In this blog post, I will cover a few related but also different topics around X2Go - the GNU/Linux based remote computing framework.

**Introduction and Catch Up**

For those, who haven't come across X2Go, so far... With X2Go [0] you can log into remote GNU/Linux machines graphically and launch headless desktop environments, seamless/published applications or access an already running desktop session (on a local Xserver or running as a headless X2Go desktop session) via X2Go's session shadowing / mirroring feature.

**Graphical backend: NXv3**

For several years, there was only one graphical backend available in X2Go, the NXv3 software. In NXv3, you have a headless or nested (it can do both) Xserver that has some remote magic built-in and is able to transfer the Xserver's graphical data to a remote client (NX proxy). Over the wire, the NX protocol allows for data compression (JPEG, PNG, etc.) and combines it with bitmap caching, so that the overall result is a fast and responsive desktop experience even on low latency and low bandwidth connections. This especially applies to X desktop environments that use many native X protocol operations for drawing windows and widget onto the screen. The more bitmaps involved (e.g. in applications with client-side rendering of window controls and such), the worse the quality of a session experience.

The current main maintainer of NVv3 (aka nx-libs [1]) is Ulrich Sibiller. Uli has my and the X2Go community's full appreciation, admiration and gratitude for all the work he does on nx-libs, constantly improving NXv3 without breaking compatibility with legacy use cases (yes, FreeNX is still alive, by the way).

[2]