CSSI Element: C++ as a service - rapid software development and dynamic interoperability with Python and beyond

CaaS aims to provide programmers and data scientists a simple and general solution to language interoperability:

- Advance interpretative technology to provide scientists a state-of-the-art C++ execution environment
- Enable functionality to provide dynamic, native-like, runtime interoperability between C++ and Python
- Allow seamless utilization of heterogeneous hardware (e.g., hardware accelerators)
- To enable rapid application development even with a complex codebase

Our approach is to generalize a high-energy physics analysis code ("Cling") to a generally accessible and fully functional tool that is part of LLVM/Clang.

- LLVM community engagement / acceptance of CaaS concept and approach
- Initial release of Clang-Repl achieved in LLVM13
- Clang-Repl based plugin (Clad) implemented and demonstrated including offload of calculations to GPU
- LibInterop design completed after extensive community discussion. Now co-developing with application developers including
  - CPPYY package enabling run-time python <-> C++ bindings
  - Xeus based Jupyter plugin supporting interoperability and data exchange between C++ and python
- Science applications include automatic differentiation, uncertainty quantification, and embedded device control

Our approach is to generalize a high-energy physics analysis code ("Cling") to a generally accessible and fully functional tool that is part of LLVM/Clang.

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPPYY</td>
<td>run-time bindings are first libinterop use case</td>
</tr>
<tr>
<td>Clad AD</td>
<td>applied to statistical analysis problems</td>
</tr>
<tr>
<td>Compiler driven</td>
<td>Uncertainty Quantification</td>
</tr>
<tr>
<td>Case Study: Simpson’s</td>
<td>Rule</td>
</tr>
<tr>
<td>Results</td>
<td></td>
</tr>
</tbody>
</table>

Thanks to this project, we have grown a diverse user community around our technology including contributors from data science and industry. We established a monthly community meeting series to discuss results and applications. Visit us at https://compiler-research.org