a target endpoint for the client(s) AV feeds
* A runnable windows application that hosts the WCF service and the monitoring UI page
* The monitoring app UI will be a simple task tray icon that has a pop-up menu for launching a control dialog window. The dialog window will provide the following functions:
  + List of registered sensors with start/stop commands for each
  + A launch button to show a diagnostic/debug screen for visualizing network connectivity and web service statistics
  + A monitor window to select a live feed to monitor
* A set of unit test classes that exercise the client/server communication APIs

## Proposed Architecture

* The client app will run on iOS devices and will be based on C# source code built under the MonoTouch framework using the MonoDevelop IDE.
* The server components will be based on WCF hosted in a running windows application (Note that IIS will NOT be a requirement so the server application will run on Windows 7 and not require a server version of the OS).
* The WCF service channel will be built in C# using the subset of the WCF framework supported by MonoTouch.
* The Silverlight WSDL parser and proxy generator will be used to emit the client connectivity code to be used on the iOS device.
* The web service will expose methods for negotiating a UDP channel when possible but that degrades to HTTP over TCP if firewalls or other network issues prevent UDP.
* The stream monitoring window on the server will be implemented using the Win32 DirectShow API and the .NET code will use the P/Invoke facility to interoperate with the non-managed code.

## Deliverables

* iOS App source code
  + Packaged as a zip file containing the complete MonoDevelop C# solution and all files required to build the client app
  + An MS Word document describing the build procedure and any additional required documentation
* Windows server application source code
  + Packaged as a zip file containing the Visual Studio 2010 solution and the service, unit test, and server application project source code.
  + An MS Word document describing the build procedure and any additional required documentation
* Live demonstration upon delivery
  + Multiple clients
  + Multiple servers

## Cost

$3500

## Vendor Experience and Expertise

Infinitek has delivered iOS solutions based on web service calls to a back-end servers that update business databases. Additionally, we have delivered streaming media applications used in home automation products. Our background in end to end solutions based on multiple technologies makes us ideally suited for delivering high-integration projects.

## Contact Information

Steve Millar, CEO Infinitek Corp

362 Valley Rd

Dawsonville, GA

30534

706-265-6317 (o)

404-275-1033 (m)

steve.millar@windstream.net