



**Implementing an
OS/COMET Model
in a
Model-View-Controller (MVC)
Architectural Framework**





MVC Architecture Components

⊕ Model

- ✧ Provides a State-Based Information “Model” of the Application Domain (Ground Station, Satellite, Payload)

⊕ View

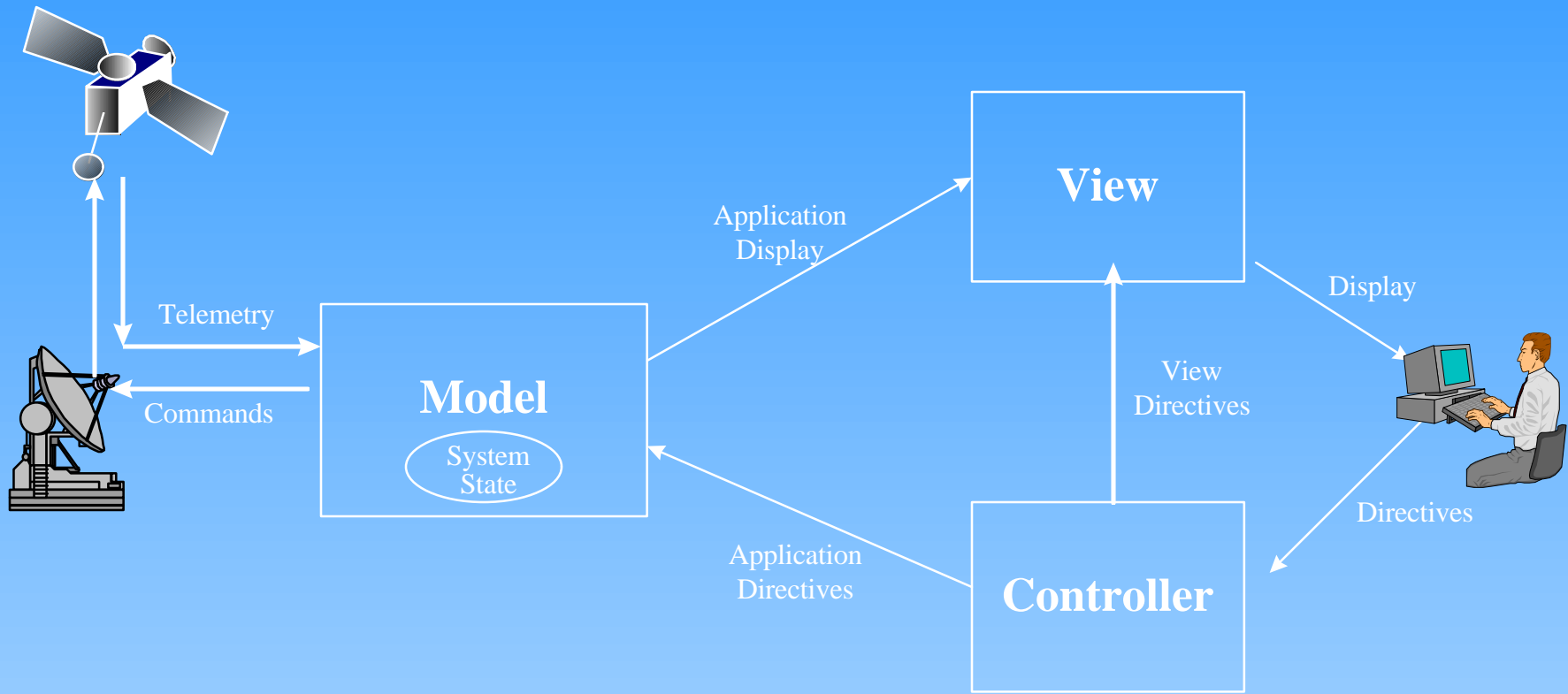
- ✧ Provides an Operator “View” into the Application Model.

⊕ Controller

- ✧ Provides Operator “Control” over the Application Model and Operator View.

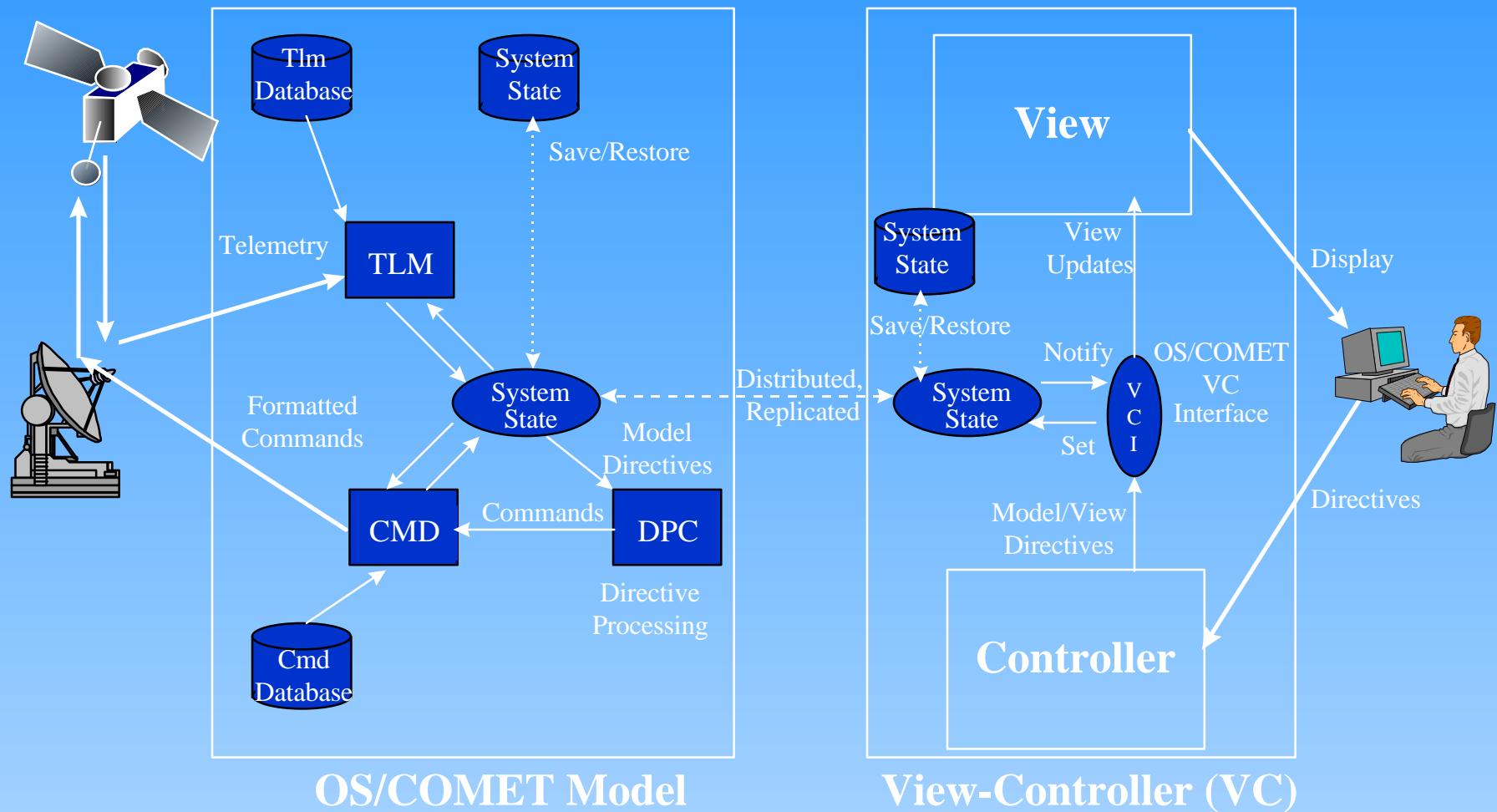


Ground Station MVC Architecture





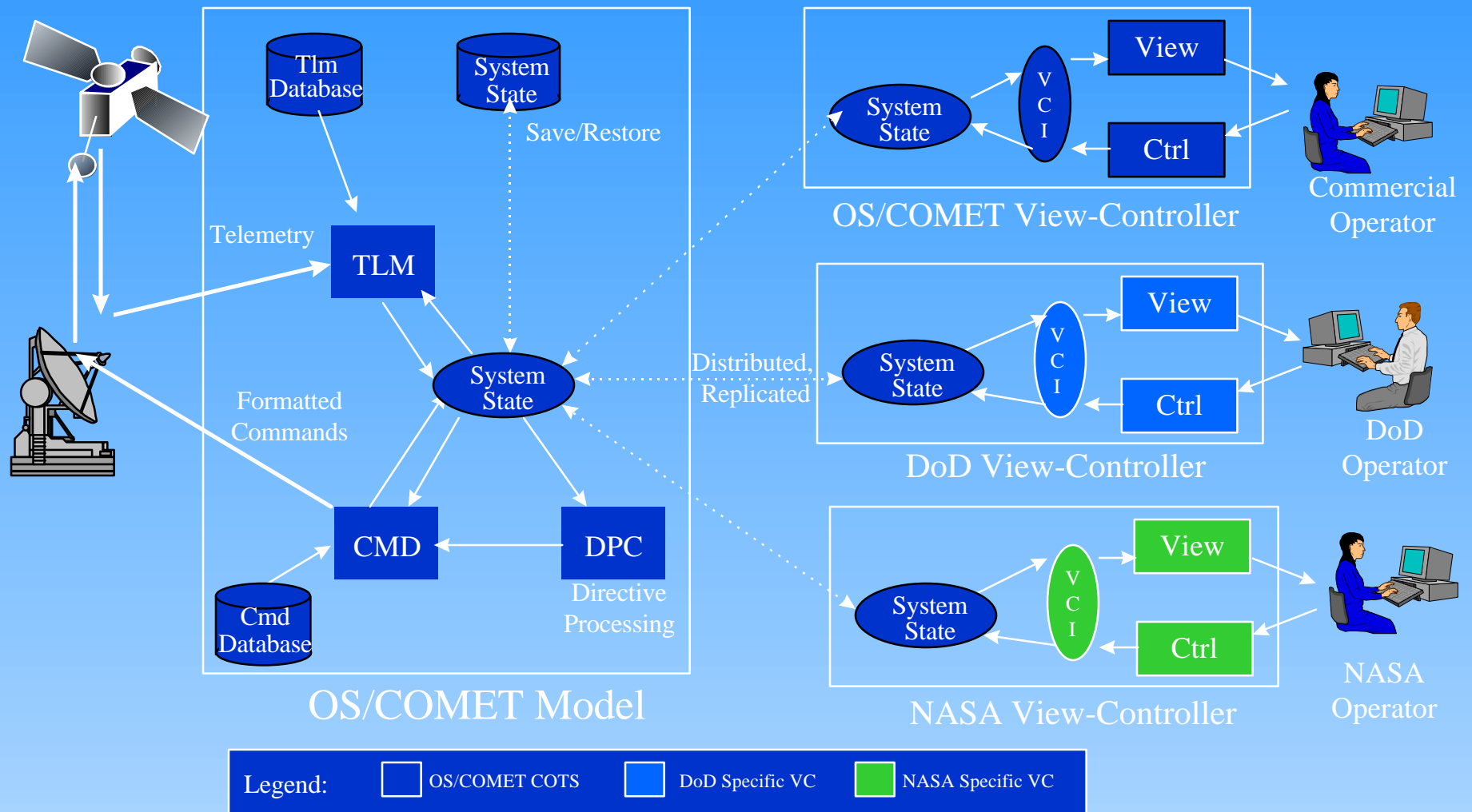
An OS/COMET Based Groundstation MVC Architecture





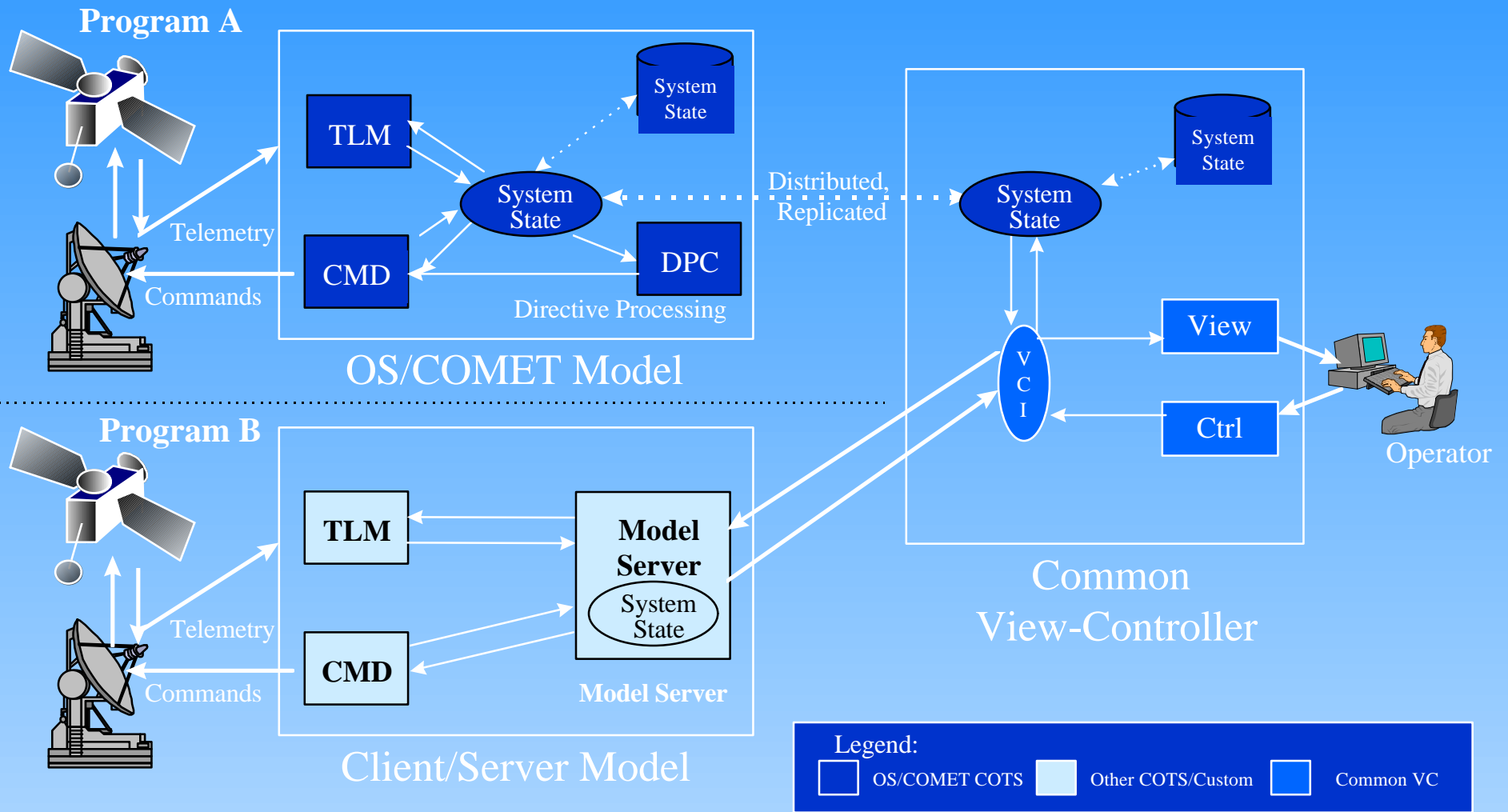
OS/COMET Model

Support of Multiple Operator Interfaces





Common Operator Interface Integrated with Multiple COTS Models





Common Operator Interface Recommendations

- ✦ A Common Operator Interface Approach Should :
 - ✦ Not Look to Industry to Develop a COTS User Interface Based on Specific Windows, Screen Positions, Colors, etc. Issues are:
 - ★ COTS Configuration Control (Shared Industry & Government Control)
 - ★ Competing Space Operations Interface Standards (AF, NASA, DoD, AIAA)
 - ★ Limited Resell Potential to other Commercial/Government Customers.
 - ★ Market-Driven. Not a Cost Effective Solution.
 - ✦ Implement View-Controller (VC) Software to Satisfy Specific Requirements.
 - ✦ Leverage from Commercially Available VC Software.
 - ✦ Publish and Configuration Control VC Interfaces.
 - ✦ Look to Industry to Support Published VC Interfaces.
 - ✦ Make Any Developed VC Software Available to Compete in the Commercial Market Place.