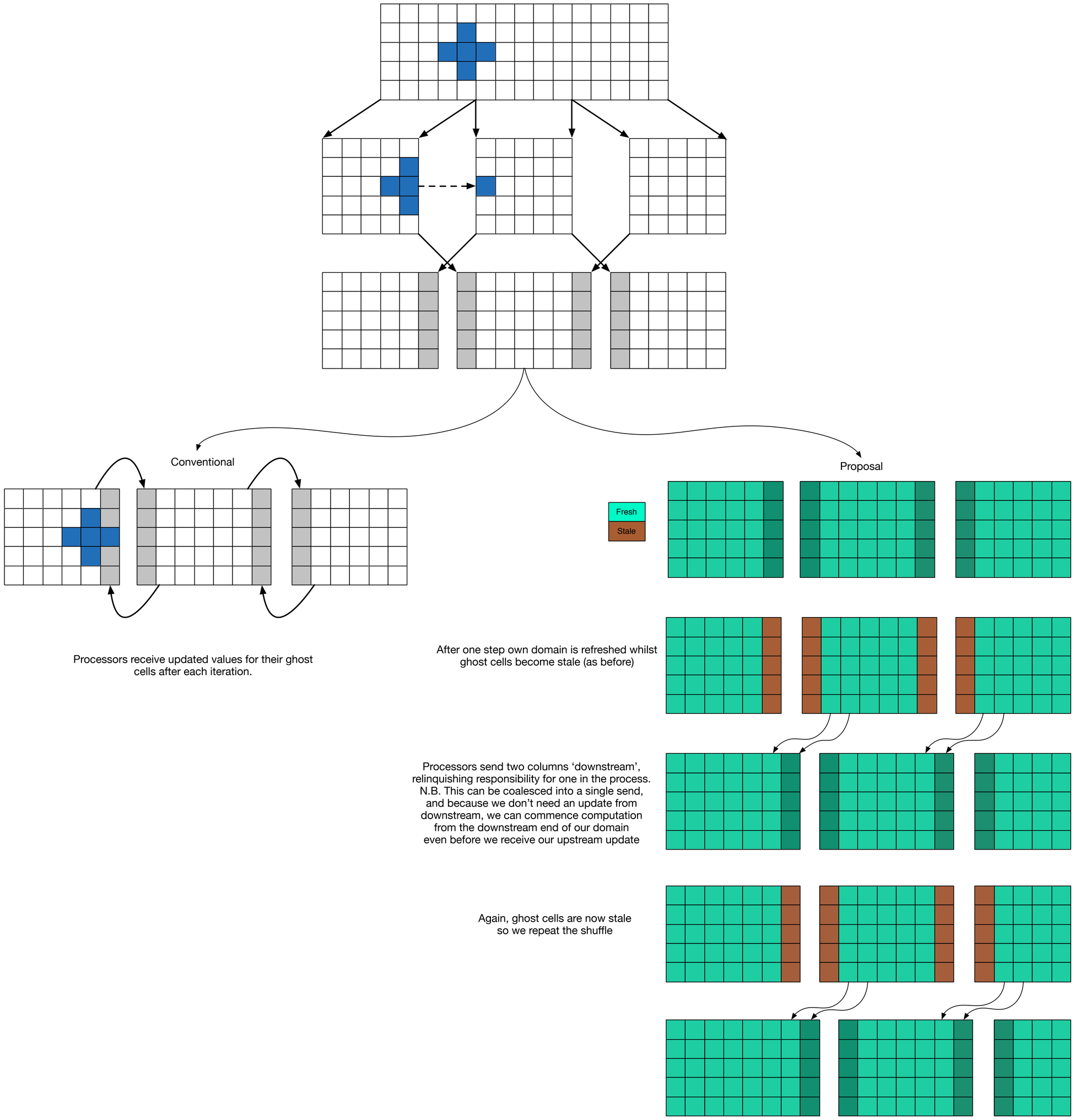

Processors receive updated values for their ghost cells after each iteration.

After one step own domain is refreshed whilst ghost cells become stale (as before)

Processors send two columns ‘downstream’, relinquishing responsibility for one in the process. N.B. This can be coalesced into a single send, and because we don’t need an update from downstream, we can commence computation from the downstream end of our domain even before we receive our upstream update

Again, ghost cells are now stale so we repeat the shuffle

Note - now we see that the responsibility of the downstream processor is growing, presumably causing a slowdown. In problems with periodic boundary conditions this would not be an issue, as there would be no ‘most downstream’ processor. Additionally, this problem can be mitigated by splitting the region, and having the shrinking upstream proc handle some of the work. This would mean that all processors have a constant area of responsibility, but may cause some performance degradation in one processor as it is in effect doing two smaller simulations. Another potential improvement is to alternate upstream and downstream directions as time progresses: