

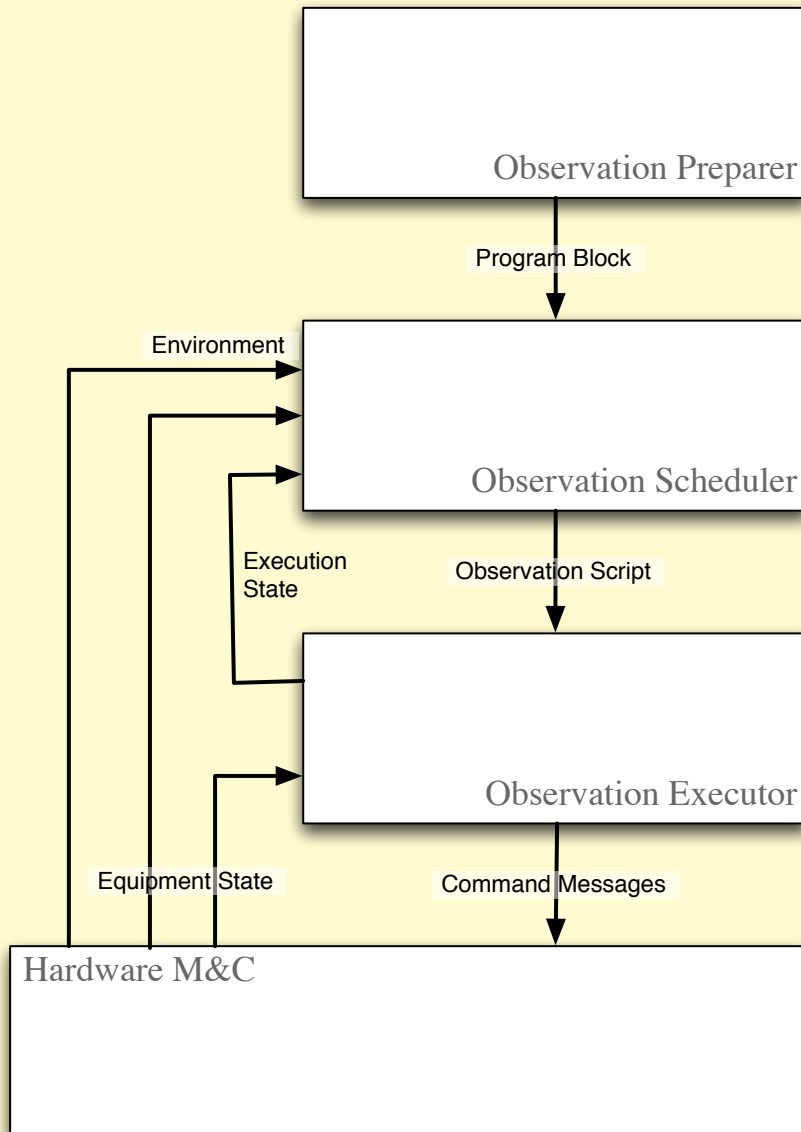
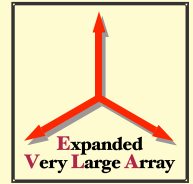
# EVLA Computing Design

---

*Subsystems I*



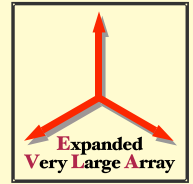
# Scope (Sub-Systems)



- Observation Preparation
- Scheduler
- Executor



# Scope (Concepts)

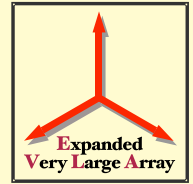


- EVLA Project Model
- Dynamic Scheduling
- Sub-Arrays
- Command Translation

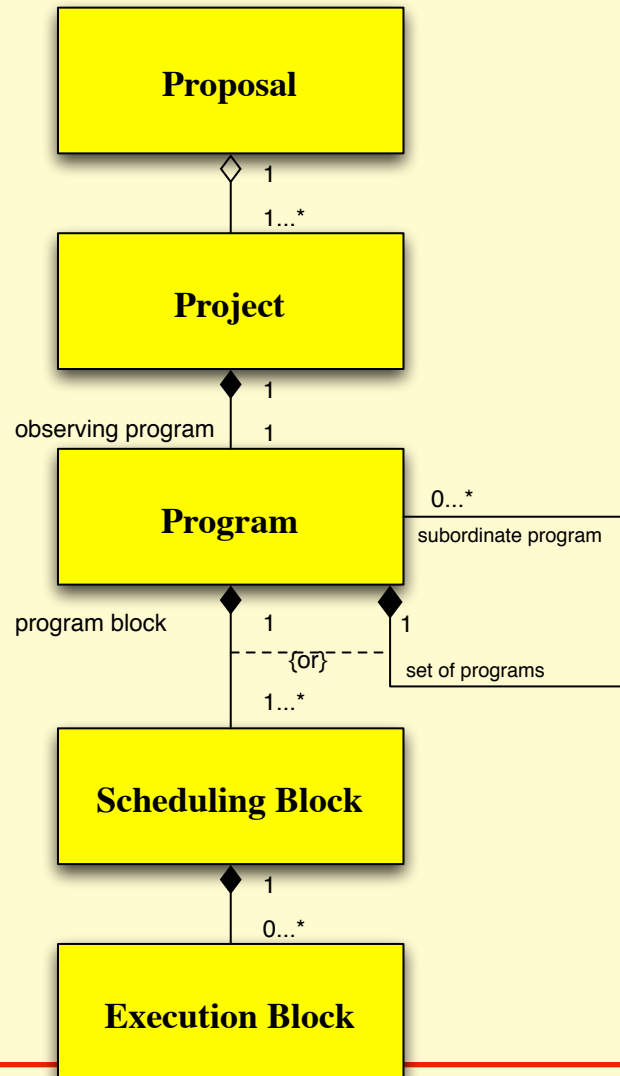


# NRAO Project Model

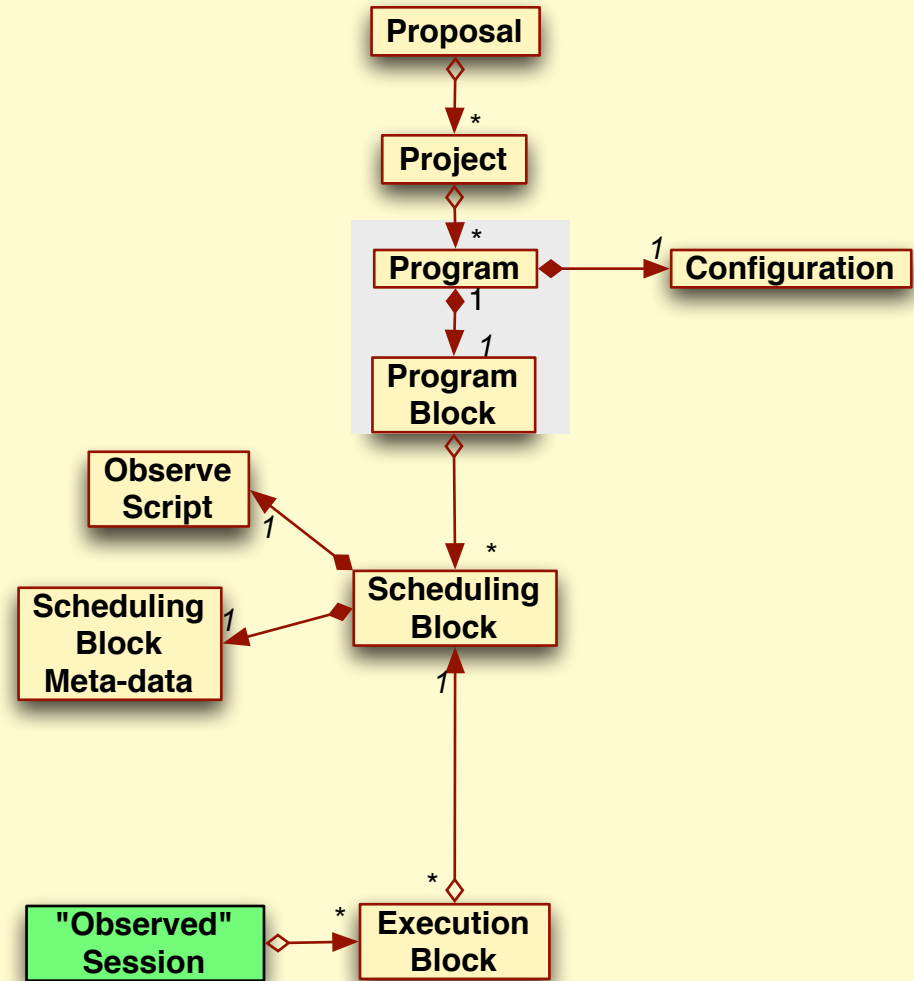
Projects → Programs → Scheduling Blocks



*E2E Model*

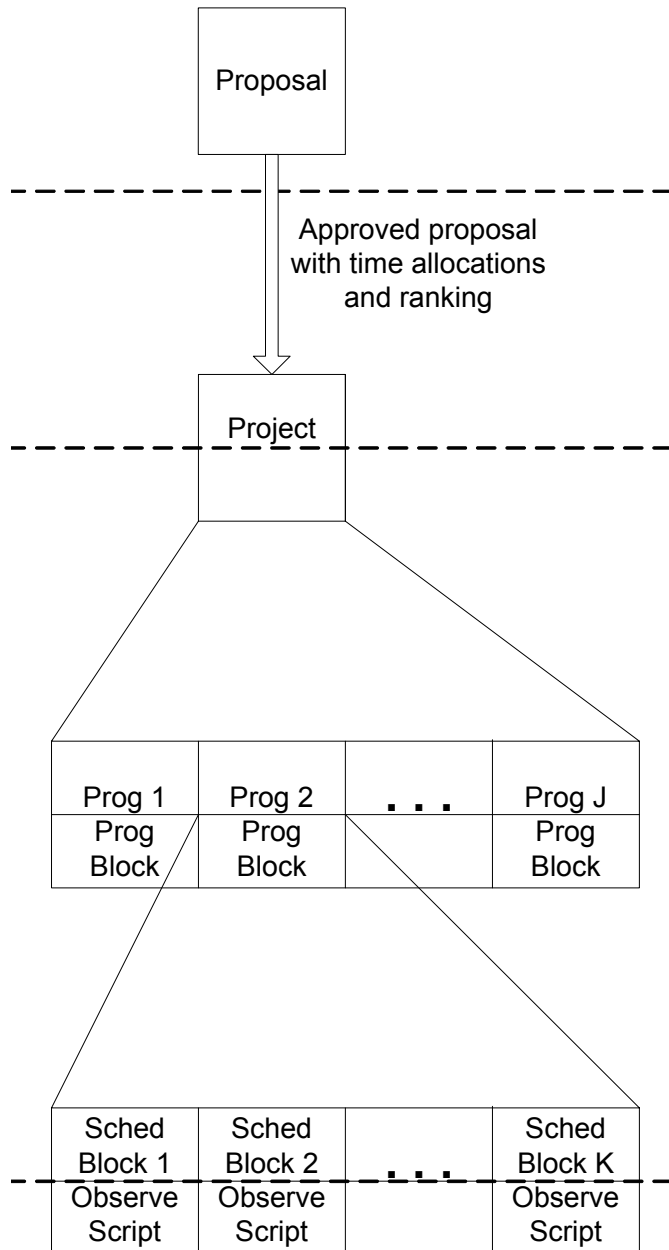
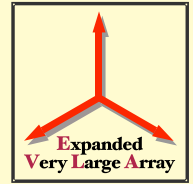


*EVLA Model*





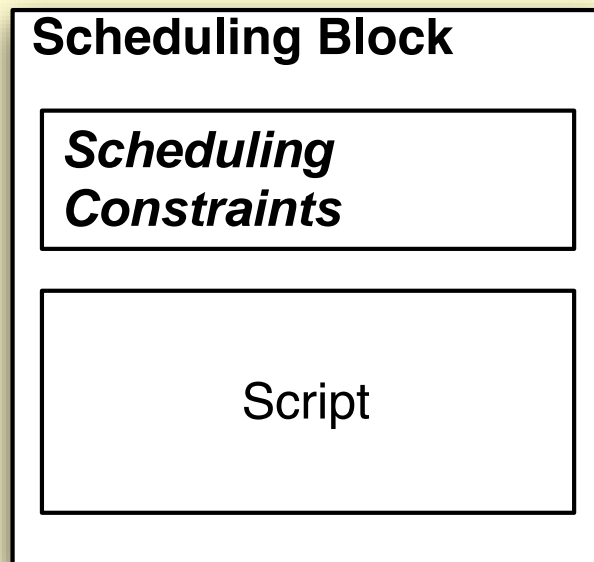
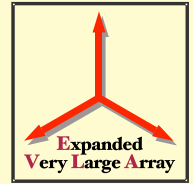
# Programs → Scheduling Blocks



- **Proposal + Allocated Time = “Project”**
- **Project + Telescope Configuration = Program**
- **A Program has a Program Block**
- **Program Blocks Organize Scheduling Blocks**



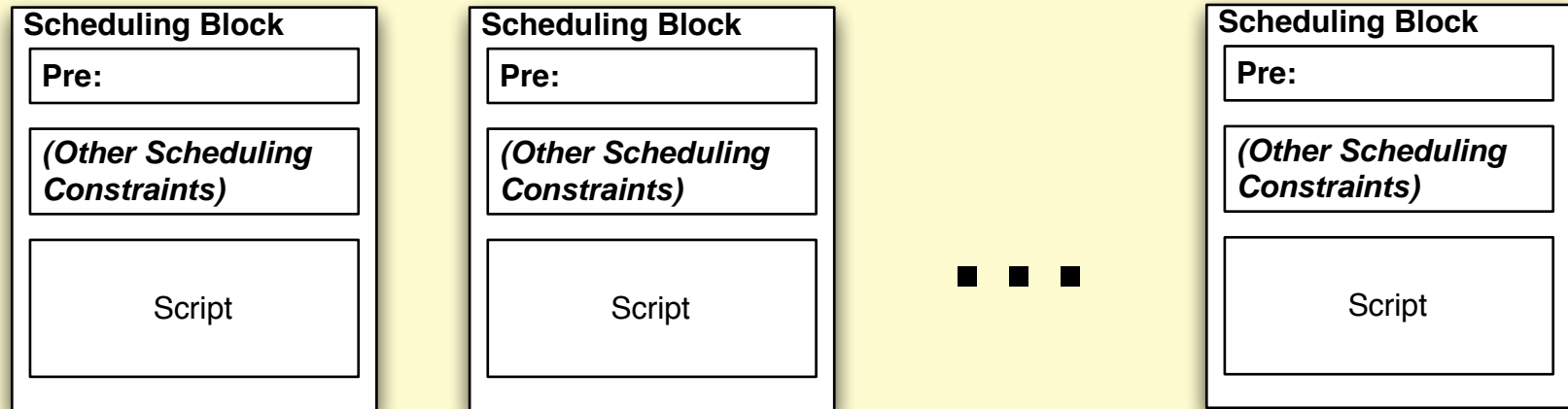
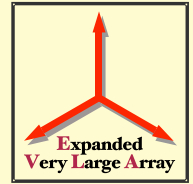
# Dynamic Scheduling



- Observations are comprised of Scheduling Blocks
- Scheduling Blocks are Atomic Units of Execution



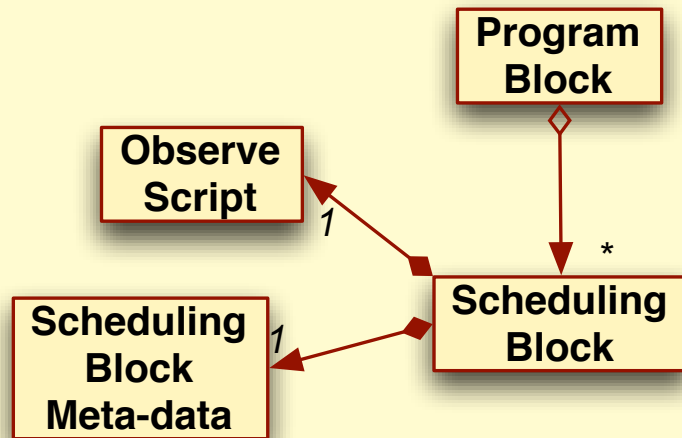
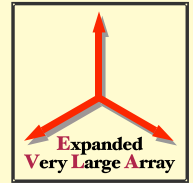
# Scheduling Block Selection



- *Scheduler Selects the Best Block from a “Pool” of All Available Blocks*

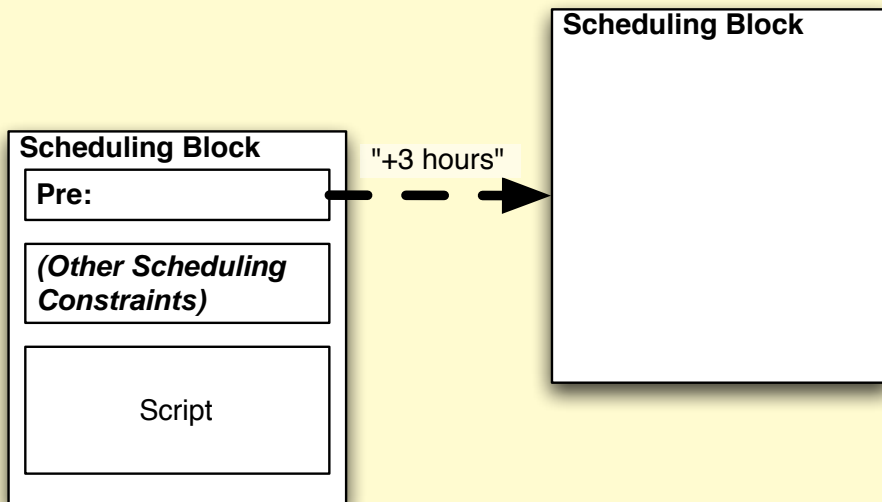


# Scheduling Block Dependencies



- Dependencies Between Scheduling Blocks may be (logically) Expressed in the Program Block

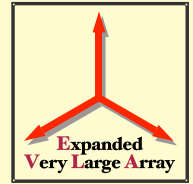
- *Implementation Note:* Scheduler is Simplified if dependencies are tracked in the Scheduling Blocks







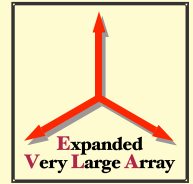
# Scheduling Block Iteration



- Absolute iteration: “repeat for N”
- UV Coverage
- RMS constraints: “repeat until  $\text{RMS} \geq n$ ”



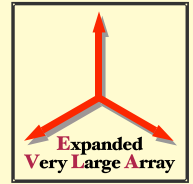
# SB Iteration: “Absolute”



- “Do this block N times.”
- An SB has an **Iteration Count**.
- Interruption of iteration does **NOT** reset the iteration count.



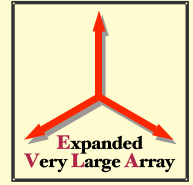
# SB Iteration: UV Coverage



- Given an LST Range, divide the LST range up into N chunks.
- Observe all N chunks...
- *Each chunk is an "LST Slot" that has the potential to be observed on a different (sidereal) day.*



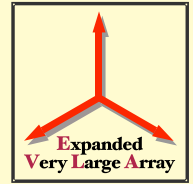
# SB Iteration: RMS Constraints



- Applied to an SB to build up integrations until the desired signal level is achieved.
- RMS values delivered by TelCal



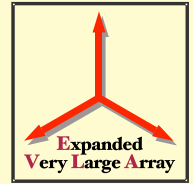
# Other SB Constraints



- Time Constraints
  - $T_{nominal}$  and  $T_{max}$
  - *LST Constraints*
- Equipment Status
- Weather
- $T_{sys}$



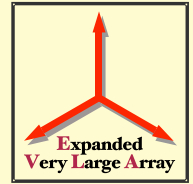
# Scheduling Block Dispatch



- **Executor Idle**
- Executor Alert (failure)
- Time Range Exceeded
- Rapid-Response Science Interrupt
- Operator (or Astronomer) Interrupt
- Environmental Conditions Alerts



# SB Pre-Emption



- **Hard Interrupt**

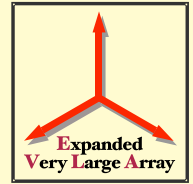
Force the currently-executing block off of a sub-array *immediately*.

- **Soft Interrupt**

Force the currently-executing block off of a sub-array after the end of the current integration.



# Sub-Arrays

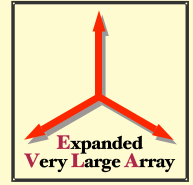


- *Sub-Array is a set of Antennas that can make baselines.*





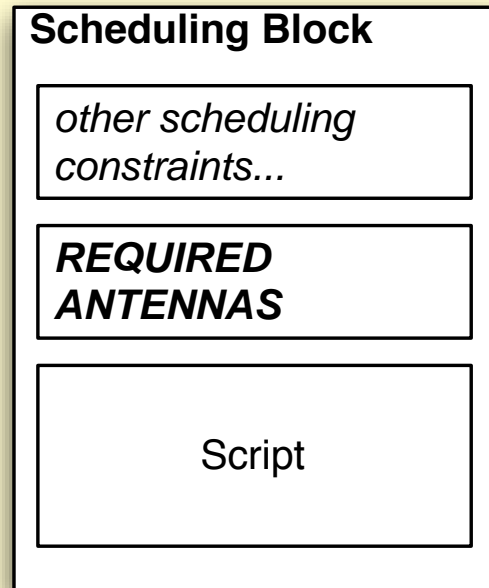
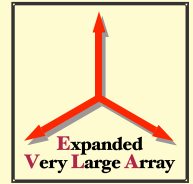
# Sub-Arrays



- **Astronomical Sub-Arrays**
  - *Created during Execution of Observation Script*
  - *Shared Script Execution Environment*
- **Administrative Sub-Arrays**
  - *Created for Independent, Parallel Execution of Different Observation Scripts*
  - *Independent Script Execution Environments*



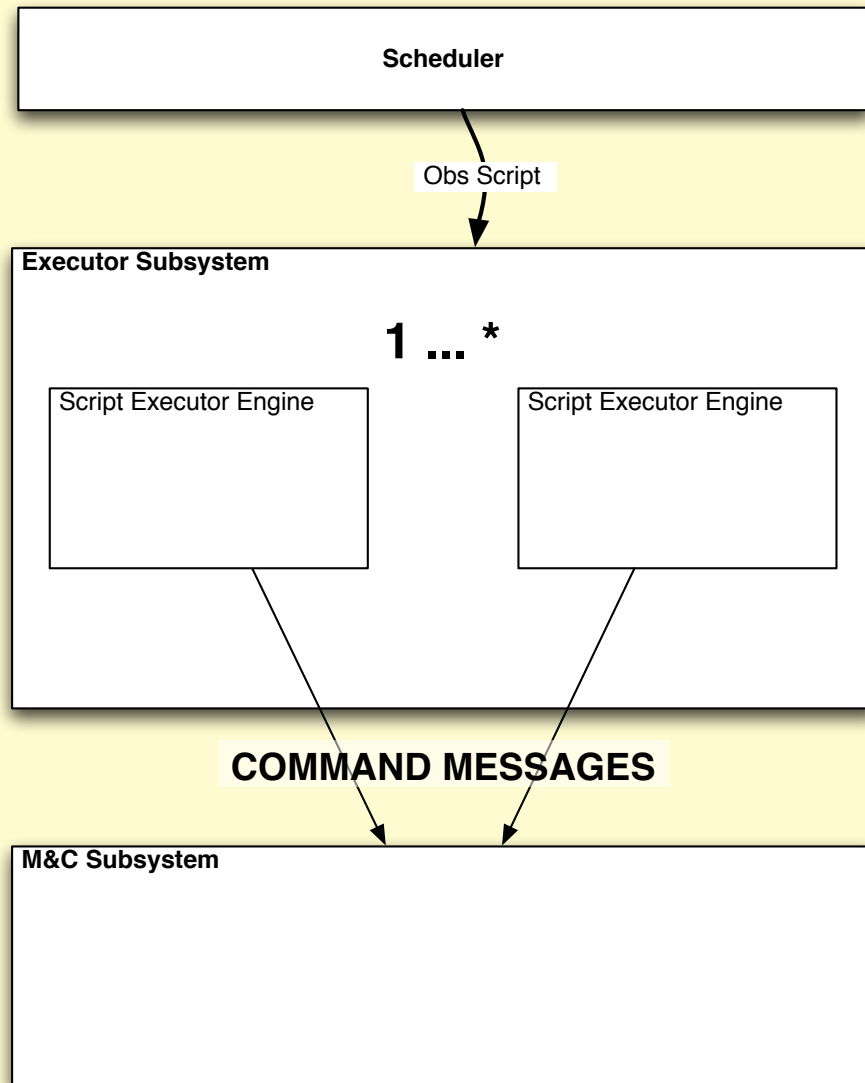
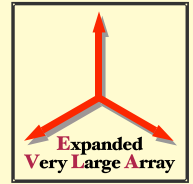
# Sub-Array Scheduling



- *Scheduler will only dispatch SBs for which the required antennas are available.*



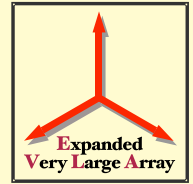
# Administrative Sub-Arrays



- *Independent, Parallel Execution of Different Observation Scripts*
- *Executor may Create a Sub-Array (with associated Executor) in Response to Scheduling Block dispatch.*
- *(Unless Sub-Array already exists.)*



# Astronomical Sub-Arrays



```
myVLA = new Subarray(VLA[0:25])
```

```
myband = LoIfSetup('10GHz', 8453.0, 8503.1)
```

```
myVLA.setLoIfSetup(myband)
```



# Command Translation

