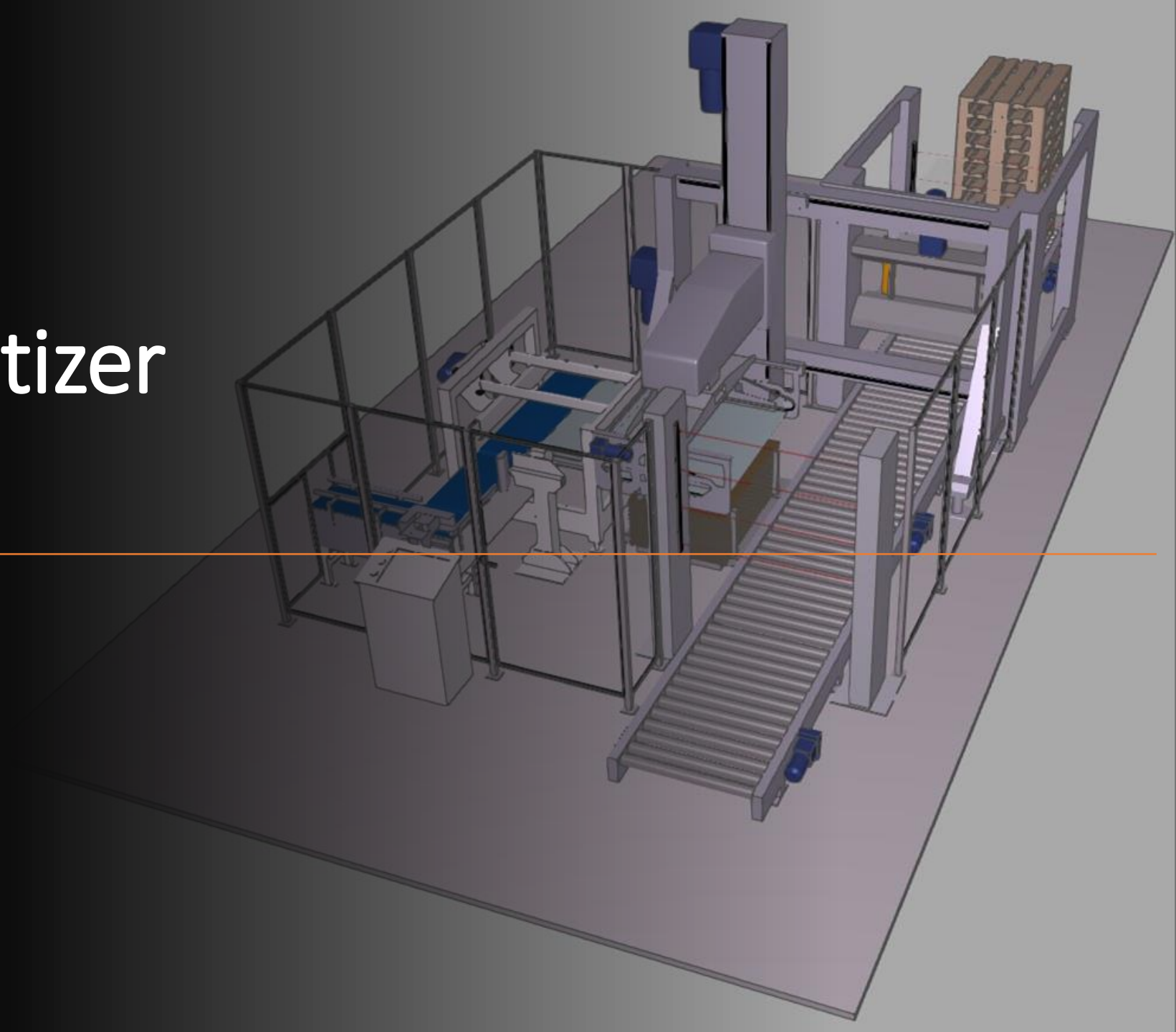
MultiBox Palletizer PROJECT



GROUP:

- Caselli Matteo
- Ricciardi Agatino
- Zattini Leonardo

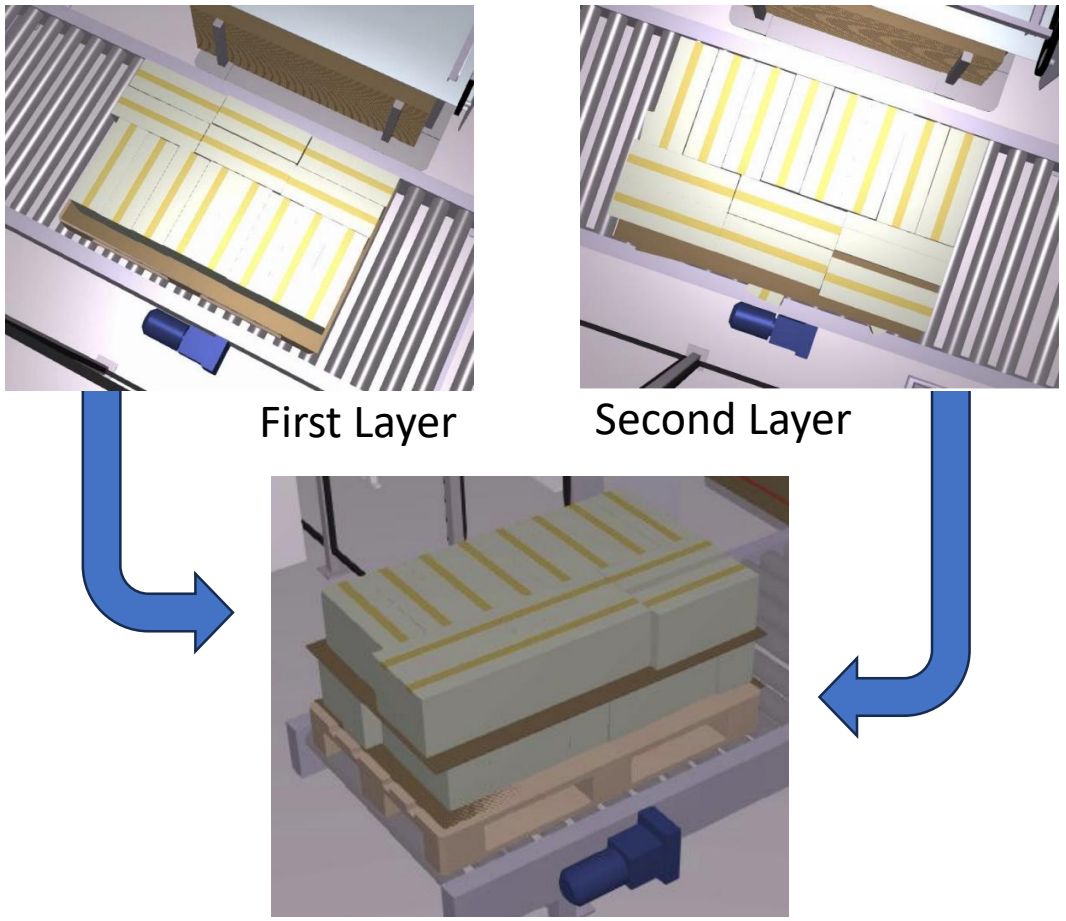
MULTIBOX PALLETIZER – SPECIFICS

The objective of this machine is to create ***pallets*** containing:

- Two ***layers*** of 14 boxes each;
- Each layer is formed by three lines of boxes with two different orientation;
- The ***box orientation*** is alternated among layers.

A ***cartboard*** must be included at the bottom of each box layer.

Box dimensions: 390x200x140 mm.



SETUP

B&R Automation Studio

Software tool for development and design of automatic solutions

Industrial Physics

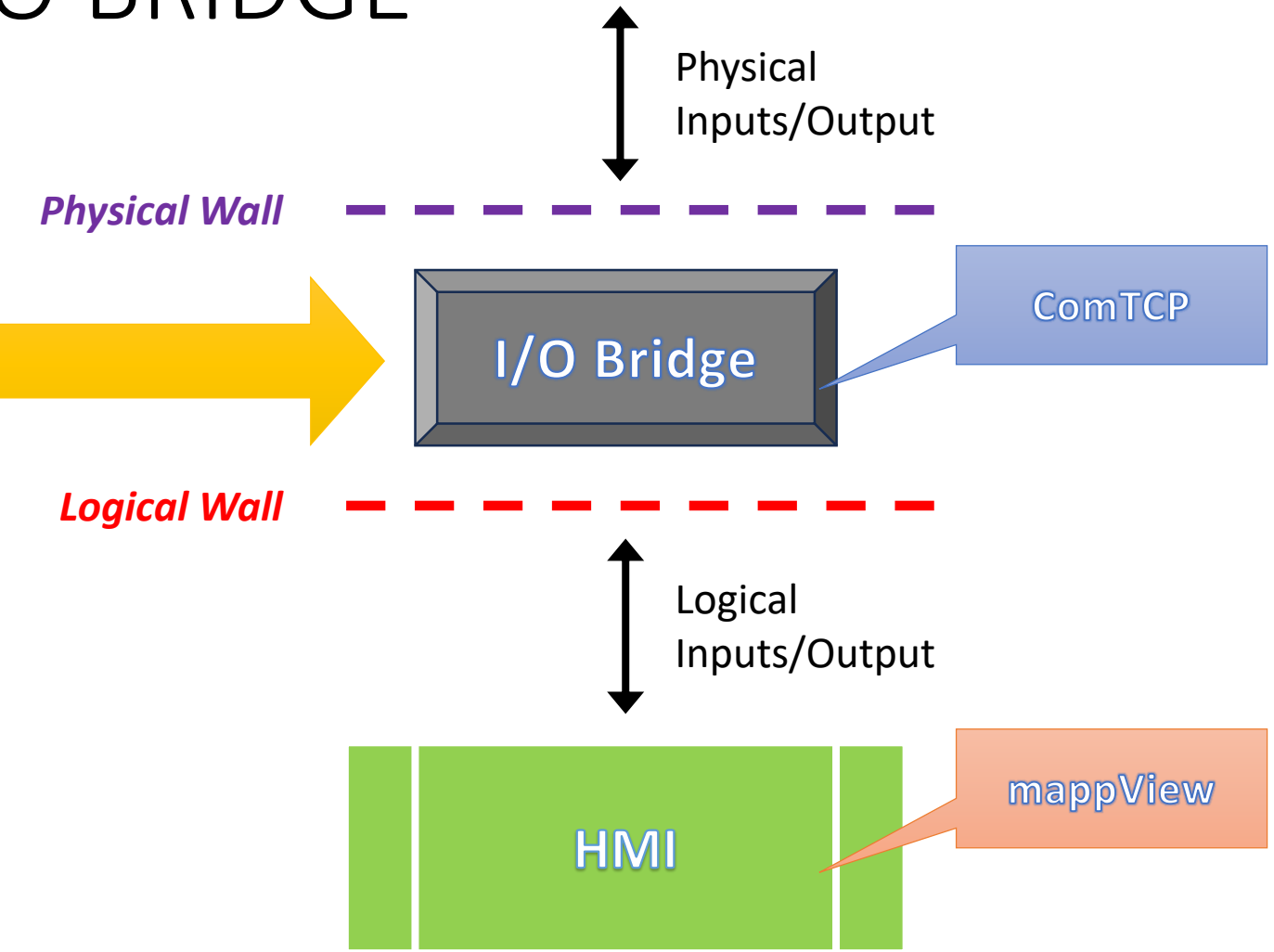
INDUSTRIAL PHYSICS

Software tool for digital design and simulation of automatic systems

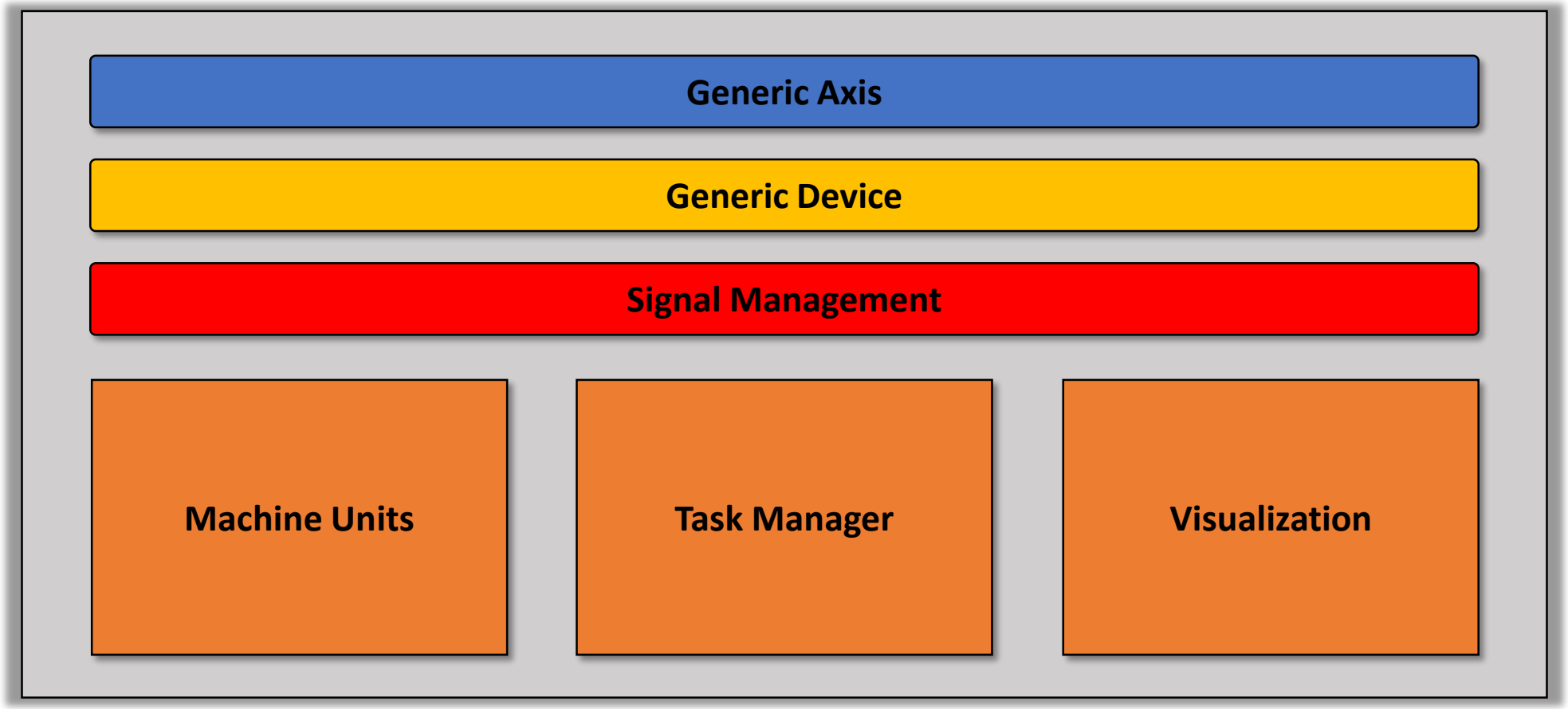
I/O BRIDGE

The **Input Bridge** allows physical variables to be mapped into logical ones.

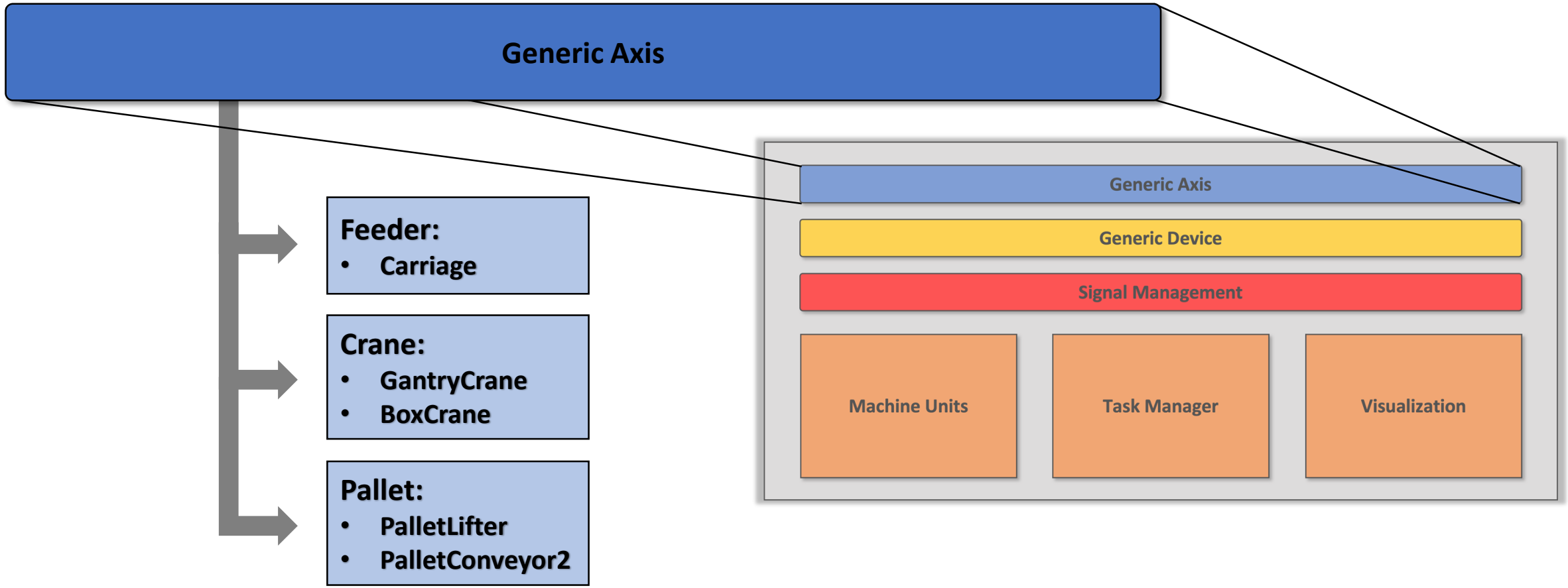
The **Output Bridge** allows logical variables to be mapped into physical ones.



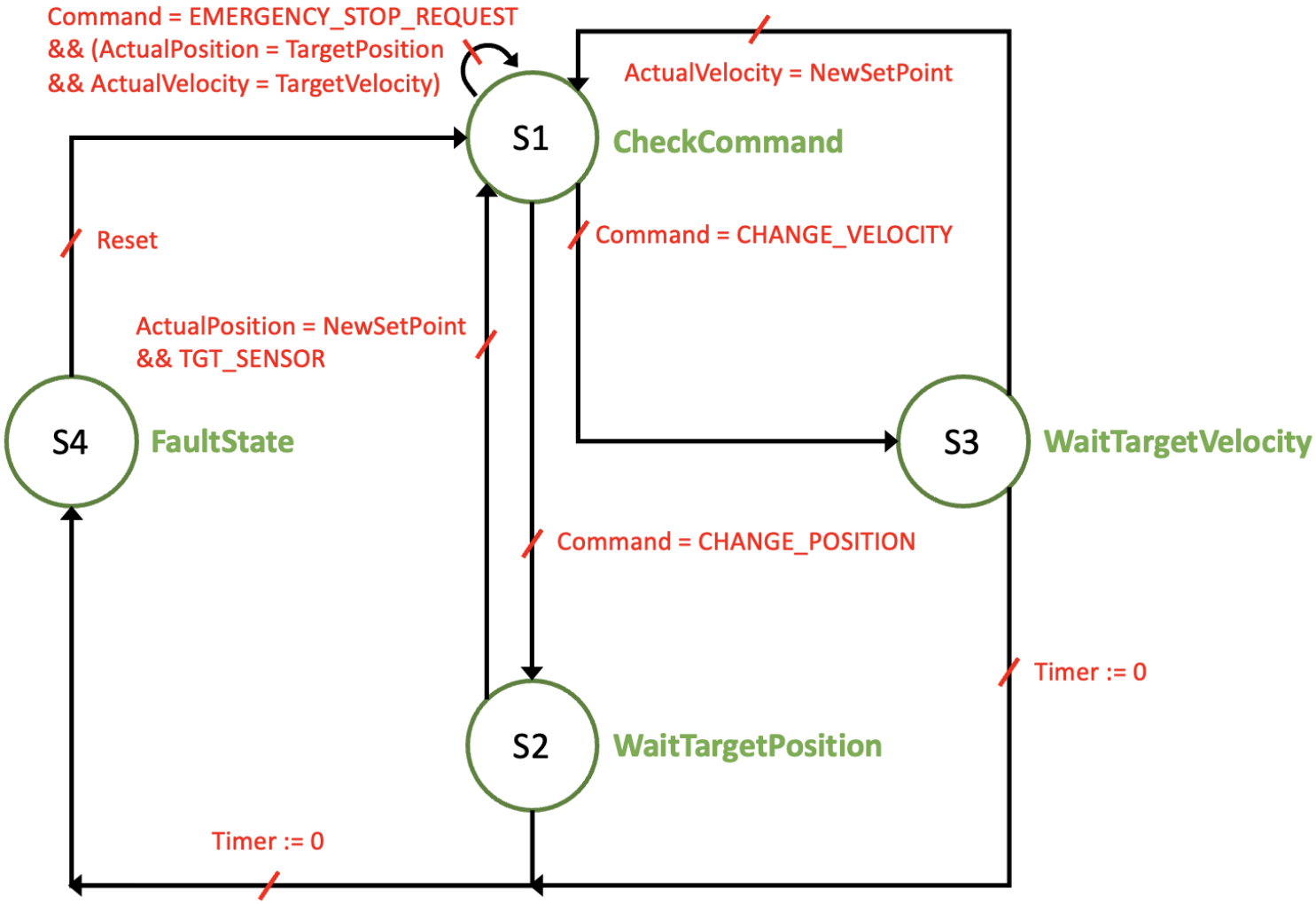
OVERALL PROJECT STRUCTURE



OVERALL PROJECT STRUCTURE

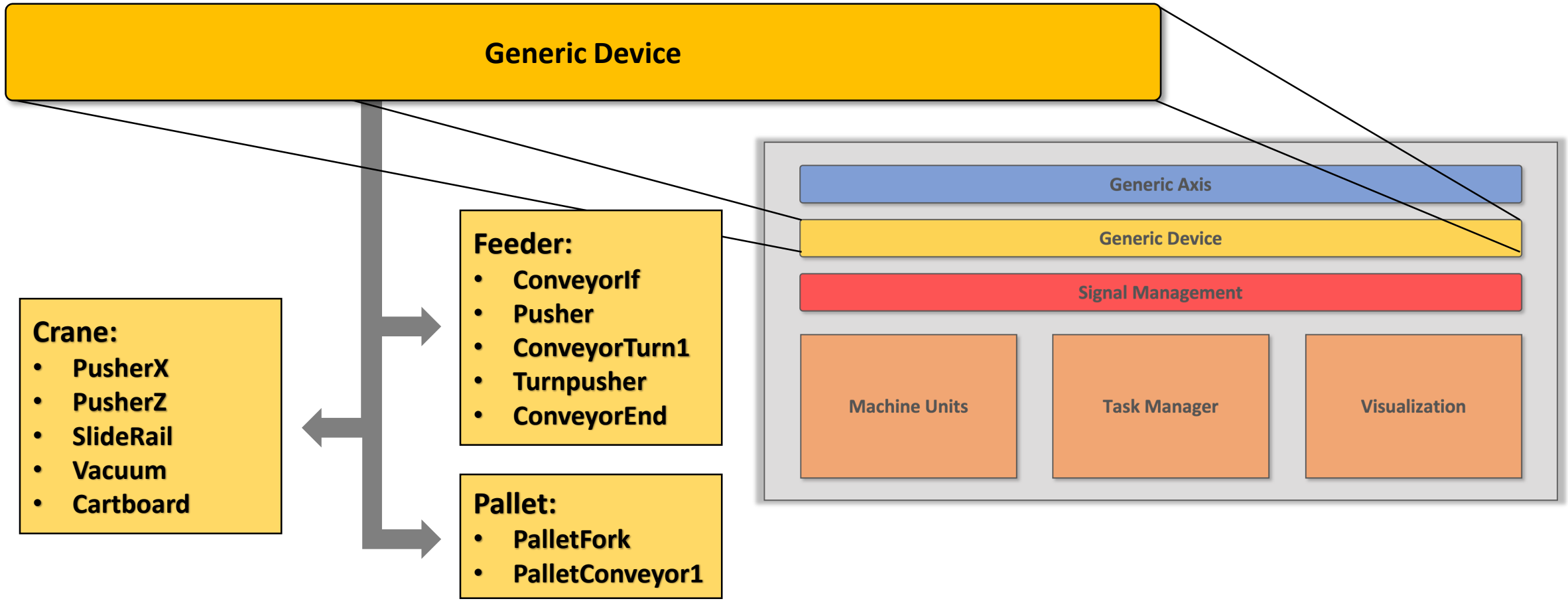


FSM DESIGN: GENERIC AXIS

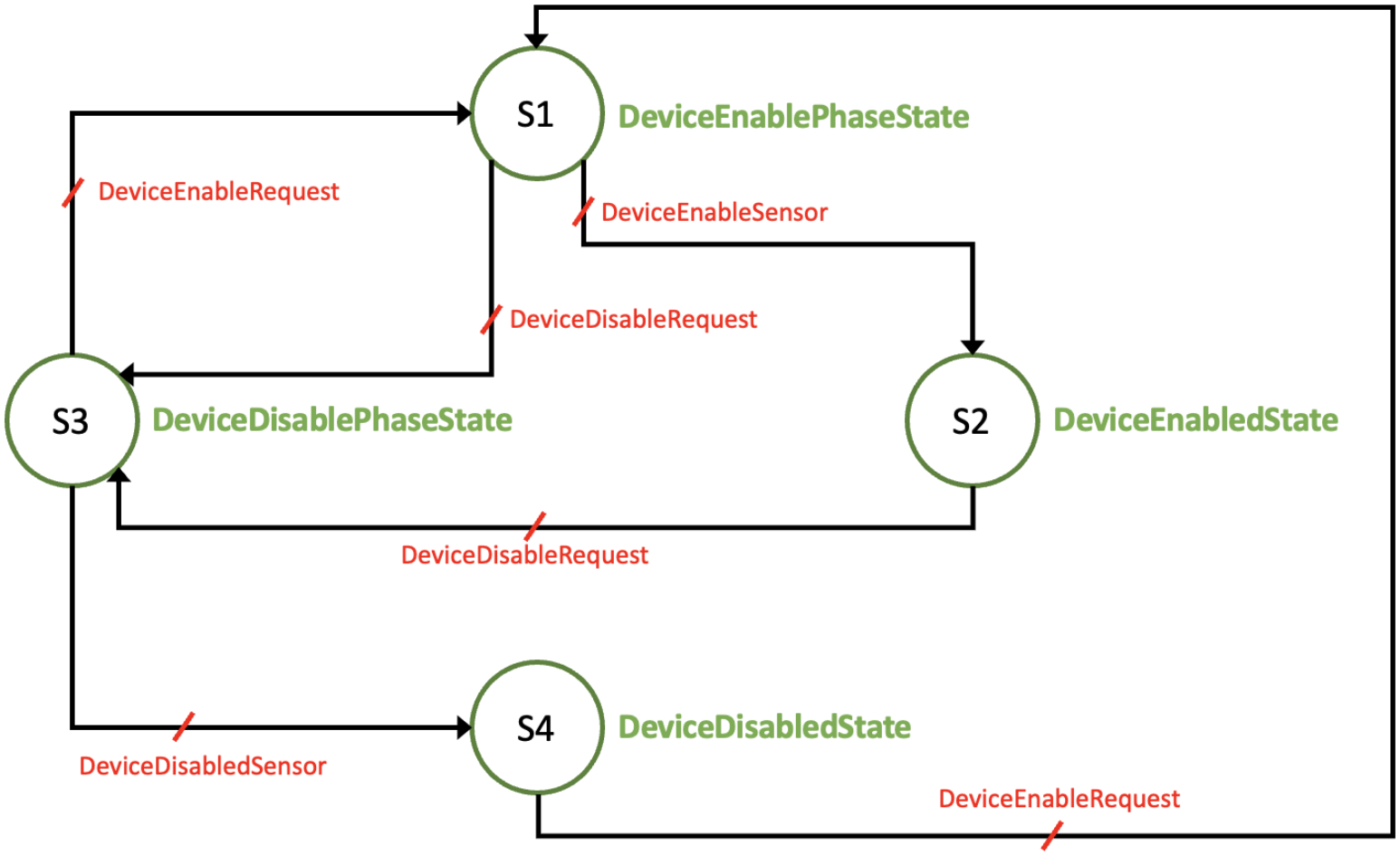


The idea is to generalize the control of a device thas must execute some specific command, such that, independently from its type, it can be controlled by standard commands.

OVERALL PROJECT STRUCTURE

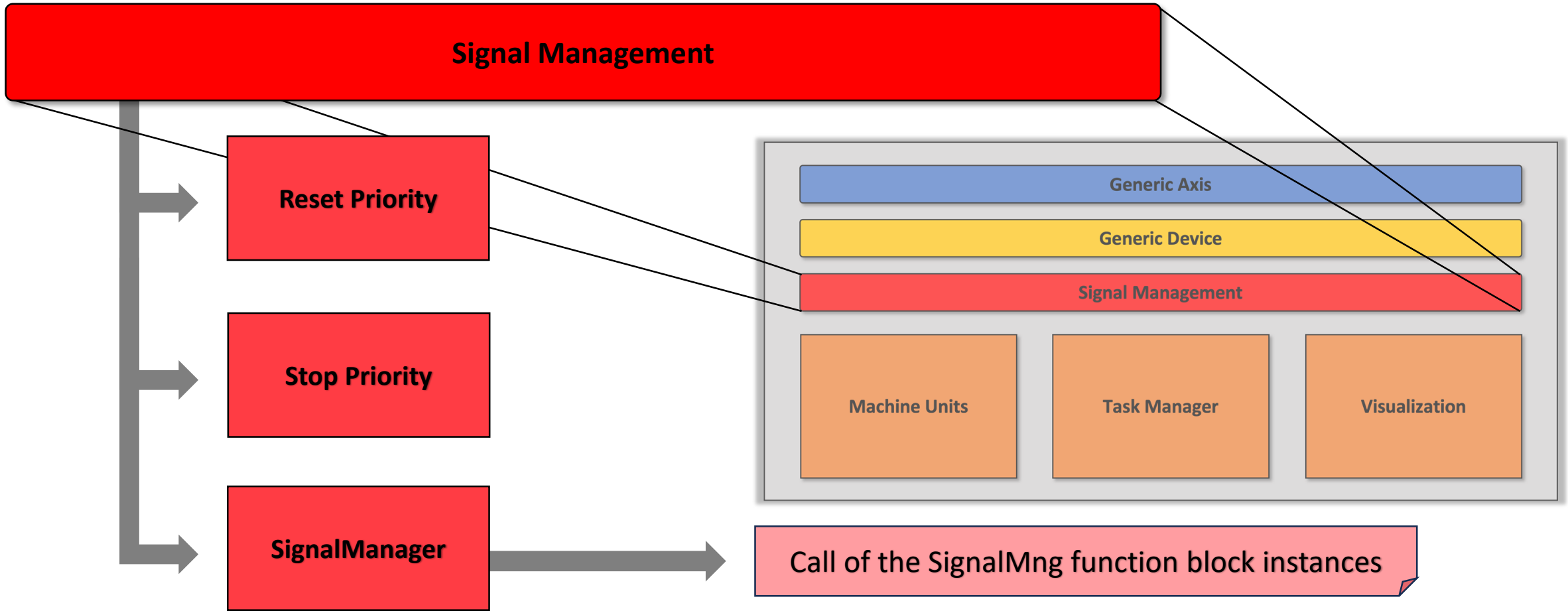


FSM DESIGN: GENERIC DEVICE



The idea is to generalize a device that can be controlled by BOOL variables, such that, independently from its actuators, it can be controlled in the same way.

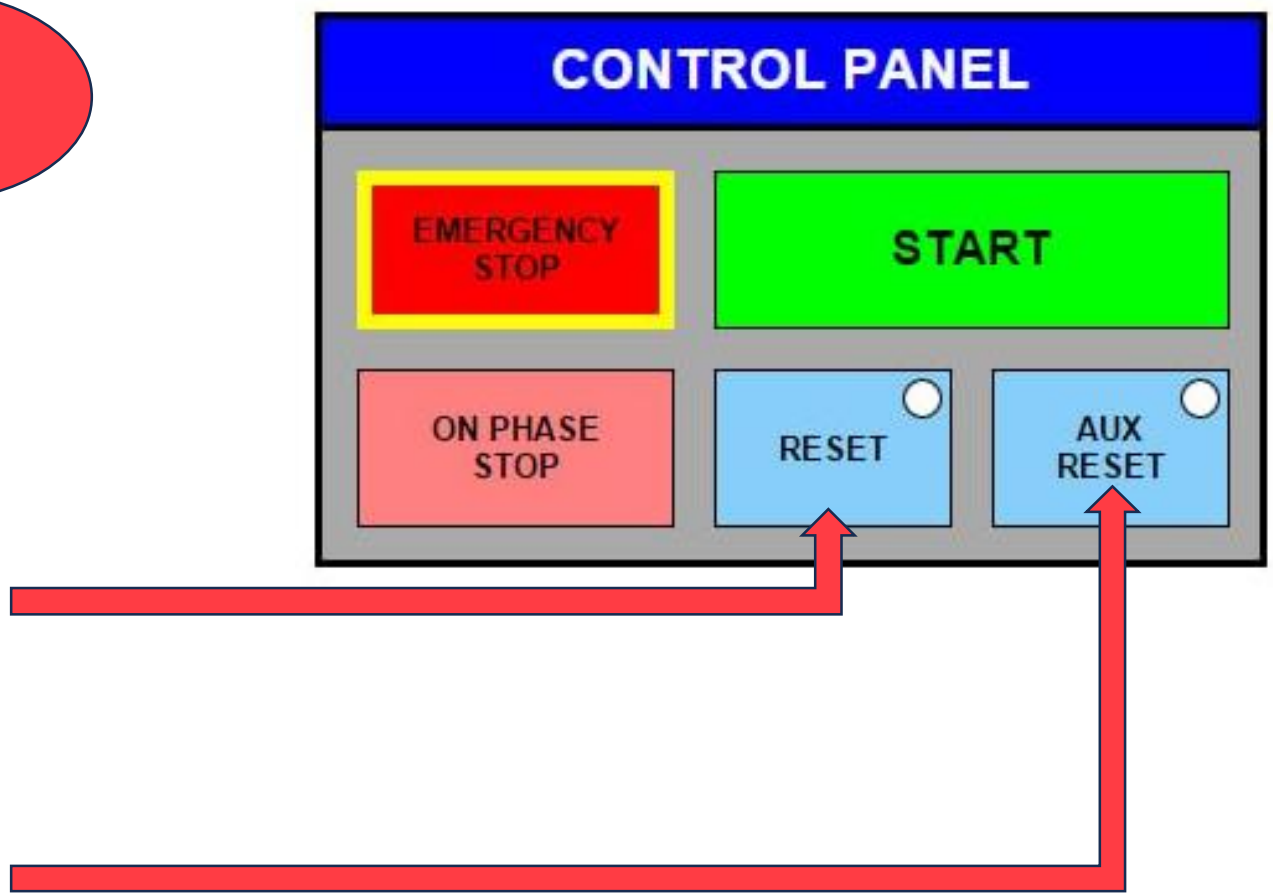
OVERALL PROJECT STRUCTURE



RESET PRIORITY

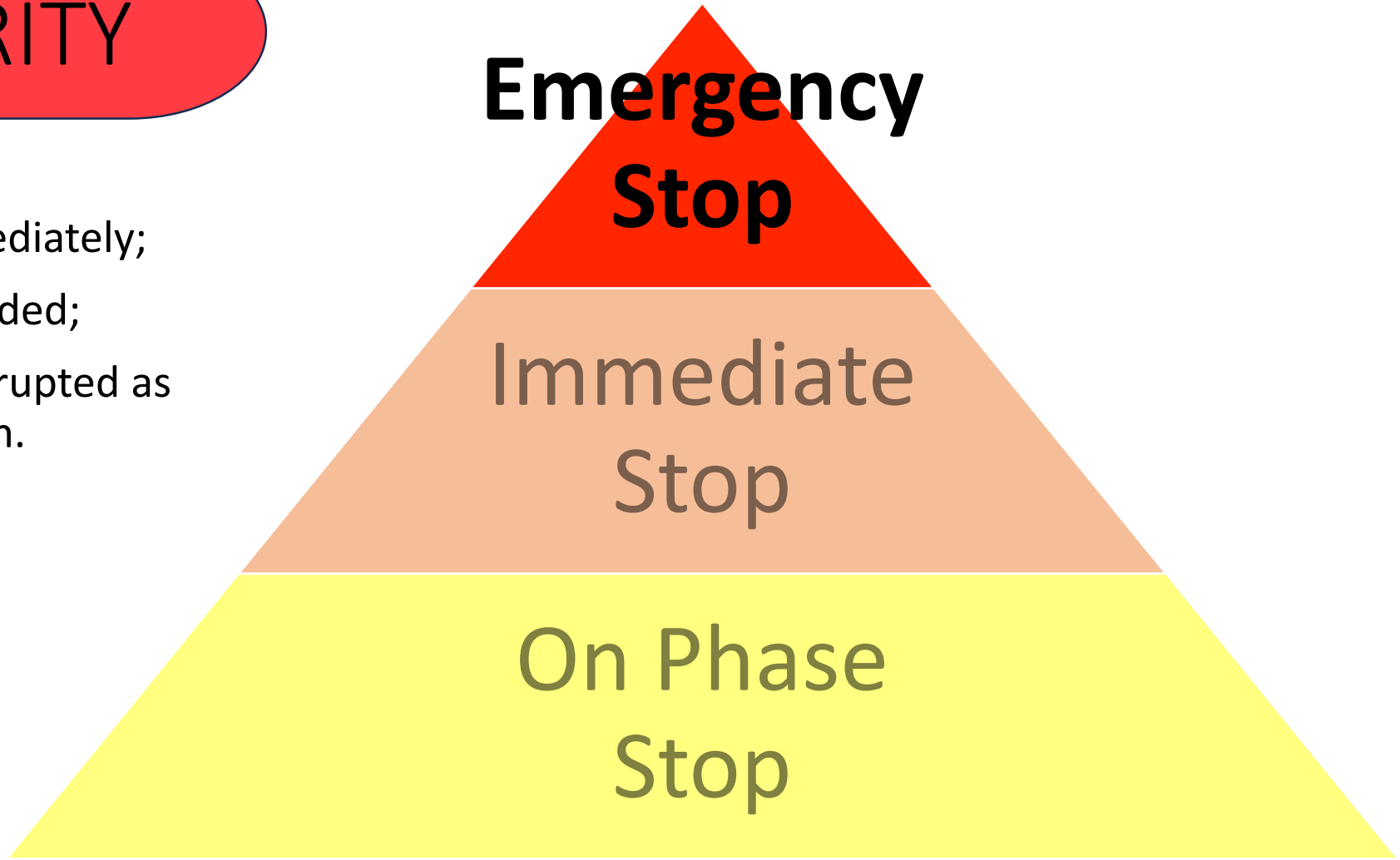
Two types of **Selective Reset** are implemented with **Dynamic Signal Priority**:

- **Conventional Reset** (RESET), everyone can push the reset button to restore the execution.
- **Key Reset** (AUX RESET), only specialized operator can activate this signal to restore the execution.



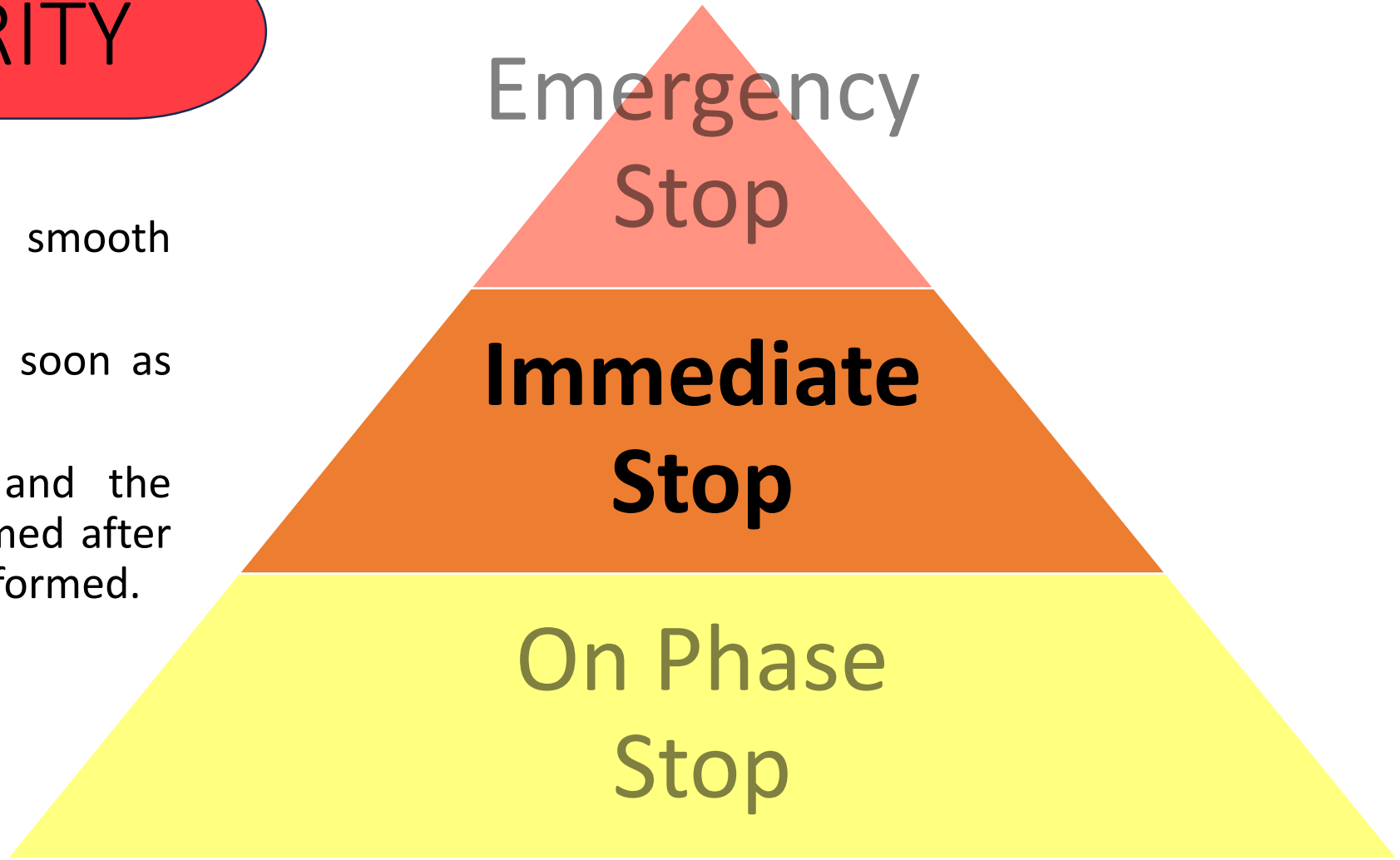
STOP PRIORITY

- Blocks the system immediately;
- A re-initialization is needed;
- Every execution is interrupted as soon as the stop is given.



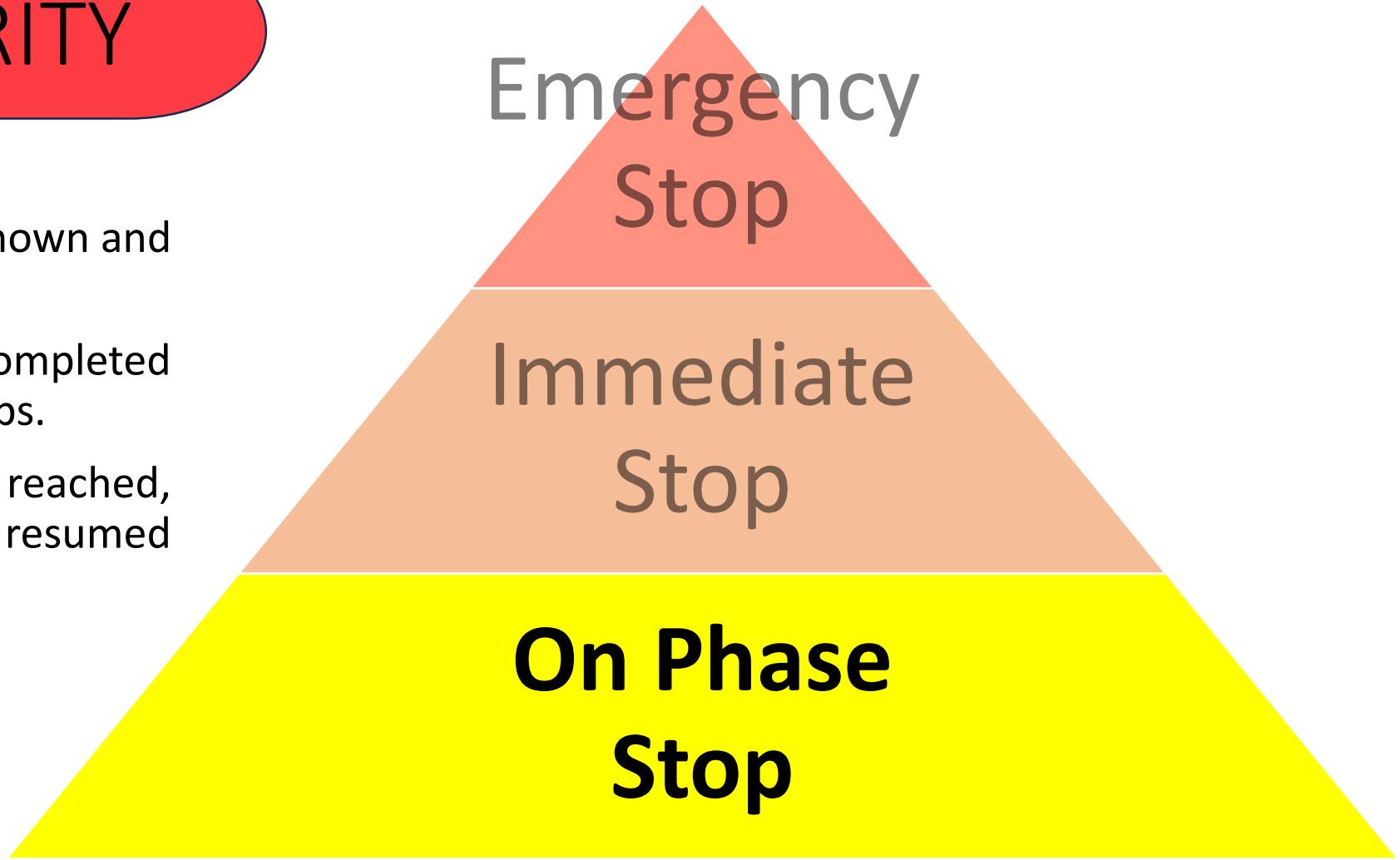
STOP PRIORITY

- Stops the system in a smooth and controlled way;
- Machine is stopped as soon as possible;
- The state is known and the execution can be resumed after an auxiliary reset is performed.



STOP PRIORITY

- Stops the system in a known and safe state;
- The operation is completed before the machine stops.
- When the safe state is reached, the execution can be resumed after a reset;

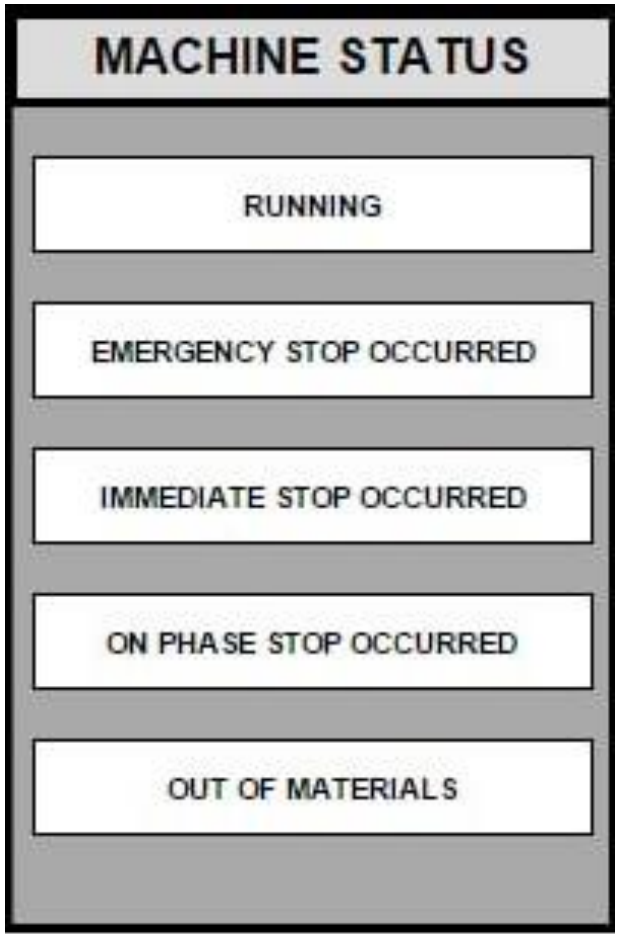


SIGNAL MANAGER

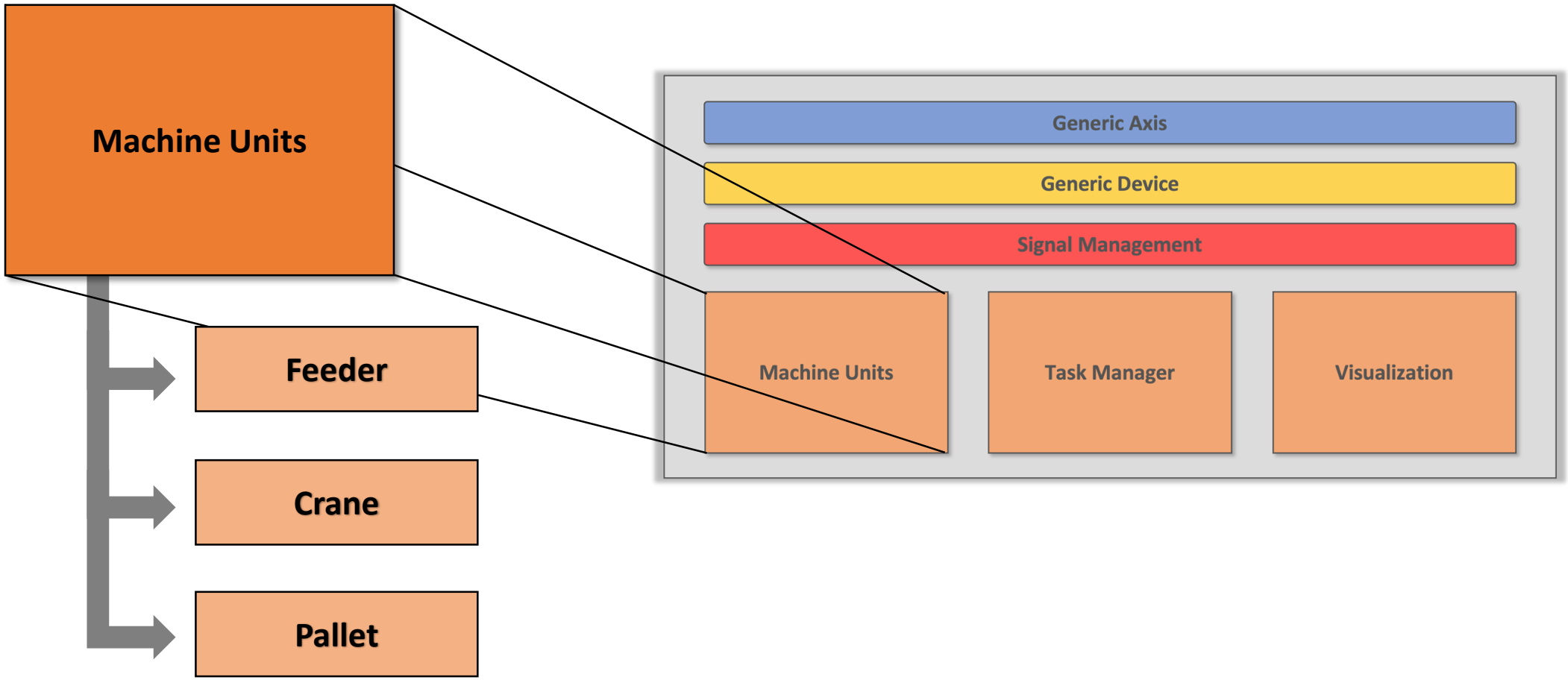
Inside the ***SignalManager Program*** there are as many instances of the ***SignalMng Function Block*** as the number of FAULTS (18), WARNINGS (2) and STOPS (3) taken into account:

- ***EmergencyStopButton*** ➢ ***Emergency Stop*** request;
- ***Fault*** ➢ ***Immediate Stop***;
- ***OnPhaseStopButton*** ➢ ***On Phase Stop*** request;
- ***Running Out of Materials*** ➢ ***On Phase Stop*** request;
- ***Materials almost Over*** ➢ ***Warning***;

The hierarchy among ***Stop Signals*** themselves and among **Stops** and ***Warnings*** is implemented.



OVERALL PROJECT STRUCTURE

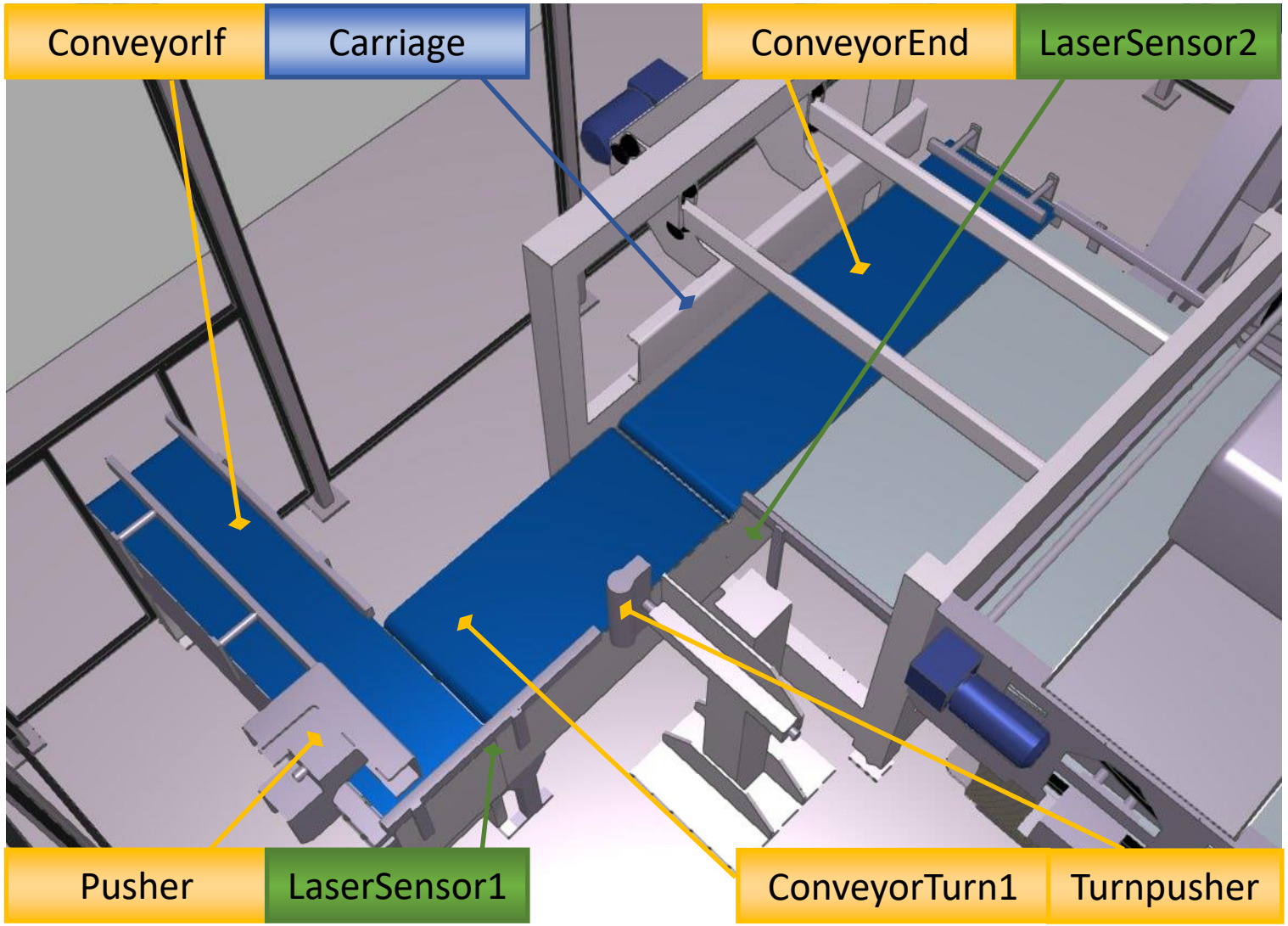


FEEDER

Generic Device:
- ConveyorIf;
- Pusher;
- ConveyorTurn1;
- Turnpusher;
- ConveyorEnd.

Generic Axis:
- Carriage.

Sensors (BOOL variables):
- LaserSensor1;
- LaserSensor2.



FEEDER - BEHAVIOUR

1. Boxes arrive singularly from ***ConveyorIf***;
2. The ***Pusher*** activates and pushes the boxes over ***ConveyorTurn1***, that arrive with a certain orientation onto ***ConveyorEnd***;
3. A layer is composed by 14 boxes placed with two different orientation, after the first disposition is done, the ***Turnpusher*** activates and changes the orientation of the boxes;
4. When the layer is ready, ***Carriage*** activates, feeding the layer to the CRANE.

Feeder States
F_StopSequence
F_StopDONE
F_ResetConfig
F_RestoreConfig1
F_RestoreConfig2
F_RestoreConfig3
F_WaitStart
F_WaitConveyorIFActive
F_WaitBoxPusherPosition
F_LayerCheck
F_LineCheck
F_WaitLaser2
F_BoxPosOk
F_WaitConveyorStop
F_Idle
F_CarriageFwdPosition
F_CarriageBwdPosition
F_WaitStartPosition

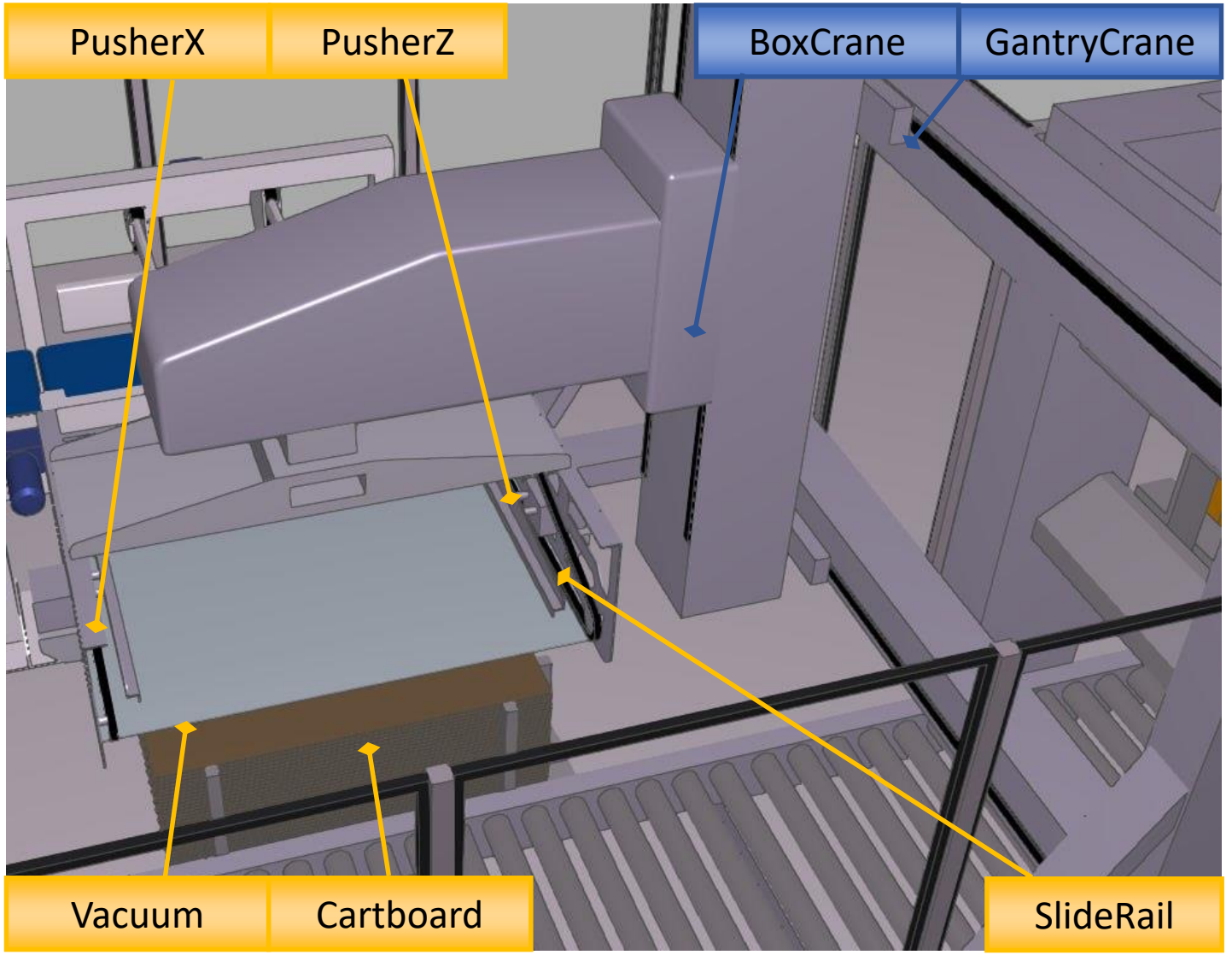
CRANE

Generic Device:

- PusherX;
- PusherZ;
- SlideRail;
- Vacuum;
- Cartboard.

Generic Axis:

- GantryCrane;
- BoxCrane.



CRANE - BEHAVIOUR

1. A layer of boxes arrives from the FEEDER;
2. The ***GantryCrane*** starts moving and reaches the set position; the ***Vacuum*** gets a ***Cartboard*** and the ***BoxCrane*** starts positioning;
3. ***PusherX*** and ***PusherZ*** are operated in order to place the ***BoxCrane*** and the ***GantryCrane*** such that the layer can be loaded onto the pallet;
4. The unloading operation is executed by releasing the ***SlideRail***; the initial position of the crane is restored to receive another layer of boxes.

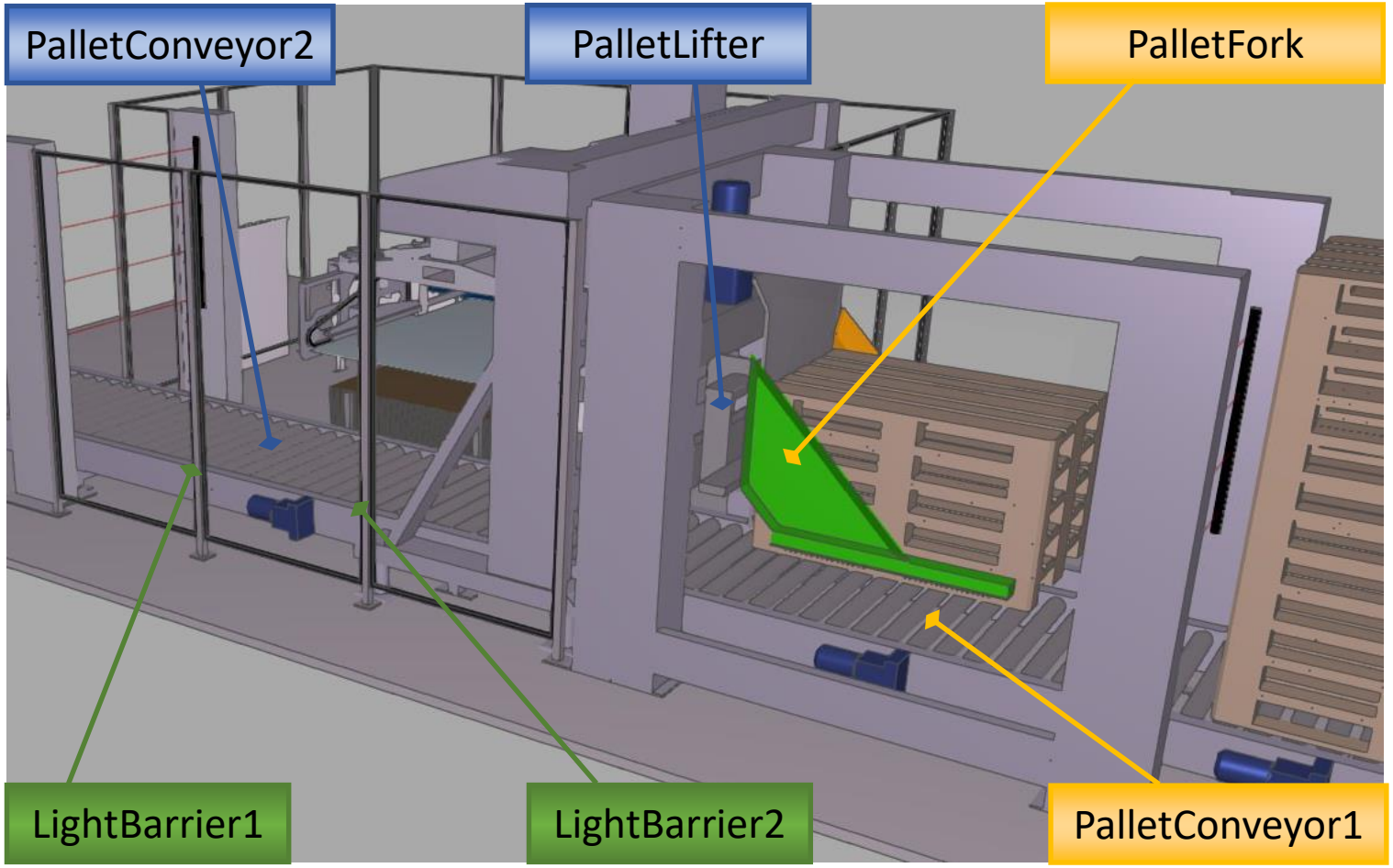
Crane States	
C_StopSequence	C_LockCartboard
C_StopDONE	C_WaitBoxes
C_ResetConfig	C_WaitSlideRailHold
C_RestoreConfig1	C_WaitPusherZPushUp
C_RestoreConfig2	C_WaitPusherXPushBwd
C_RestoreConfig3	C_Idle
C_WaitStart	C_WaitPusherZPushDown
C_WaitGantryCraneStartPosition	C_WaitBoxCraneCartboardPos
C_WaitCartboardLoadPosition	C_WaitGantryCraneCartboardPos
C_WaitCartboardLoad	C_CartboardPosOk
C_WaitBoxCraneStartPosition	C_BoxesPosOk
C_WaitPusherZStartDown	C_OpenSlideRail
C_WaitPusherXStartFwd	C_PusherFwd

PALLET

Generic Device:
- PalletConveyor1;
- PalletFork.

Generic Axis:
- PalletLifter;
- PalletConveyor2.

Sensors (BOOL variables):
- LightBarrier2;
- LightBarrier1.

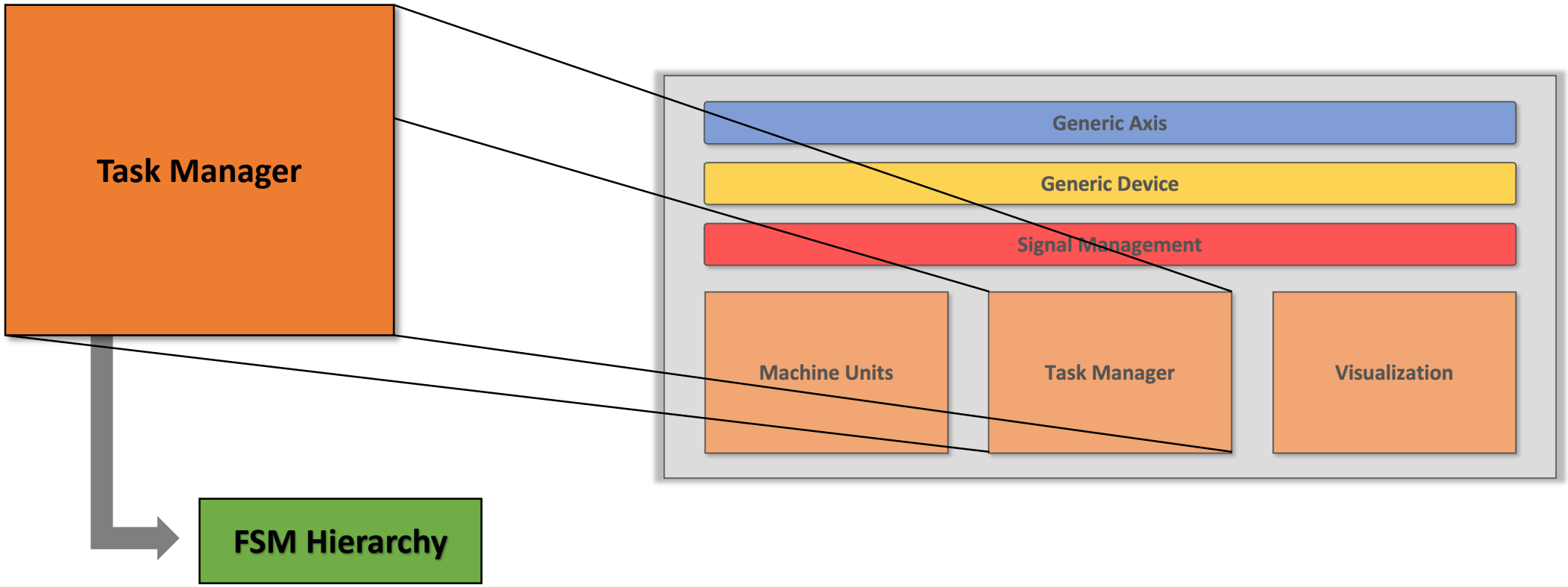


PALLET - BEHAVIOUR

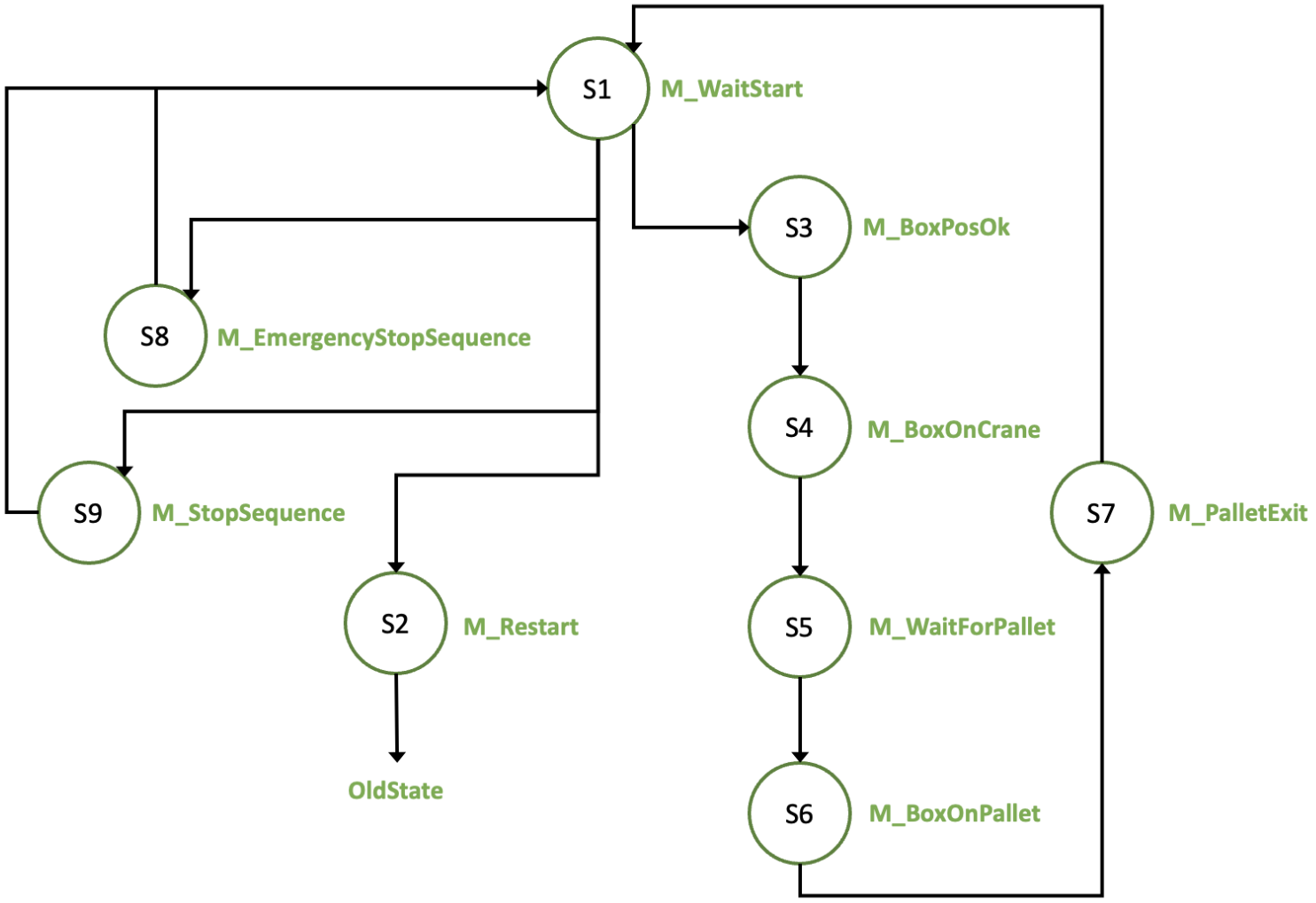
1. The ***PalletLifter*** moves down, when in position, ***PalletFork*** activates and deposits one pallet on top of ***PalletConveyor1***;
2. The ***PalletConveyor1*** moves until the empty pallet reaches ***PalletConveyor2***;
3. Velocity of ***PalletConveyor2*** slows down until the pallet reaches the set point, ready to receive the layer of boxes from the CRANE;
4. Once everything is placed, the conveyor restarts sending out the pallet to the WAREHOUSE.

Pallet States	
P_StopSequence	P_WaitLifterIdle
P_StopDONE	P_WaitConveyor1
P_ResetConfig1	P_WaitConveyor2
P_ResetConfig2	P_WaitLightBarrier2
P_ResetConfig3	P_WaitStopConveyor1
P_RestoreConfig1	P_WaitConveyor2SlowDown
P_RestoreConfig2	P_WaitLightBarrier1
P_RestoreConfig3	P_WaitStopConveyor2
P_WaitStart	P_WaitBoxes
P_WaitLifterDown	P_WaitRestartConveyor2
P_ForkOpen	P_WaitNOTLightBarrier2
P_WaitLifterUp	P_WaitExitVelocity
P_ForkClose	

OVERALL PROJECT STRUCTURE



FSM DESIGN: TASK MANAGER

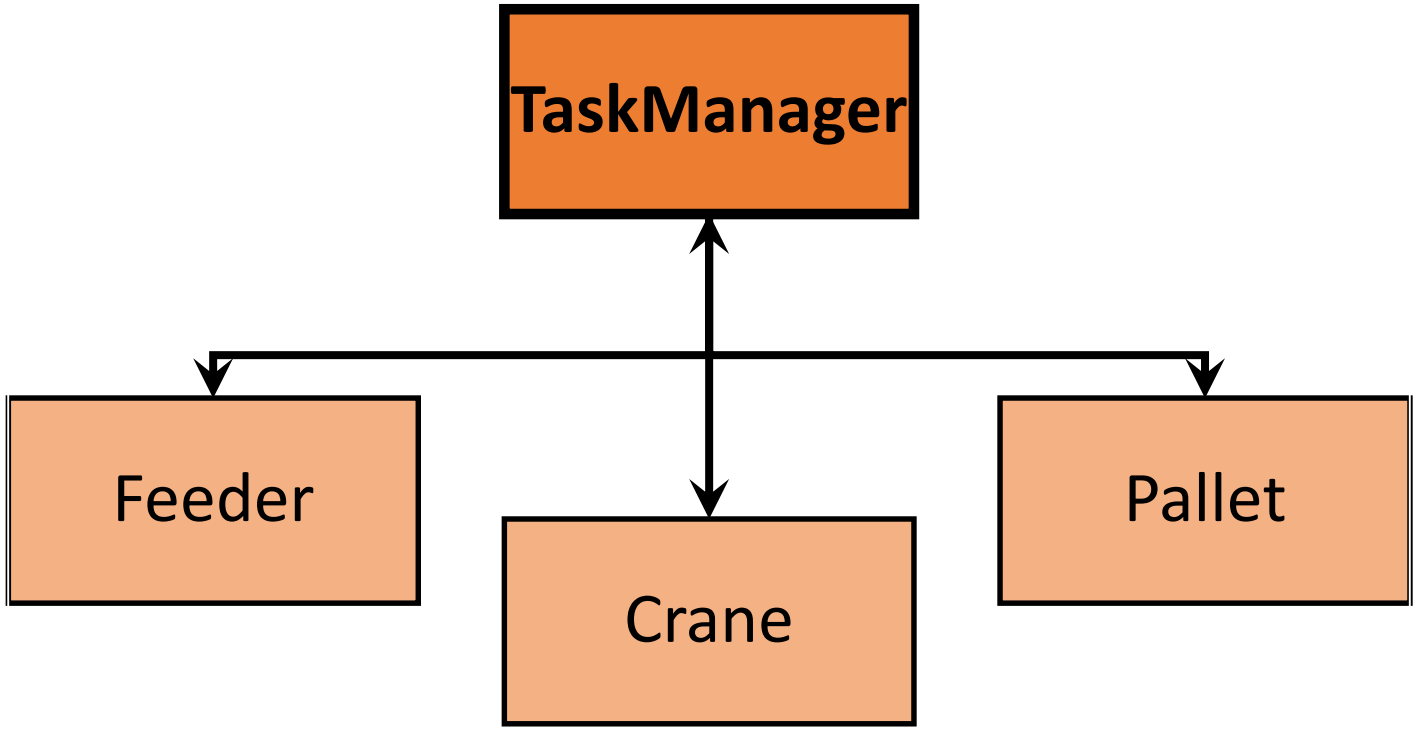


TaskManager States
M_EmergencyStopSequence
M_StopSequence
M_Restart
M_WaitStart
M_BoxPosOk
M_BoxOnCrane
M_WaitForPallet
M_BoxOnPallet
M_PalletExit

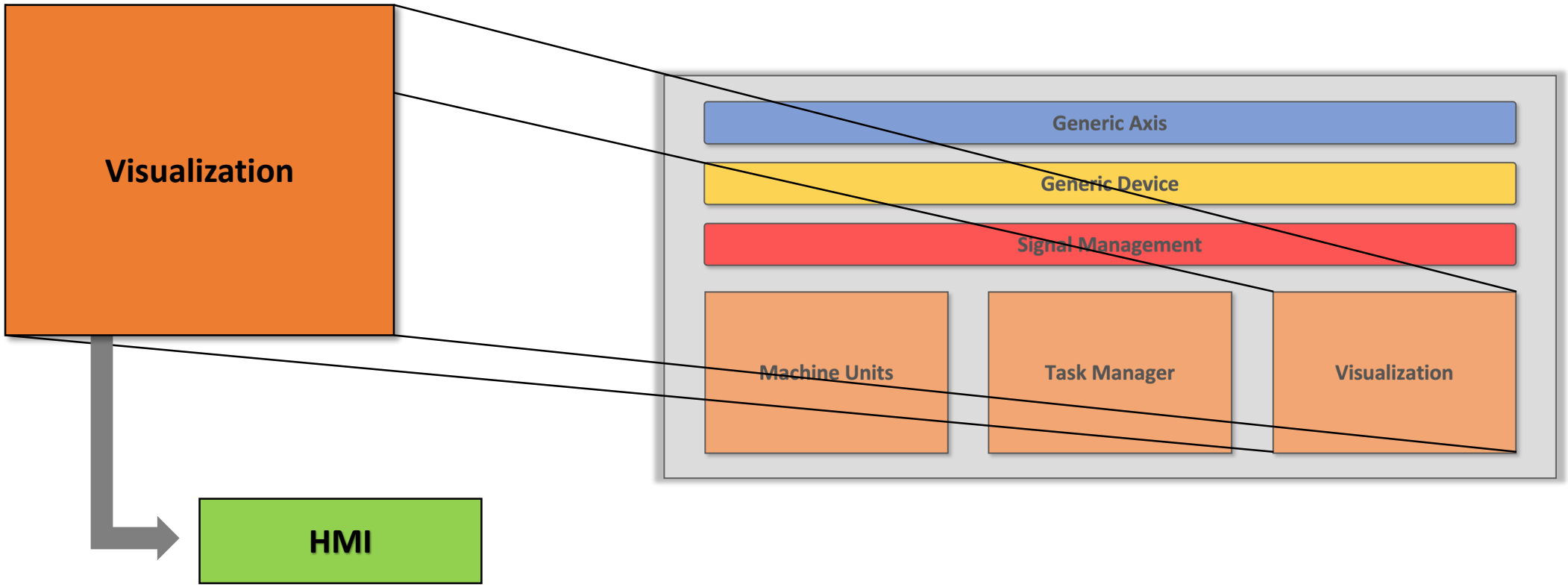
FSM HIERARCHY

The ***TaskManager Program*** coordinates the behaviour of the entire machine.

Feeder, ***Crane*** and ***Pallet*** FSM do not communicate among them, but only via the ***TaskManager***.

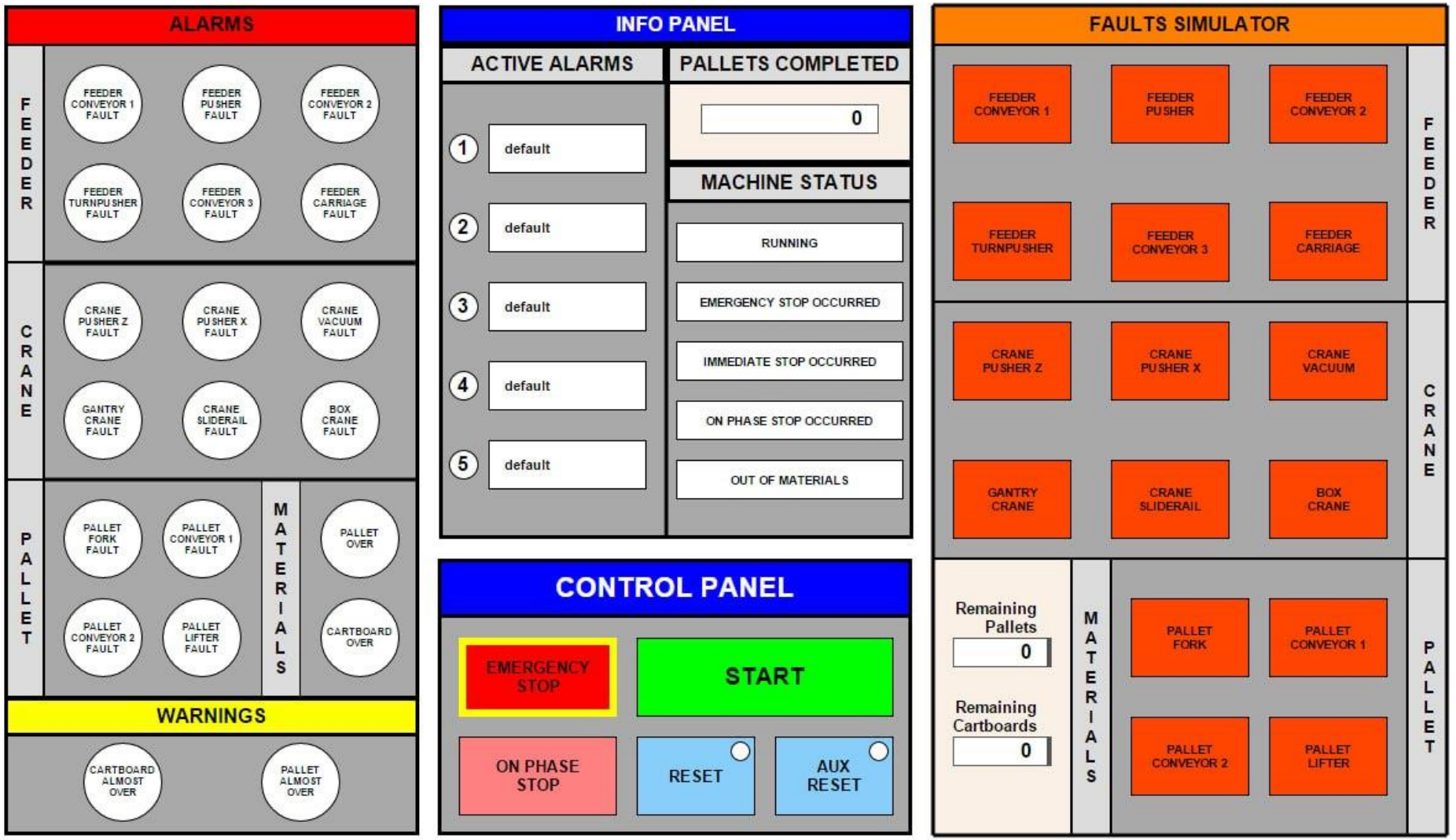


OVERALL PROJECT STRUCTURE

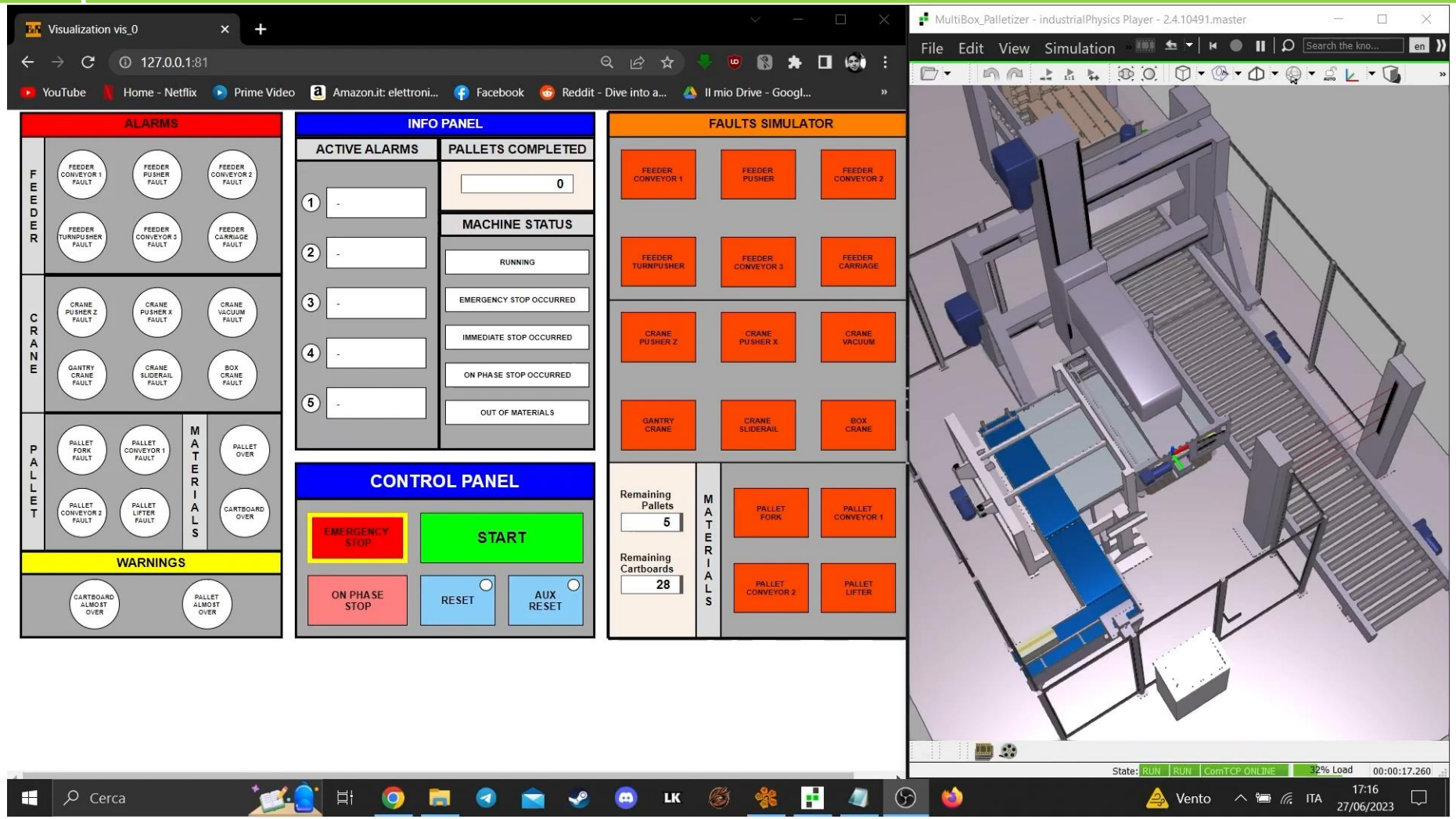


HMI

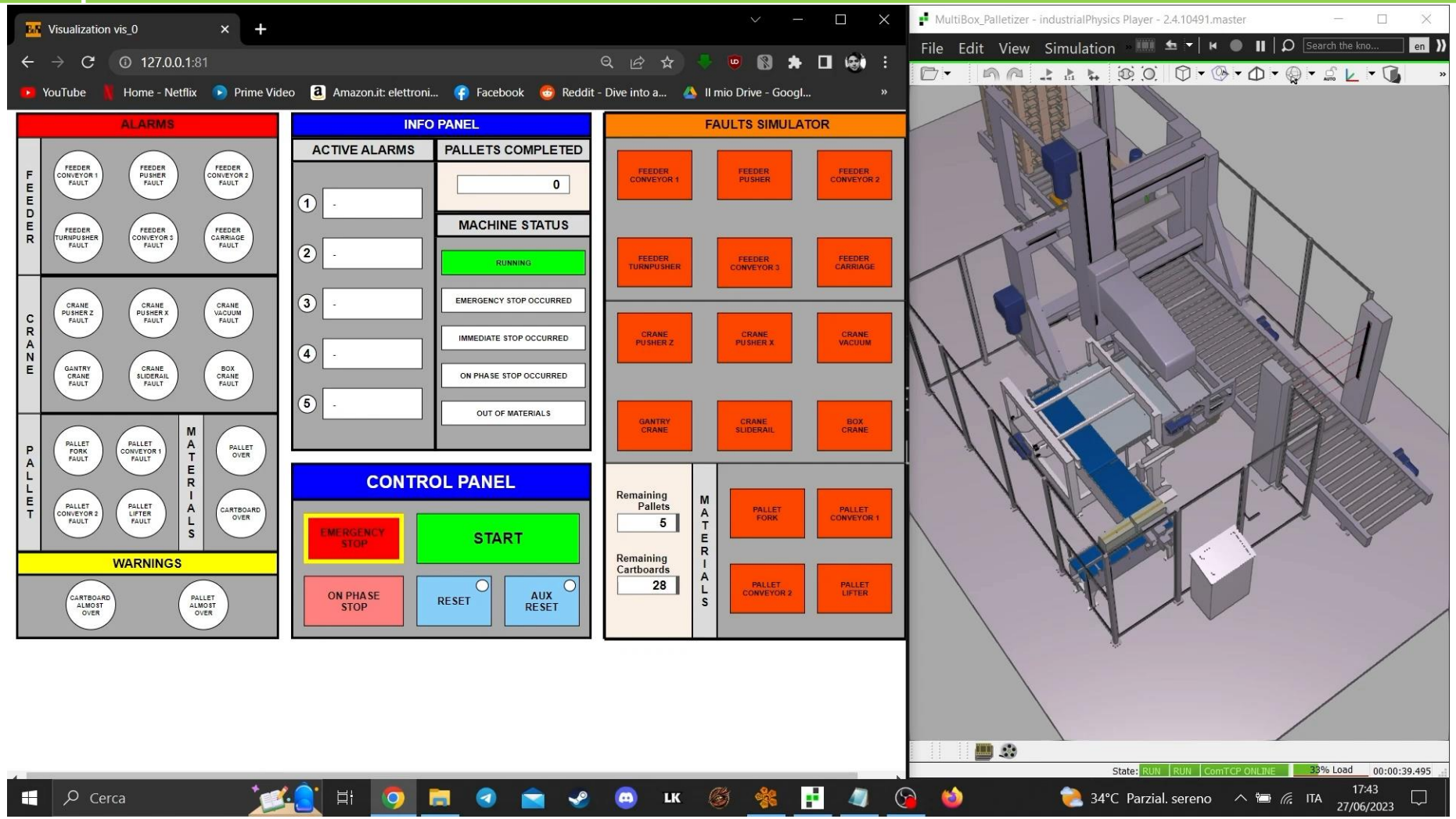
MULTIBOX PALLETIZER - HMI



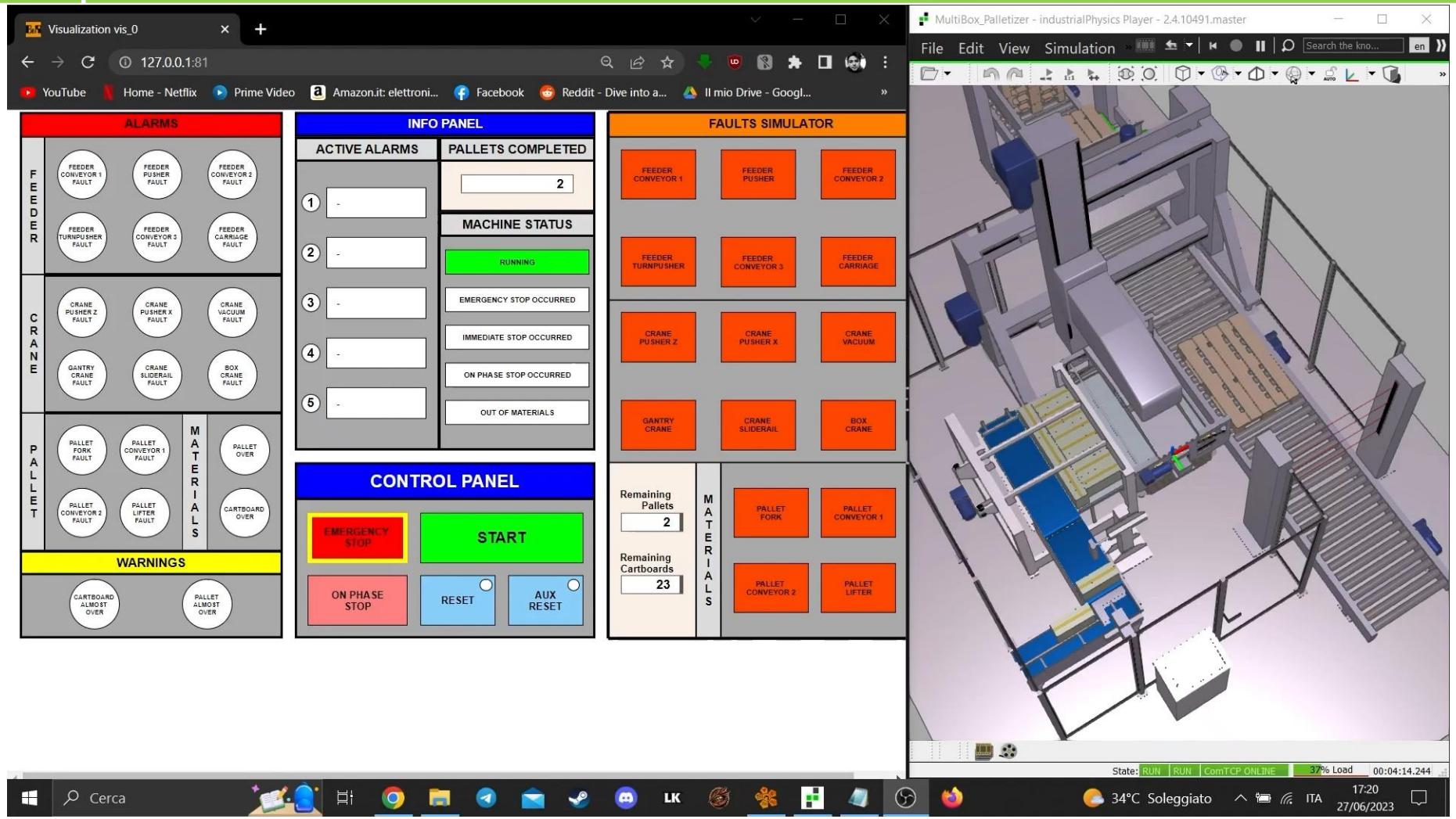
EXECUTION IN NOMINAL CONDITIONS



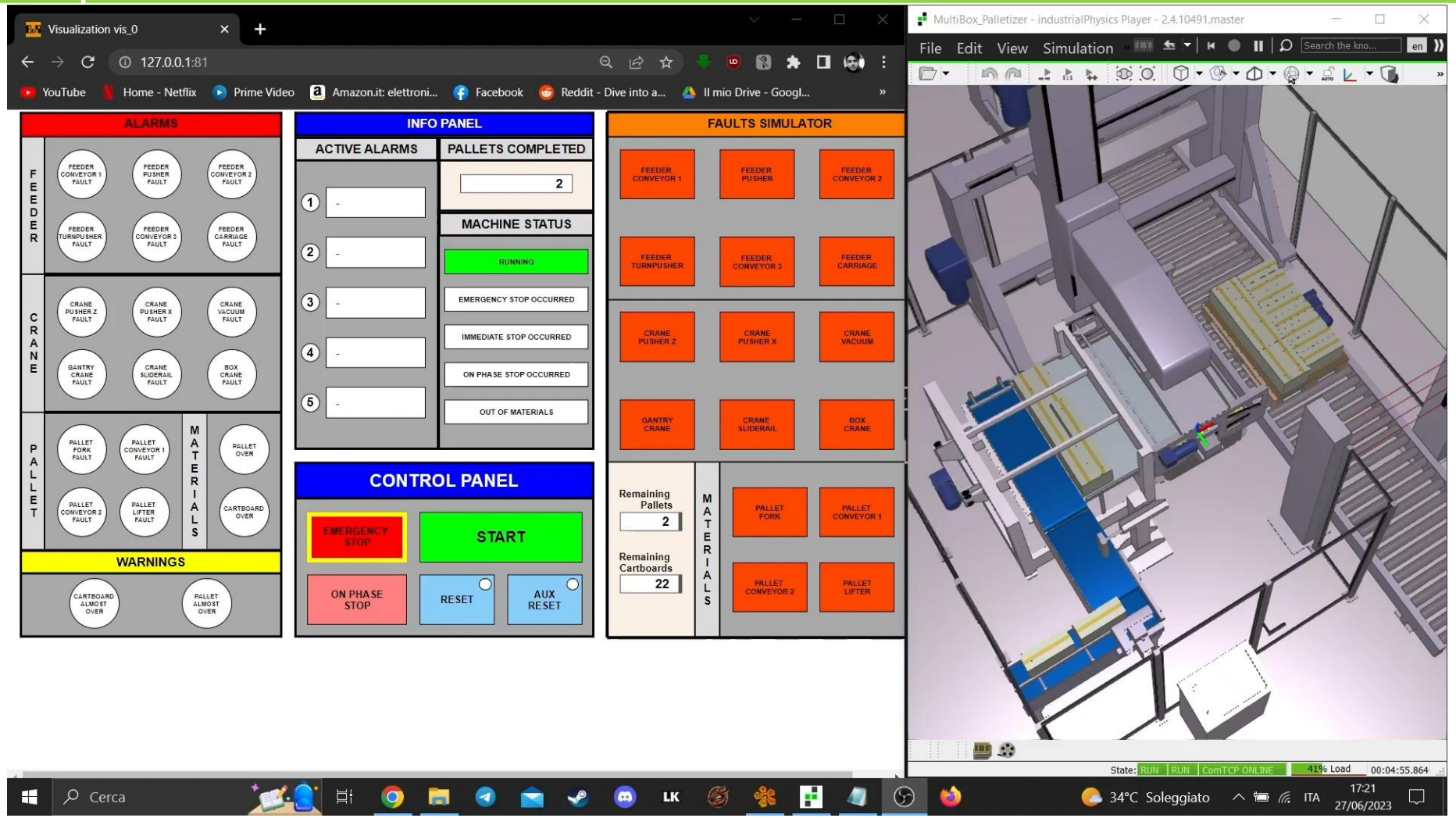
WARNING + ON PHASE STOP



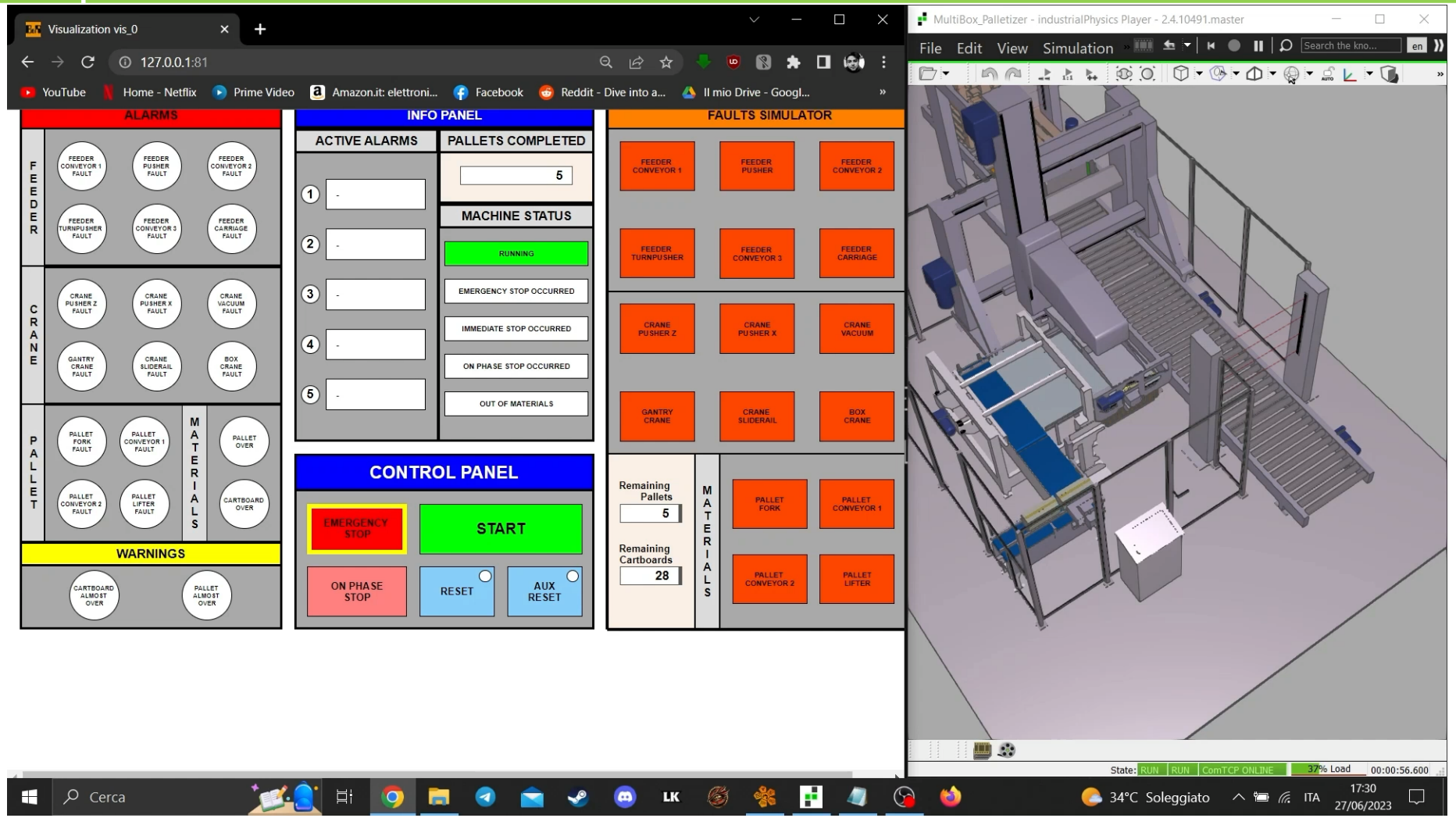
ON PHASE STOP + RESET



IMMEDIATE STOP + AUX RESET



EMERGENCY STOP + AUX RESET



STOP PRIORITY HANDLING