# HOW EMBRACING A COMMON TECH STACK CAN IMPROVE THE LEGACY SOFTWARE MIGRATION EXPERIENCE

*C. D. Burgoyne, C. R. Albiston, R. G. Beeler, M. Fedorov, J. J. Mello, E. R. Pernice, M. Shor*
***Lawrence Livermore National Laboratory (LLNL)***

## Abstract

Over the last several years, the National Ignition Facility (NIF), the world's largest and most energetic laser, has regularly conducted approximately 400 shots per year. Each experiment is defined by up to 48 unique pulse shapes, with each pulse shape potentially having thousands of configurable data points. The importance of accurately representing small changes in pulse shape, illustrated by the historic ignition experiment in December 2022, highlights the necessity for pulse designers at NIF to have access to robust, easy to use, and accurate design software that can integrate with the existing and future ecosystem of software at NIF. To develop and maintain this type of complex software, the Shot Data Systems (SDS) group has recently embraced leveraging a common set of recommended technologies and frameworks for software development across their suite of applications. This paper will detail SDS's experience migrating an existing legacy Java Swing-based pulse shape editor into a modern web application leveraging technologies recommended by the common tech stack, including Spring Boot, TypeScript, React and Docker with Kubernetes, as well as discuss how embracing a common set of technologies influenced the migration path, improved the developer experience, and how it will benefit the extensibility and maintainability of the application for years to come.

## INTRODUCTION

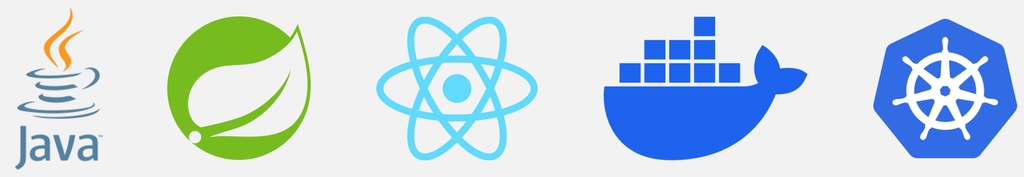
- Pulse Shape Editor (PSE) was a legacy desktop application used to design pulses for use on NIF
- PSE was determined to require a migration to new technology due to security vulnerabilities and the difficulty to update in-place due to architectural decisions by original developers. The new application would be called Pulse Shape Tool (PST)
- Creating a new application in today's technology ecosystem can be difficult to research and bootstrap due to the proliferation of solutions available

## REQUIREMENTS

- Single page web application
- Edit and save data in a database
- Allow import and export of data to and from the database and Excel or CSV
- Plot and compare multiple pulse points and graph in GUI in real-time
- Complex data integrity checking
- Permission management
- Ability to easily integrate with other internal applications using REST APIs

## CHOOSING A TECHNOLOGY STACK

- There are an incredible number of solutions available to create web applications
- Recommended tech stacks, also known as endorsed tech stacks or common tech stacks, is one strategy that can be used to streamline this decision
  - —Recommended tech stacks are sets of technology endorsed by a group that aims to provide a guarantee that choosing a given set of technologies is supported in advance and that there is sufficient developer knowledge and support within an organization for the technologies
- In 2020, an architectural board within SDS released a set of various endorsed tech stacks based on their prior successful application within SDS
- PST developers chose the initial tech stack based on the recommendations provided by the board. Technologies included:
  - —Java
  - —Spring Boot
  - —Oracle DB
  - —React
  - —KendoReact
  - —Docker
  - —Kubernetes




## MODIFYING THE TECHNOLOGY STACK

### Switching KendoReact for AG Grid React

- Developers used KendoReact in the past, and felt it was difficult to customize in the ways necessary for typical custom use cases in our internal applications
- PST only needed to use the grid from KendoReact and asked: Could there be something better for our use case?

| Solution | Positives | Negatives |
|---|---|---|
| KendoReact | - PST developers had experience<br>- History of use on internal applications<br>- Many customizable components available | - Requires license<br>- Needed many customizations |
| PrimeReact | - History of use on internal applications<br>- Many customizable components available<br>- Free version available with enterprise level add-ons | - No past PST developer experience<br>- Needed many customizations |
| AG Grid React | - Most popular by npm downloads (Fig. 1)<br>- Excellent documentation<br>- Prototyping showed most necessary functionality for requirements was built in<br>- Easy to customize<br>- Free version available with enterprise level add-ons | - No past PST developer experience<br>- Only a grid, could not be leveraged for additional components |

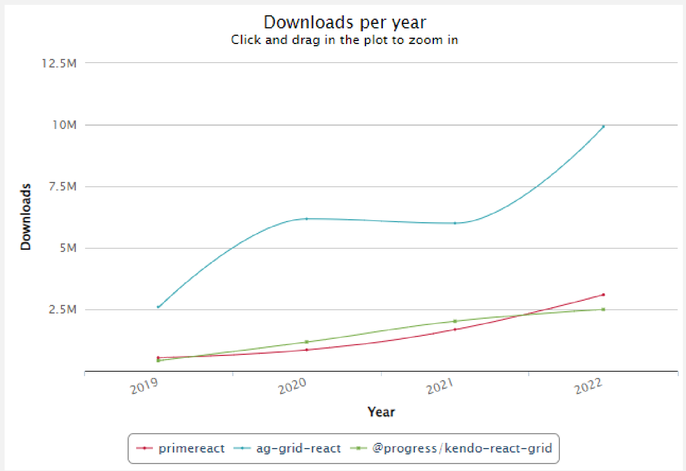



Fig. 1: historical npm download statistics for primereact, ag-grid-react, and @progress/kendo-react-grid packages from 2019 - 2022

- Based on prototyping, and consideration of pros and cons, developers decided to use AG Grid React

AG Grid

### Adding ReCharts

- PST must be able to plot pulses in real-time and without a comprehensive component library, a charting library needed to be added for this purpose
- Four options were evaluated, and two moved on to prototyping: nivo and recharts

| Solution | Positives | Negatives |
|---|---|---|
| Recharts | - Highest usage based on npm downloads (Fig. 2)<br>- Customizable<br>- Open source<br>- Excellent official documentation with detailed examples<br>- Examples included the zoom, pan and brush functionality necessary for requirements | - No past PST developer experience |
| Nivo | - Aesthetic<br>- Customizable<br>- Open source<br>- Good official documentation with interactive examples | - No past PST developer experience<br>- Complex<br>- Zoom, pan, brush functionalities were not implemented in the way we needed, no examples |

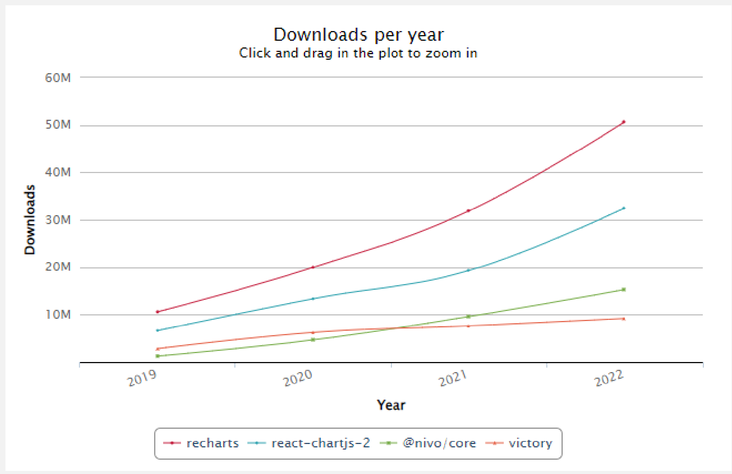



Fig. 2: historical npm download statistics for recharts, react-chartjs, @nivo/core and victory packages from 2019 - 2022

- Based on prototyping, and consideration of pros and cons, namely zoom, pan, and brush support at the time, developers decided to use Recharts

<Recharts />

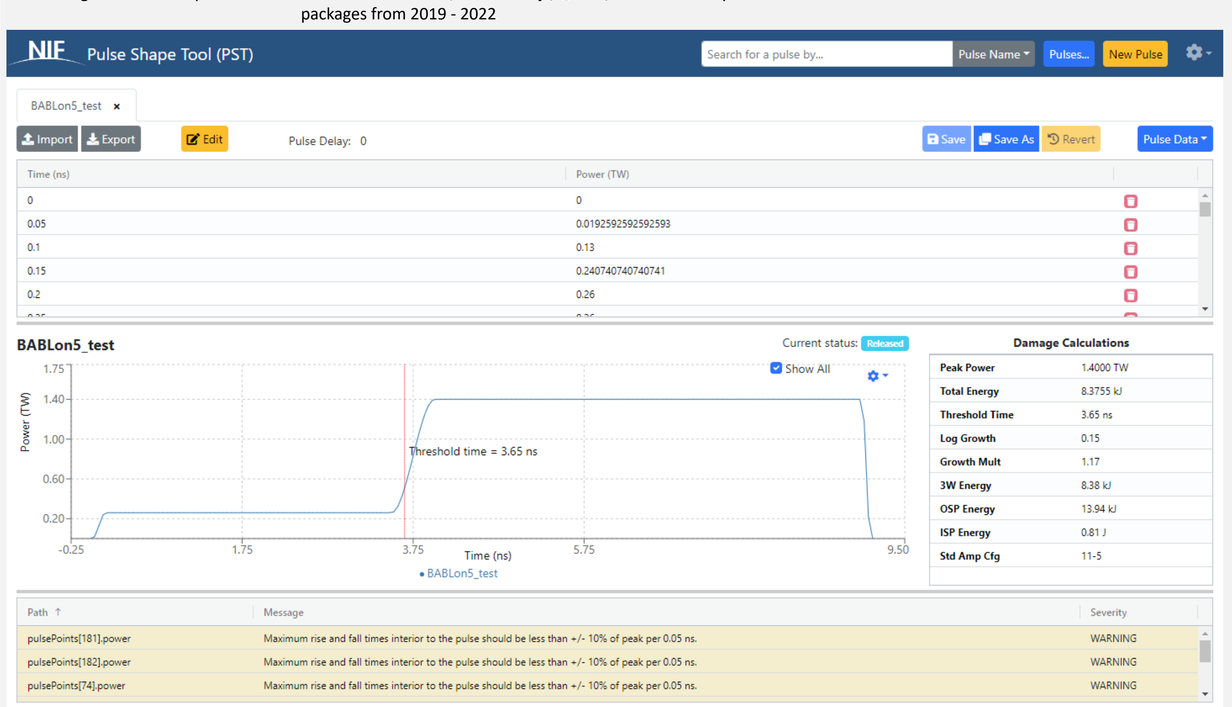



Fig. 3: Migration was completed and in early 2022, PST was delivered to customers

## LESSONS LEARNED

- Development for PST began in late 2020, PST was delivered to users in early 2022 (Fig. 3)
- Using the recommended tech stacks, of which developers could leverage prior experience, resulted in a development process that was fast and efficient and final product was found to be of high quality with very few bug reports
- There are still positives and negatives to consider before adopting recommended tech stacks

### Potential Benefits

- Reduced project research and bootstrapping times
- Simplified development of internal tools for future use such as seed projects and internal component libraries
- Streamlined update process for new versions
- Overall developer experience may be more enjoyable, resulting in more satisfied developers
  - —More time coding!

### Potential Drawbacks

- Updating for breaking changes can be time consuming when many applications use the same technologies
- Developers need to be proactive at research and ensure they are making modifications to best meet requirements

## CURRENT STATUS AND FUTURE WORK

- PST project is now in maintenance phase
- SDS continues to keep the PST tech stack up to date. Currently the following tasks are taking place:
  - —Upgrading Java from v11 to v17 (and later v21)
  - —Updating React to v18
- Future maintenance tasks for PST will include
  - —Updating to Spring Boot 6
  - —Updating authentication to use OAuth2.0
- Teams within SDS continue to use the idea of common recommended tech stacks for new applications and migrations, taking advantage of lessons learned from PST, such as seed projects and internal component libraries
- In late 2023, SDS will establish a new architectural review board to research and share knowledge with other developers on recommended tech stacks and their benefits, as well as other up-to-date developer best practices

LLNL-POST-854379


This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.