

Portfolio

As an autodidact, I find the best ways to keep up with technology is to create new projects. Anything from small libraries to large distributed platforms. To me, they are all learning opportunities. Each project will see the cutting board often. Whether it's a cleaning the code (like PHPBenchTime and ContextDeleter), or complete scrap and rebuild (like

Each project will see the cutting board often. Whether it's a cleaning the code (like PHPBenchTime and ContextDeleter), or complete scrap and rebuild (like [Sphire](#) and ServerStatusEmitter).

Below, I detail a few of the projects I've created - as well as the road it took to where it is (or where it's going).

[Sphire](#) has been a labor of love - and frustration, which I initiated around 2015. It has seen three scrap-and-rebuild cycles as of this writing (May 2017). What started out as a simple social network (think Facebook-lite) has morphed into a pet project of wonderful proportions. Boasting a central API bus that drives both the mobile applications and the web frontend. Behind that bus is a fairly well-architected backend written in a plethora of languages (PHP, Go, and Python are some of the standouts) and utilizing a mix of databases (MySQL, MongoDB, Aerospike).

The first iteration of [Sphire](#) was a monolithic Laravel4 application. The second iteration became a vanilla PHP application (still monolithic) with stored MySQL procedures driving all CRUD operations. The third iteration was a pure Python and MongoDB application. The fourth, I've described above.

That being said, each of those iterations came with changes in what the platform was and what it did - not just the tech driving it.