Tier2 Submit Web Application Challenges

A web-based version of Tier2Submit presents a number of challenges – some for developers, some for states that host the application, and some for both.

Overview

Producing a full-fledged web-based version of Tier2 Submit would likely require us to abandon the existing codebase and start from scratch with a new programming language and database system. It would be a major undertaking even if we hosted the application, but having to target multiple operating systems and server configurations raises the software's complexity at least one order of magnitude beyond the desktop Tier2Submit. We know of no federal agency that has attempted such a project.

For states, the challenge would be in assuming the task of hosting and administering the application. This would involve hardware, software, and personnel requirements.

Deployment

Downloading and installing the existing desktop Tier2 Submit requires little technical expertise, and one can begin using the program immediately. A web application, however, has many more "moving parts" that must be properly configured. We would aim to simplify this process as much as possible by packaging the application inside a virtual machine or application container. Nevertheless, installation would have to be done by an experienced system administrator, who would also perform some additional configuration.

User management

In the current Tier2 Submit, all users have full rights to view and edit all data. In contrast, users of the web application would log in with a username and password that will determine the rights which they are granted.

While the application must provide some level of automation for account signup, password changes, etc., it would still be incumbent on someone to, at minimum, approve account requests and reinstate disabled accounts. Note that users would not be limited to submitters; they would also include LEPC members, for example.

Security

This is another consideration not faced in the desktop environment. With web servers under attack, web applications must take care to avoid vulnerabilities, to detect and take action against hacking attempts, and to maintain a logfile of login attempts. In addition to monitoring and acting on that information, system administrators may choose to add intrusion-prevention software.

Scaling Resources

When many users are accessing the system simultaneously, performance may deteriorate, and users may experience long waits for the system to respond. To avoid this, the system administrator must monitor disk usage and CPU load, and if necessary add servers or swap out a CPU for a faster one. Such measures could also be taken when high demand is anticipated, such as near the March 1 deadline. To the extent possible, the application would be designed to permit such flexibility. As an alternative, some states may choose to distribute

the load over two or more deployments, such as one for facilities in the west part of the state and another for facilities in the east.

Tradeoffs

It is possible to reduce the burden on states by adding bells and whistles to the software. Some desired features, however, may conflict with others. As with any software, the benefits of adding a feature must be weighed against possible negative impacts on system stability and performance. Finally, we may lack the resources to add everything we'd like, especially in an initial release.