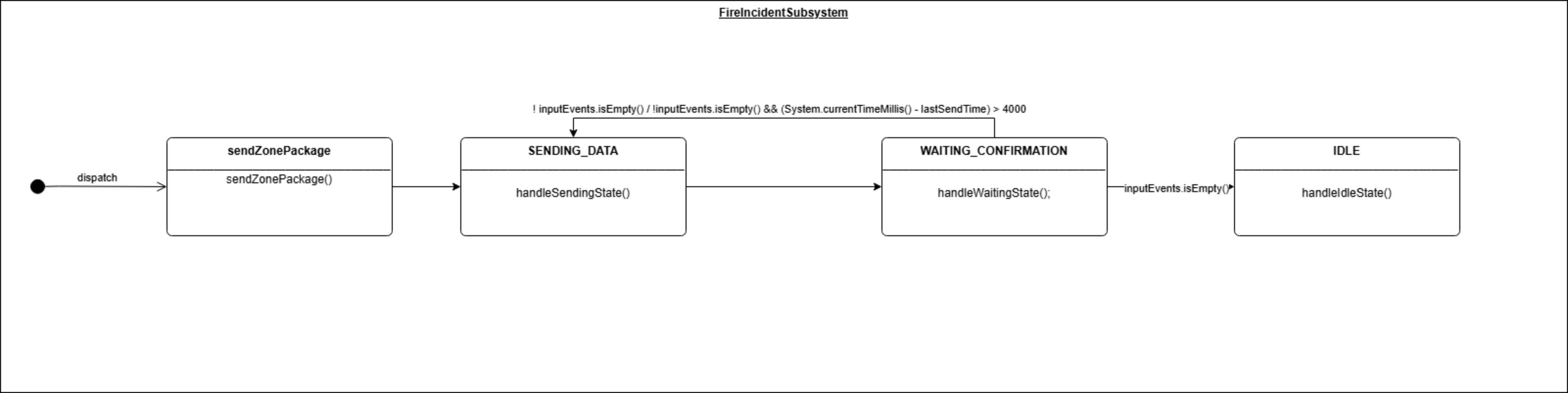
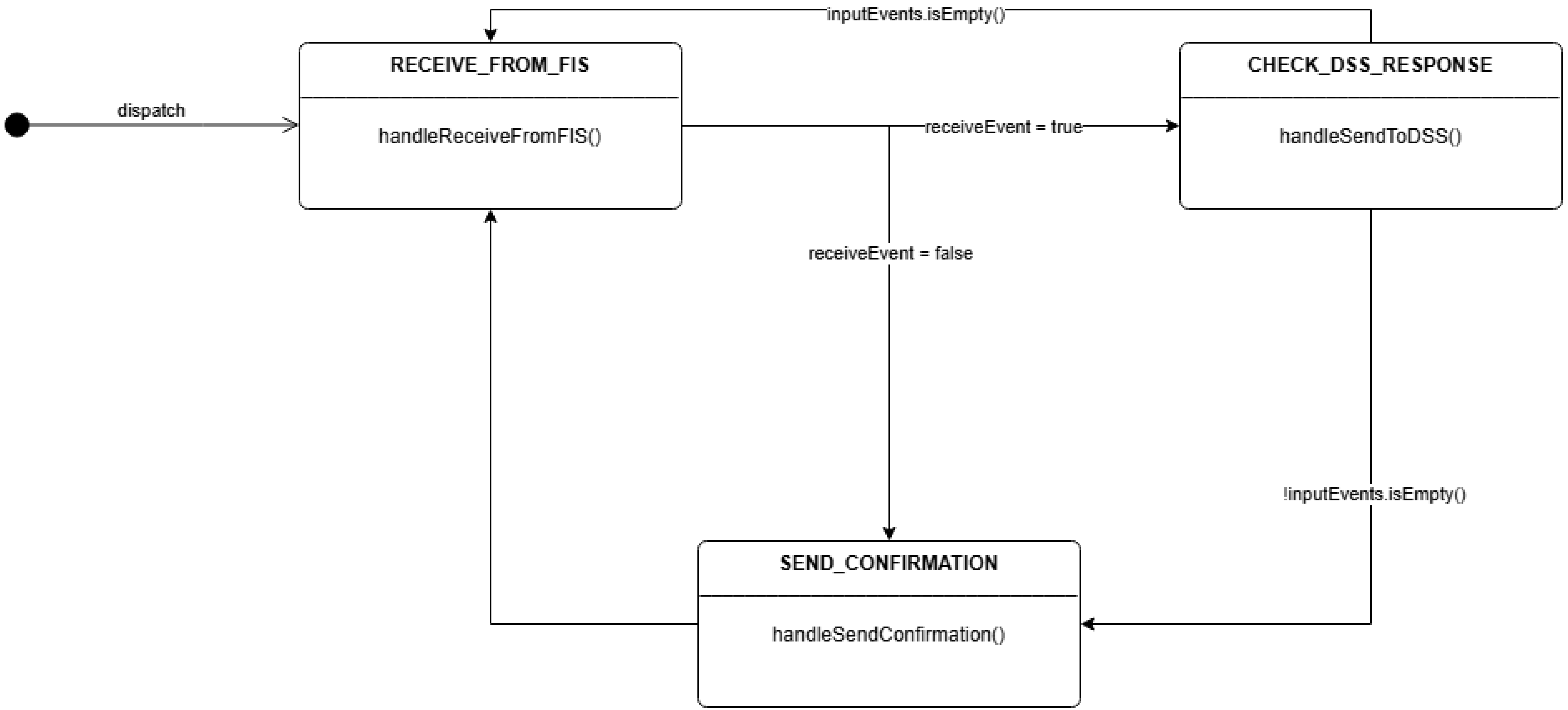
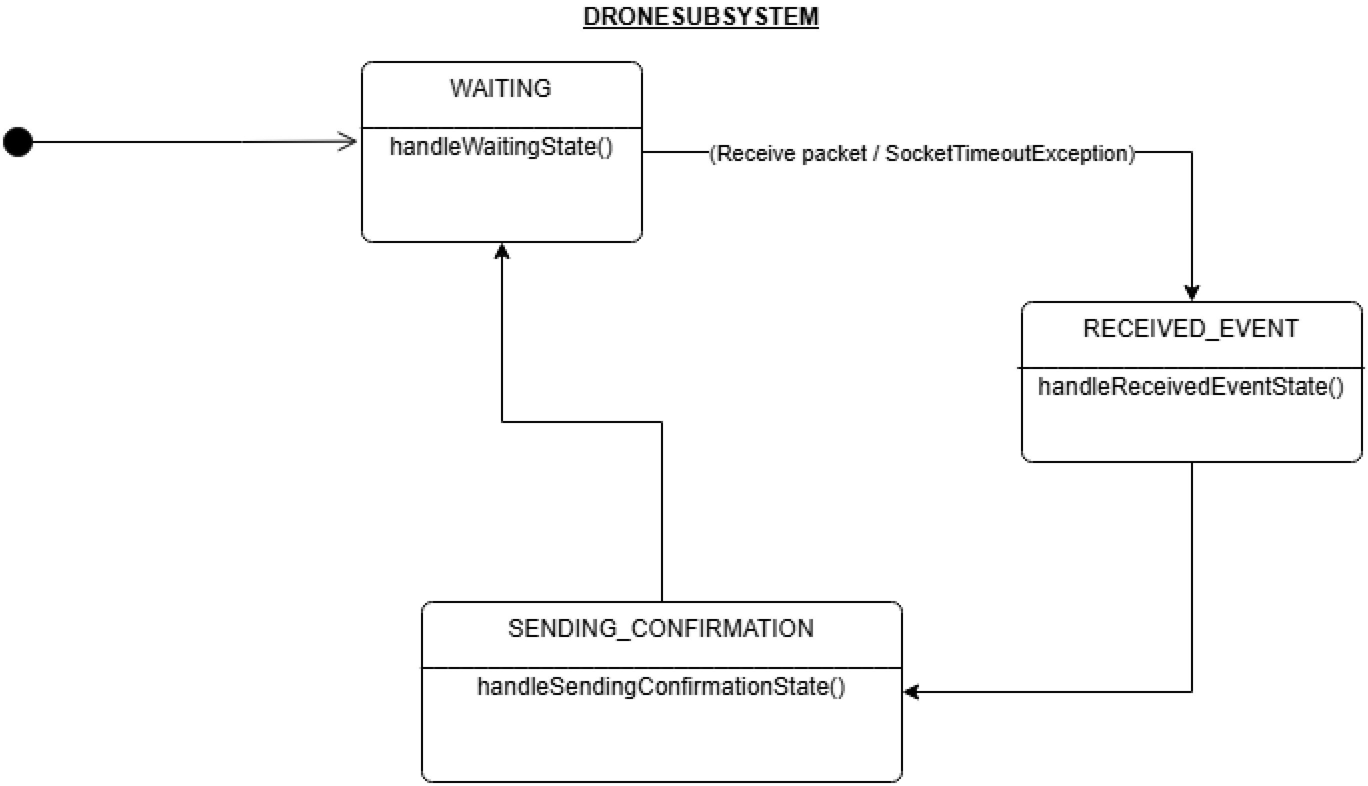
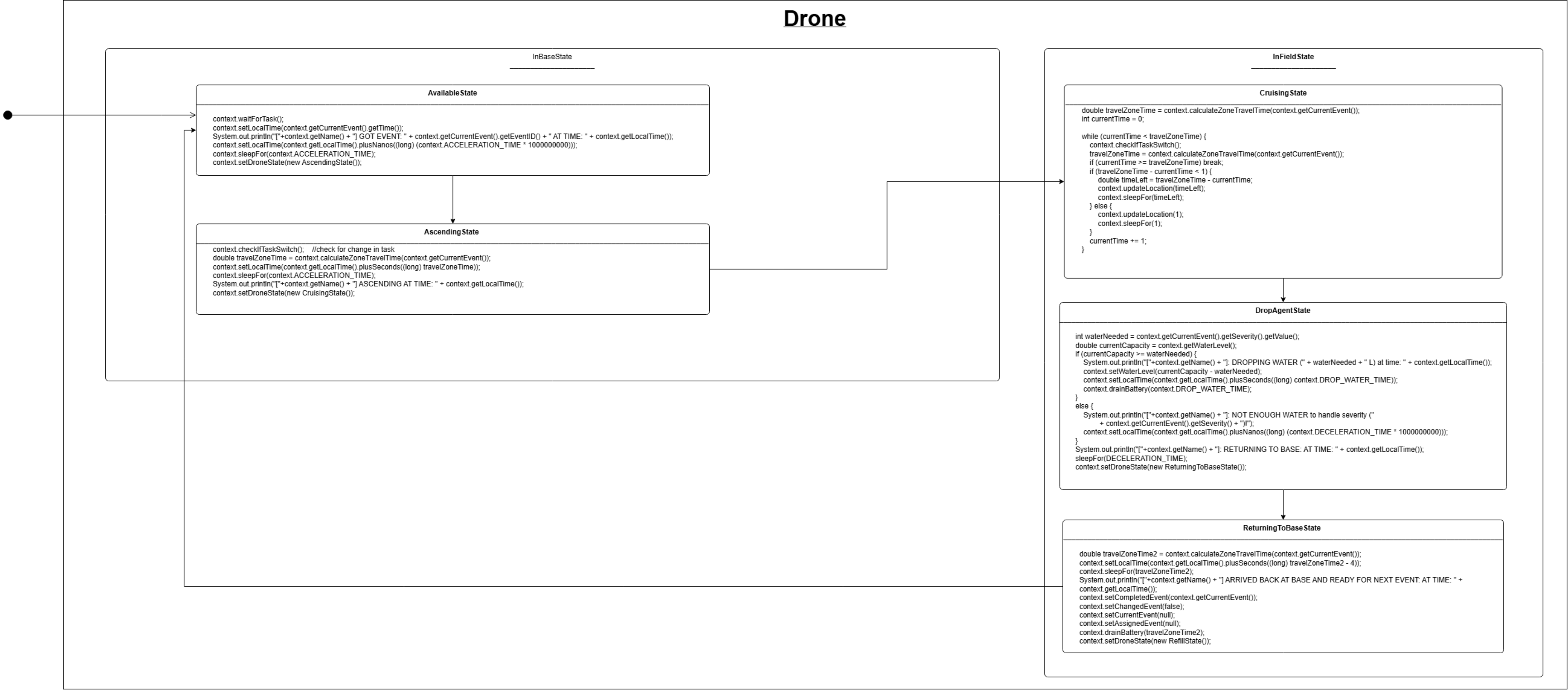

FireIncidentSubsystem
dispatch
sendZonePackage
sendZonePackage()
SENDING_DATA
handleSendingState()
! inputEvents.isEmpty() / !inputEvents.isEmpty() && (System.currentTimeMillis() - lastSendTime) > 4000
WAITING_CONFIRMATION
handleWaitingState();
inputEvents.isEmpty()
IDLE
handleIdleState()

Scheduler


inputEvents.isEmpty()
RECEIVE_FROM_FIS
handleReceiveFromFIS()
CHECK_DSS_RESPONSE
handleSendToDSS()
dispatch
receiveEvent = true
receiveEvent = false
!inputEvents.isEmpty()
SEND_CONFIRMATION
handleSendConfirmation()


DRONESUBSYSTEM
WAITING
handleWaitingState()
(Receive packet / SocketTimeoutException)
RECEIVED_EVENT
handleReceivedEventState()
SENDING_CONFIRMATION
handleSendingConfirmationState()



AvailableState

```
context.waitForTask();
context.setLocalTime(context.getCurrentEvent().getTime());
System.out.println("["+context.getName() + "] GOT EVENT: " + context.getCurrentEvent().getEventID() + " AT TIME: " + context.getLocalTime());
context.setLocalTime(context.getLocalTime().plusNanos((long) (context.ACCELERATION_TIME * 1000000000)));
context.sleepFor(context.ACCELERATION_TIME);
context.setDroneState(new AscendingState());
```

AscendingState

```
context.checkIfTaskSwitch();    //check for change in task
double travelZoneTime = context.calculateZoneTravelTime(context.getCurrentEvent());
context.setLocalTime(context.getLocalTime().plusSeconds((long) travelZoneTime));
context.sleepFor(context.ACCELERATION_TIME);
System.out.println("["+context.getName() + "] ASCENDING AT TIME: " + context.getLocalTime());
context.setDroneState(new CruisingState());
```

CruisingState

```
double travelZoneTime = context.calculateZoneTravelTime(context.getCurrentEvent());
int currentTime = 0;

while (currentTime < travelZoneTime) {
   context.checkIfTaskSwitch();
   travelZoneTime = context.calculateZoneTravelTime(context.getCurrentEvent());
   if (currentTime >= travelZoneTime) break;
   if (travelZoneTime - currentTime < 1) {
      double timeLeft = travelZoneTime - currentTime;
      context.updateLocation(timeLeft);
      context.sleepFor(timeLeft);
   } else {
      context.updateLocation(1);
      context.sleepFor(1);
   }
   currentTime += 1;
}
```

DropAgentState

```
int waterNeeded = context.getCurrentEvent().getSeverity().getValue();
double currentCapacity = context.getWaterLevel();
if (currentCapacity >= waterNeeded) {
   System.out.println("["+context.getName() + "]: DROPPING WATER (" + waterNeeded + " L) at time: " + context.getLocalTime());
   context.setWaterLevel(currentCapacity - waterNeeded);
   context.setLocalTime(context.getLocalTime().plusSeconds((long) context.DROP_WATER_TIME));
   context.drainBattery(context.DROP_WATER_TIME);
}
else {
   System.out.println("["+context.getName() + "]: NOT ENOUGH WATER to handle severity ("
         + context.getCurrentEvent().getSeverity() + ")!");
   context.setLocalTime(context.getLocalTime().plusNanos((long) (context.DECELERATION_TIME * 1000000000)));
}
System.out.println("["+context.getName() + "]: RETURNING TO BASE: AT TIME: " + context.getLocalTime());
sleepFor(DECELERATION_TIME);
context.setDroneState(new ReturningToBaseState());
```

ReturningToBaseState

```
double travelZoneTime2 = context.calculateZoneTravelTime(context.getCurrentEvent());
context.setLocalTime(context.getLocalTime().plusSeconds((long) travelZoneTime2 - 4));
context.sleepFor(travelZoneTime2);
System.out.println("["+context.getName() + "] ARRIVED BACK AT BASE AND READY FOR NEXT EVENT: AT TIME: " +
context.getLocalTime());
context.setCompletedEvent(context.getCurrentEvent());
context.setChangedEvent(false);
context.setCurrentEvent(null);
context.setAssignedEvent(null);
context.drainBattery(travelZoneTime2);
context.setDroneState(new RefillState());
```