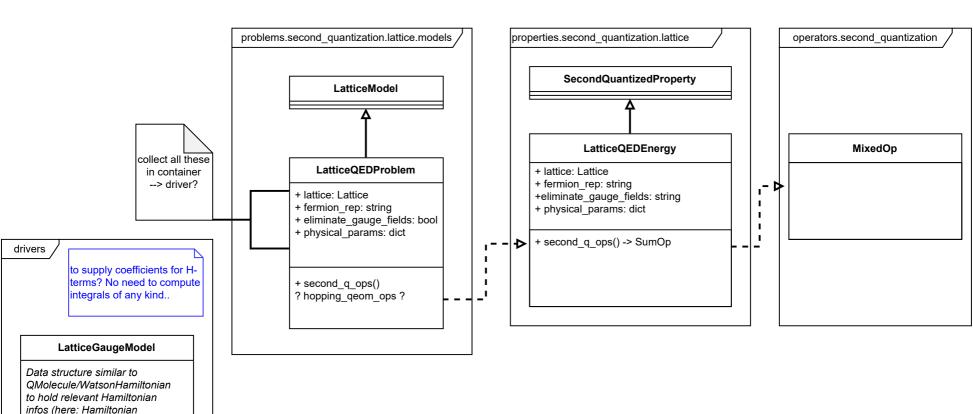
Implementation of LatticeQED problem:

- 1. most general case: quantum link model (with gauge fields) in d dimensions (limit to 1 or 2 dimensions first)
- 2. as special cases:
 - Wilson and staggered fermions
 - in 1D: possibility to eliminate gauge fields
- 3. start with Wilson Hamiltonian in 1D and 2D + QLM for gauge fields
- 4. need to enforce Gauss' law: e.g., as an additional Hamiltonian term
- 5. User should have the following options to specify:
 - o lattice (includes: dim, lattice constant, num sites)
 - fermion_rep (Wilson or staggered)
 - physical params (mass, coupling, etc.)
 - o whether to add Gauss' operator to Hamiltonian (energy penalty) or return as a separate operator
 - o for fermion rep == 'wilson': representation of gamma-matrices
 - for dim == 1: whether to eliminate gauge fields
- 6. combinations of operators:
 - FermionOp * FermionOp
 - FermioOp * SpinOp * FermionOp
 - SpinOp * SpinOp
- 7. introduce new operator.second quantization.mixed op
 - o mixing/composition is always as a tensor product since different types of operators must act on distinct Hilber spaces
 - MixedOp = List[FermionicOp, SpinOp, FermionicOp]
 - o need to specify WHICH qubits each op acts on? Allow for cases where two FermionicOps act on different registers?
 - o can this be made compatible with the current implementation of mappers?
- 8. return list of operators as

mapper?

 List[MixedOp(FermionOp, FermionOp), MixedOp(FermionOp, SpinOp, FermionOp), ...], each element is term in operator



[spin[self_energy[coeff, index], tunnel[coeff, index]], boson[self_energy[coeff, index]], couple[coeff, index]]]

+ lattice: Lattice
+ fermion_rep: string
+elimiinate_gauge_fields: bool
+ physical_params: dict
+ SpinS: Union[int, fraction]

parameters). E.g., array[array]: