Application Code Manager RA Libraries

Jacob Berg & Brian Askjaer

Rockwell Automation

expanding **human possibility**™

Agenda

1 PackML

2 Framework B

3 DEMO

Create a project and your own libarary

4 Emulate3D and ACM

Rockwell Automation

Pack ML

Rockwell Automation

The History

1994 – OMAC is formed as The Open Modular Architecture Controls User's Group to provide an organization for companies to work together and:

- Collectively derive ***common solutions for both technical and non-technical issues*** in the development, implementation, and commercialization of open, **modular architecture control technologies**
- ***Promote open, modular architecture*** control development among control technology providers and adoption among end users, OEMs, and system integrators
- Act as a repository for open, ***modular architecture control requirements and operating experience*** from users, software developers, hardware builders and OEMs in manufacturing applications
- Facilitate the accelerated development and convergence of industry- and government-developed technology guidelines to one set that satisfies common use requirements
- Collaborate with users groups around the world in pursuit of common international technology guidelines.

OMAC

The Organization for Machine Automation and Control

Rockwell Automation

How is OMAC

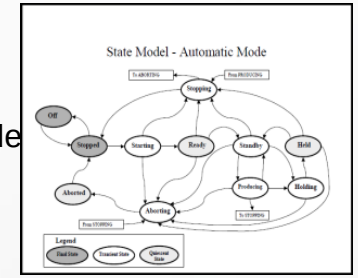
Technology Providers

End Users & Original Equipment Manufacturers

Rockwell Automation

The History

PackML v2.2 Released, include 10 state model



PackML v3.0 Released, 17 state model

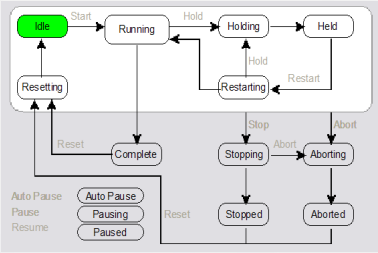
PackML ISA-TR88.00.02 Released

OMAC template available for download

Latest template released

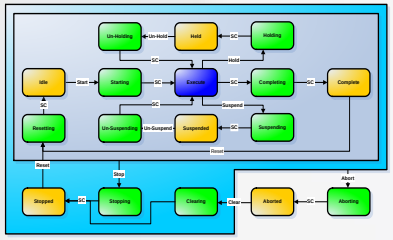
2003 2004 2005 2006 2007 2008 2009 2010 … 2015 … 2018

First versions of current template are created and installed in manufacturing sites, using S88 11 state model



Various revisions and executions across multiple projects…

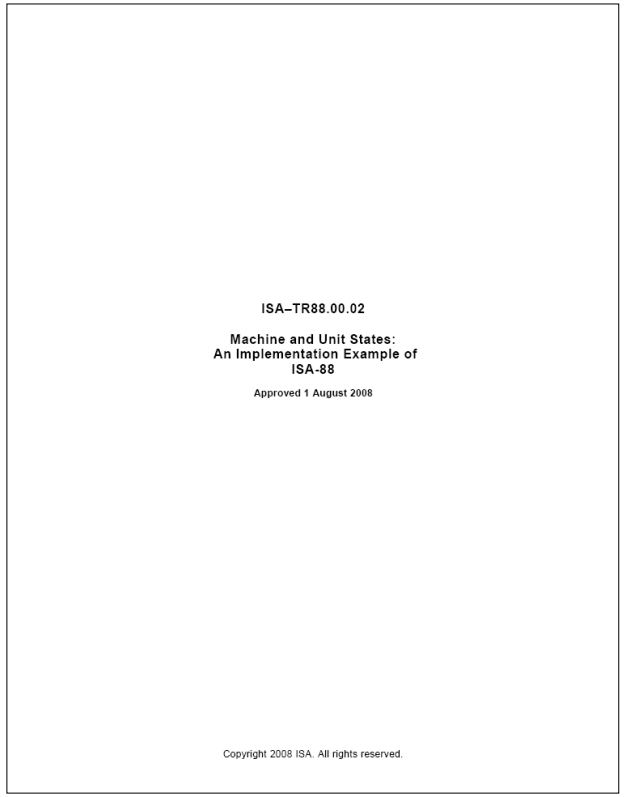
Template created using latest PackML Standard, ISA-TR88.00.02



Rockwell Automation

What is PackML?

- **Pack**aging **M**achine **L**anguage
  - Programming standard developed in early 2000's by the OMAC Packaging Workgroup Committee
- ISA-TR88.00.02-2008
  - Machine and Unit States: An Implementation Example of ISA-88
  - V3.0 released in 2008
- Combined with ISA-S88 Part 5, PackML provides a consistent machine programming approach for discrete automated equipment


ISA–TR88.00.02

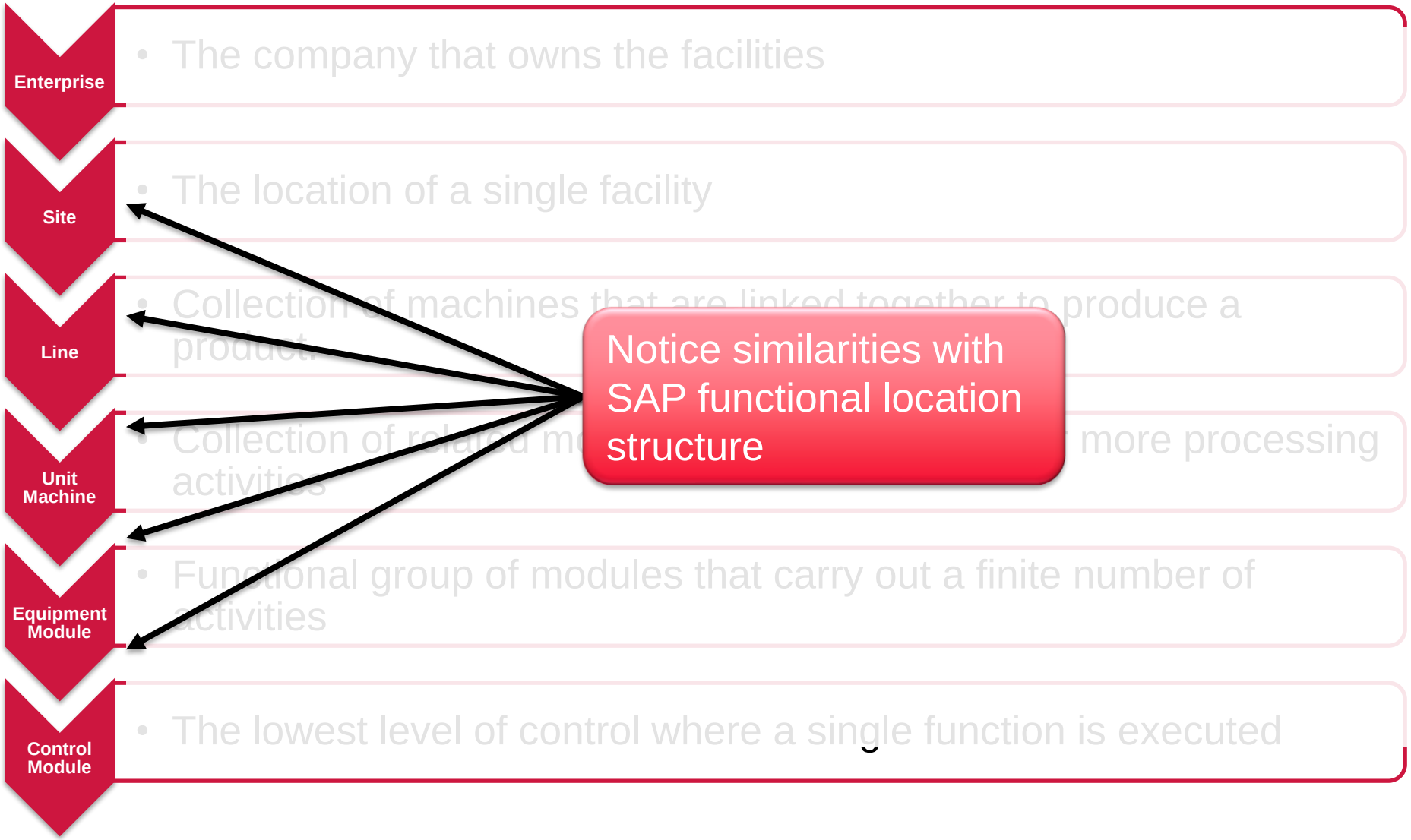
Machine and Unit States:
An Implementation Example of
ISA-88

Approved 1 August 2008

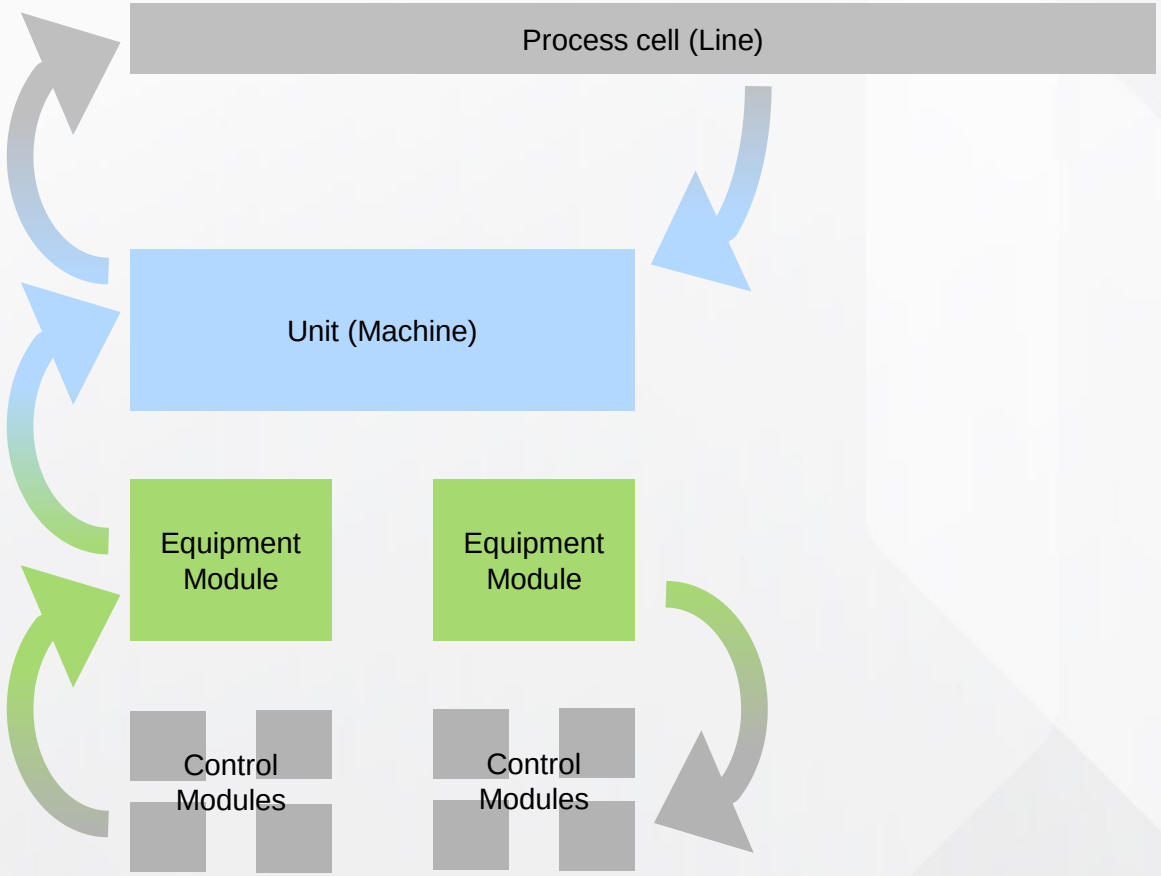
International Society of Automation
http://www.isa.org/Template.cfm?Section=Standards2&template=/Ecommerce/ProductDisplay.cfm&ProductID=9999

Rockwell Automation

Physical Hierarchy Descriptions



Structure of PackML



Unit

Unit (Machine)

Unit (or **machine**): A collection of related CMs and EMs that can carry out one or more processing activities. The **unit** corresponds to the logical grouping of mechanical and electrical assemblies that historically has been called a **machine**. It should be understood that in discrete applications, the term Machine is commonly used to describe a complete piece of equipment as provided by a machine builder, and could technically consist of more than one unit as defined by the standard. A **unit**, according to the ISA-88.01 definition, acts on one batch or lot at any given time on a process application; however, on a discrete machine there is a difference in that multiple batches, typically referred to a products or parts, can be acted upon at the same time.

Equipment Module

Equipment Module

Equipment Module (EM): A functional group of EMs, CMs, or both that can carry out a finite number of activities.

The primary purpose of control in an EM is to coordinate the functions of other EMs and lower-level CMs. An EM may be commanded by a process cell, unit, operator, or by another EM.

An EM can be part of a unit, or it can be a common resource within a process cell. A common resource can be either exclusive-use, where only one controlling entity can take ownership at a time, or a shared-use resource, where multiple controlling entities can share EM ownership. The EM's control interface will usually include a method to establish which higher-level entity (or entities in the case of share-use) has ownership of the EM at any given point, in order to prevent conflicting commands from multiple sources. The EM combines all the physical processing and control equipment required to perform the manufacturing process. The scope of the EM is defined by the finite tasks that it is designed to carry out.

Some examples of EMs are as follows:

- A valve matrix used for material transfer between units (shared resource of process cell)
- A level control for a tank (EM within a specific unit)
- A vertical form-fill-seal machine's "sealing jaws control" (EM within a discrete unit)

Control Module

Control Modules

Control module (**CM**): A regulating device, state-oriented device or a combination thereof (typically, a collection of sensors, actuators, and other CMs) that is operated as a single device. A CM typically controls or monitors one device state such as a motor running or stopped, or a valve open or closed, or one process variable such as a flow rate, level, pressure, count, or quantity.

Control at this level usually directly manipulates actuators and other CMs. A CM can direct commands to other CMs, or to actuators that have been configured as part of the CM. Equipment-specific manipulation of CMs and actuators control the process. The CM is the lowest level of equipment in the Physical Model that can carry out basic control.

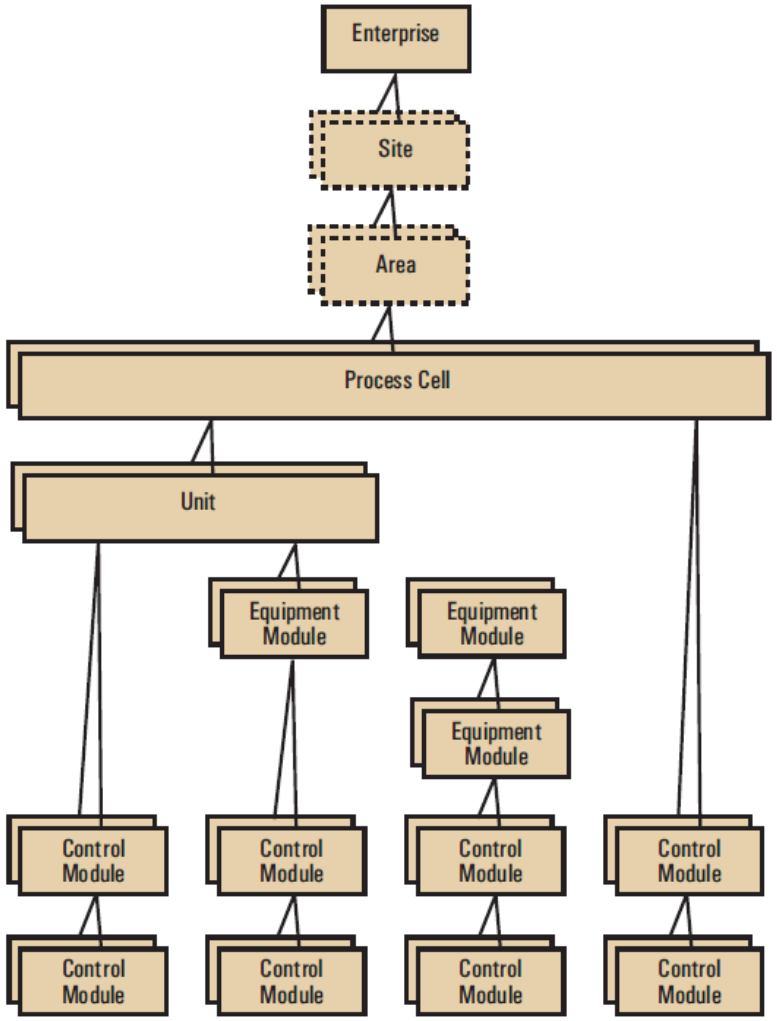
What follows is a list of devices or functions where CMs can be applied:

- An individual sensor or actuator
- A state-oriented device that consists of an on/off automatic block valve with position feedback switches and that is operated via the set point of the device
- A header that contains several on/off automatic block valves that coordinates the valves to direct flow to one or several destinations based upon the set point directed to the header CM
- A servo-controlled electronic gear or cam function (that is, a discrete unit), including its interlock and permissives.

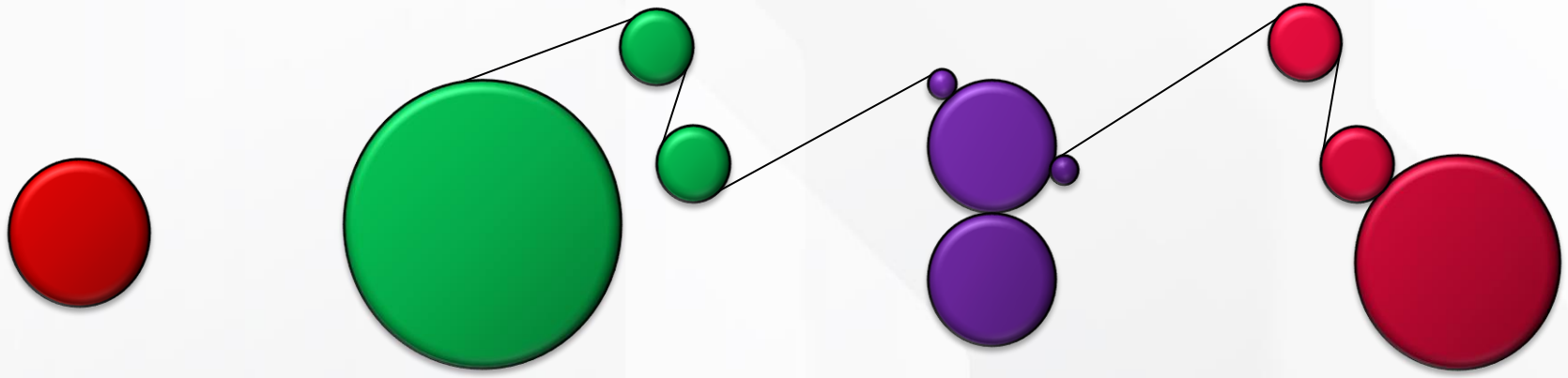
The following list contains some typical CMs that you might find in a programming library:

- Analog output
- Analog input with scaling and alarms

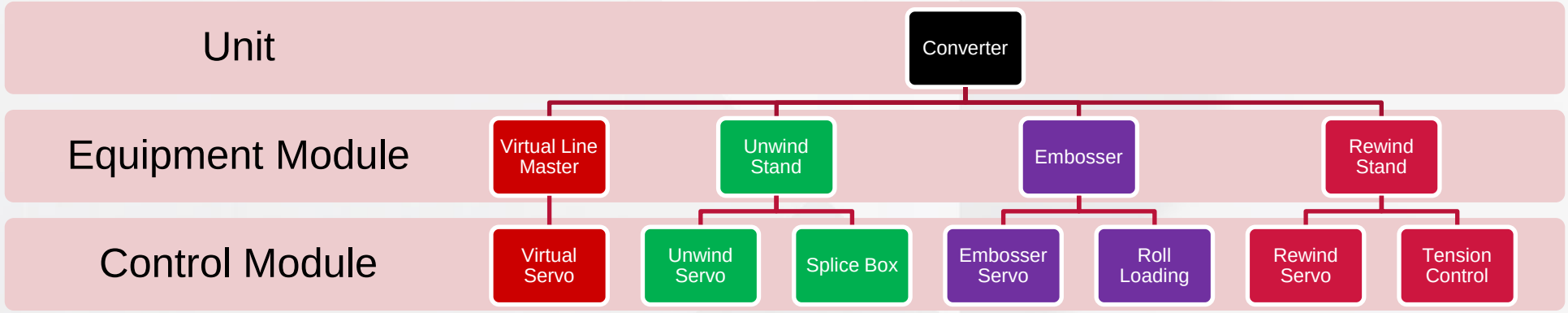
Example of a complete factory



Equipment Decomposition Example



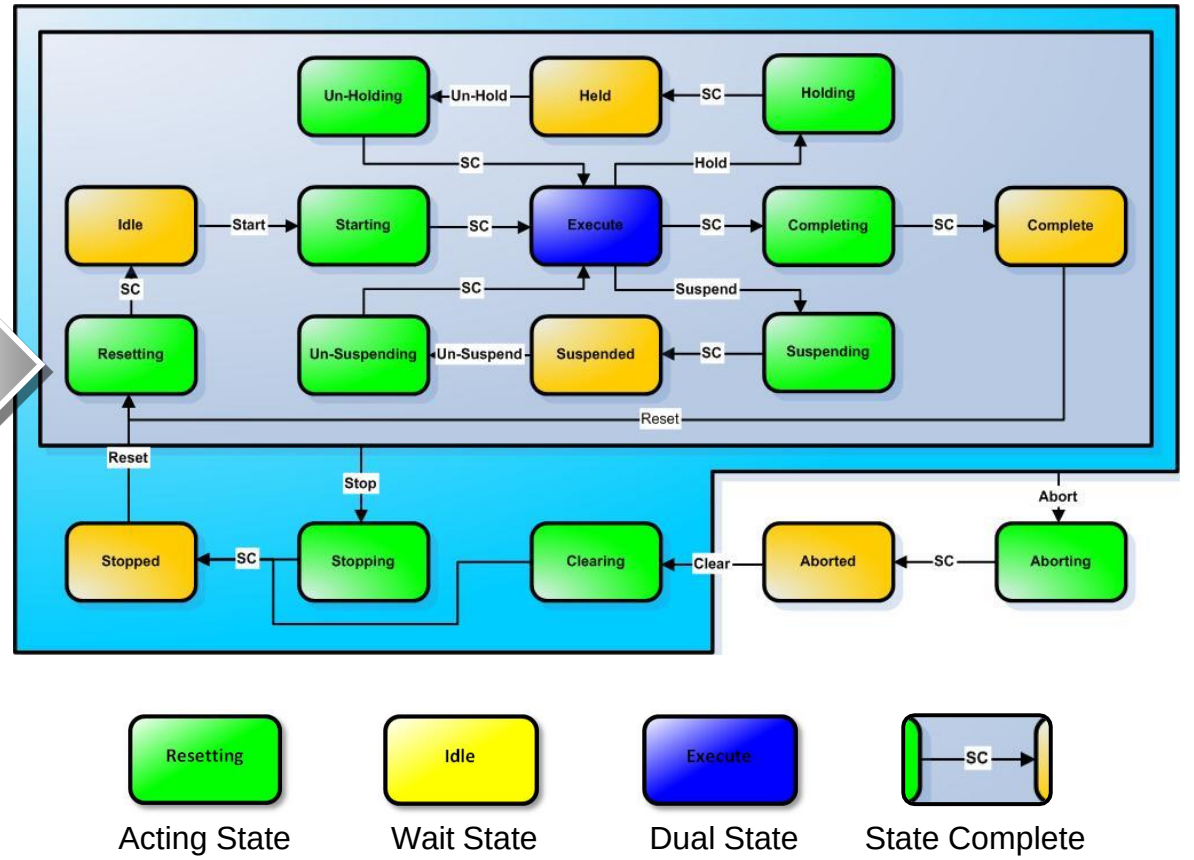
Virtual Line Master | Unwind Stand | Embosser | Rewind Stand



PackML Structure

Base State Model

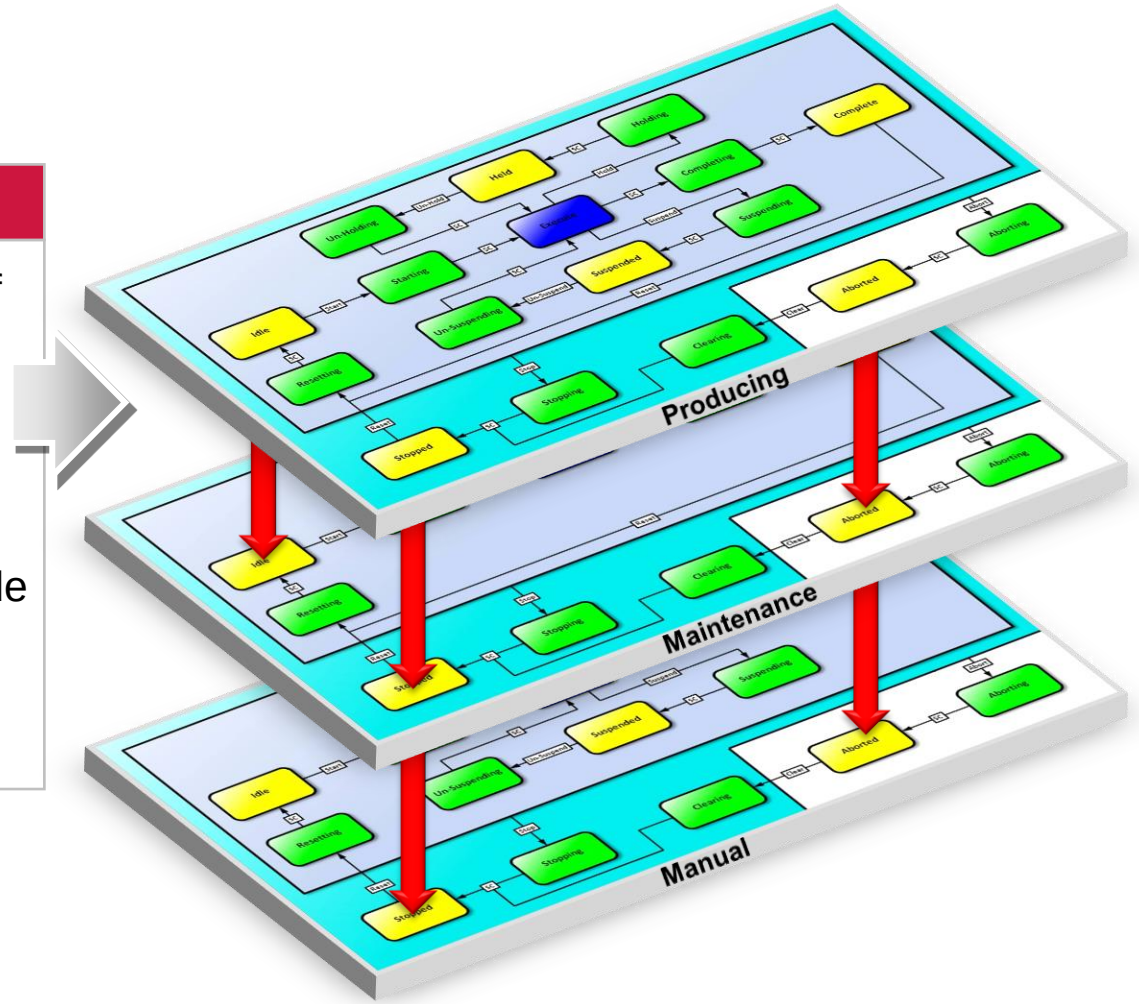
- Completely defines behavior of machine for a specific mode
- 17 predefined states with state transitions and state commands
- All modes defined by subset of base state model
- See end of presentation for formal definition of each state



PackML Structure

Mode Manager

- Manages current mode of machine
- Allows mode transitions only when in correct state
- Different subset of states may be used in each mode
- Procedure within each state can be unique for mode state resides in



Rockwell Automation

Framework B

Rockwell Automation

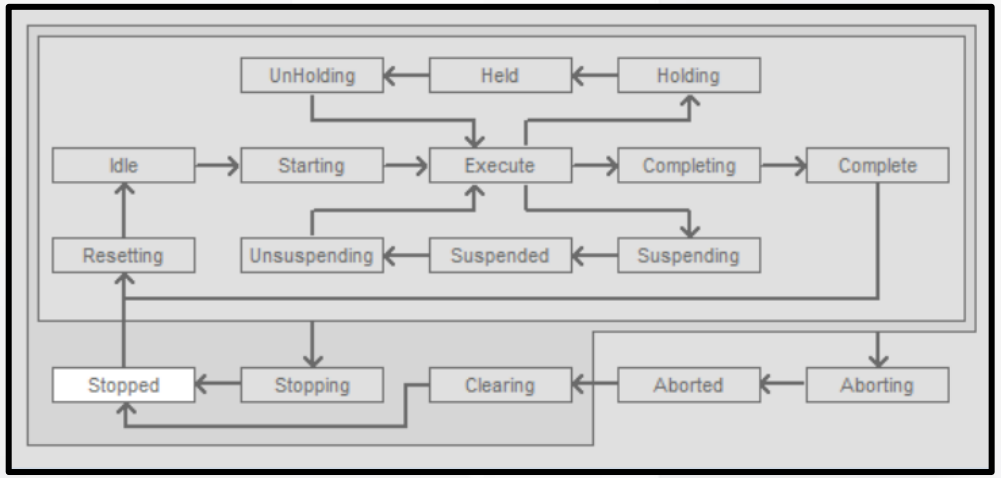
What is Framework B ?

- Equipment Framework ‘B’ provides machine code framework based on international standards where the Physical Model (equipment hierarchy) is derived from ISA-88 and the Procedural Model follows ISATR88.00.02-2015 Mode and State models.

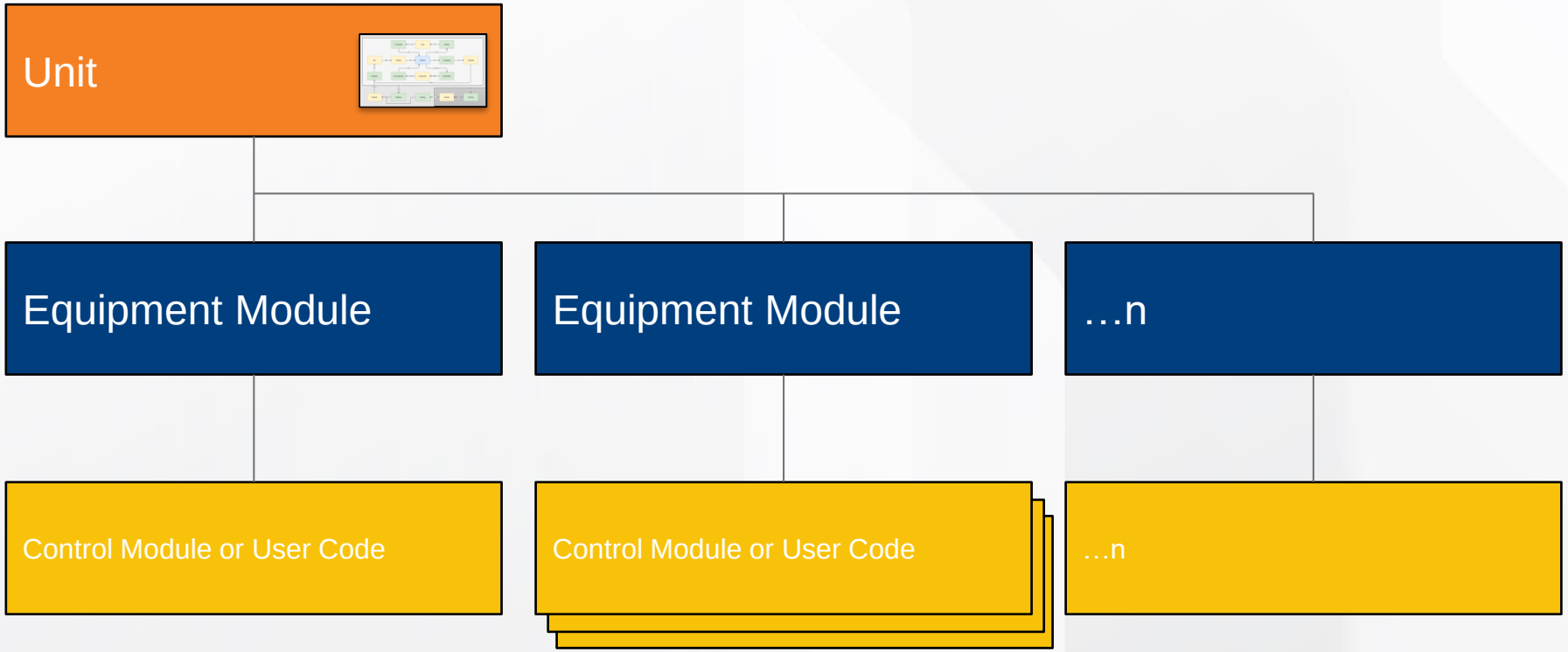
- Framework B include 3 modules
  - Unit
  - Equipment Supervisor
  - Equipment Module

What is PackML Framework ?

- Framework for handling states on the machine
- ISA TR88.00.02 PackML - 17 state model is used
    - Standard defined machine states and operational flow



Framework B – Physical Model



A Unit can control up to 31 Equipment Modules

Framework B - Unit

The Unit Module provides the following features/functionalities:

- Mode and state handling.
- Handling HMI commands and status for the Mode and State
- Controls physical inputs that drive machine state
- Blocked or Starved monitoring
- Fault monitoring

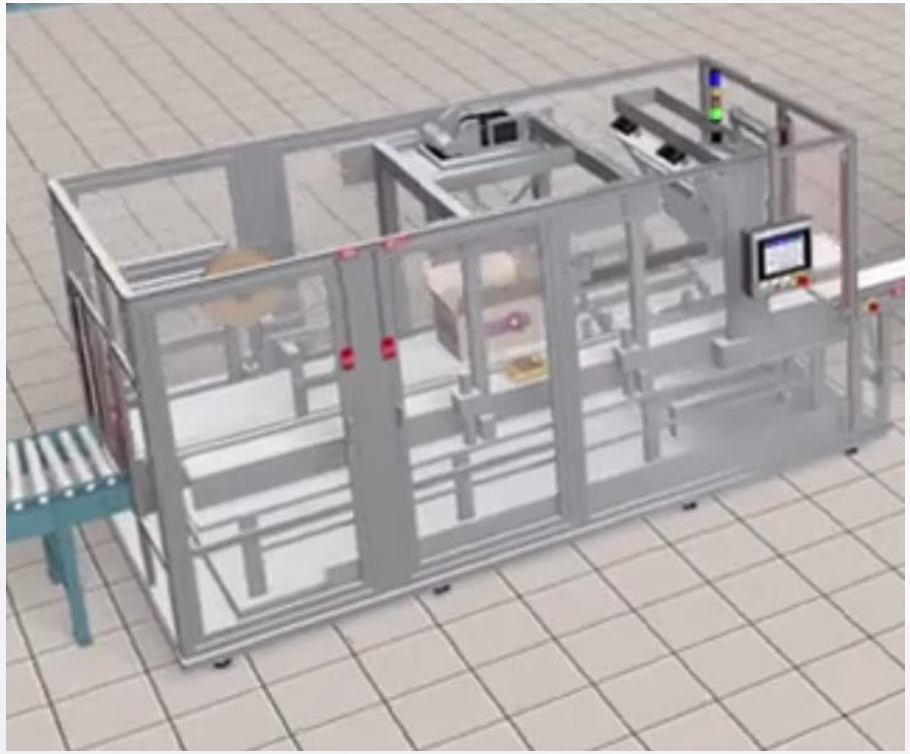
Framework B - Equipment Module

Equipment Module

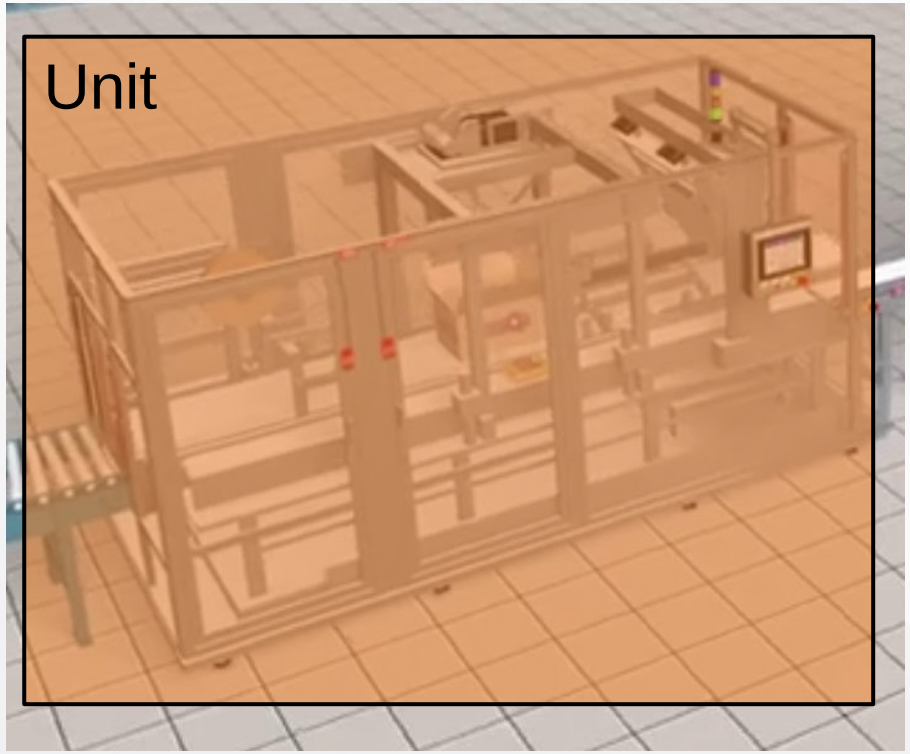
The Equipment Module provides the following features/functionalities:

- Basic Equipment Phase routines for sequencing control code execution
  - Fault clearing – Clearing
  - Initialization – Resetting
  - Runtime – Starting, Execute, Unholding and Unsuspending
  - Termination (end runtime) – Aborting, Stopping, Holding, Suspending and Competing
- Simulation
  - Fault generation
  - Delayed state complete response
- Reporting Fault, availability and state complete to unit

Example - Packing Machine

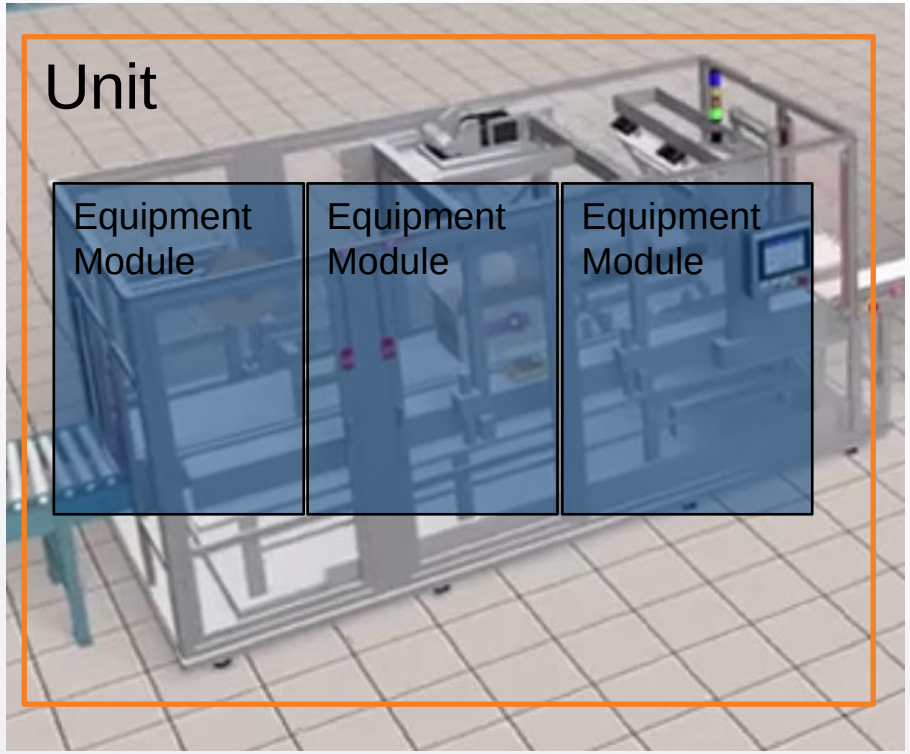


Example - Packing Machine - Unit



Unit 

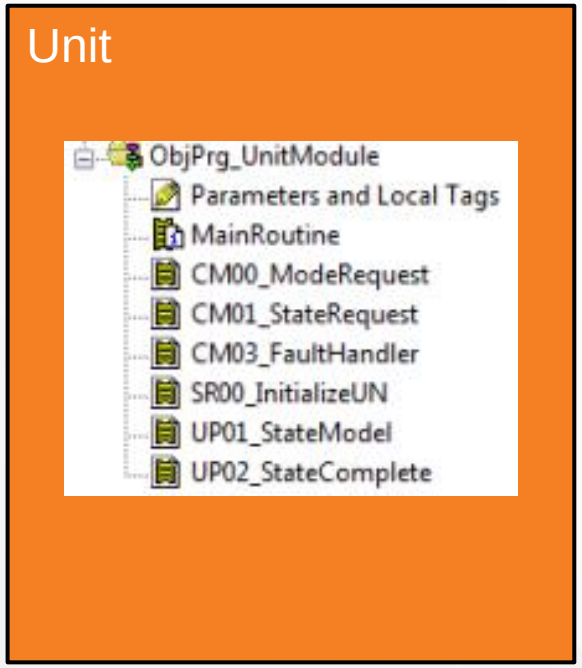
Machine #1 Equipment Module

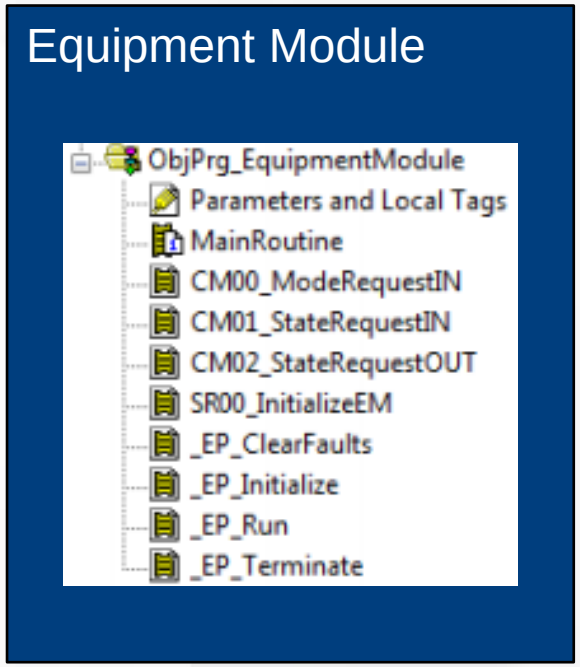


Unit

Equipment Module

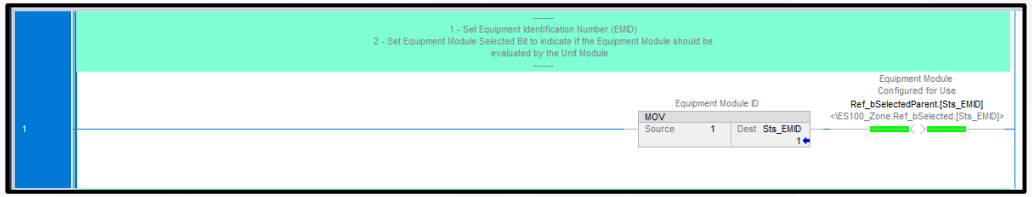
Framework B – Program Structure

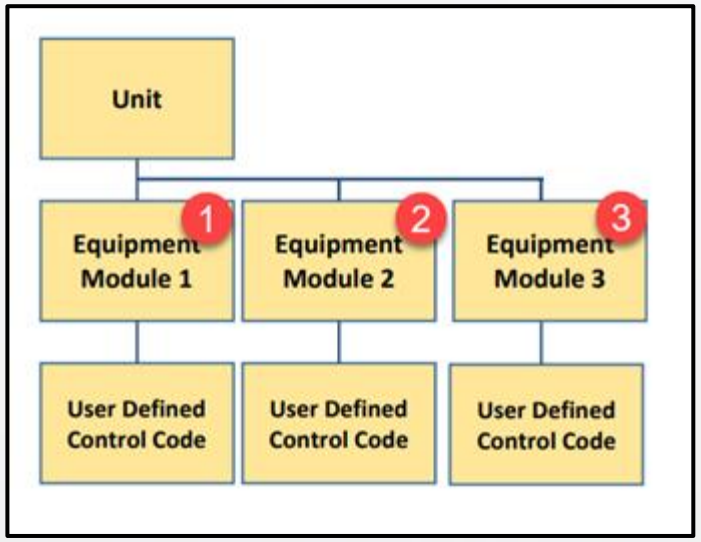




Eqiument Module – Sts_EMID

- The ID for the Equipment (child) is related to the parrent and must be uniq numbered.

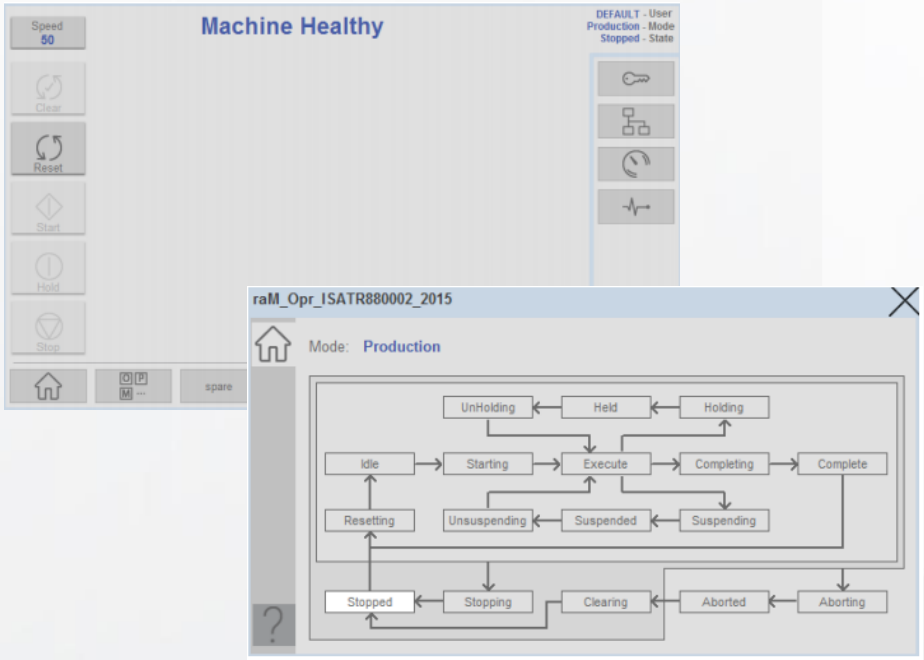


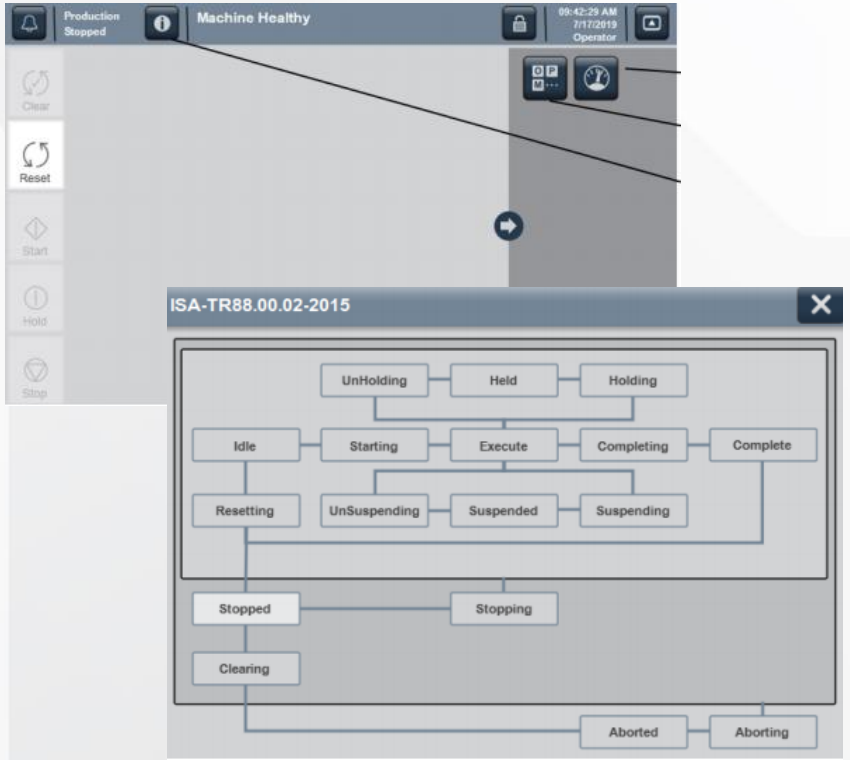


- Unit can Control up to 31 equipment modules

Framework B -

Basic visualization is provided for FactoryTalk View Studio and Studio5000 View Designer



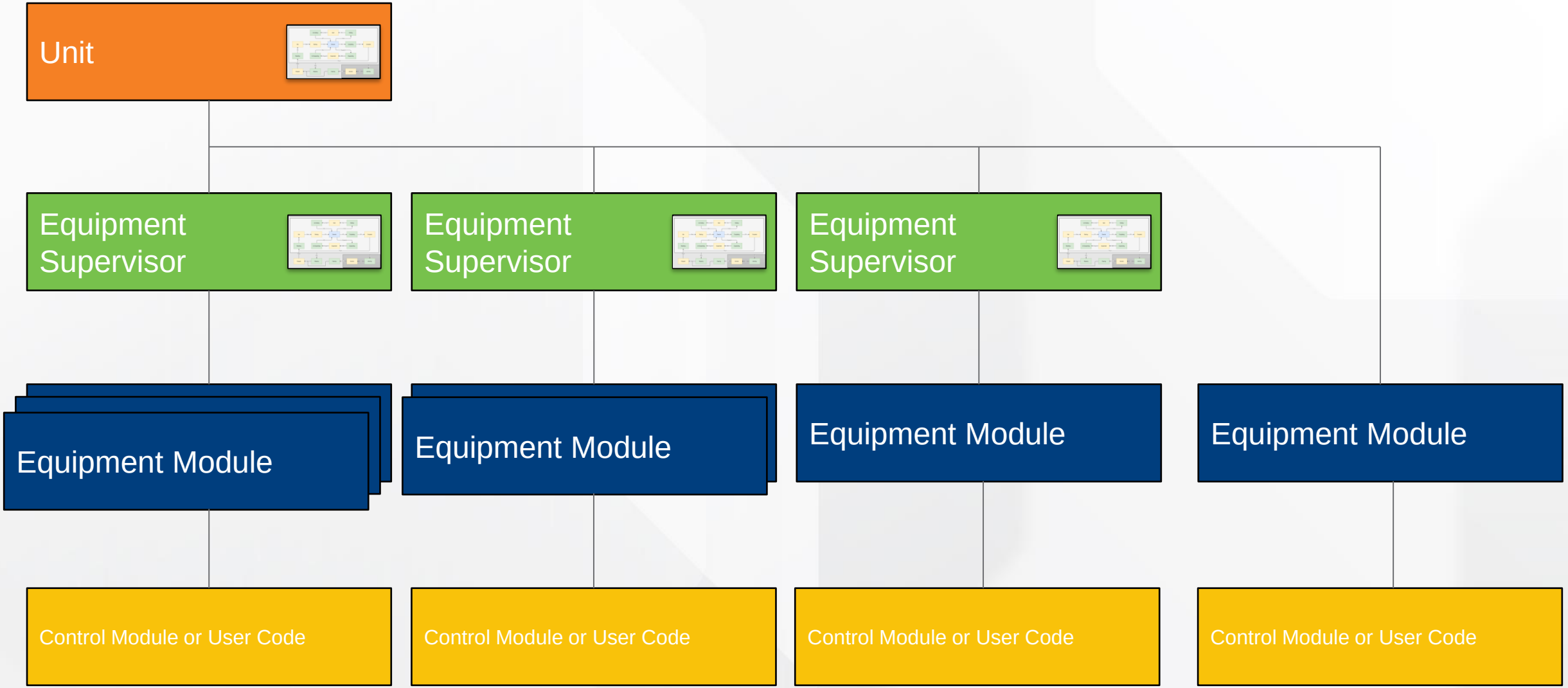


Rockwell Automation

Multi Zone Support

Rockwell Automation

Topology - Framework B V5.0



Equipment Supervisor

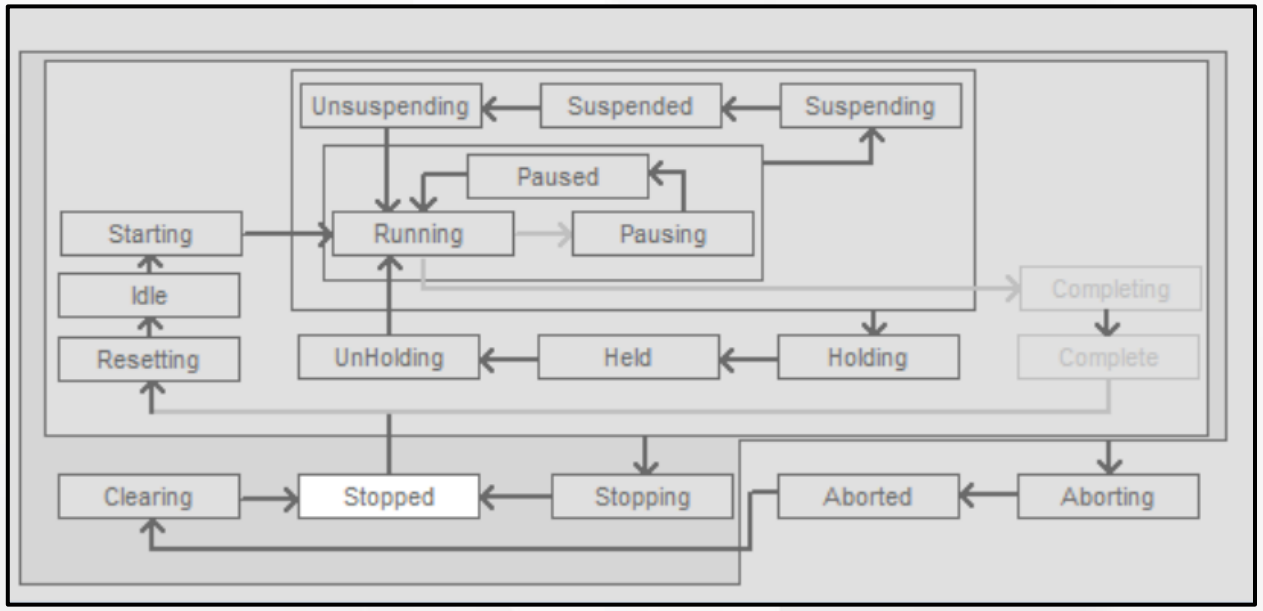
- The Equipment Supervisor Module represents a machine or a part of a machine which can control **subordinate equipment modules**. The Supervisor can operate in either an independent manner section (using its own state model) or in a synchronized, linked manner with a Unit Module.

- The point off using this is that if one Equipment Module under an Equipment Supervisor goes into Abort, Stop, Hold or Suspend the hole Unit does not follow, and can still be operative with other Equipment modules.

- An Equipment Supervisor can also be used if a Unit (Machine) has several Safety Zones. If one is activated, the unit **continues operation** on other Safety Zones

Equipment Supervisor – State model

- ISA880001AnnexD_2010
  - 19 states

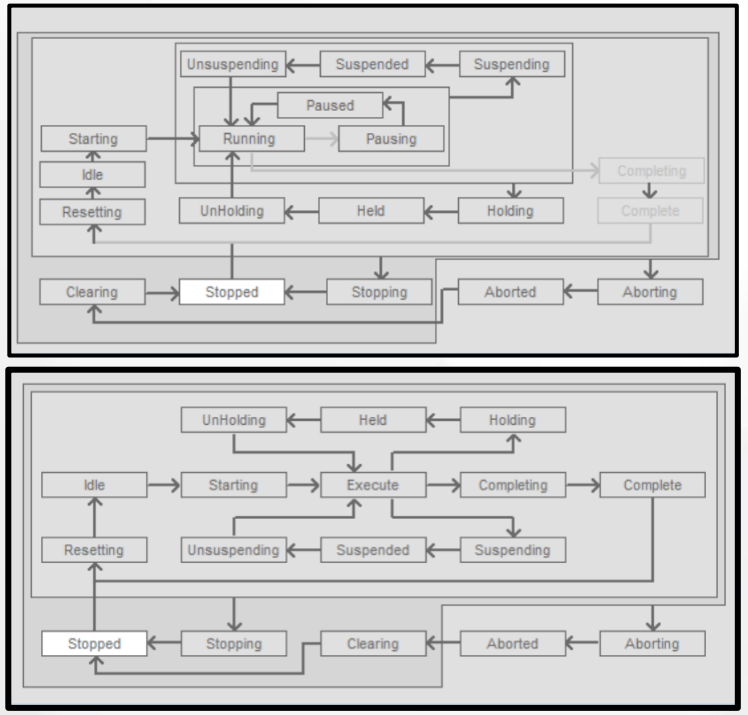


Equipment Supervisor – State model

- ISA880001AnnexD_2010
  - 19 states

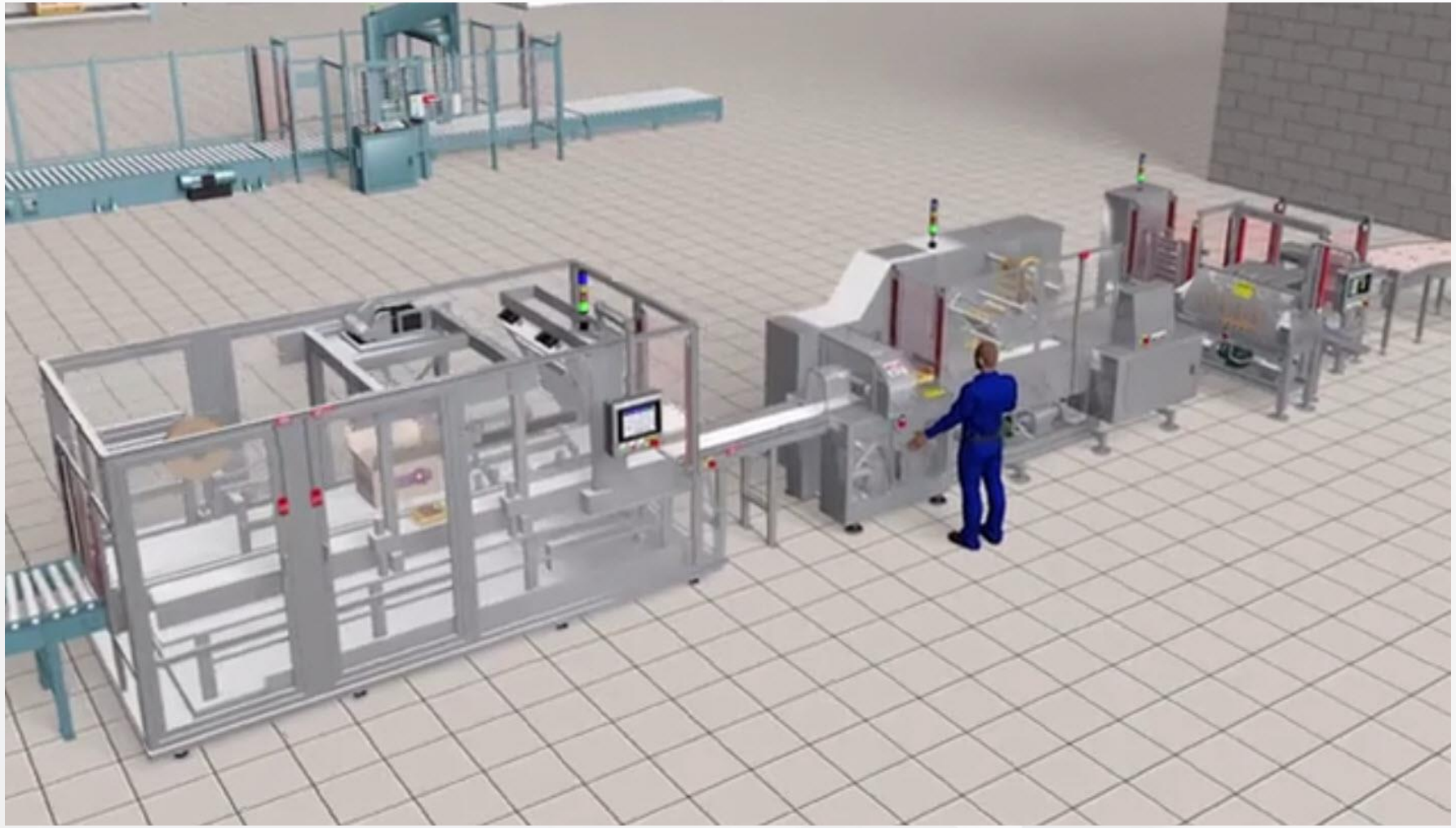
- Remote/Unit Control
  - Synchronized with the unit
  - Only 17 states is used

- Local Control
  - Controlled asynchronous to the Unit
  - Optioned to shift mode
  - Optioned to use all 19 states

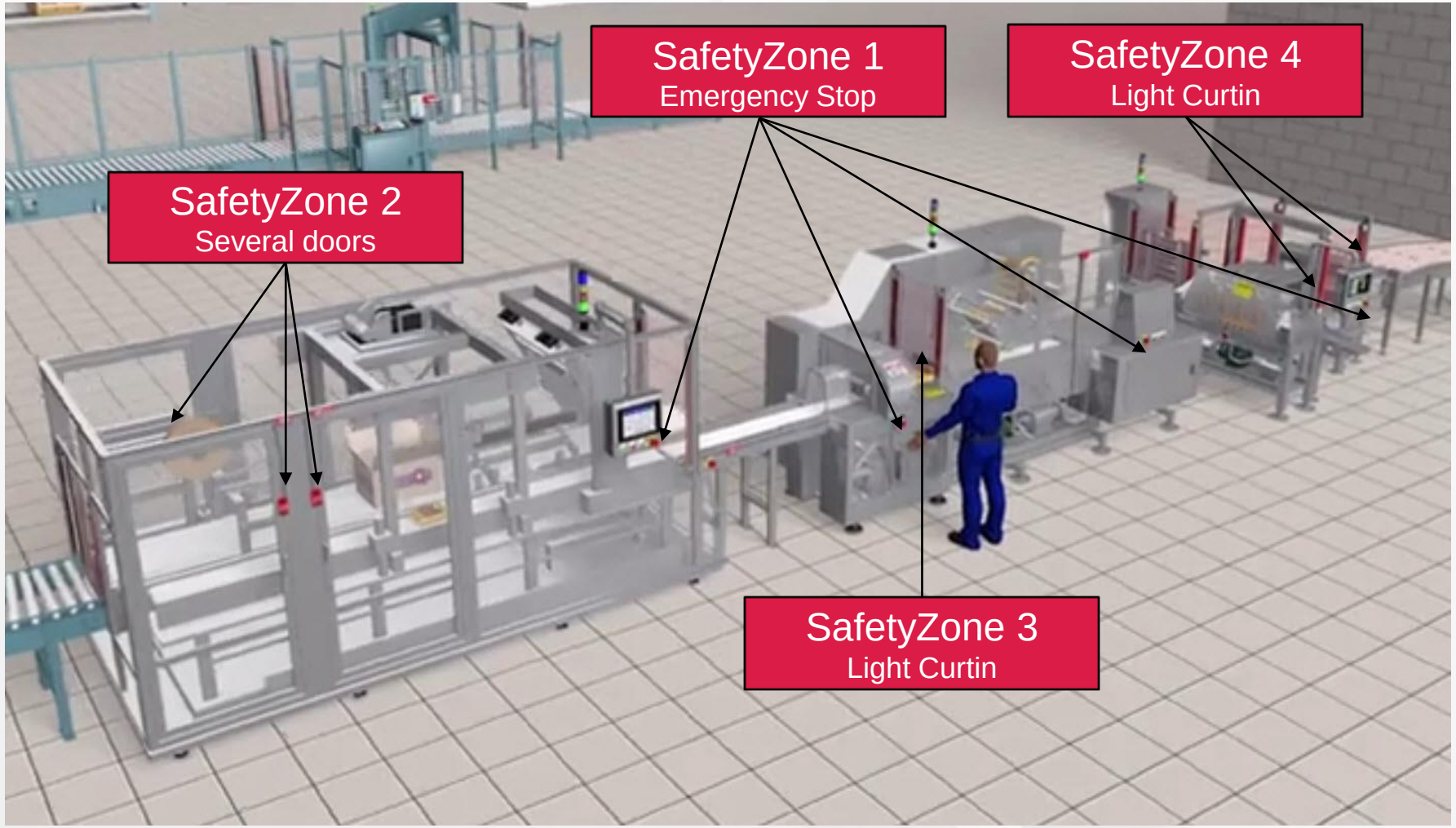


- The Equipment Supervisor will resync to follow the Unit when transit from Local to Remote control

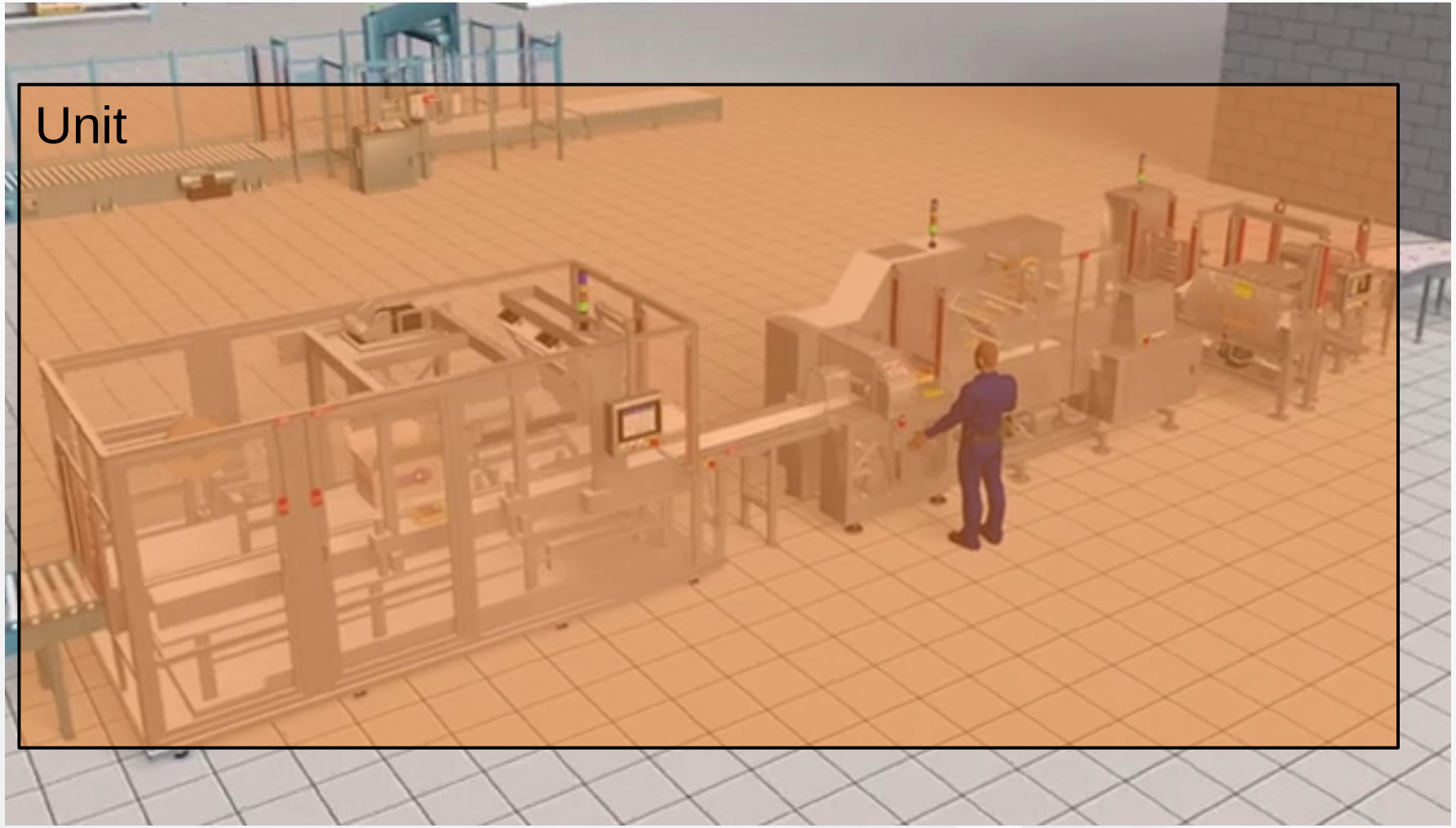
Machine #1 Safety zone example



Machine #1 overview

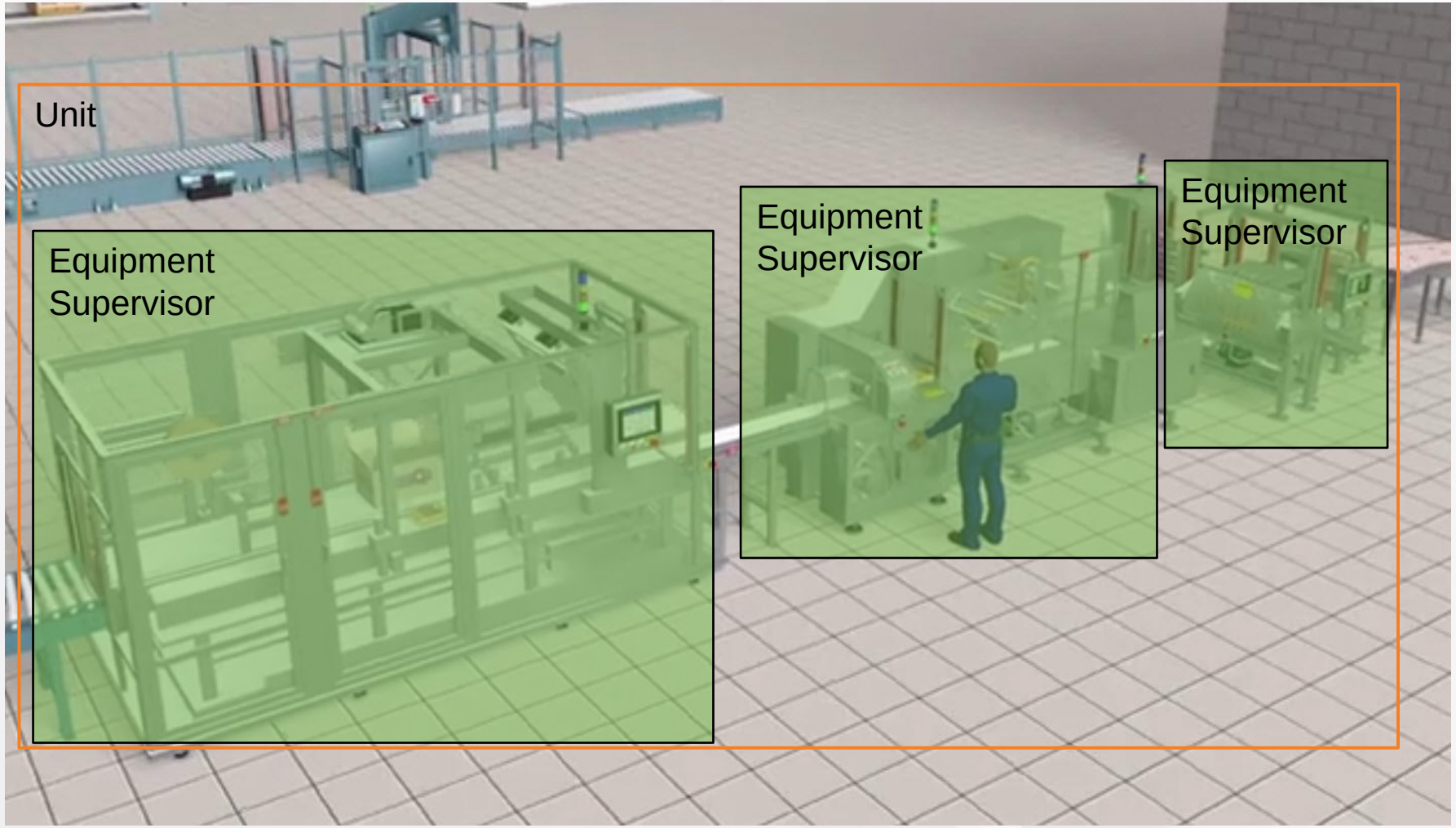


Machine #1 Unit



Unit 

Machine #1 Equipment Supervisor

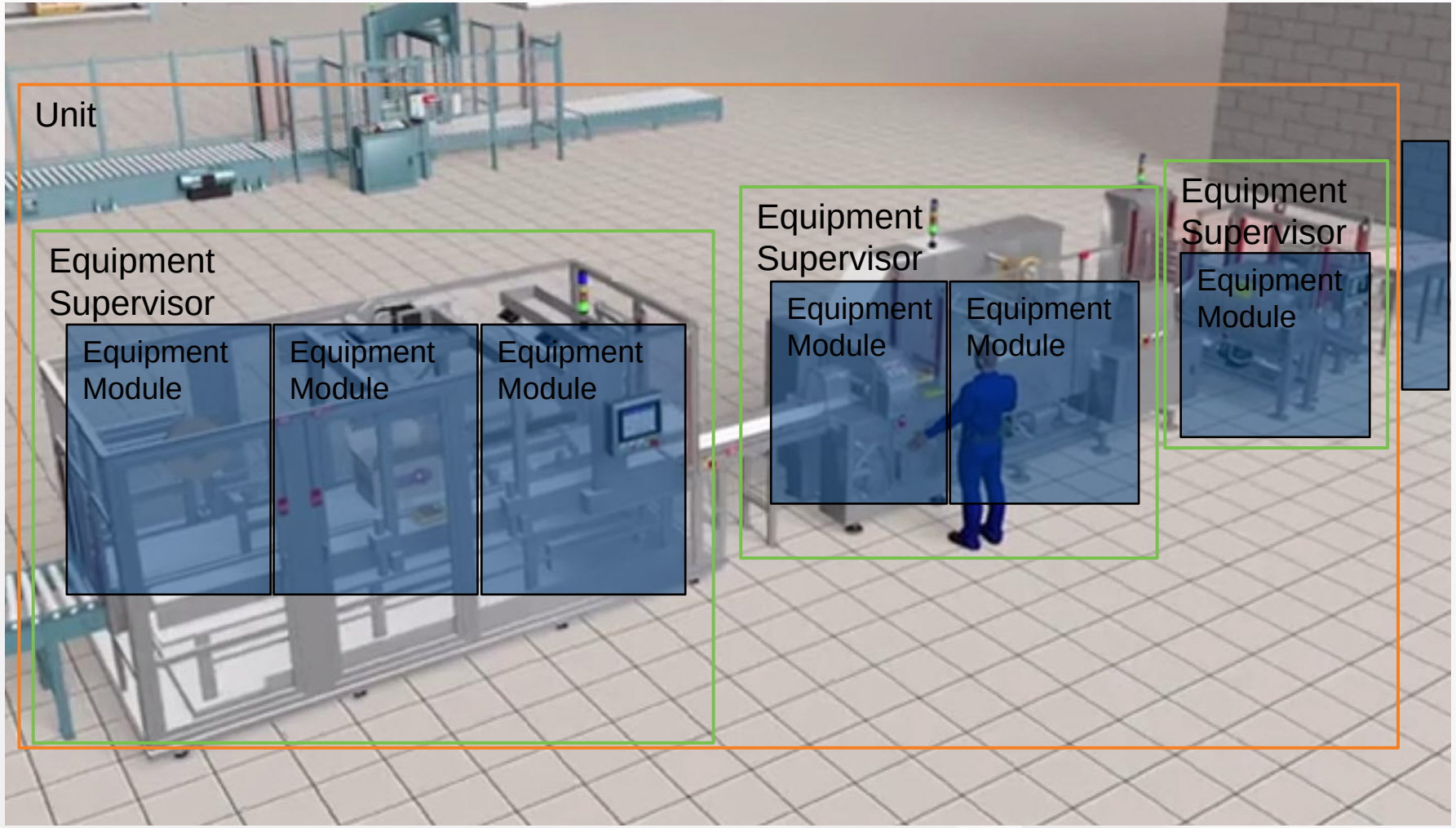


Unit

Equipment Supervisor

Rockwell Automation

Machine #1 Equipment Module



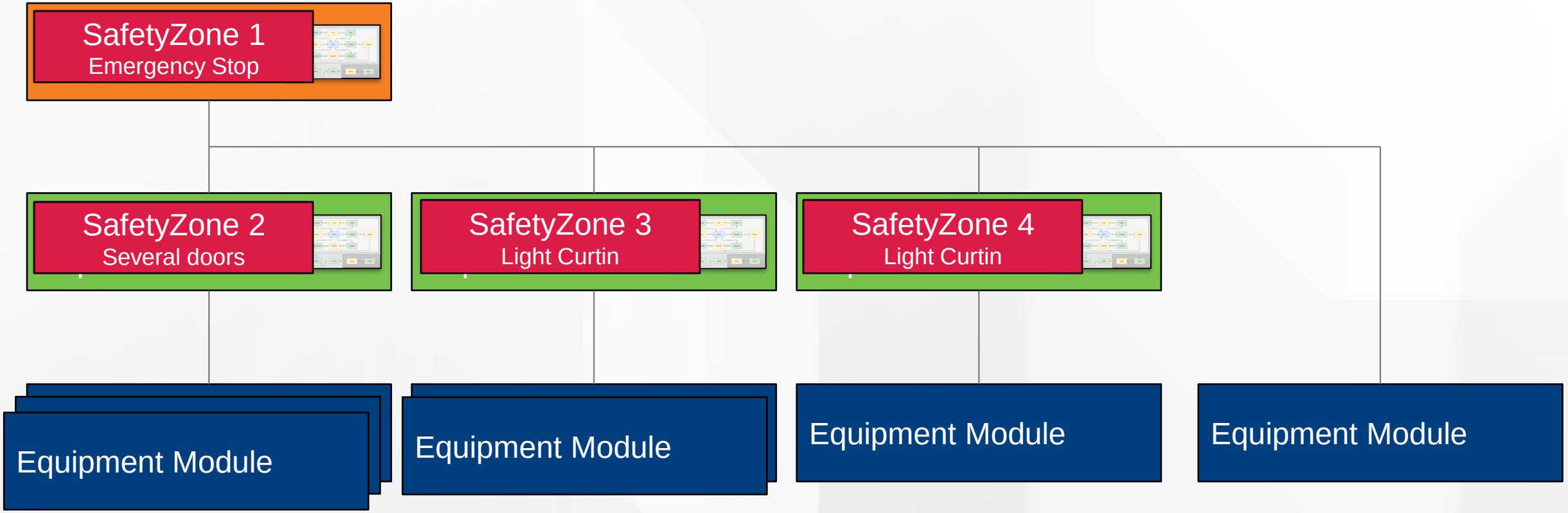
Unit

Equipment Supervisor

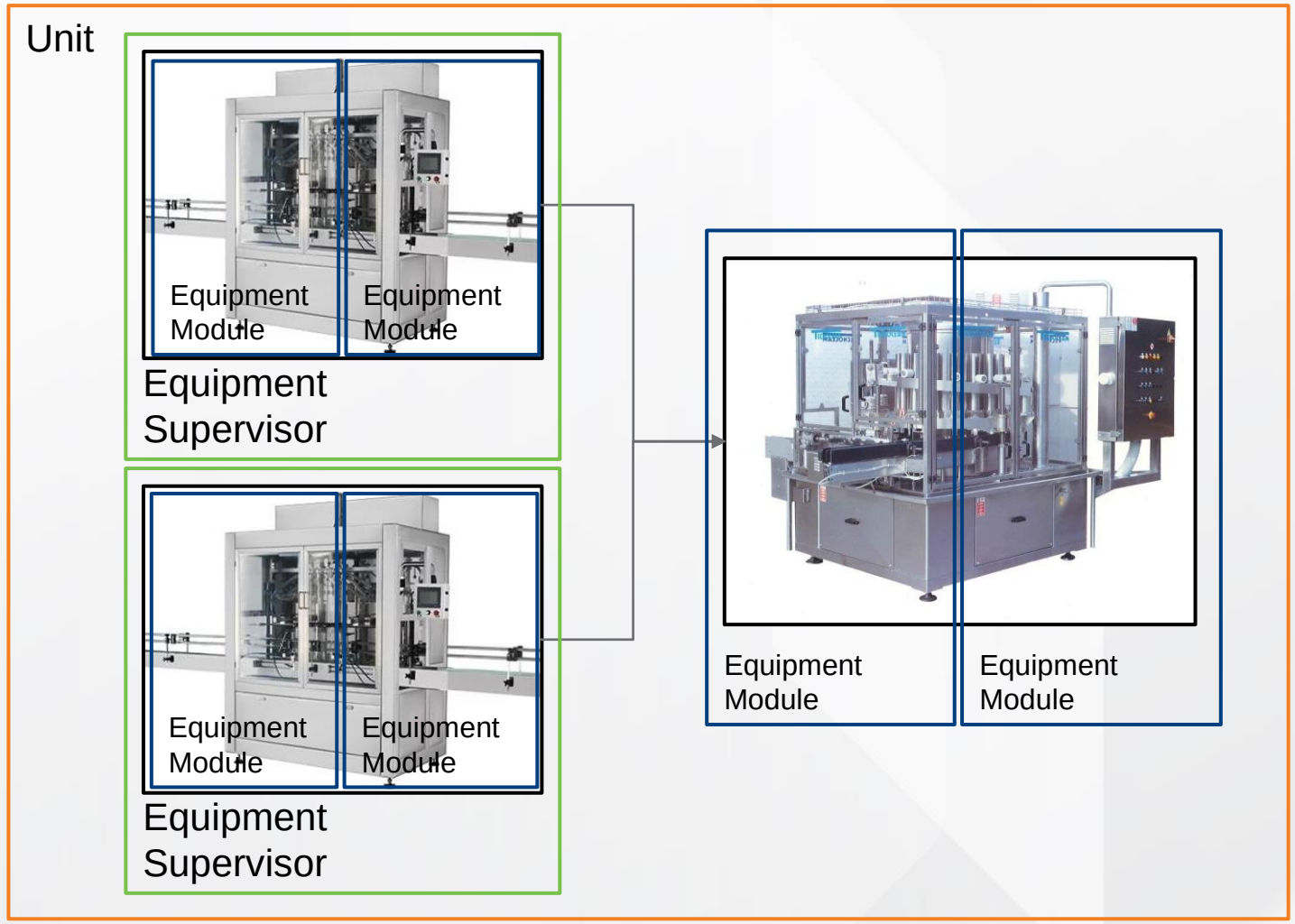
Equipment Module

Rockwell Automation

Machine #1 Topologi



Machine #2 Parallel Production



Unit

Equipment Supervisor

Equipment Module

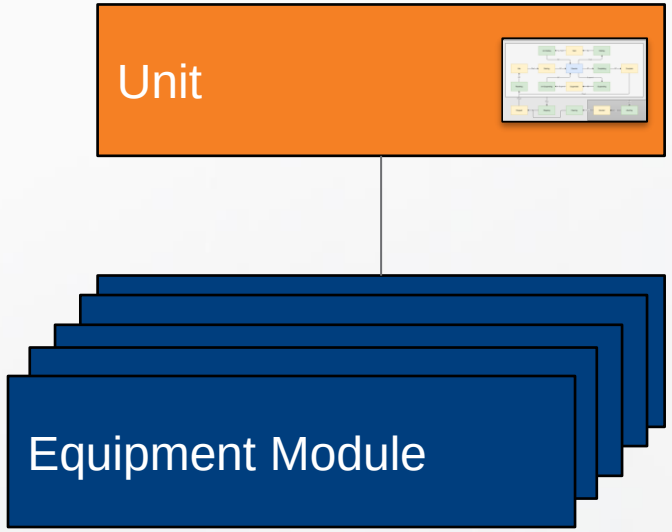
Benefits

Set one Equipment Supervisor in local and the other still runs production. Could be used to purpose of maintenance etc.

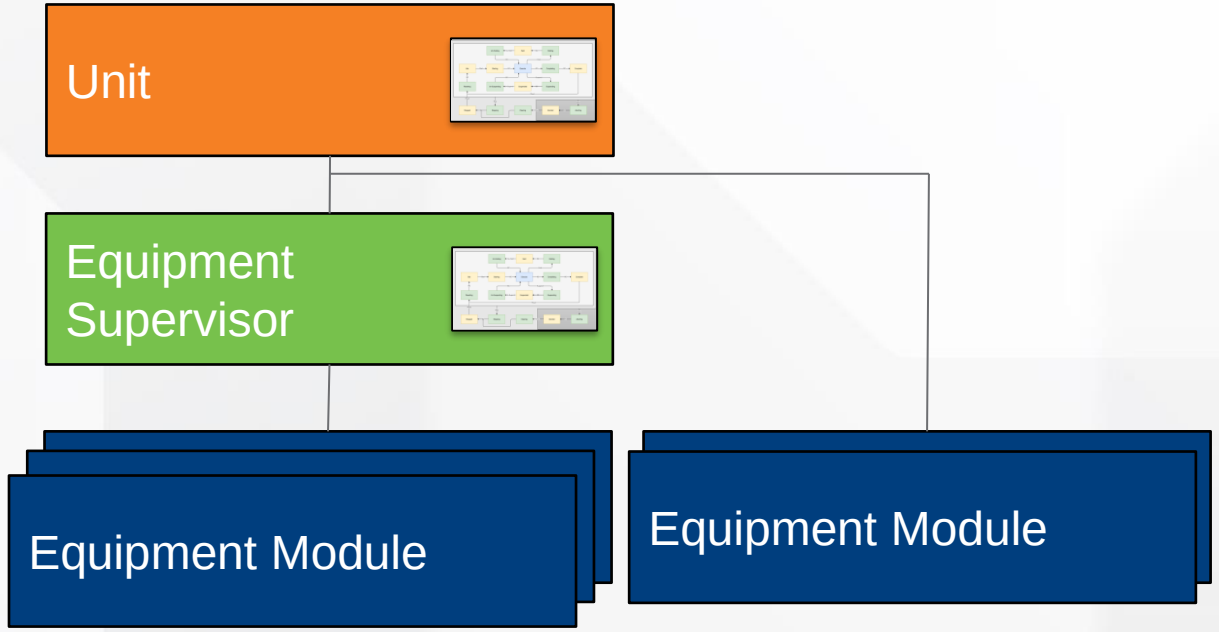
Rockwell Automation

Easy to reuse or move Eqiument Module beetween Unit and ES

- Structure on the Unit and Equipment supervisor is equal.



Name	Usage	Alias For
Ref_bStateCompleteParent	Local	\UN01_Unit.Ref_bStateComplete
Ref_bSelectedParent	Local	\UN01_Unit.Ref_bSelected
Ref_bFaultedParent	Local	\UN01_Unit.Ref_bFaulted
Inp_bState	Local	\UN01_Unit.Out_bState
Inp_bMode	Local	\UN01_Unit.Out_bMode

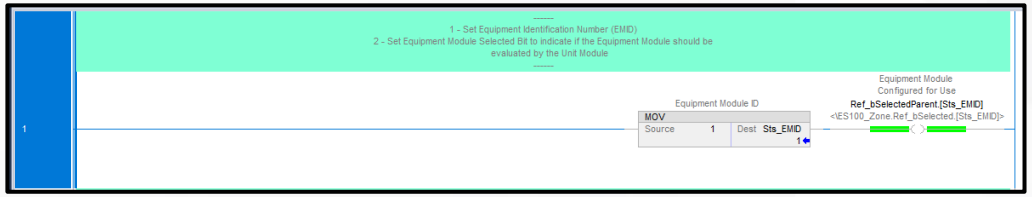


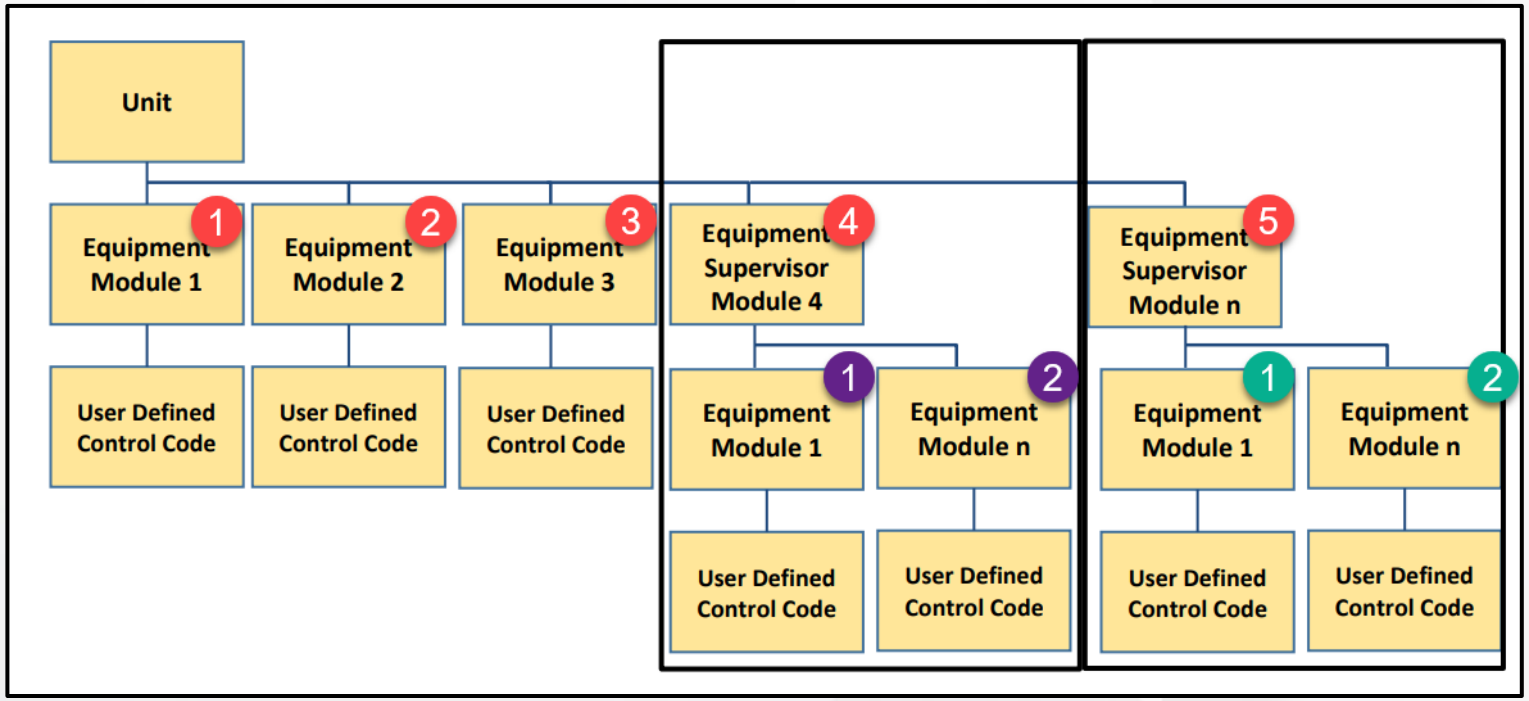
Name	Usage	Alias For
Ref_bStateCompleteParent	Local	\ES100_Zone.Ref_bStateComplete
Ref_bSelectedParent	Local	\ES100_Zone.Ref_bSelected
Ref_bFaultedParent	Local	\ES100_Zone.Ref_bFaulted
Inp_bState	Local	\ES100_Zone.Out_bState
Inp_bMode	Local	\ES100_Zone.Out_bMode

- Be sure that the Sts_EMID for Eqiument Module is uniq for the new parrent

Eqiument Module – Sts_EMID

- The ID for the Module (child) is related to the parrent and must be uniq numbered.





- Unit can be used with up to 31 equipment modules or/and equipment supervisors
- Equipment supervisor can control up to 31 equipment modules
- It is possible to have 31^{31} Equipment modules under a Unit

Rockwell Automation

MBL - Demo

Rockwell Automation

Demo

Framework B project with Motion Object

1. Create an ACM project
2. Create a Controller and HMI
3. Create Unit, Equipment Supervisor, Equipment Module
4. Create a Control Module with Virtual Axis
5. Generate ACD
6. Create HMI
7. Show Application

Rockwell Automation

Create your own library - Demo

Rockwell Automation

Demo

Create motor starter Library

1. Create a simple motor starter in logix designer
2. Make a library in Design library
3. Use the library in Application code manager
4. Generate ACD

Rockwell Automation

E3D - Demo

Rockwell Automation

Demo

Emulate 3D

1. Create project in E3D
2. Make a simulation in E3D
3. Make the PLC code from the E3D simulaion
4. Generate the Code in Application Code Manager