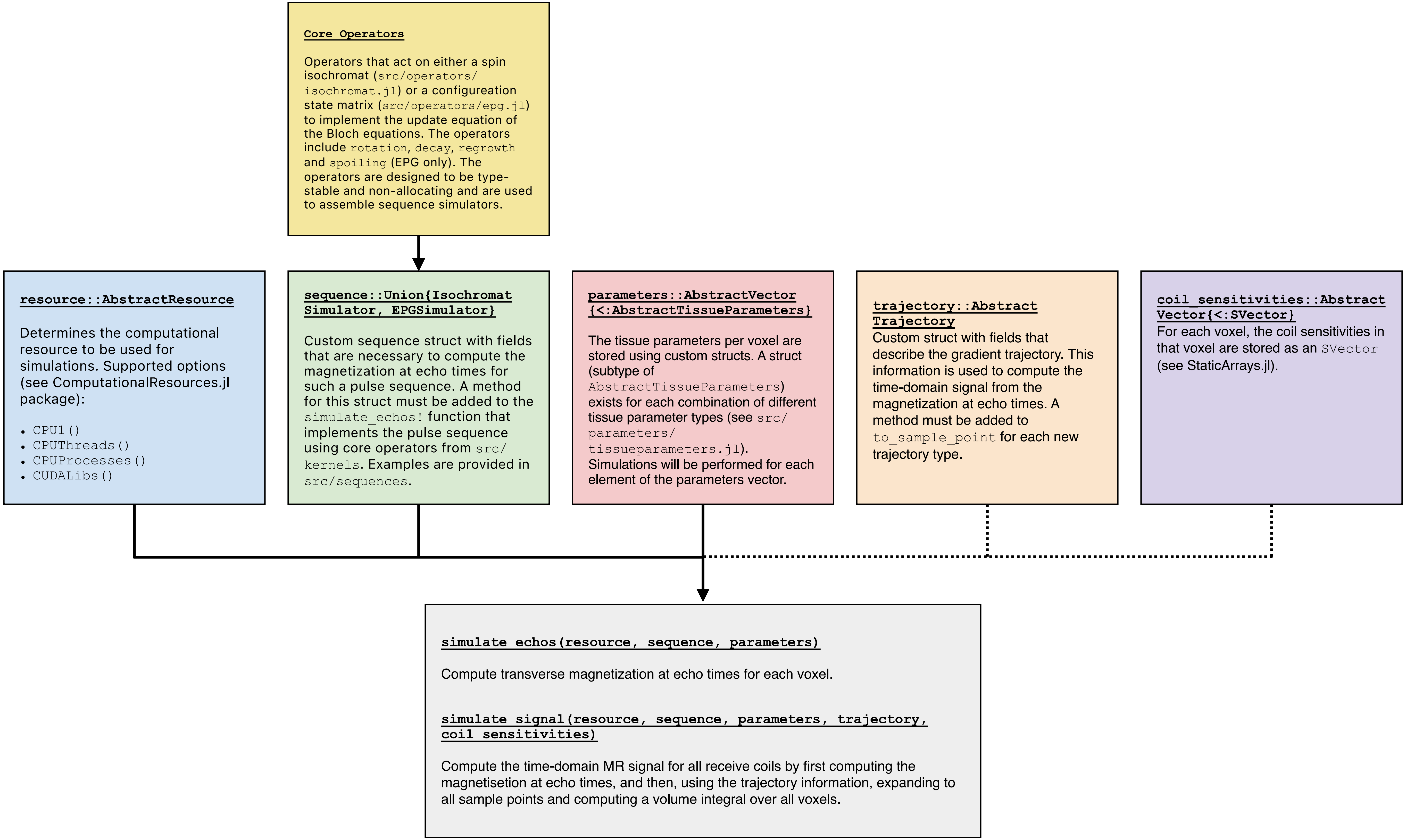
Core Operators
Operators that act on either a spin isochromat (src/operators/isochromat.jl) or a configuration state matrix (src/operators/epg.jl) to implement the update equation of the Bloch equations. The operators include rotation, decay, regrowth and spoiling (EPG only). The operators are designed to be type-stable and non-allocating and are used to assemble sequence simulators.
resource::AbstractResource
Determines the computational resource to be used for simulations. Supported options (see ComputationalResources.jl package):
• CPU1()
• CPUThreads()
• CPUProcesses()
• CUDALibs()
sequence::Union{IsochromatSimulator, EPGSimulator}
Custom sequence struct with fields that are necessary to compute the magnetization at echo times for such a pulse sequence. A method for this struct must be added to the simulate_echos! function that implements the pulse sequence using core operators from src/kernels. Examples are provided in src/sequences.
parameters::AbstractVector{<:AbstractTissueParameters}
The tissue parameters per voxel are stored using custom structs. A struct (subtype of AbstractTissueParameters) exists for each combination of different tissue parameter types (see src/parameters/tissueparameters.jl). Simulations will be performed for each element of the parameters vector.
trajectory::AbstractTrajectory
Custom struct with fields that describe the gradient trajectory. This information is used to compute the time-domain signal from the magnetization at echo times. A method must be added to to_sample_point for each new trajectory type.
coil_sensitivities::AbstractVector{<:SVector}
For each voxel, the coil sensitivities in that voxel are stored as an SVector (see StaticArrays.jl).
simulate_echos(resource, sequence, parameters)
Compute transverse magnetization at echo times for each voxel.
simulate_signal(resource, sequence, parameters, trajectory, coil_sensitivities)
Compute the time-domain MR signal for all receive coils by first computing the magnetisetion at echo times, and then, using the trajectory information, expanding to all sample points and computing a volume integral over all voxels.