Schedule Update:

Everything is going as scheduled so far, I was able to test Jellyfin on 4 different platforms (windows browsers, IOS app, Roku, Windows native app). And I had some tests with ffmpeg which is used by Jellyfin for transcoding. I have tested hardware acceleration options in Jellyfin and I will be comparing them to the manual ffmpeg test result.

technical challenges met, solved and remaining.

I'm using a system from 2012 as my Jellyfin server, this makes the tests painfully slow but also is a perfect demonstration on hardware support limits. Also I'm not able to test Jellyfin on all supported platforms since I don't have access to them.

adjustment to the proposal

the proposal suggests that open source and free development is one of the reasons of Jellyfin limitations, however since the proposal the Jellyfin has expanded its team by requesting more people to join the development team (link). This issue will not be addressed.

More Updates:

The main solution for Jellyfin will be a form of passive transcoding while the system is not being used, this should be advertised as an option

which makes video playback much better but comes with computational and space issues.