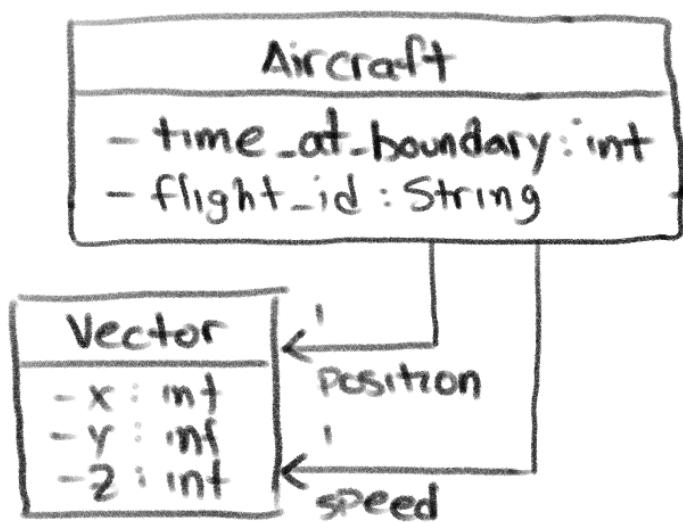
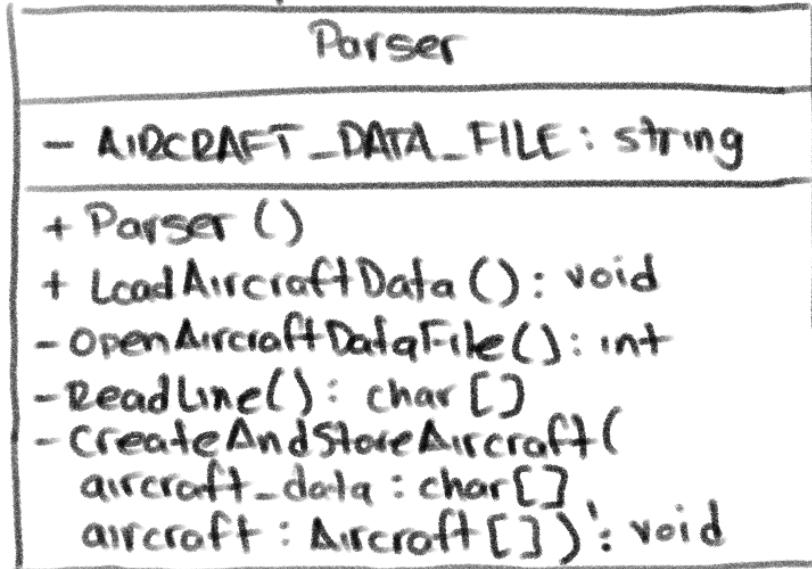
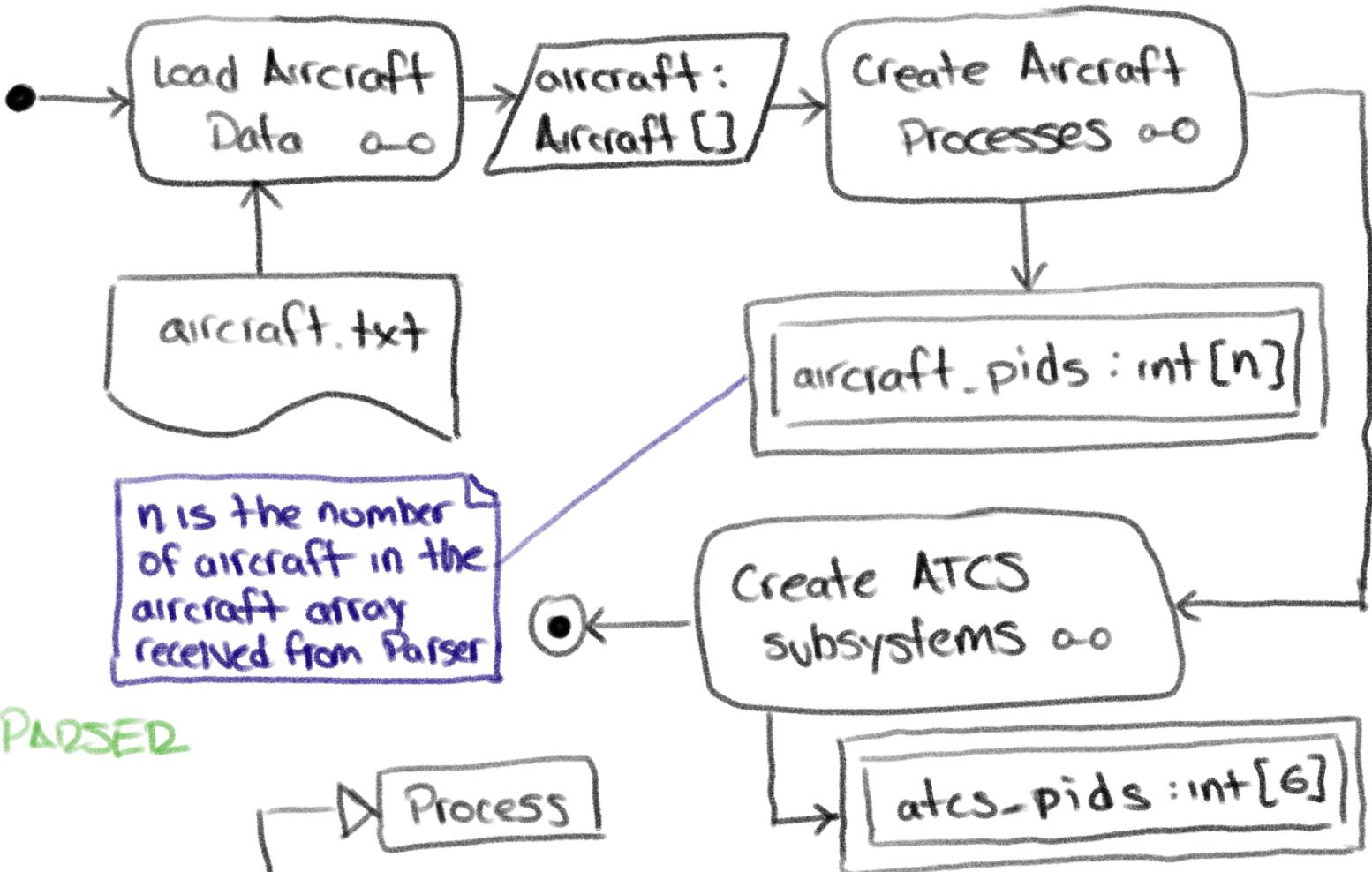
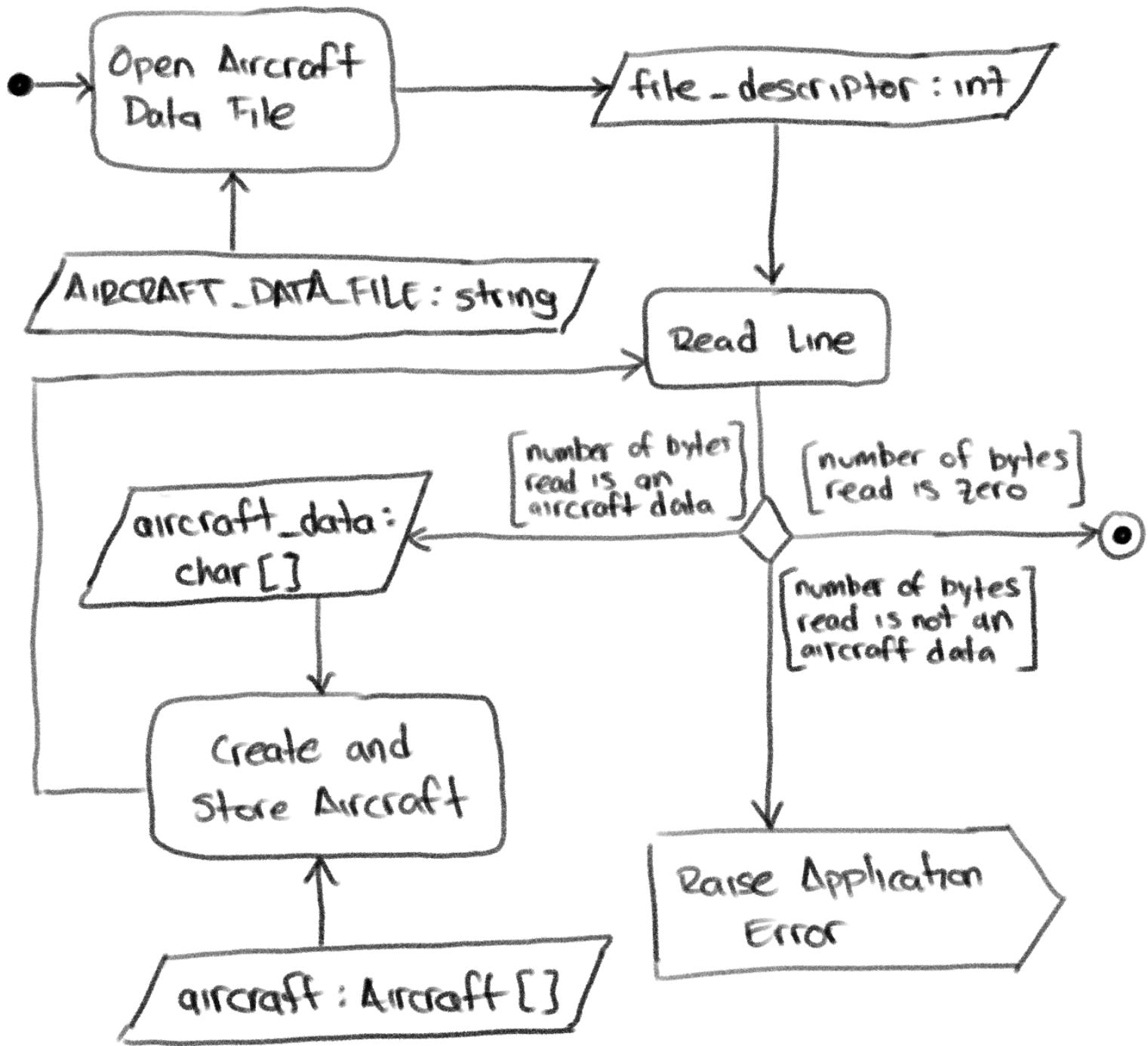


## CONFIGURE SIMULATION



# LOAD AIRCRAFT DATA

PAGE 2



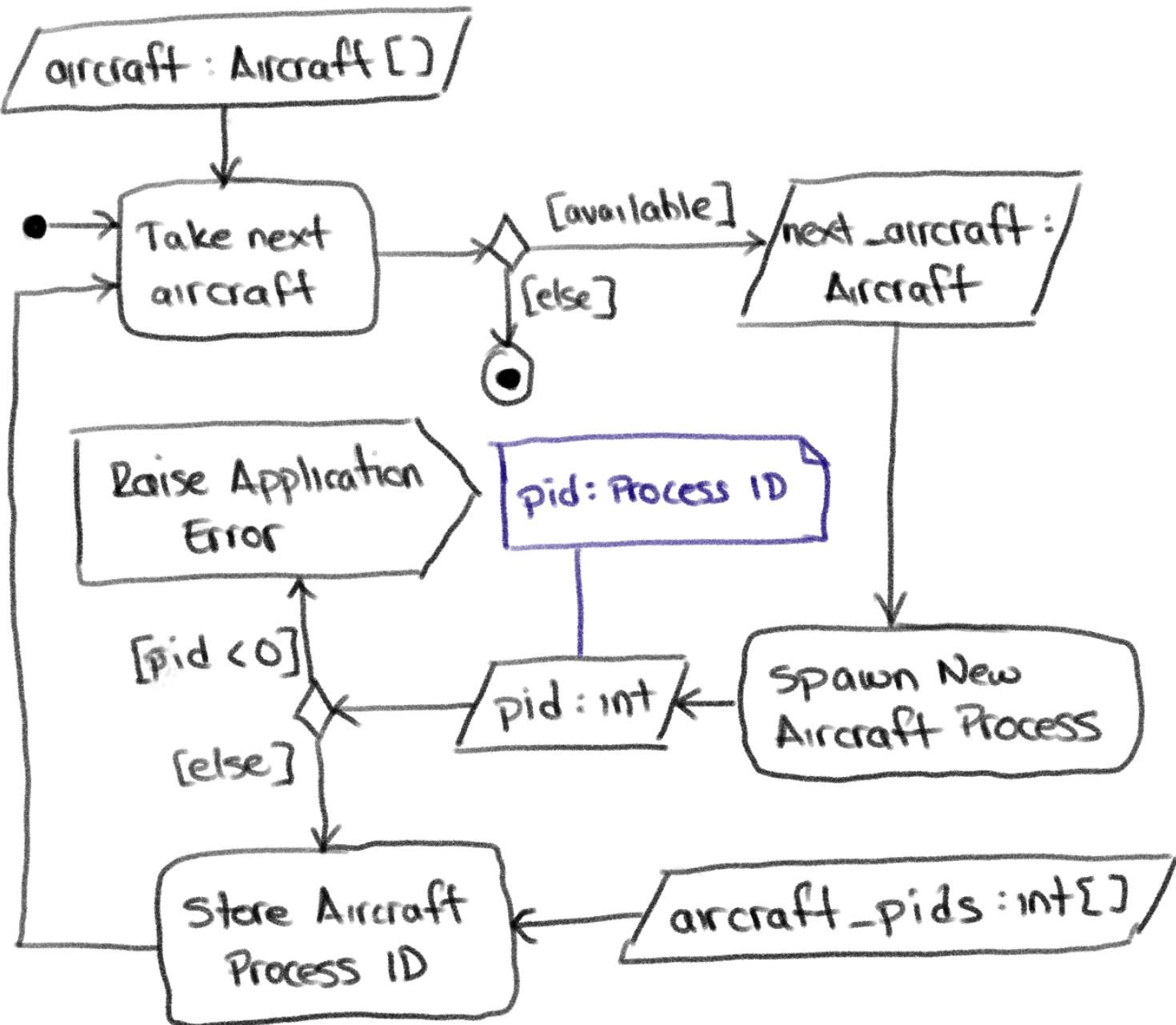
# RESOURCE MANAGER



- aircraft\_pids : int[]  
- atcs\_pids : int[]

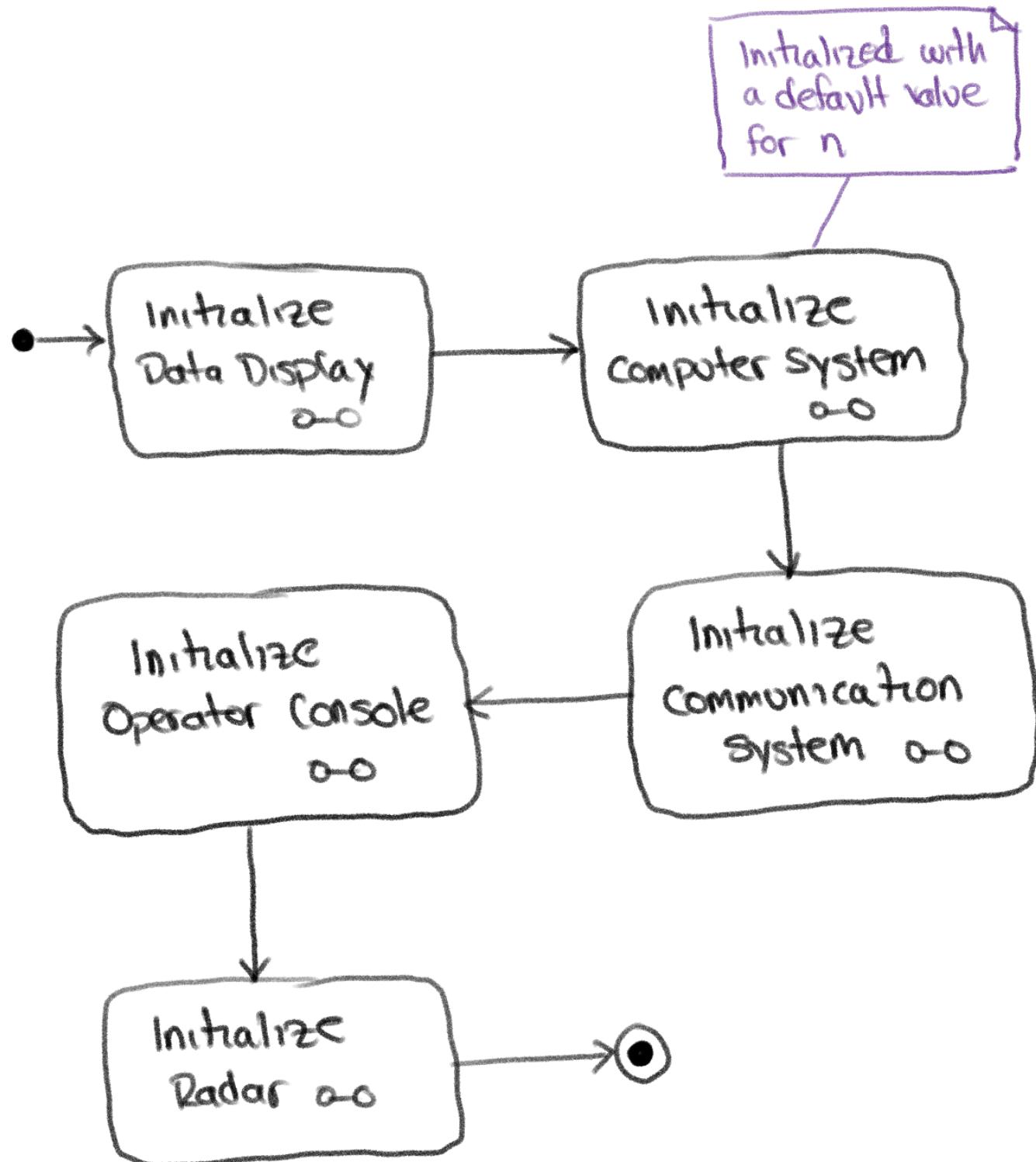
+ ResourceManager()  
+ CreateAircraftProcesses (aircraft: Aircraft[]): void  
+ CreateATCSSubsystems (): void  
- TakeNextAircraft (aircraft: Aircraft[]): Aircraft  
- SpawnNewAircraftProcess (aircraft: Aircraft): int  
- StoreAircraftProcessID (pid: int): void  
- InitializeDataDisplay (): void  
- InitializeComputerSystem (): void  
- InitializeCommunicationSystem (): void  
- InitializeOperatorConsole (): void  
- InitializeRadar (): void  
- StoreATCSProcessID (pid: int): void

## CREATE AIRCRAFT PROCESSES



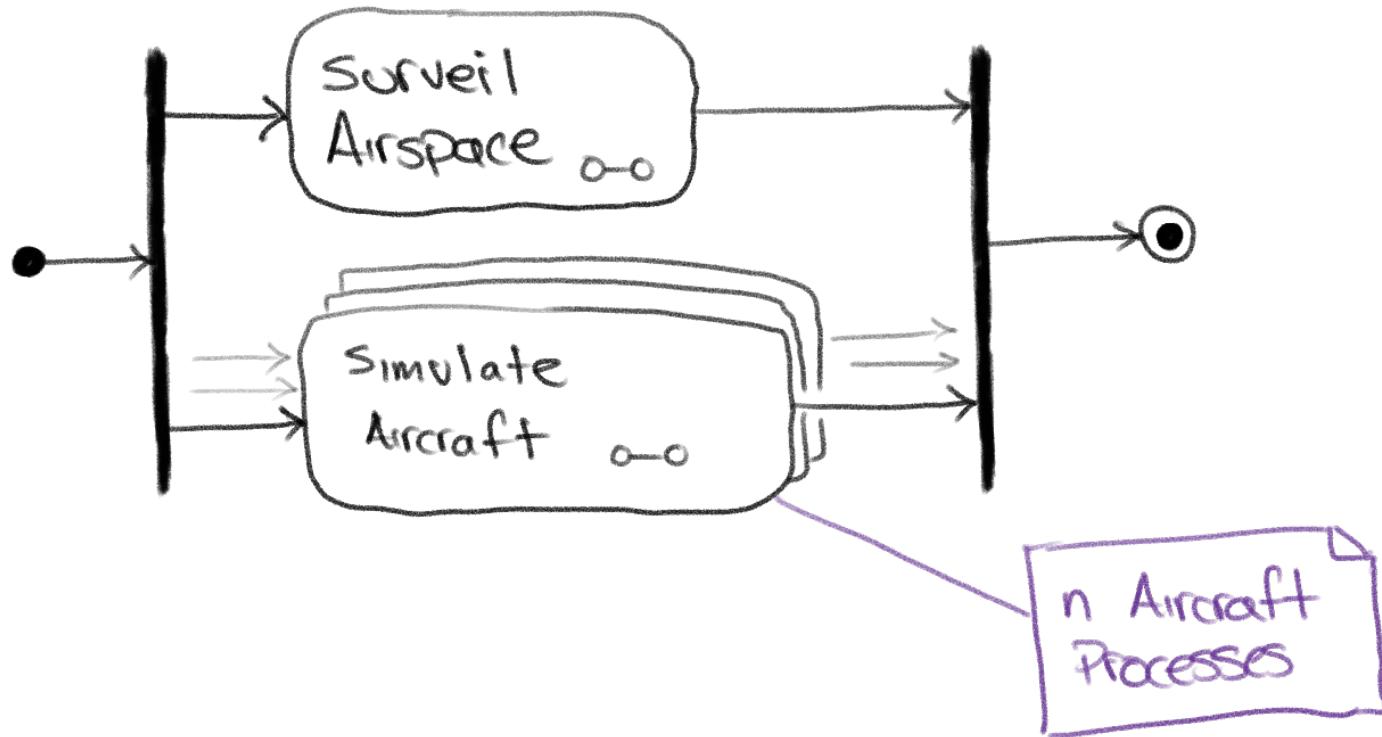
# CREATE ATCS SUBSYSTEMS

PAGE 5

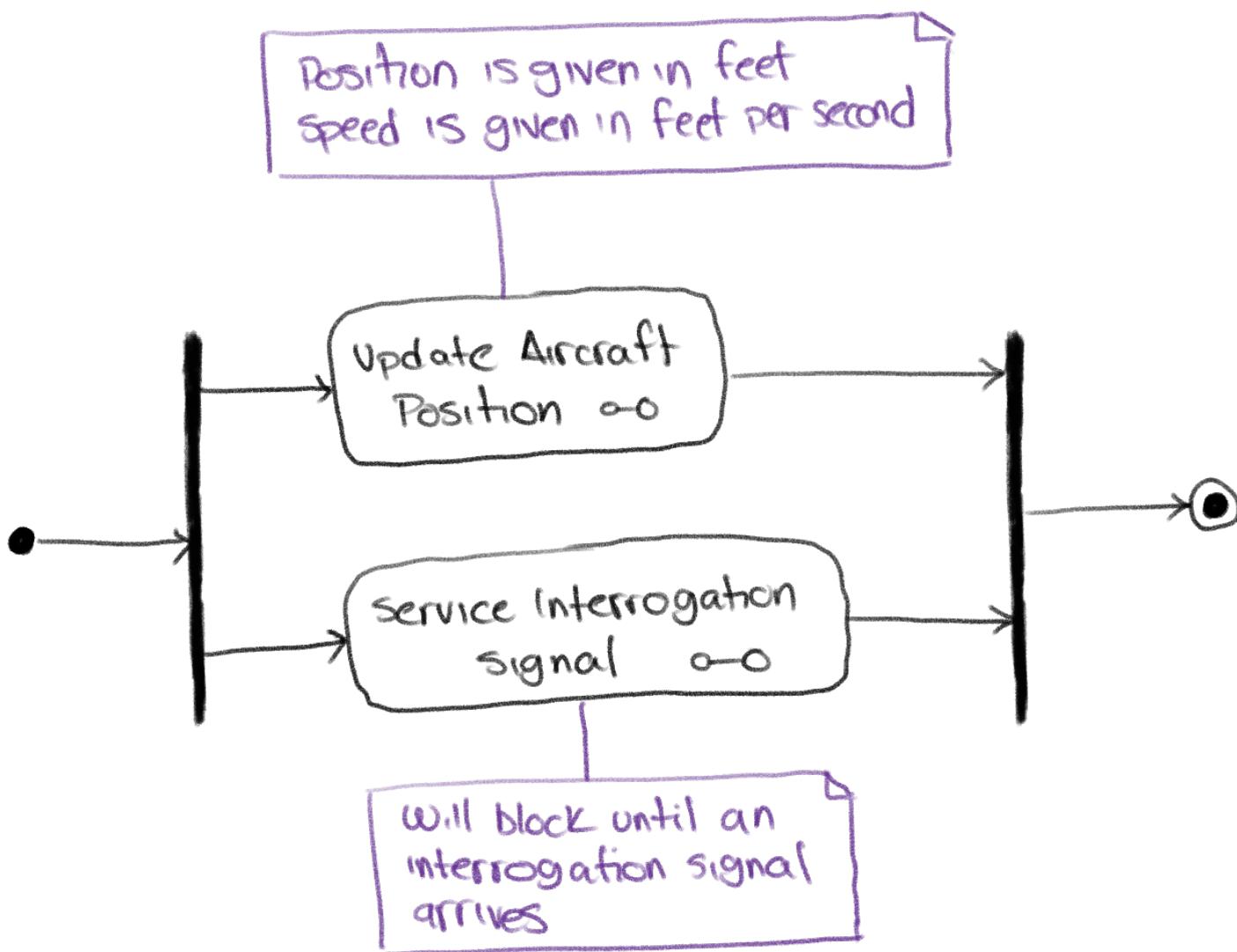


## RUN SIMULATION

PAGE 6

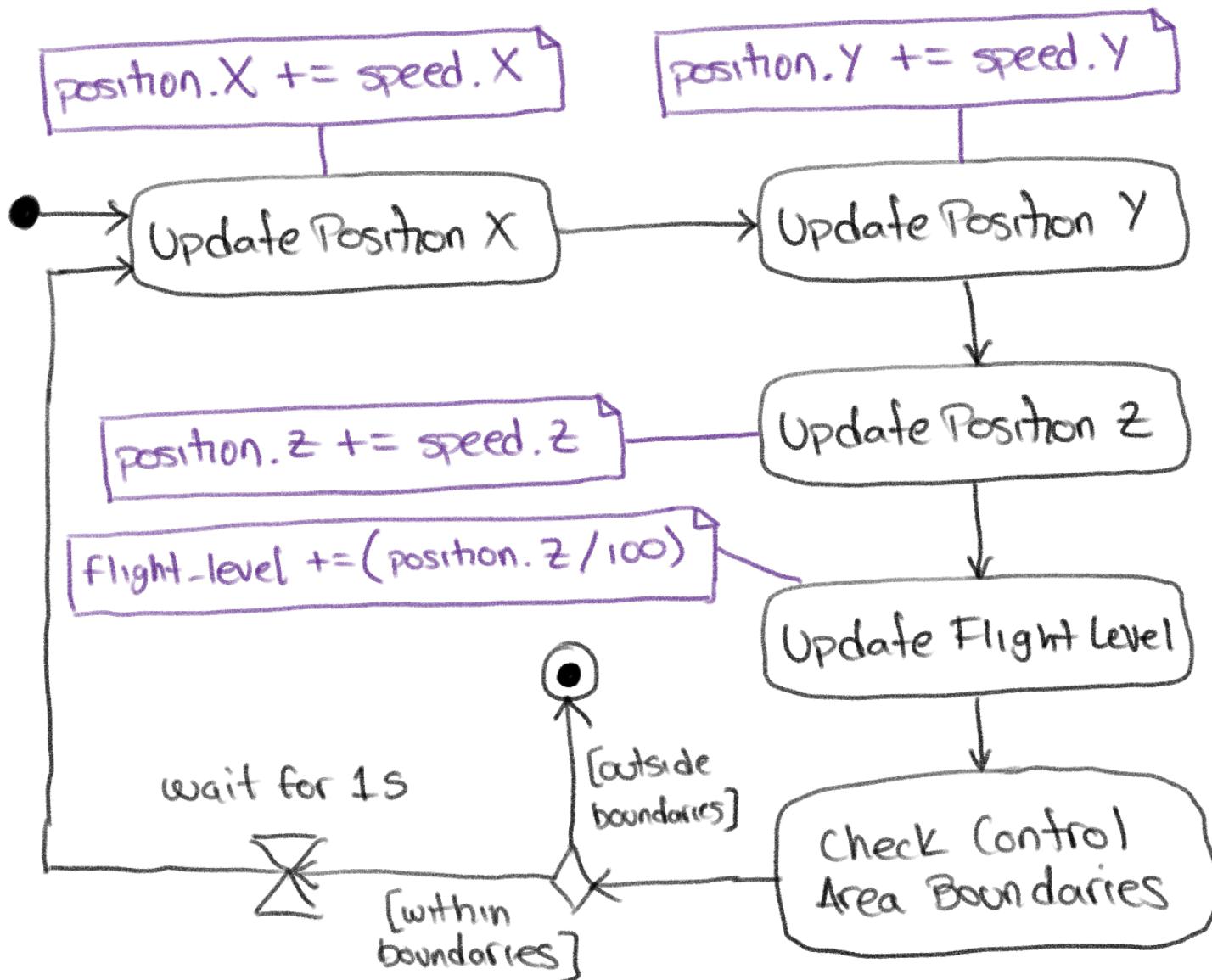


## SIMULATE AIRCRAFT

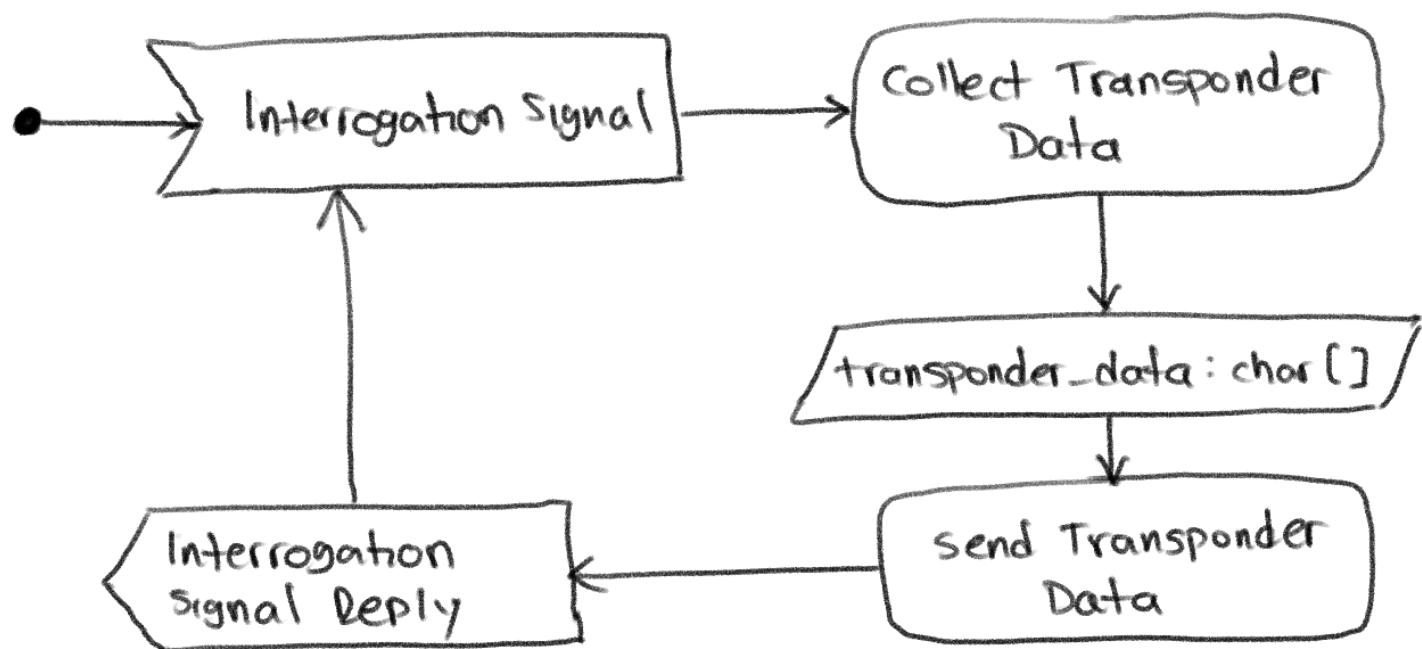


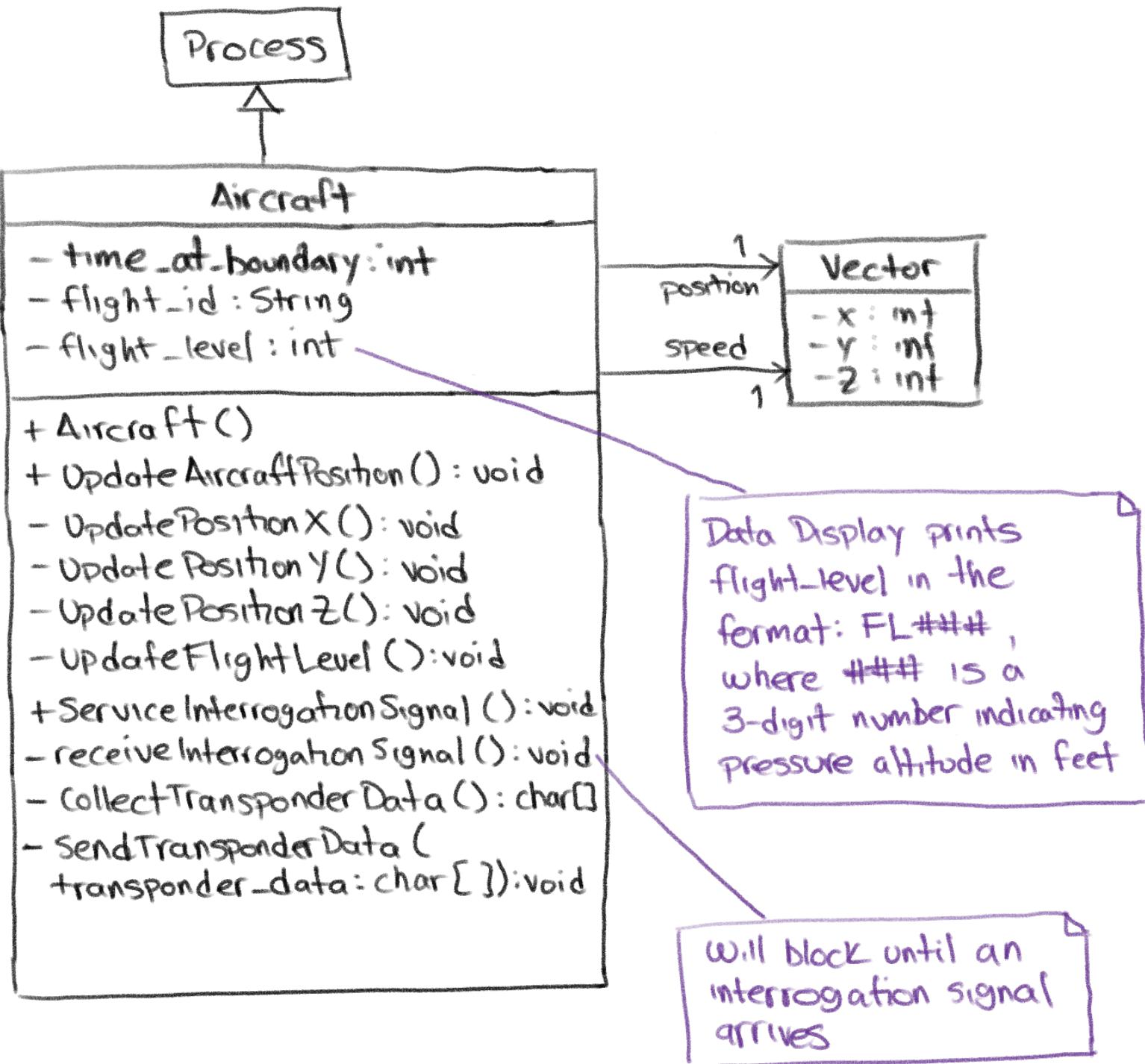
# UPDATE AIRCRAFT POSITION

PAGE 7



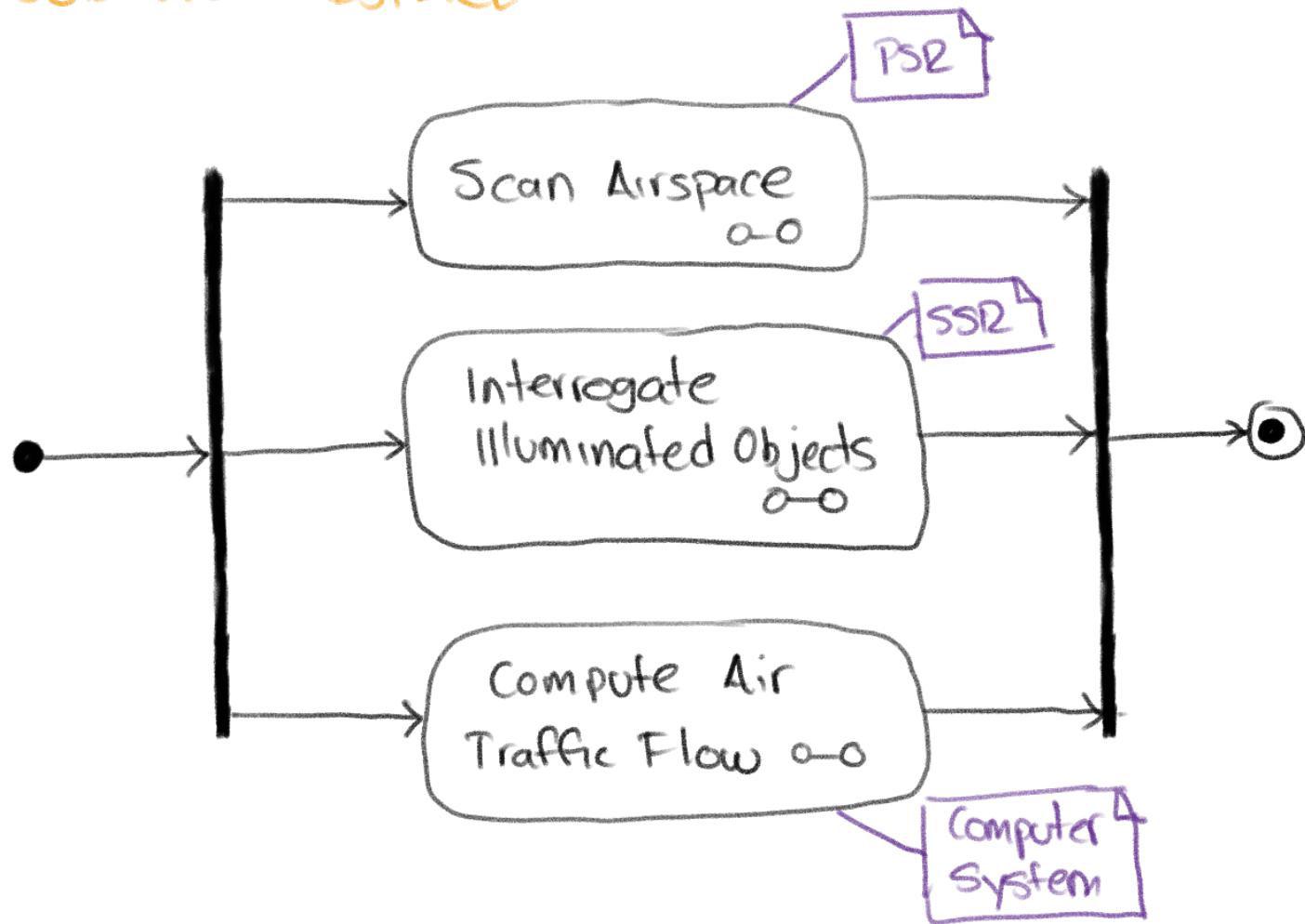
# SERVICE INTERROGATION SIGNAL



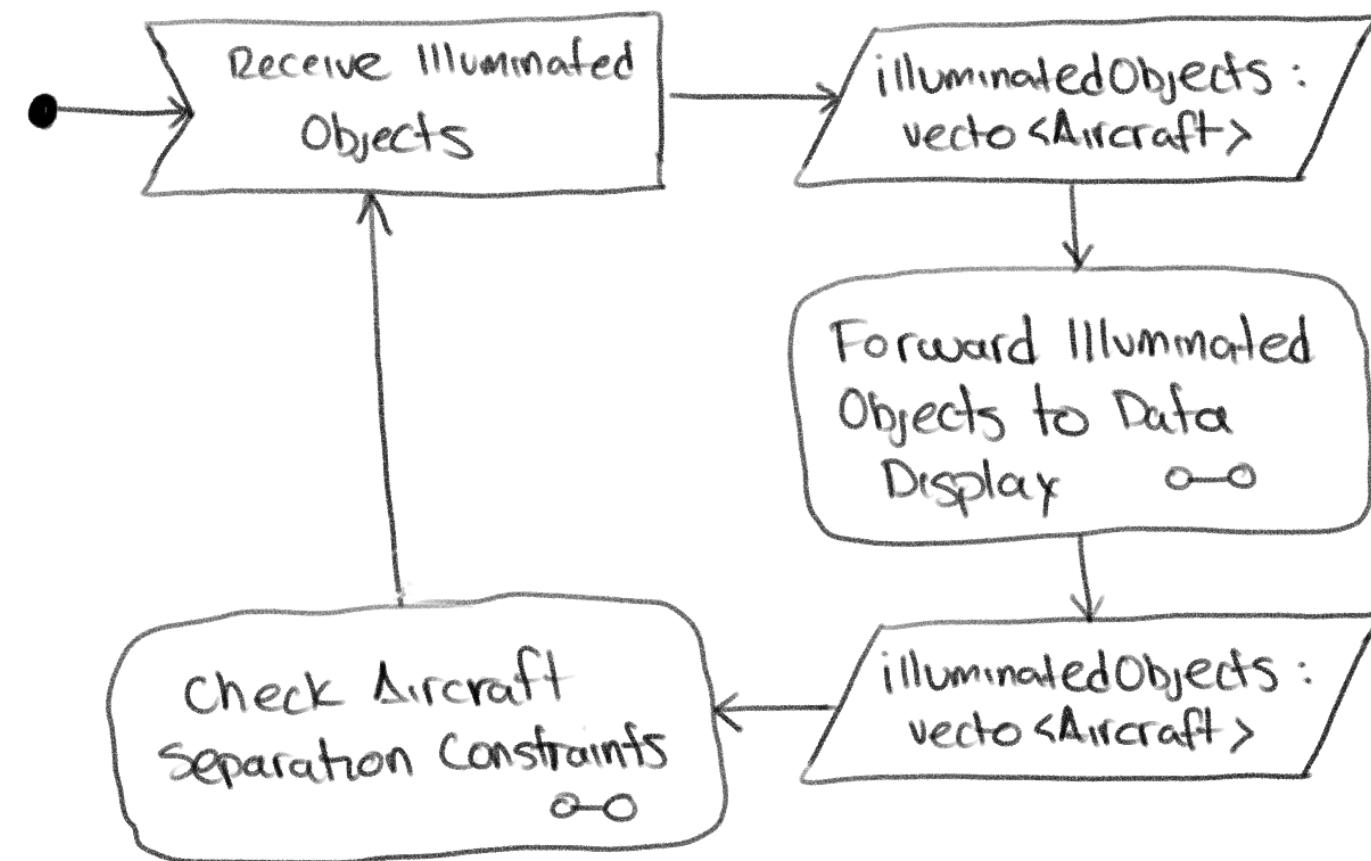


# SURVEIL AIRSPACE

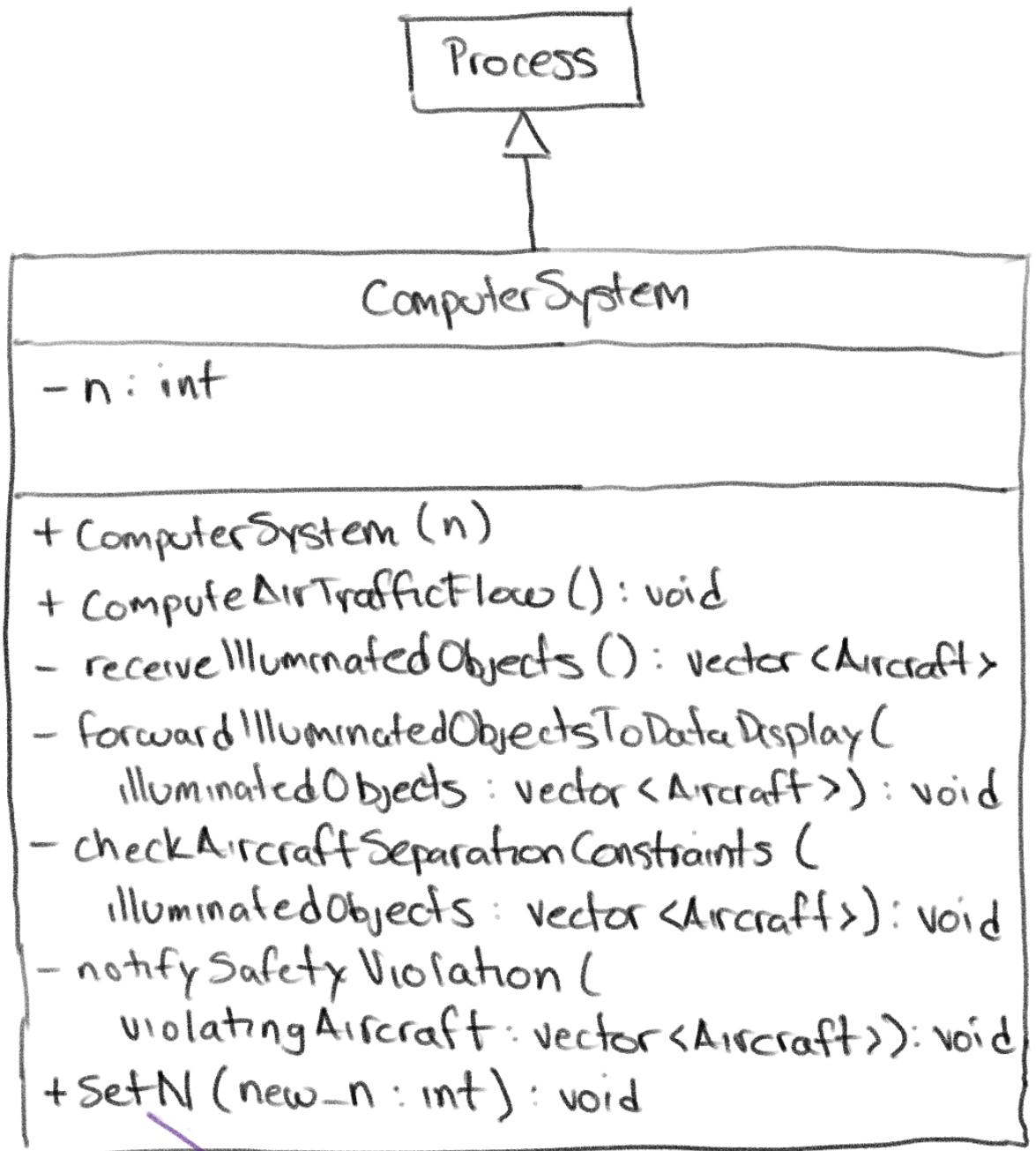
PAGE 9



# COMPUTE AIR TRAFFIC FLOW



# COMPUTER SYSTEM



Multiple options here:

- 1) Receive new `n` through interprocess communication
- 2) `n` is defined in shared memory
- 3) Operator Console has a reference to the `Computer System` object
- 4) Operator Console requests setting `n` through the Resource Manager

CHECK AIRCRAFT SEPARATION CONSTRAINTS

PAGE 11