Allowing MPI Tool Builders to Forget About Fortran

Soren Rasmussen, University of Oregon
Martin Schulz, Lawrence Livermore National Lab
Kathryn Mohror, Lawrence Livermore National Lab

The Background

- The MPI Profiling Interface allows tool users to intercept MPI routines and complete the routine using PMPI

  (Kathryn discussed this on Monday and made my job easier, thanks!)

- Simple example

```
int MPI_Init(int *argc, char ***argv) {
    int err;

    /* Insert Profiling Code */

    err = PMPI_Init(&argc, &argv);

    /* Wrap-up Profiling Code */

    return err;
}
```

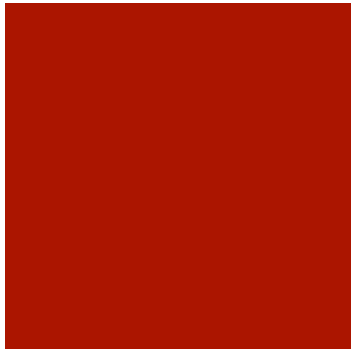
The Issue

- Profiling tools written in C intercept MPI calls from Fortran and have to handle all interoperability
- Lurking Fortran and C incompatibilities in MPI!
  - But things have been working fine, right?
    - Tools exist that assist with interoperability, such as wrap.py, would require MPICH (see next bullet)
  - Things might work, but it's not guaranteed in the standard
- Paper "*Implementing the MPI-3.0 Fortran 2008 Binding*" by J. Zhang, B. Long, et al. discusses MPICH's solution to interoperability
  - MPICH specific solution

The Incompatibilities

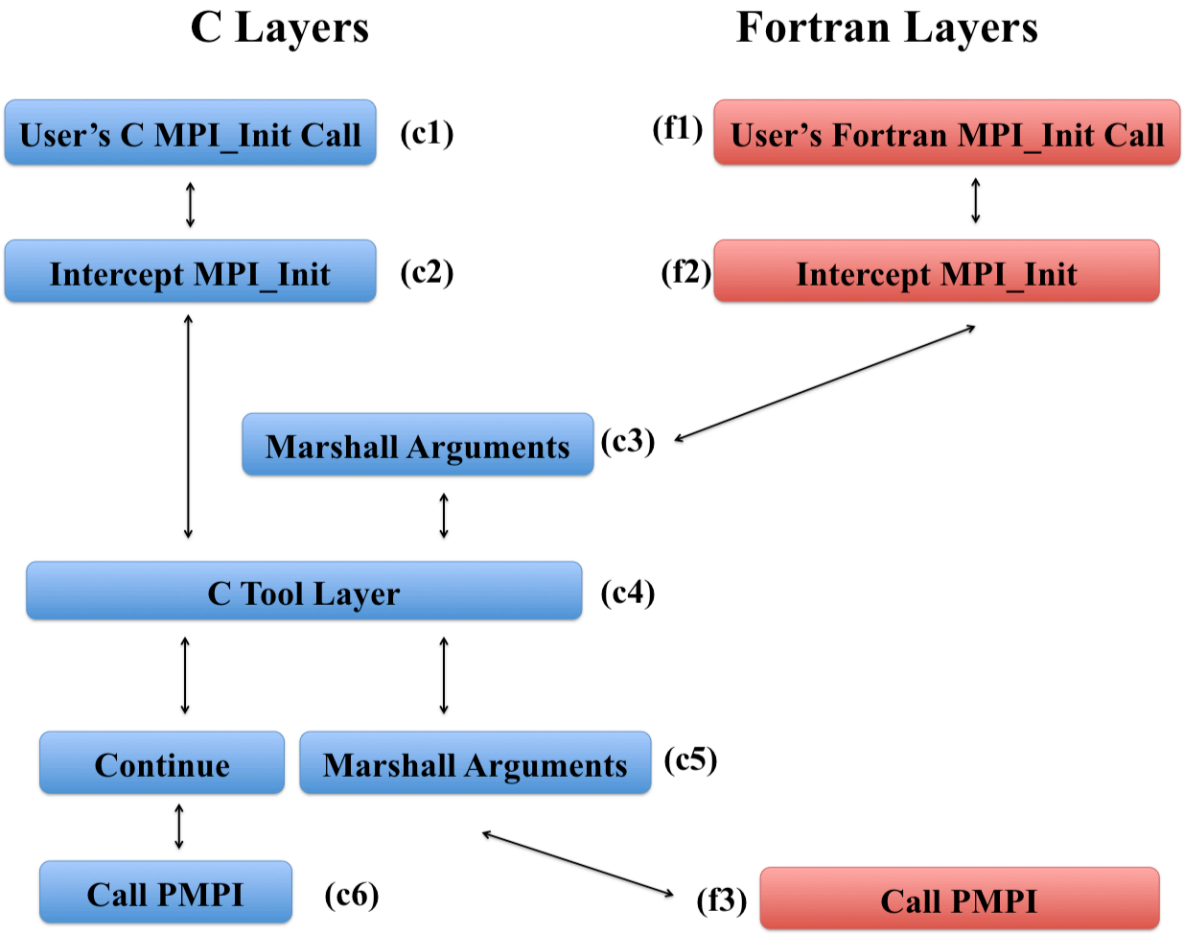
- Logical Type
- Error Return Types
- Fortran Only Routines
- Procedures and Callbacks
- Attributes
- Descriptors and Arrays
- Marshaling of MPI Handles
- Constants
- Character Strings

WMPI: a prototype wrapper solution

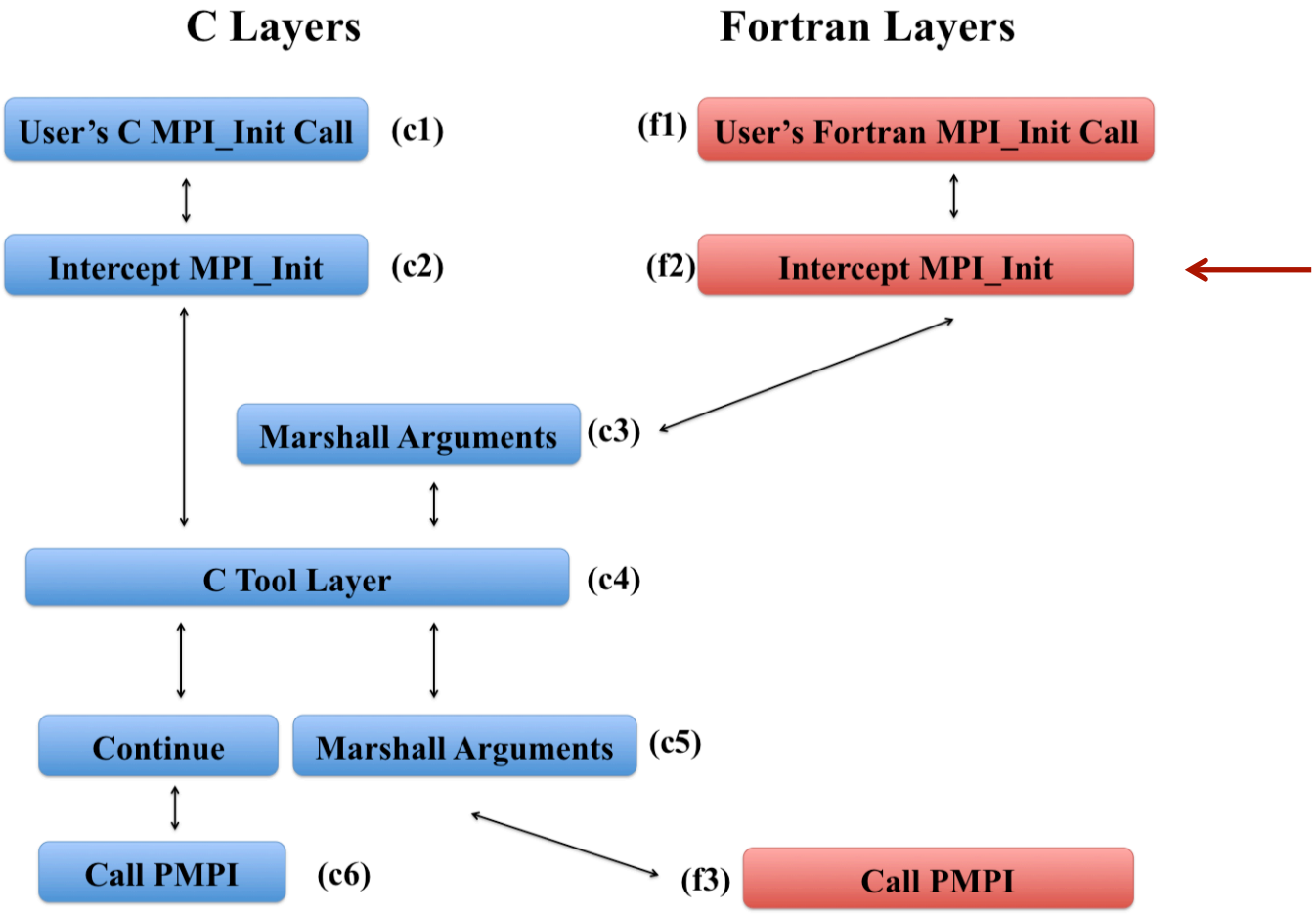


- WMPI generates wrappers that deal with the interoperability issues
  - Stratego/XT is used to create Abstract Syntax Tree (AST) from MPI Fortran 2008 header file
    - WMPI parses the AST, transforms the AST, and generates wrappers
- Handles all marshaling of arguments between Fortran and C layers
- Once TS 29113 is fully implemented, will provide access to Fortran's array descriptor to C

Wrapper Architecture



Wrapper Architecture



1. Intercept MPI Routine (f2)

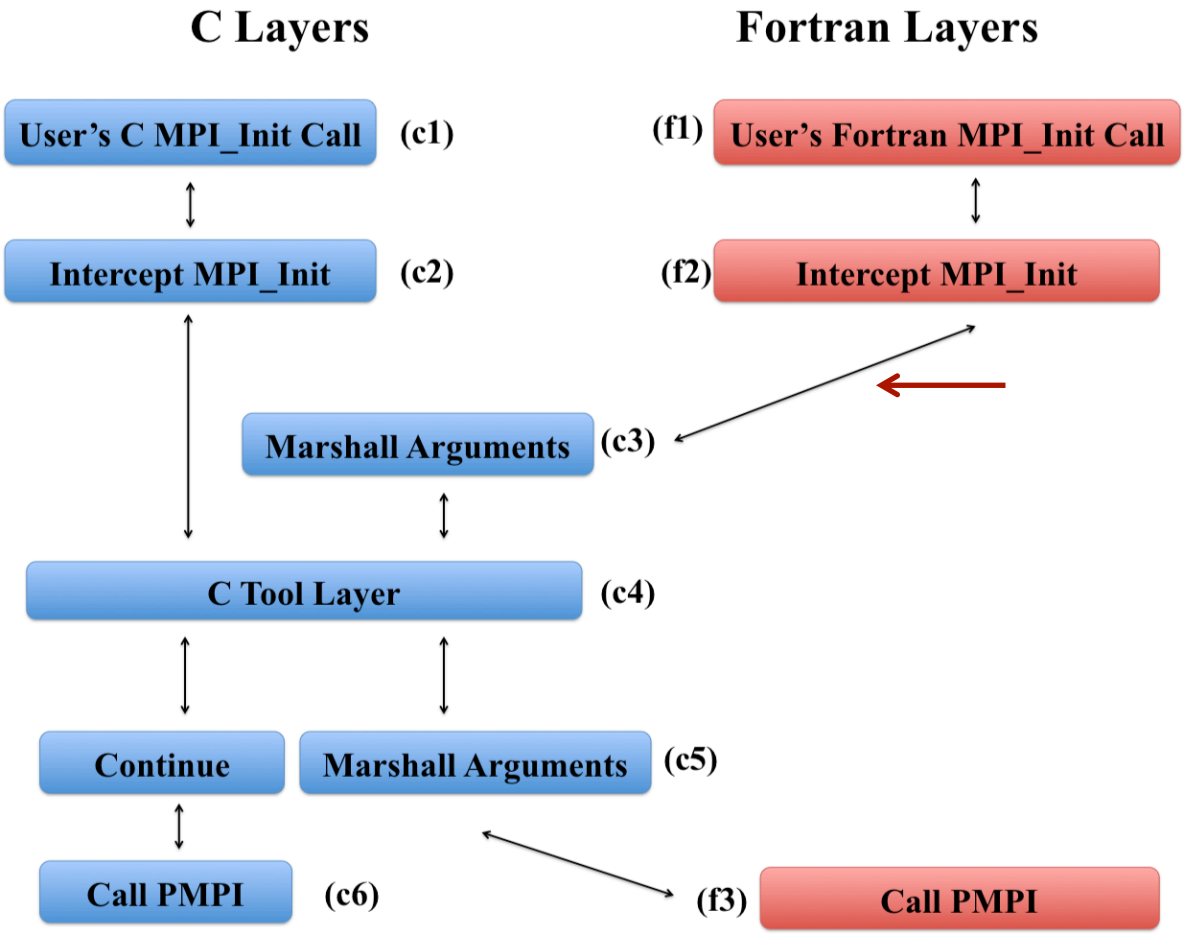
Made-up and simplified MPI routine for example

```
subroutine MPI_Routine(a_value, comm, ierr)
  integer :: a_value
  type(MPI_Comm) :: comm
  integer(C_INT) :: c_value, lang_info
  integer, intent(OUT), optional :: ierr

  c_value = a_value
  lang_info = getProceduralValue()

  WMPI_Routine_f(c_value, comm, lang_info)

  a_value = c_value
  if (present(ierr)) ierr = lang_info
end subroutine
```

Issues

- Optional ierr
- Procedural values

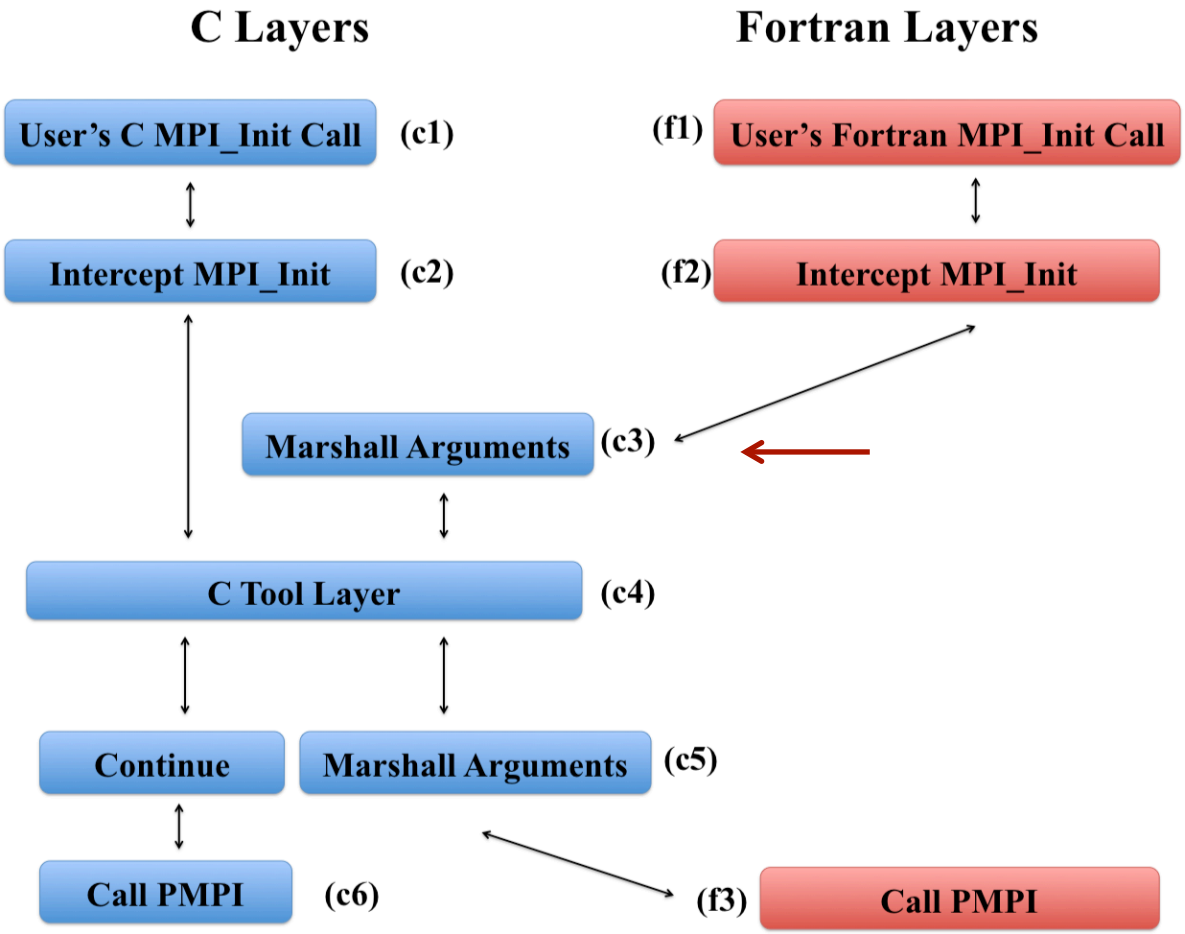
Wrapper Architecture



2. Pass to C (Interface)

```
interface
subroutine WMPI_Routine_f(a_value, comm, lang_info) &
  bind(c, name = "WMPI_Routine_f")
  implicit none

  integer(C_INT), intent(INOUT), value :: a_value, lang_info
  type(MPI_Comm), intent(INOUT) :: comm

end subroutine
end interface
```

Wrapper Architecture



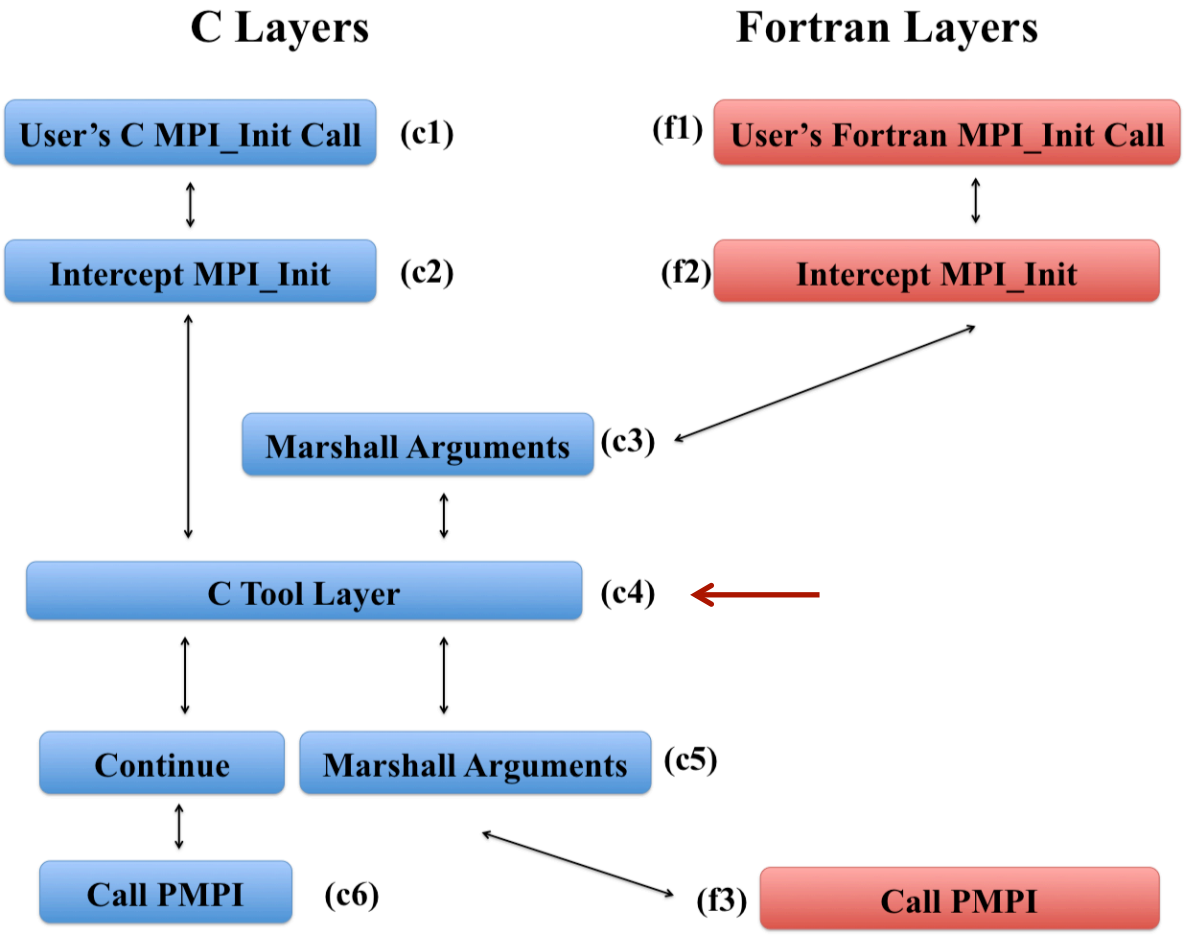
3. Marshall Arguments (c3)

```
int WMPI_Routine_f(int value, MPI_Fint *f_comm, int lang_info)
{
    MPI_Comm c_comm;

    c_comm = MPI_Comm_f2c(*f_comm);
    err = wmpi_routine(value, c_comm, lang_info);
    *f_comm = MPI_Comm_c2f(bc_comm);

return err;
}
```

Issues

- f2c/c2f functions: supposed to be called from C

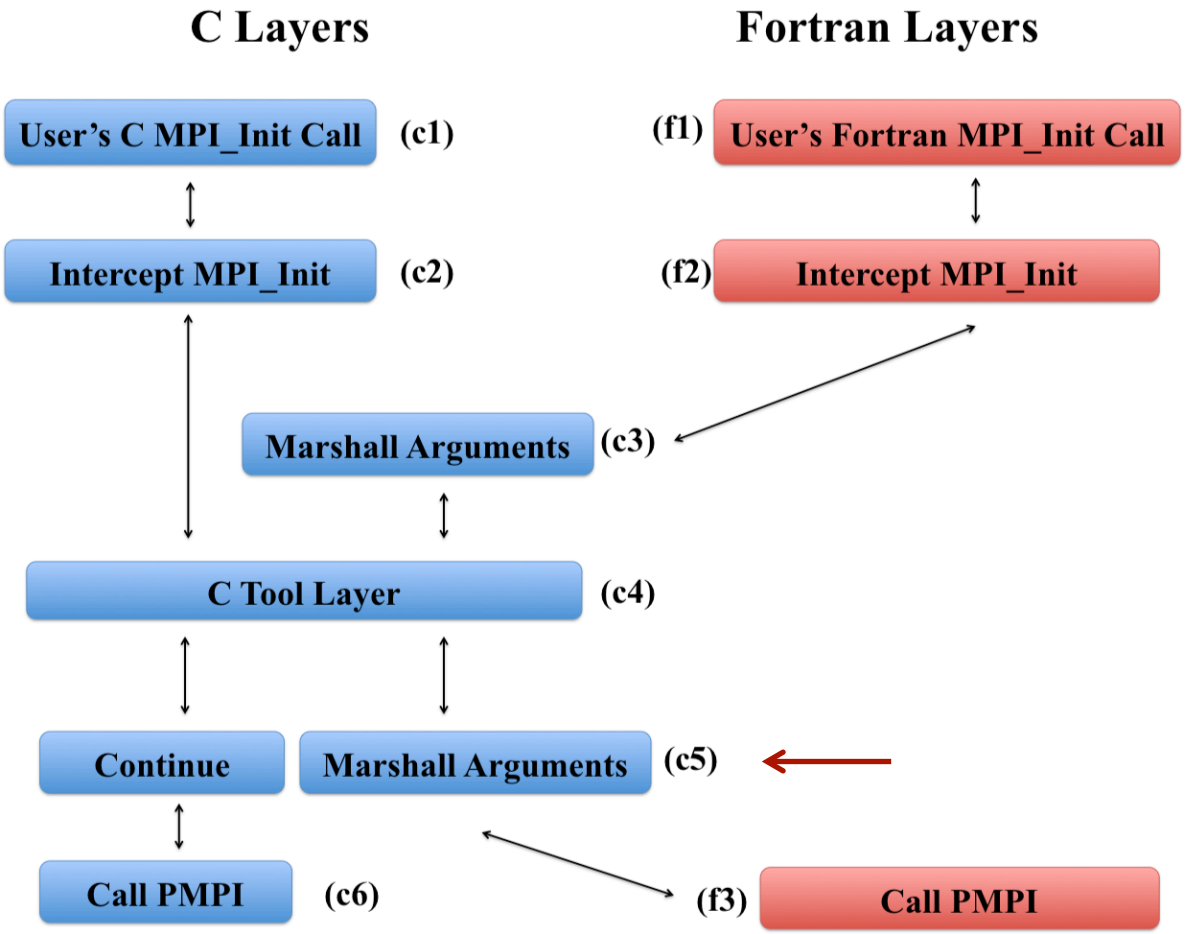
Wrapper Architecture



4. C Tool Layer (c4)

```
int WMPI_Routine(int value, MPI_Fint *f_comm, int lang_info)
{
  int ierror;

  /* Insert Profiling Code */
  ierror = WPMPI_Routine(value, f_comm, lang_info);
  /* Wrap-up Profiling Code */

  return ierror;
}
```

Wrapper Architecture



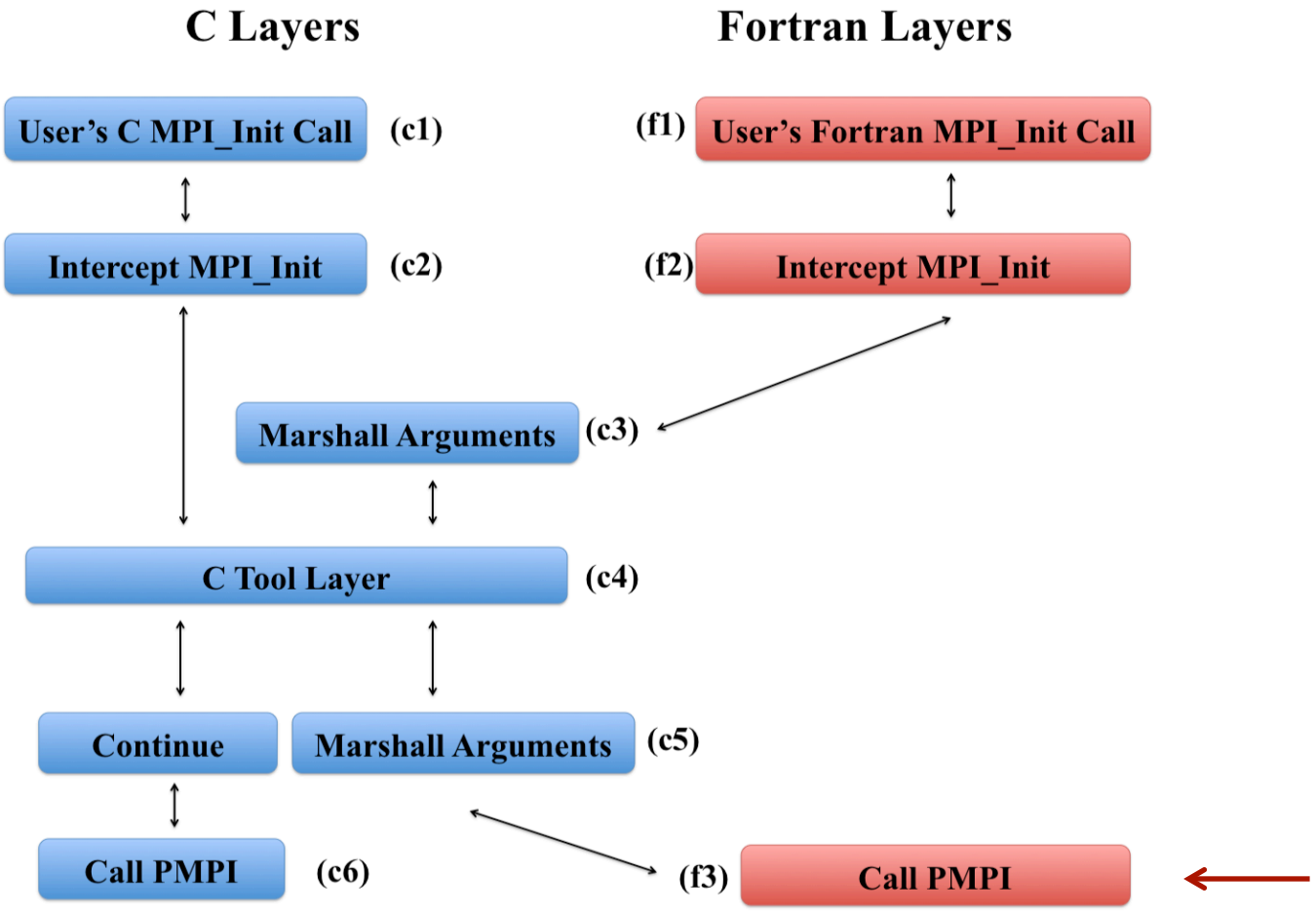
5. Marshall Arguments (c5)

- Marshall arguments back to Fortran variables
- Call PMPI with correct procedure postfix ('','_fts','_f08','_f08ts')
  - *fts* or *ts* indicates the routine has one or more choice buffer dummy arguments, i.e. TYPE(*), DIMENSION(..)
  - The MPI Standard states that callback procedure interfaces and callbacks are not interoperable with C and must be completed with the same postfix as original routine.

OR

- Call PMPI from C

Wrapper Architecture



6. Call PMPI from Fortran (f3)

```
Subroutine WPMPI_Routine_F(value,comm,lang_info) &
  bind (c, name=’WPMPI_Routine_F')
  implicit none

  type(MPI_Comm), intent(IN):: comm
  integer(C_INT), intent(INOUT), value :: value, lang_info

  call PMPI_Routine(value, comm, lang_info)

end subroutine PMPI_Comm_rank_F
```

Incompatibilities

- Logical Type
- Error Return Types
- Fortran Only Routines
- Procedures and Callbacks
- Attributes
- Descriptors and Arrays
- Marshaling of MPI Handles
- Constants
- Character Strings

Future Work

- Complete full implementation of prototype
  - Will require full adoption of TS 29113 by Fortran compilers
- Make Timing Runs
  - What is performance impact of wrappers?
- Explore additional functionality that wrappers can offer?

Thanks to

Lawrence Livermore National Laboratory

Department of Energy

Jeff Squares