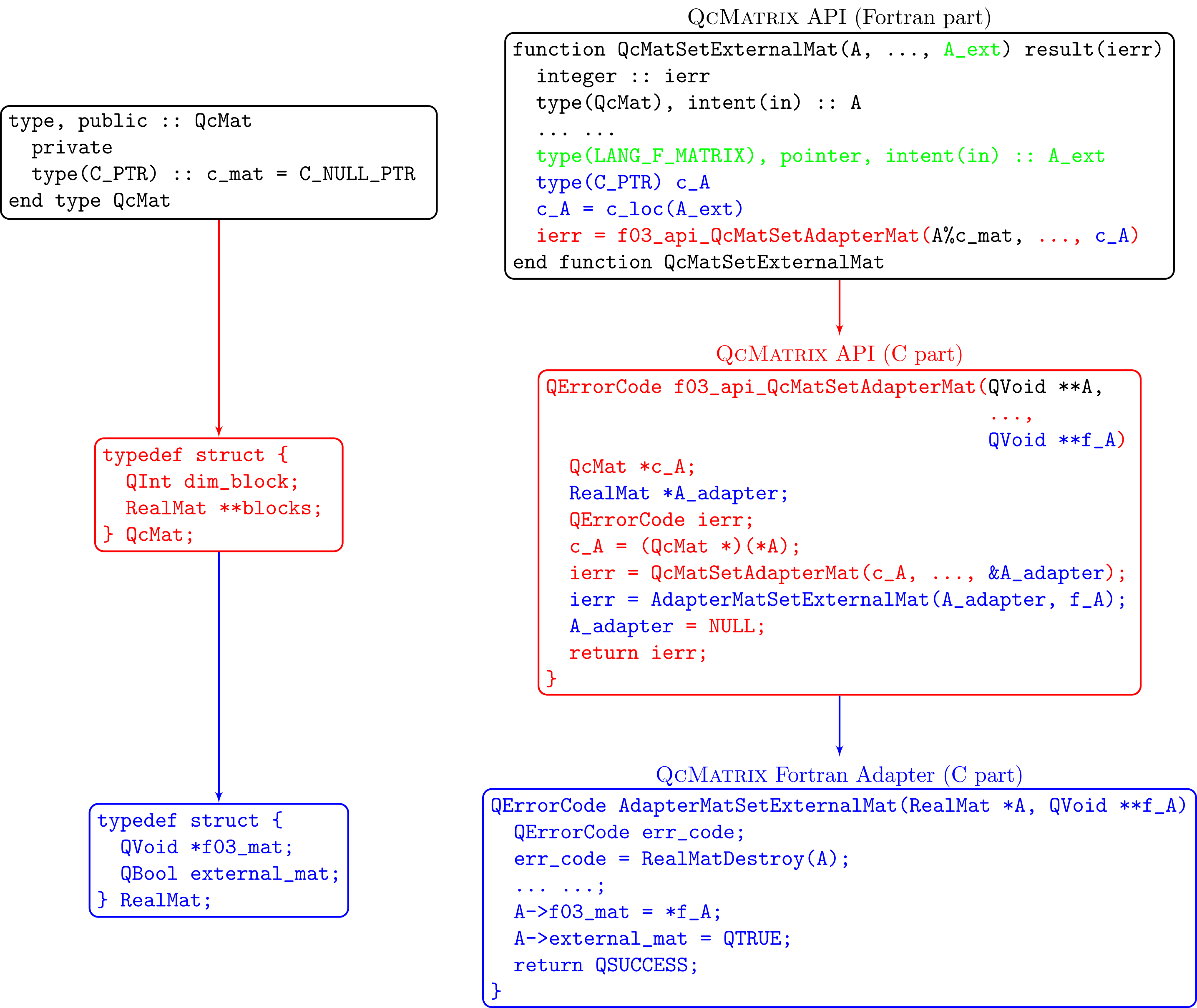

QcMatrix API (Fortran part)

```
function QcMatSetExternalMat(A, ..., A_ext) result(ierr)
  integer :: ierr
  type(QcMat), intent(in) :: A
  ... ...
  type(LANG_F_MATRIX), pointer, intent(in) :: A_ext
  type(C_PTR) c_A
  c_A = c_loc(A_ext)
  ierr = f03_api_QcMatSetAdapterMat(A%c_mat, ..., c_A)
end function QcMatSetExternalMat
```

```
type, public :: QcMat
  private
  type(C_PTR) :: c_mat = C_NULL_PTR
end type QcMat
```

QcMatrix API (C part)

```
QErrorCode f03_api_QcMatSetAdapterMat(QVoid **A,
                                      ...,
                                      QVoid **f_A)
  QcMat *c_A;
  RealMat *A_adapter;
  QErrorCode ierr;
  c_A = (QcMat *)(*A);
  ierr = QcMatSetAdapterMat(c_A, ..., &A_adapter);
  ierr = AdapterMatSetExternalMat(A_adapter, f_A);
  A_adapter = NULL;
  return ierr;
}
```

```
typedef struct {
  QInt dim_block;
  RealMat **blocks;
} QcMat;
```

QcMatrix Fortran Adapter (C part)

```
QErrorCode AdapterMatSetExternalMat(RealMat *A, QVoid **f_A)
  QErrorCode err_code;
  err_code = RealMatDestroy(A);
  ... ...;
  A->f03_mat = *f_A;
  A->external_mat = QTRUE;
  return QSUCCESS;
}
```

```
typedef struct {
  QVoid *f03_mat;
  QBool external_mat;
} RealMat;
```