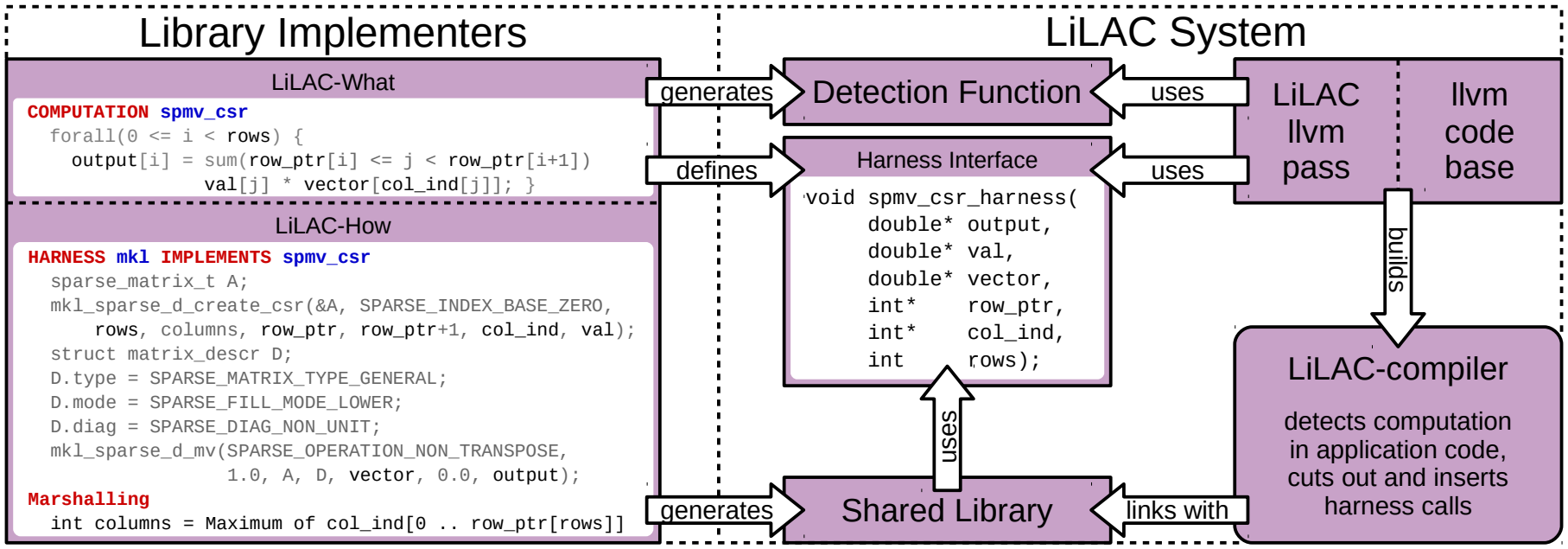
Library Implementers
LiLAC System
LiLAC-What
COMPUTATION spmv_csr
forall(0 <= i < rows) {
output[i] = sum(row_ptr[i] <= j < row_ptr[i+1])
val[j] * vector[col_ind[j]]; }
LiLAC-How
HARNESS mkl IMPLEMENTS spmv_csr
sparse_matrix_t A;
mkl_sparse_d_create_csr(&A, SPARSE_INDEX_BASE_ZERO,
rows, columns, row_ptr, row_ptr+1, col_ind, val);
struct matrix_descr D;
D.type = SPARSE_MATRIX_TYPE_GENERAL;
D.mode = SPARSE_FILL_MODE_LOWER;
D.diag = SPARSE_DIAG_NON_UNIT;
mkl_sparse_d_mv(SPARSE_OPERATION_NON_TRANSPOSE,
1.0, A, D, vector, 0.0, output);
Marshalling
int columns = Maximum of col_ind[0 .. row_ptr[rows]]
generates
Detection Function
uses
LiLAC llvm pass
llvm code base
defines
Harness Interface
void spmv_csr_harness(
double* output,
double* val,
double* vector,
int* row_ptr,
int* col_ind,
int rows);
uses
builds
LiLAC-compiler
detects computation in application code, cuts out and inserts harness calls
uses
generates
Shared Library
links with