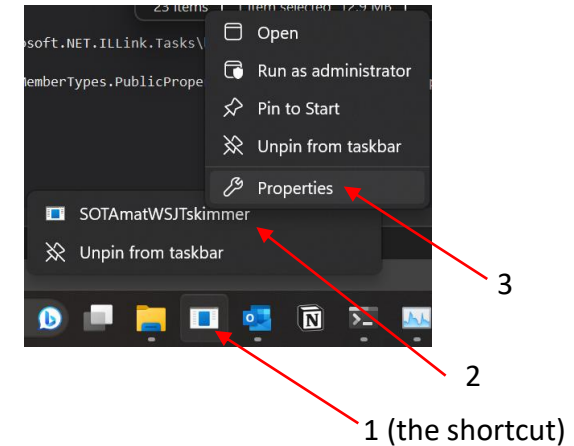
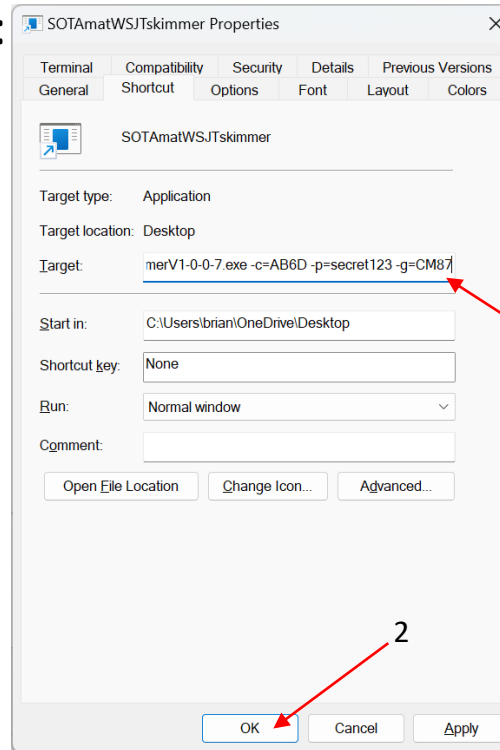


Setting Up SOTAmat's Skimmer Plugin on Windows

1. Copy the `SOTAmatSkimmer.EXE` file somewhere on your computer
2. Create a Windows shortcut to the plugin
(I like to drag-n-drop the EXE onto the Windows Task Bar, which *automatically* creates a shortcut to the EXE on the taskbar)
3. Right-click the shortcut, SOTAmatSkimmer, and select "Properties":
4. Add your command line parameters in the "Properties" window at the very end of the "Target:" field:
(To see command line options documentation, run the EXE with the "--help" parameter)



Setting Up SOTAmat's Skimmer Plugin on Linux / Mac

1. Copy the `SOTAmatSkimmer` file somewhere on your computer

2. In a terminal window run the commands:

A. Check to see if the `SOTAmatSkimmer` is set as an executable (it should be):

```
Prompt> ls -l SOTAmatSkimmer
```

```
-rw-rw-rw- 1 mathewb mathewb 15119416 Apr 28 19:09 SOTAmatSkimmer
```

B. If it isn't an executable already (no "X" in the left columns) then change it:

```
Prompt> chmod 550 SOTAmatSkimmer
```

```
Prompt> ls -l SOTAmatSkimmer
```

```
-r-xr-x--- 1 mathewb mathewb 15119416 Apr 28 19:09 SOTAmatSkimmer*
```

C. Run the program with command line options:

```
Prompt> SOTAmatSkimmer -c=AB6D -p=mysecret -g=CM87
```

(Optional: write a script that runs the program with your command line options, and/or set environment variables for your callsign, password, and gridsquare -- see next page)

Required configuration properties

- Your **callsign**
(with the “-c” parameter, or via the SOTAMAT_CALLSIGN environment variable)
- Your SOTAmat **password** from the web site
(with the “-p” parameter, or SOTAMAT_PASSWORD environment variable)
- Your **antenna’s Gridsquare**
(with the “-g” parameter using 4 to 6 characters such as “CM87”, or the SOTAMAT_GRIDSQLURE environment variable)
(Note: if you are using a remote station with WebSDR use the antenna’s gridsqure, not your own location!)

Command Line Documentation

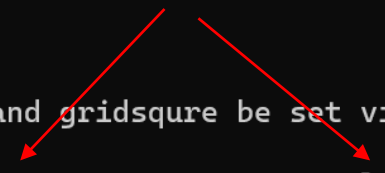
```
DOS-box - SOTAmatWSJTskin  X  +  v

C:\Users\brian\OneDrive\Desktop>SOTAmatWSJTskimmer.exe --help
SOTAmat WSJT-X Skimmer v1.0.0.7, by Brian Mathews AB6D. WsjtxUdpLib by Tom Fanning M0LTE.
This utility connects to WSJT-X, filters reception reports, and sends SOTAmat messages to the SOTAmat server.
Usage: SOTAmatWSJTskimmer <options>
Options:
  -h, --help
  -v, --version
  -d, --debug [default: false]
  -l, --log [default: false]
  -a=<DNS-name or IP-address>, --address=<name/ip> [default: 127.0.0.1]
  -port=<port>, --port=<port> [default: 2237]
  -m, --multicast [default: false]
  -c=<SOTAmat user callsign>, --callsign=<callsign> (required)
    [or use SOTAMAT_CALLSIGN environment variable]
  -p=<SOTAmat user password>, --password=<SOTAmat user password> (required)
    [or use SOTAMAT_PASSWORD environment variable]
  -g=<your gridsquare>, --gridsquare=<your gridsquare> (required)
    [or use SOTAMAT_GRIDSQUARE environment variable]

Examples:
  SOTAmatWSJTskimmer          [Note: requires callsign, password, and gridsquare be set via environment variables]
  SOTAmatWSJTskimmer -c=AA1ABC -p=mysecret -g=CM89
  SOTAmatWSJTskimmer -c=AA1ABC -p=mysecret -g=CM89 -a=192.168.0.101 -port=2237 --multicast
[type CTRL-C to exit]

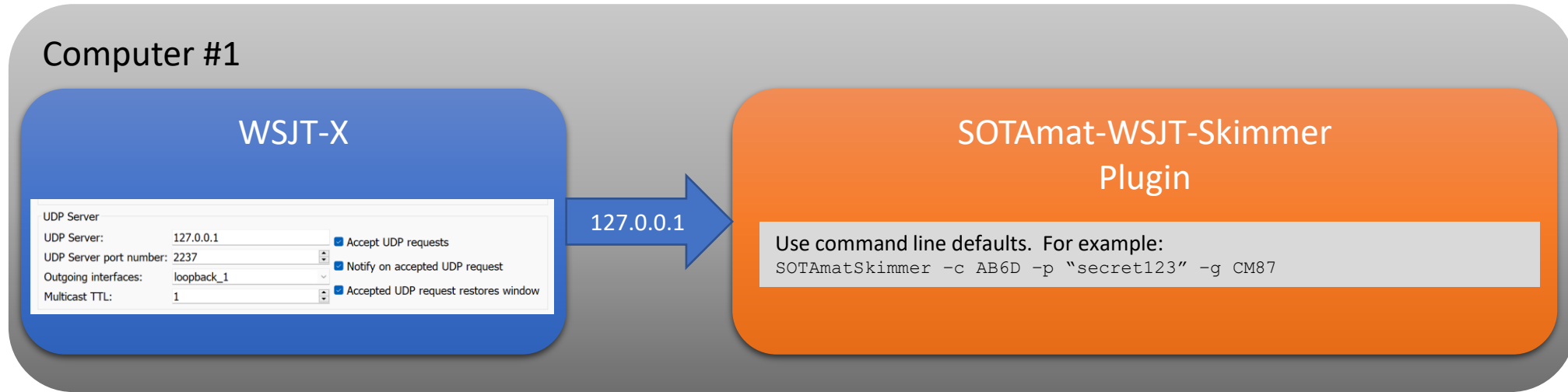
Attempting to connect K6LWM to WSJT-X via direct UDP at address 127.0.0.1...
```

Documentation bug: multicast should be 224.0.0.1



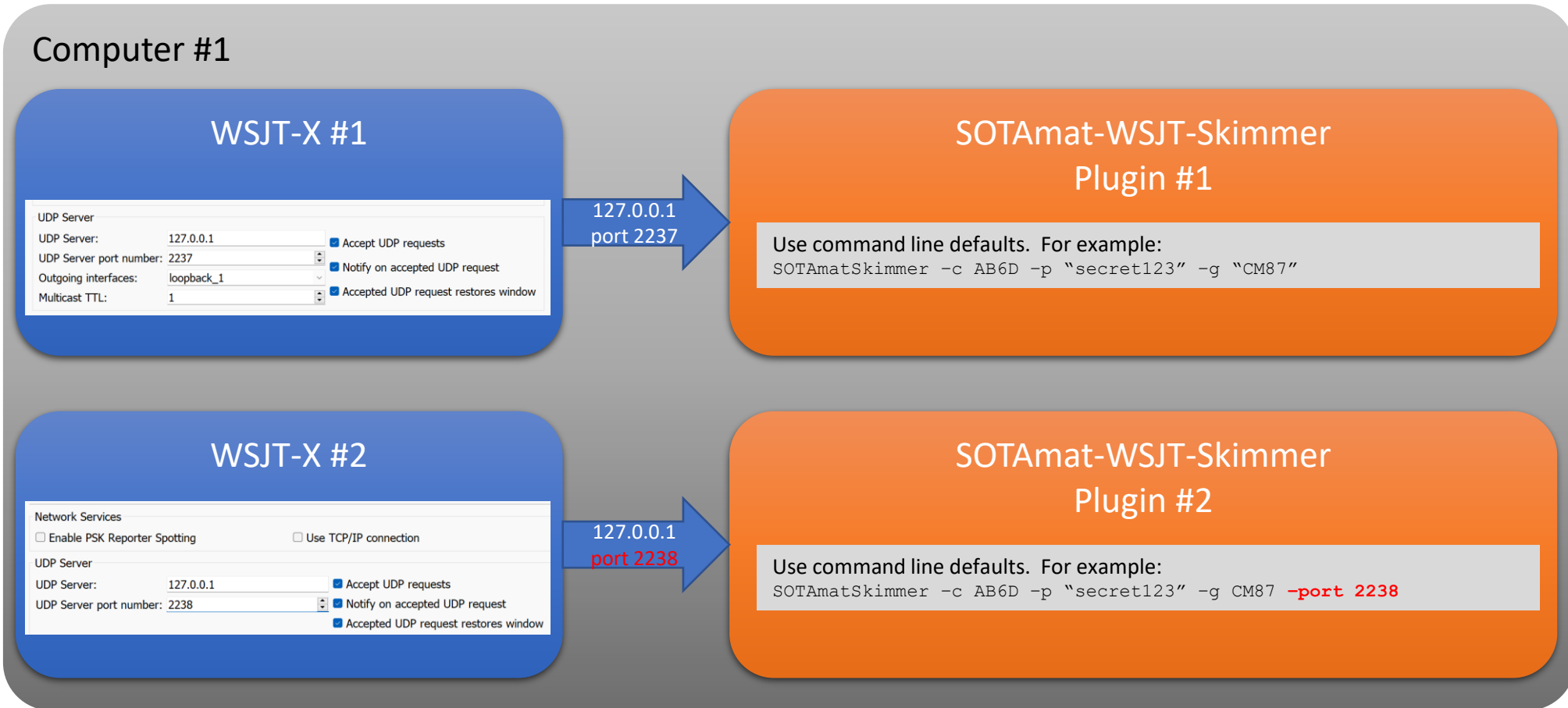
Unicast connection:

single computer, single instance, single plugin



Unicast connection:

single computer, two instances, single plugin for each



Unicast connection:

Two computers, single instance, single plugin

Computer #1 – Example IP Address: 192.168.50.111

WSJT-X

UDP Server
UDP Server: 192.168.50.222 ☒ Accept UDP requests
UDP Server port number: 2237 ☒ Notify on accepted UDP request
Outgoing interfaces: loopback_1 ☒ Accepted UDP request restores window
Multicast TTL: 1

192.168.50.222

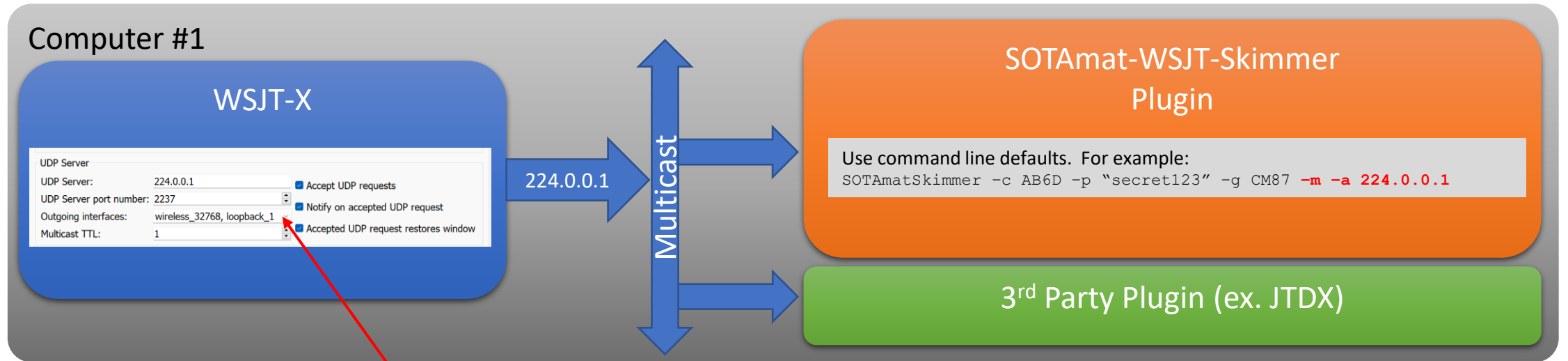
Computer #2 – Example IP Address: **192.168.50.222**

SOTAmat-WSJT-Skimmer
Plugin

Use command line defaults. For example:

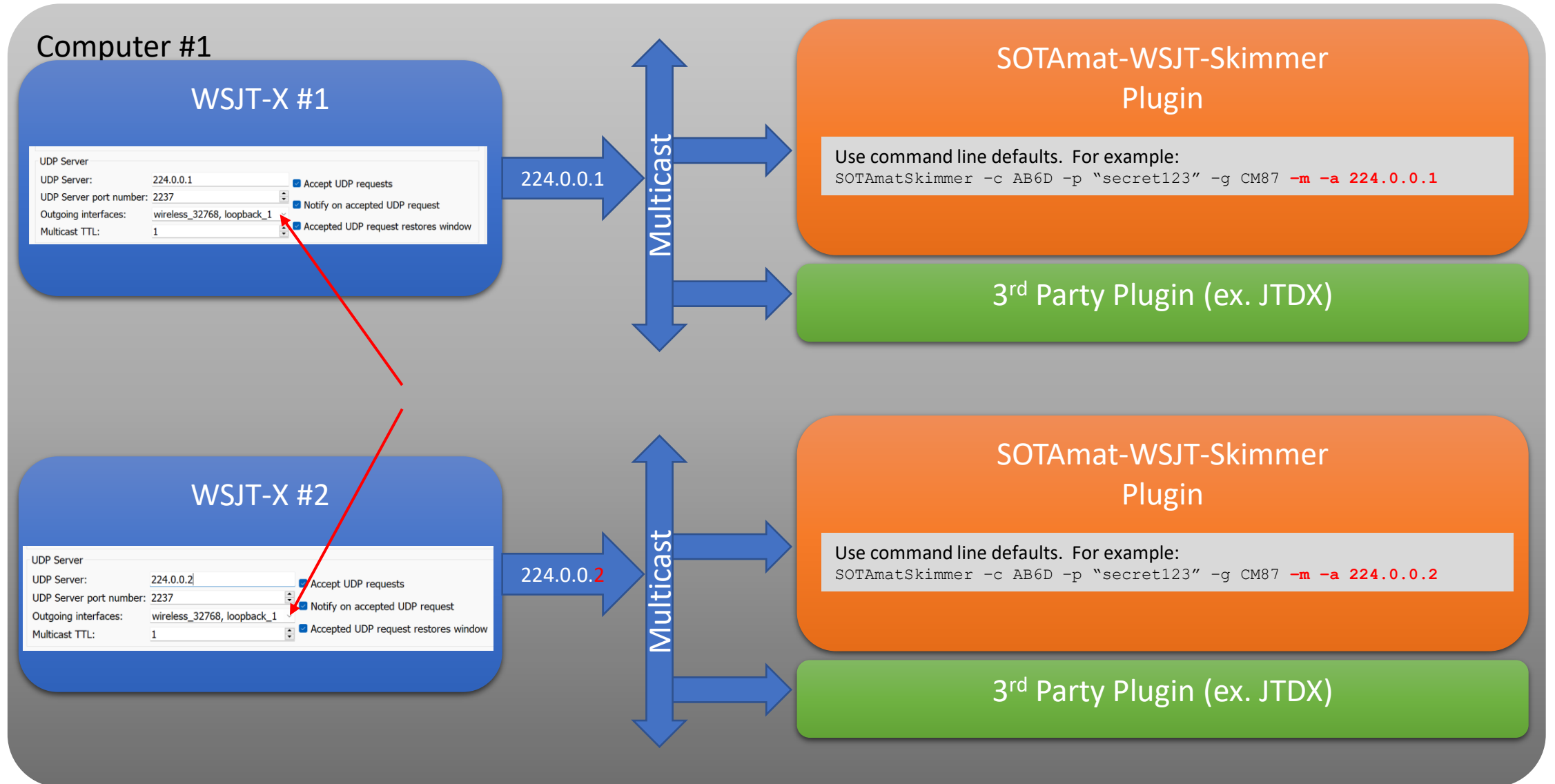
```
SOTAmatSkimmer -c AB6D -p "secret123" -g CM87 -a 192.168.50.222
```

Multicast connection: use IP address in range 224.0.0.1 thru 224.0.0.255
single computer, single instance, two plugins

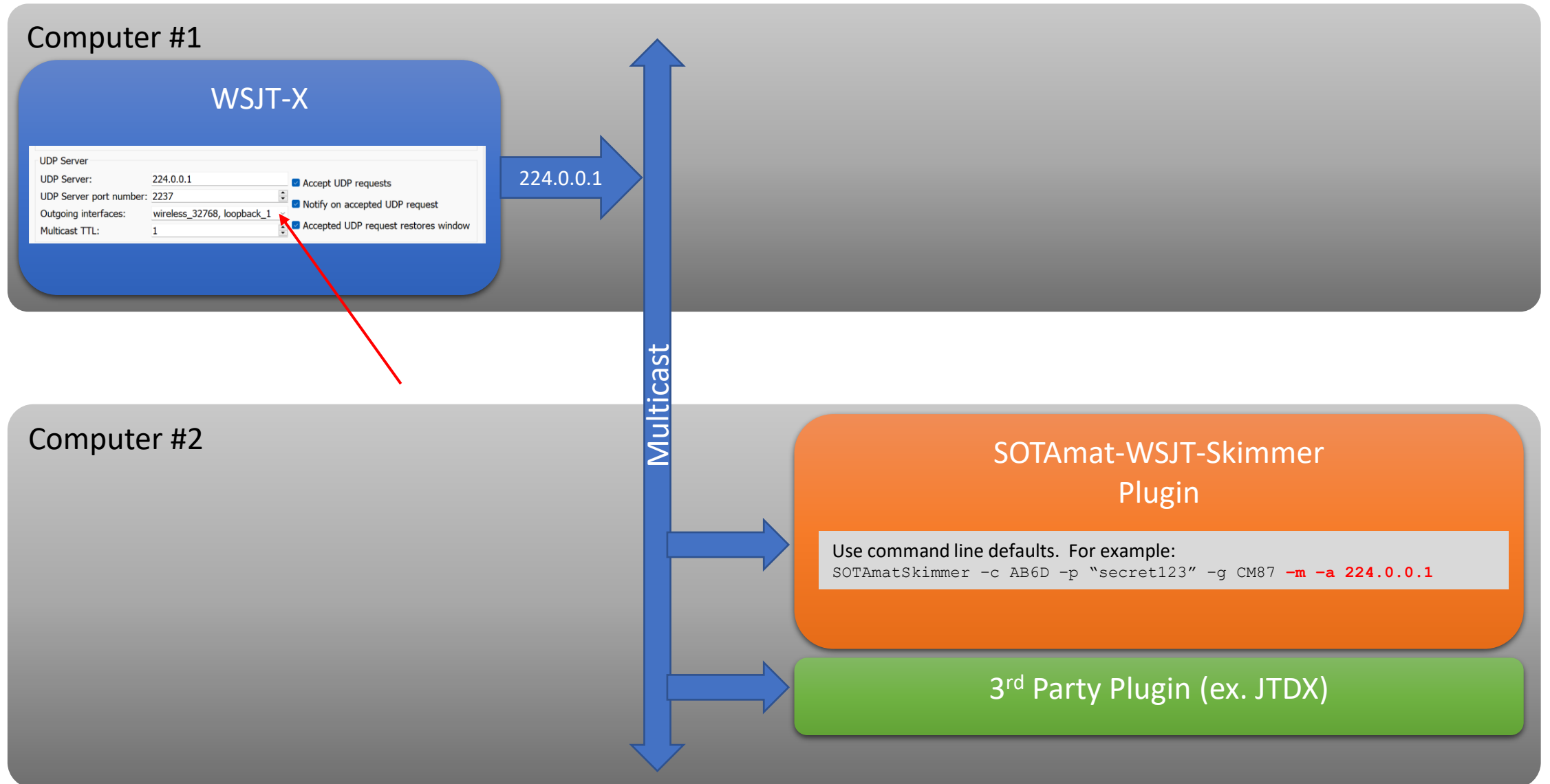


Note: enable all your interfaces

Multicast connection: use IP address in range 224.0.0.1 thru 224.0.0.255
single computer, two instances, two plugins each instance



Multicast connection: use IP address in range 224.0.0.1 thru 224.0.0.255
two computers, one instance, two plugins



Connecting to a Remote SDR with <http://websdr.org> (part 1)

1. Install “VB-Audio Virtual Cable” from <https://vb-audio.com/Cable/>
Configure it so the Windows system audio goes to the Virtual Audio Cable output.
2. **Optional:** if you want to connect to multiple radios you’ll need to direct multiple web browser instances to different virtual sound cards, with each directed at a different WSJT-X instance.
Here is software that does that: <https://github.com/audiorouterdev/audio-router>
3. Configure WSJT-X to:
 - Configure your settings to use a “NONE” radio, and VOX PTT.
 - Use the Virtual Audio Cable as the microphone input,
 - Because there is no real radio to get frequency info from, manually select the band/frequency that you will tune your remote SDR radio to inside WSJT. WSJT will give this frequency to SOTAmat when reporting.
4. Run the SOTAmatSkimmer.EXE with appropriate parameters
5. Configure your web browser for WebSDR:
 - Install a browser plugin such as “Auto Refresh Plus” to refresh the browser page automatically every ~90 min.
 - Change the browser settings to white-list audio auto-play for your selected SDR URL (to eliminate the need to click)
[Normally browsers block media (audio) auto-play. Since the “Auto Refresh Plus” will reload the page, you need auto-play to work. The setting is different for every browser, and in most browsers is a very hidden feature.
You will need to research (Google) how to white-list auto-play allowed URL’s for your specific browser, and add your WebSDR’s set of URL’s.]
 - Set the WebSDR site and URL parameters as the default web page to load on launch of the browser.
 - The URL parameters should specify the frequency/mode to use on launch (“?tune=14074usb” or just “?tune=14074”)

Audio Delay: Remote SDR with <http://websdr.org> (part 2)

- **NOTE: Audio Delay is an issue.**

Since the audio from the remote receiver is getting compressed, transmitted over the internet, received by the browser, decoded, shuttled through Virtual Audio Cable, and finally delivered to WSJT-X there may be significant DELAY between the atomic clock time WSJT-X uses (from your computer Operating System NTP time synchronization) and the received audio which has been delayed. This can reduce decode effectiveness.

- Problem:

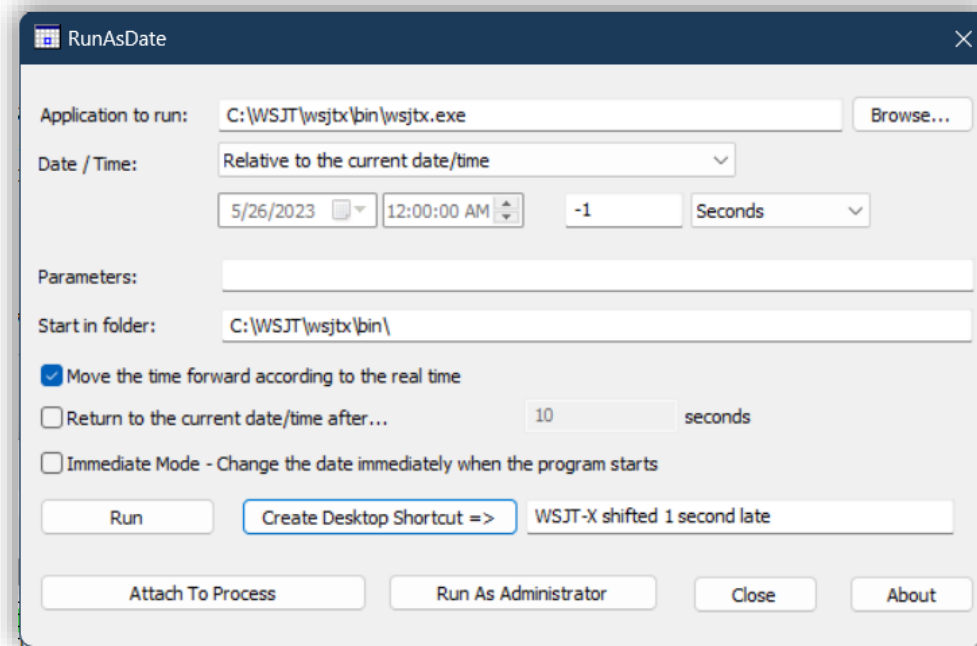
- We can't add negative delay to an audio stream, thus we need to add a positive delay to the processing by WSJT-X.
- WSJT-X doesn't have a setting for adding a processing delay. It simply uses the computer's system time.
- Many of the operating system time sync programs (Dimension 4, Meinberg, NTPd, Windows Time Sync, etc.) are meant to keep the system clock as accurate as possible. None of these programs (to my knowledge) allows a time offset from the actual time.

Correcting Audio Delay: Remote SDR with <http://websdr.org> (part 3)

- Solution #1:

Use the app “RunAsDate” https://www.nirsoft.net/utils/run_as_date.html

- Pros: it leaves the system time as actual time, but it will report a different time to a single application (WSJT-X!)
- Cons: The app only allows integer second offsets. “-1” is OK, “-0.8” does not work.
- Cons: you can’t run WSJT-X directly, you must configure the “RunAsDate” app to launch it for you.
- Must use in combination with a system time correction service (Meinberg, Dimension4, BktTimeSync, etc.)
- Configure “RunAsDate” to add a relative offset. My example settings (I use the Desktop Shortcut to launch WSJT-X):

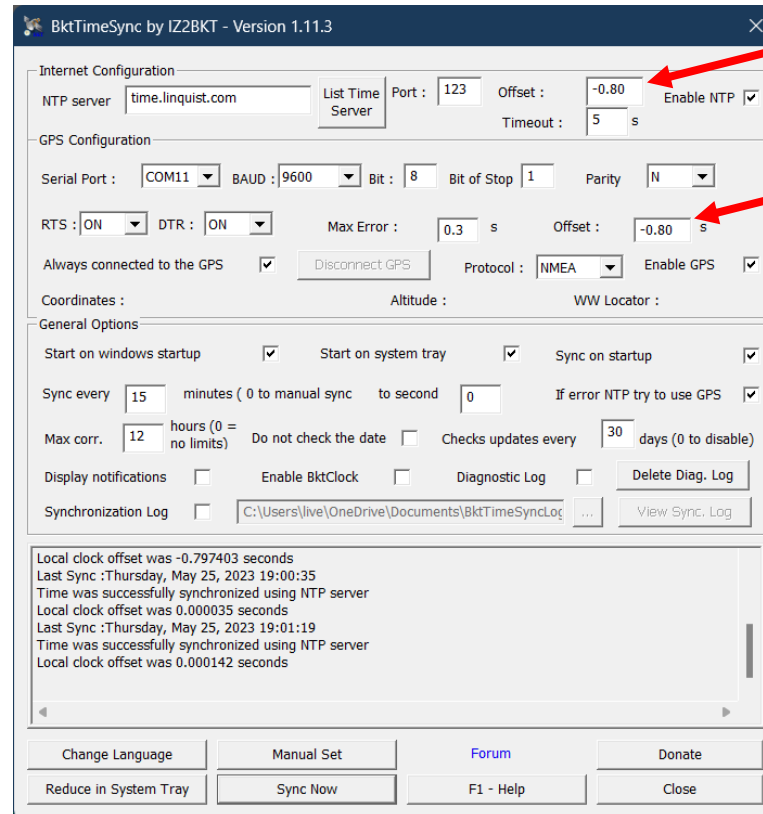


Correcting Audio Delay: Remote SDR with <http://websdr.org> (part 4)

- Solution #2:

Use the app “BktTimeSync”: <https://www.maniaradio.it/en/bkttimesync.html>

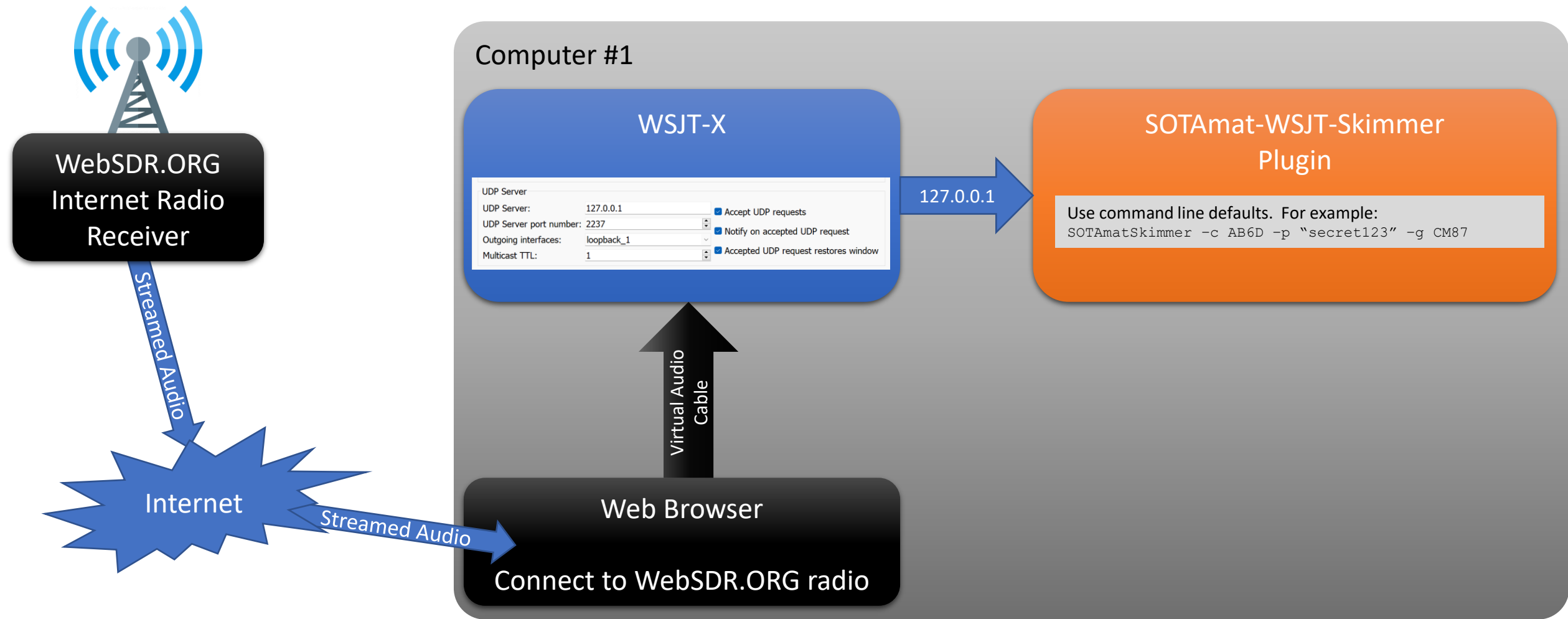
- It is a system time correction service (connects to NTP server), but allows for an offset to be applied!
- Cons: it applies the “wrong” time to all applications, not just the one WSJT-X instance you want.
- Pros: allows for the offset to be a fraction of a second! For example: -0.8 seconds
- Pros: you don’t need to configure it to launch WSJT-X since it is setting the system time. Launch WSJT-X yourself!



Other solutions to the Audio Delay issue (part 5)

- There is a Windows application called “**JTSync**” <http://www.dxshell.com/jtsync.html> that collects many FT8 messages decoded by WSJT-X and computes the median & average time sync offset for the received messages. The app allows you to measure the actual audio delay.
- The app also allows you to apply a **one-time** system-wide time shift to the operating system clock
- This would be a fantastic solution except for some issues:
 - The app requires human input to work, it can’t be configured to automatically apply a time offset after launch.
 - Since the app shifts the system-time, you must disable all other time sync programs (Dimension 4, Meinburg, NTPd, Windows built-in Time Sync, etc.) since you WANT your system time to be “wrong” by the amount JTSync computes for you... those other apps will just shift the time back to accurate time!
 - When you turn off time sync programs, the system will start to drift again. There is no automatic periodic correction built into JTSync. This means the program will only work properly for a few hours or so.
 - The default configuration needs to be modified to remove the network based time servers (removing that sync option) since otherwise the app will sync the clock to the actual time.
 - Both JTSync and SOTAmatSkimmer need to connect to WSJT-X at the same time, thus you can’t use WSJT-X’s default Unicast network configuration (127.0.0.1). You must re-configure all 3 applications (WSJT-X, SOTAmatSkimmer, and JTSync) to use a shared multicast UDP address on the network.

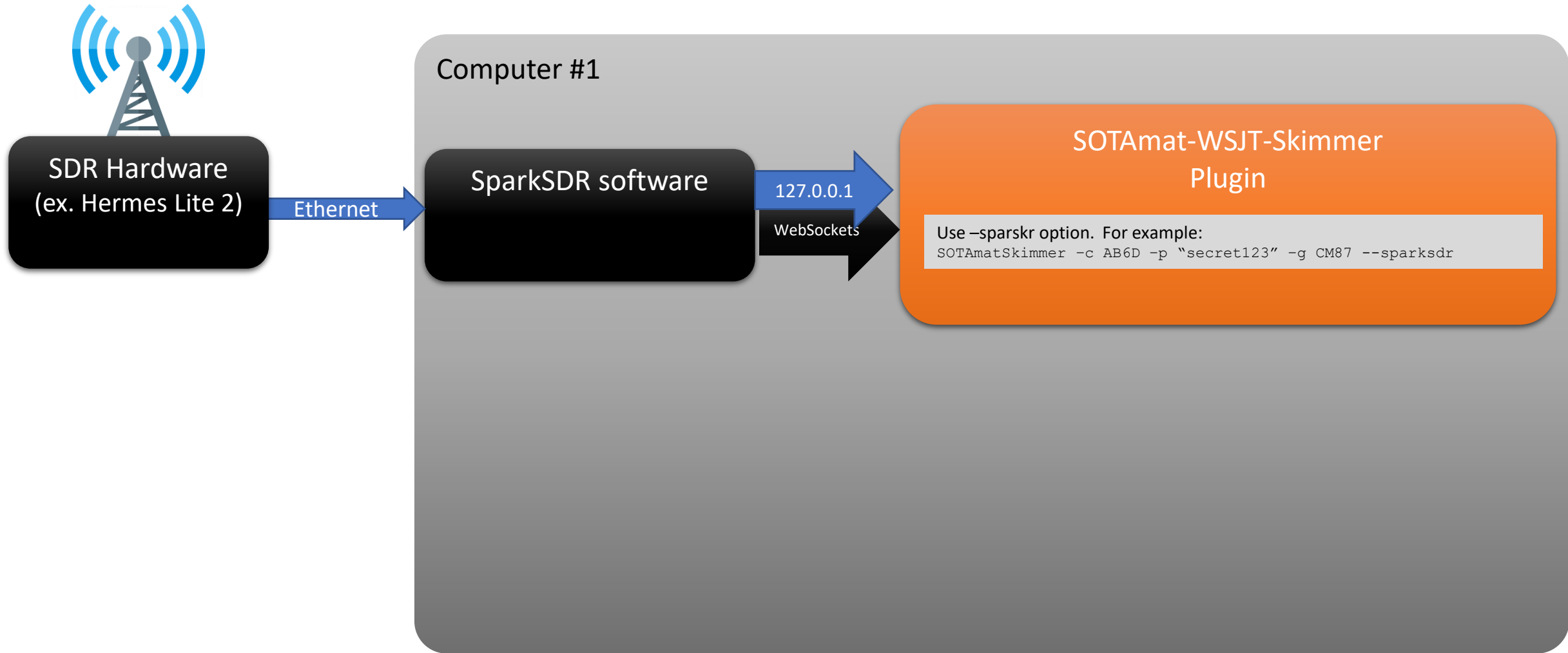
Connecting to a remote WebSDR radio



Reducing Spotting latency with SparkSDR

- SparkSDR is software that can work with hardware software defined radios (HW SDR's) such as the "Hermes Lite 2" radio. The combination of SDR HW and SparkSDR is that you can skim (monitor) multiple bands and multiple modes (FT8, FT4, PSK31, etc.) simultaneously (each mode/band combination is a "slice") from a single radio with a single computer.
- SparkSDR is special in that it decodes SOTAmat's FT8 "Free Text" messages and reports them to PSK Reporter, which in turn sends them to SOTAmat's server for processing.
- The PSK Reporter protocol requires skimmers like SparkSDR to cache 5 minutes worth of reception reports before sending the bundle to the PSK Reporter server. For SOTA/POTA this (up to) 5 minute delay is not always desirable.
- SOTAmatSkimmer plugin can connect to SparkSDR directly via a special SparkSDR websocket connection and receive reception reports. Since the SOTAmat plugin immediately reports spots to SOTAmat, it has no 5-minute delay.
- The SparkSDR software can be configured (the default) to also report to PSK Reporter (redundantly) or not. The SOTAmat server knows how to ignore duplicate reports from different sources already.
- SOTAmatSkimmer version 1.1.1.12 added direct support for SparkSDR

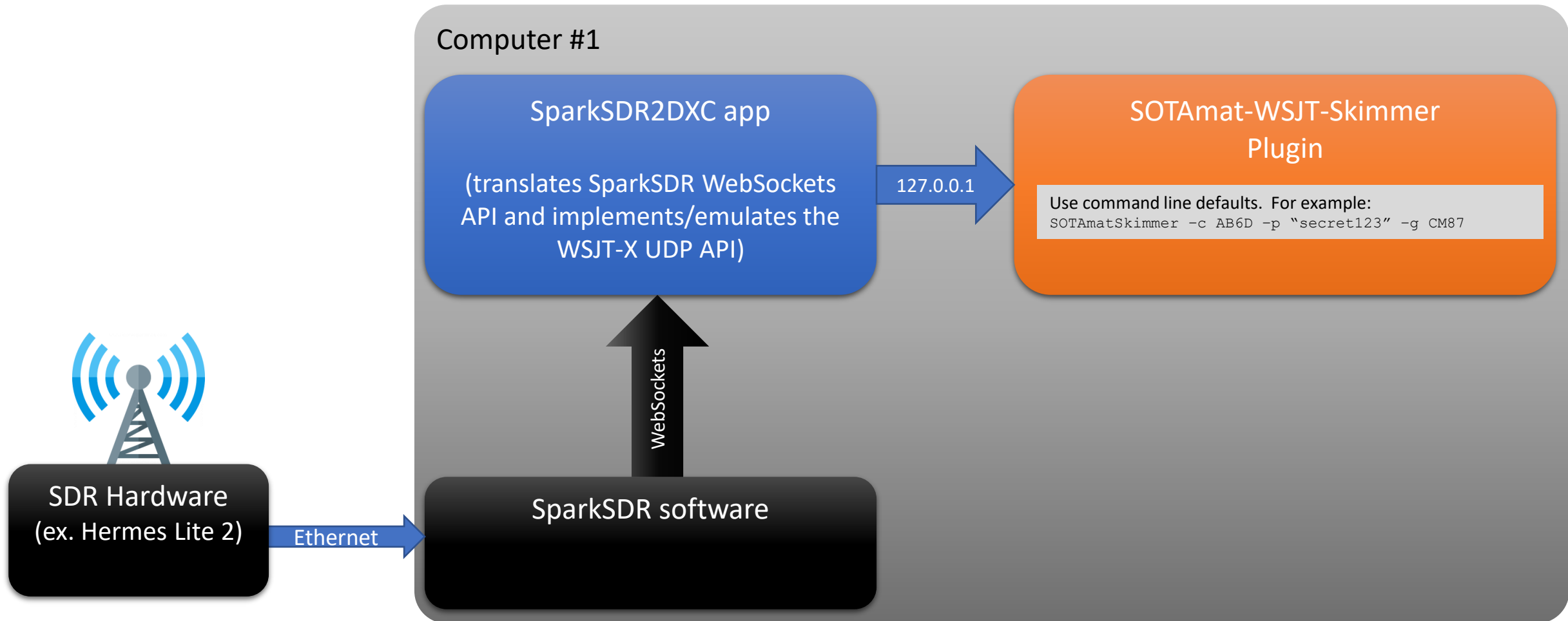
Connecting to SparkSDR via SOTAmatSkimmer (New way)



Reducing Spotting latency with SparkSDR: OLD WAY

- SparkSDR has an optional module (program) that allows SparkSDR to emulate/translate the UDP API of the popular WSJT-X application. This means plugins designed to connect to WSJT-X can also connect to the SparkSDR emulator/translator module.

Connecting to SparkSDR via WSJT-UDP emulator/translator: OLD WAY



More Information on SparkSDR UDP emulator (OLD WAY)

- Discussion thread (search for “UDP”): <https://groups.google.com/g/sparksdr/c/SgclcMcthaY?pli=1>
- SparkSDR adapter download:
<https://drive.google.com/file/d/1xhoOb7ISHz1IZG9hemlVf7O67FKalVMK/view>
- Main SparkSDR web site (documentation on SparkSDR and the main download):
<https://www.sparksdr.com/>