

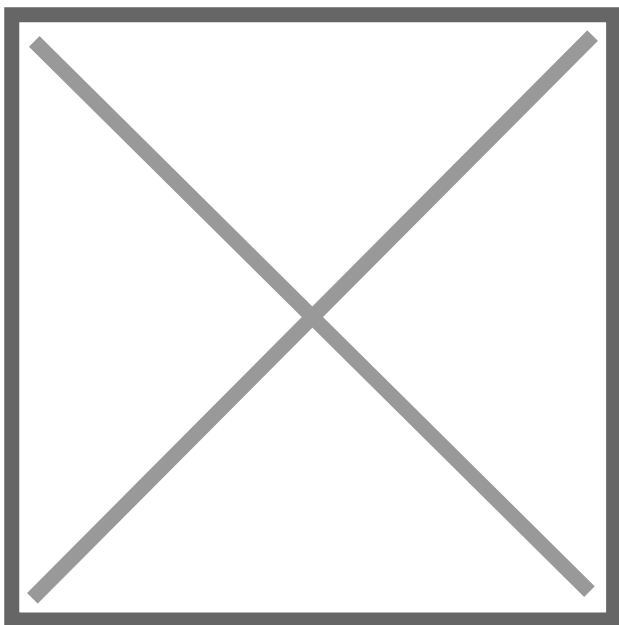
Updates and Simulations

The Material Tracking Layer builds entropy as it operates, following the direction of time. That is, the outcome of a particular event may depend on events that happened before. For this reason, out of order events are not permitted and any conflicting spatial-temporal information will be discarded by the layer. It is up to the sender to guarantee events in the past are not submitted after future events.

Assuming the updates are submitted in proper sequence, the caller can issue multiple requests without waiting for any callbacks. In some other cases, however, there may be temporal dependencies that are not resolved yet. In these cases, the caller will be instructed by the API to wait, as the input data for the next stage is still not completely known to the system yet.

This case is caused by simulations. A simulation agent is an external system that uses a particular subset of REST calls, which tell the system a simulation must start or has ended, and what was the outcome of the simulation in terms of material changes. The system must be aware of simulations because when the system is under simulation, any new material operations should be deferred until the simulation is over. The outcome of these operations may depend on the results produced by the simulation.

For instance, consider this scenario:

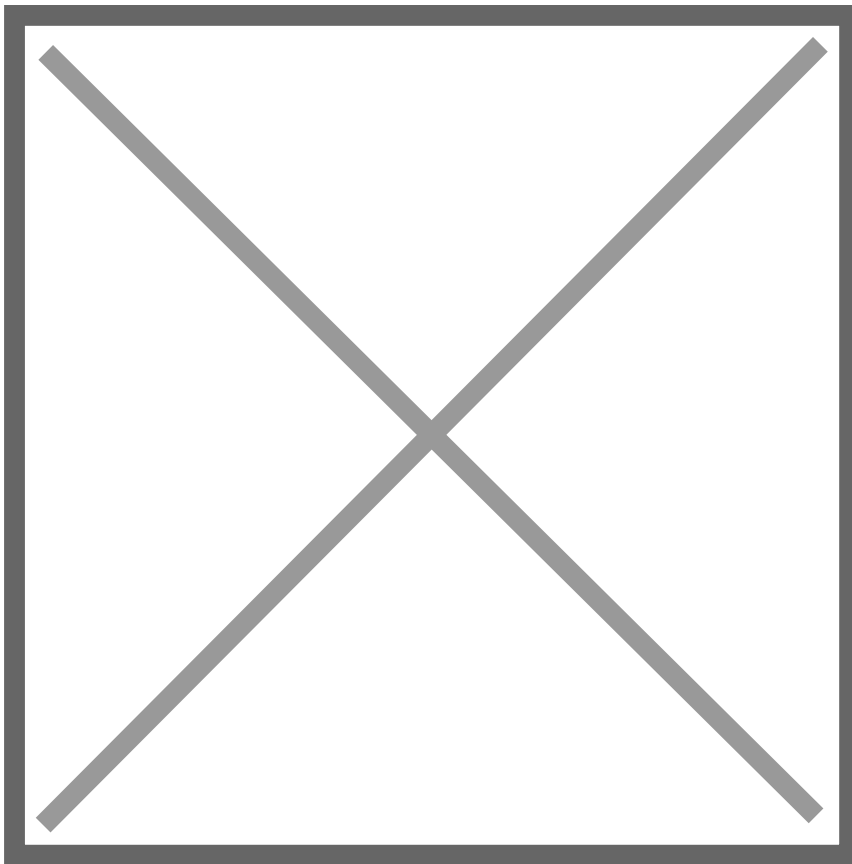


1. A dump event containing material M occurs at simulation time $\text{sim_t} = 0$, system time $\text{sys_t} = 0$

2. A simulation agent computes the new location of material M from system time $\text{sys_t} = 0$ to $\text{sys_t} = 200$
3. An excavator load event occurs at simulation time $\text{sim_t} = 100$, system time $\text{sys_t} = 100$

The outcome of step (3) depends on the outcome of the preceding step, which is a simulation that is performed by a third-party agent. In this example, the simulation took longer to compute than the physical process to happen, so the excavator load event arrived at the system before there is knowledge of what the excavator could be picking up.

These dependencies are compounded when there are multiple simulation steps chained together. For this reason, the API uses an interaction model that guarantees simulation steps are performed only after their preconditions have been committed to the material tracking layer.



In some cases, when submitting a new batch of material tracking operations, the API may instruct the caller to wait for a callback to submit the batch.

The image at the left shows two simulation steps S1 and S2, where S2 depends on S1. If the event that triggers S2 is received before S1 completes, the system will return WAIT on any attempt to queue the S2 simulation and the caller will be notified when S1 is complete via a callback.

