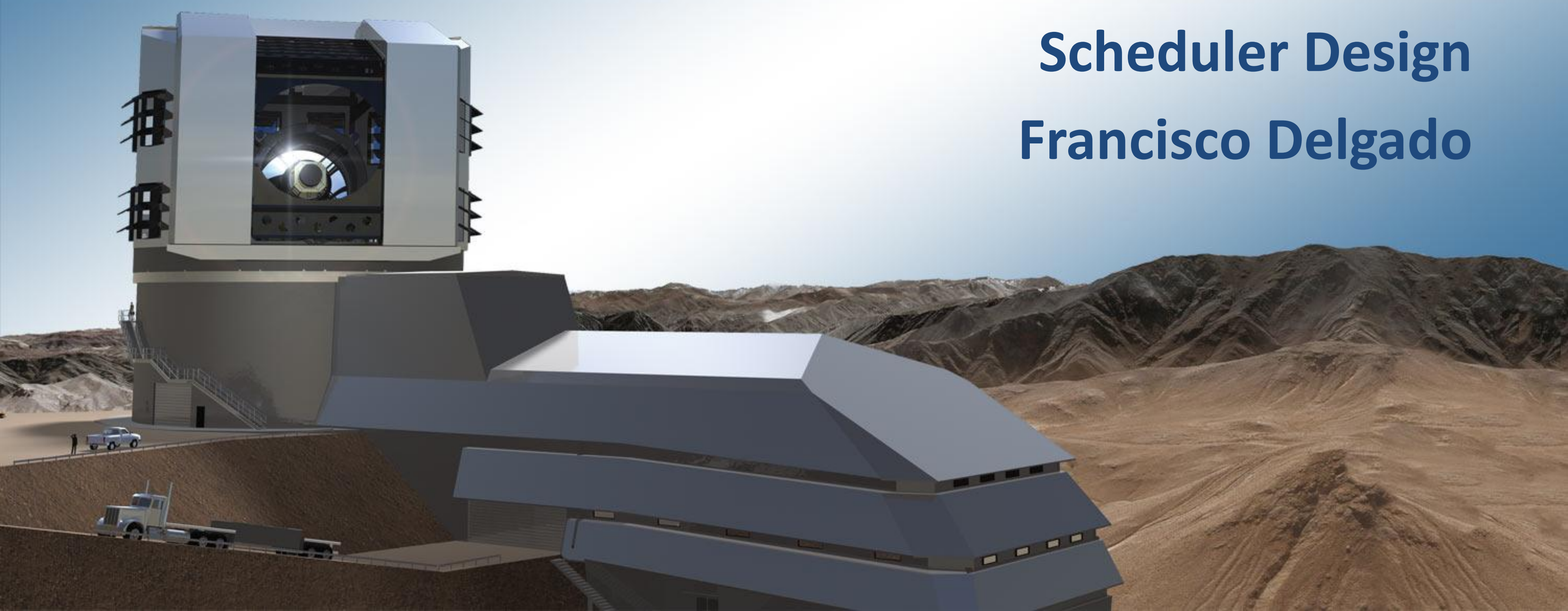
# LSST Telescope and Site Observatory Control System Interface Review

## Scheduler Design

Francisco Delgado



# Addressing the Charge

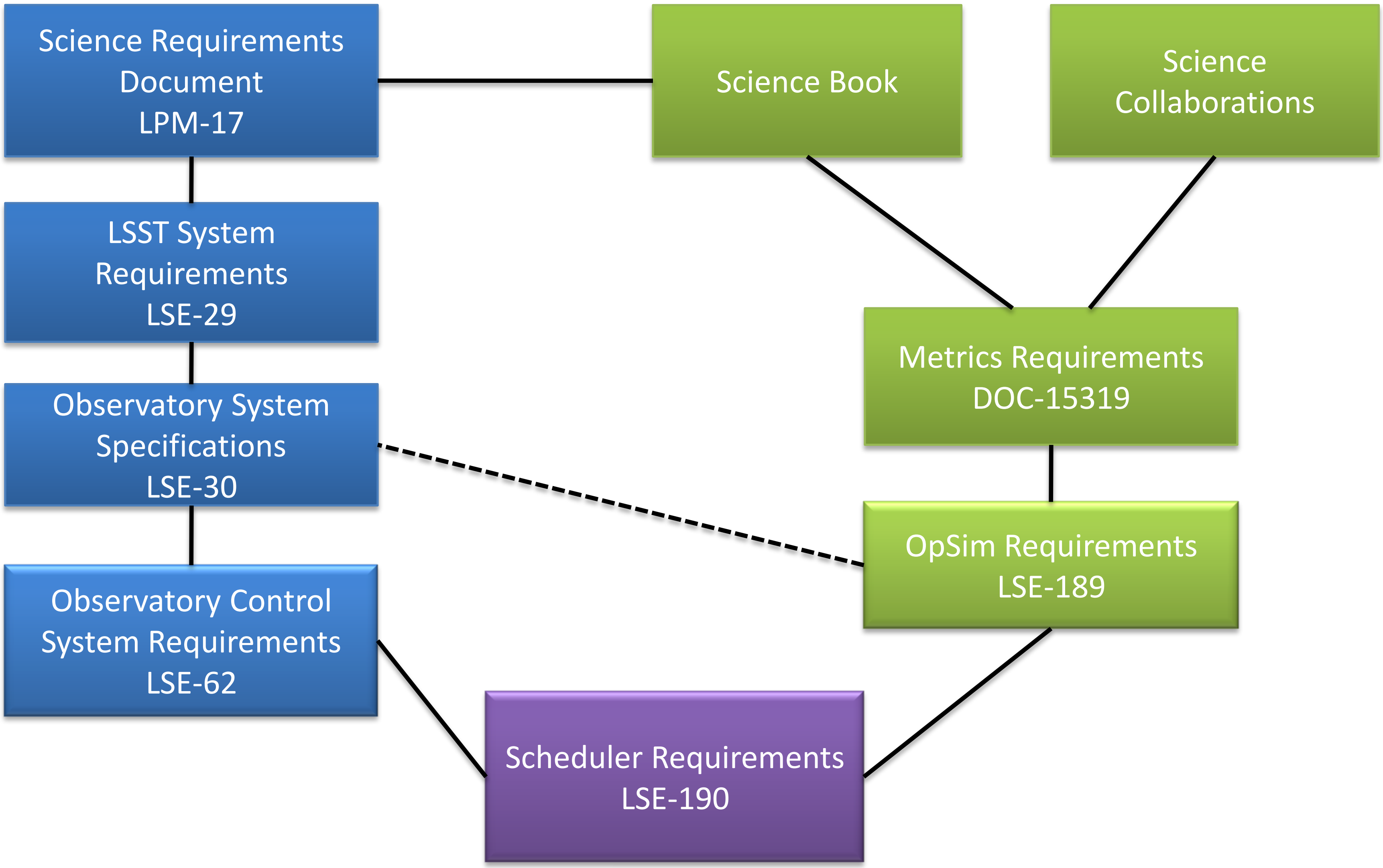
2. **Is the OCS design mature enough** to support (i) the analysis of compliance with the requirements and (ii) the definition of interfaces?

9. Are the plans for implementing the OCS are adequate and realistic, including budget, schedule, and organization/management structure? **Are the deliverables for the Scheduler and the Operations Simulator well defined and the corresponding resources properly aligned between the OCS and Systems Engineering teams?** Are the deliverables for communication middleware well defined and the assigned resources adequate?

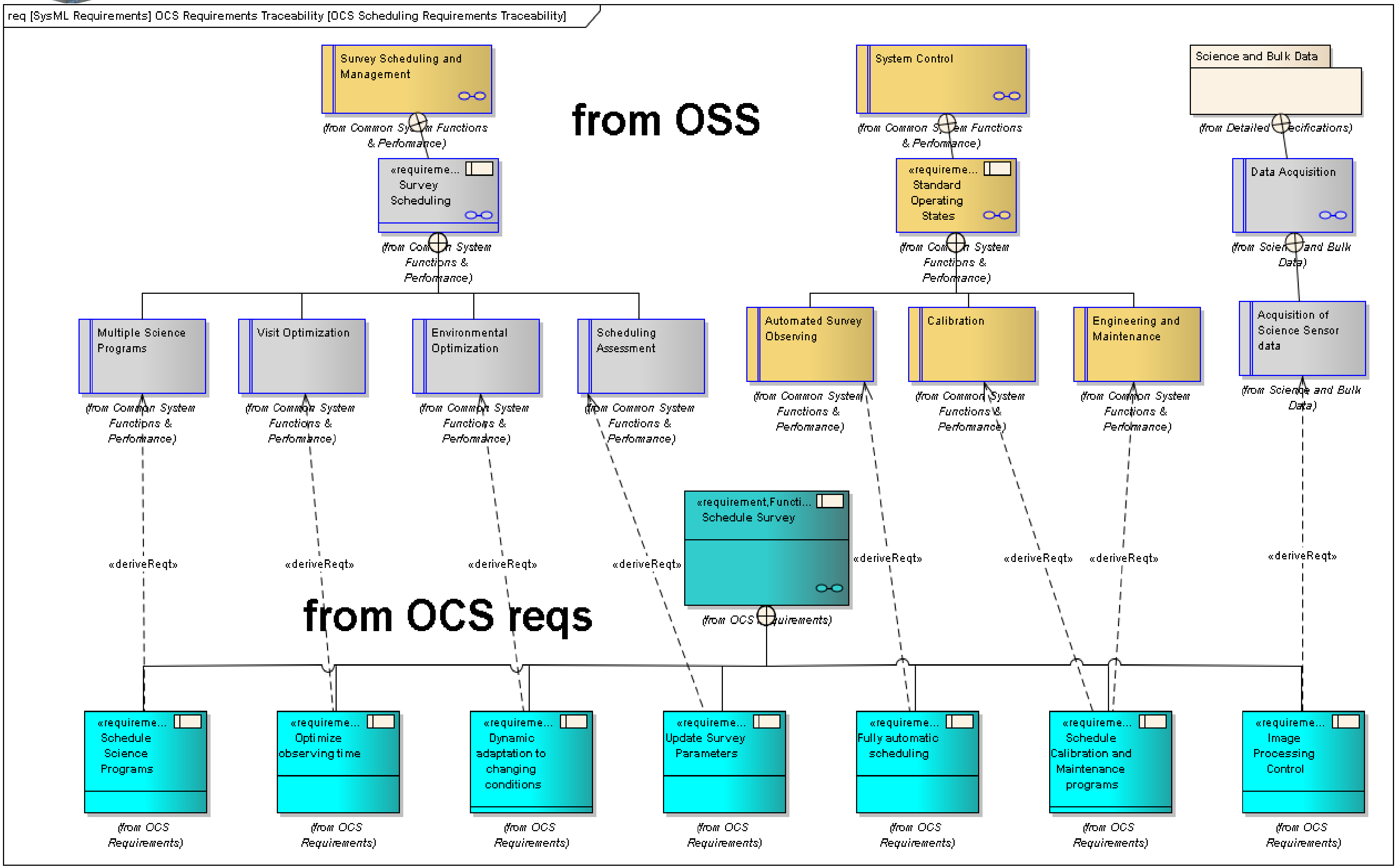
# Scheduling the LSST Survey

- LSST as a robotic observatory
- Survey is automatic
- Multiple science goals
- Combine area distribution with temporal sampling
- Dynamic adaptation to weather
- Flexibility for survey adjustments during operations
- Flexibility for changes in science programs

# Requirements Flow down



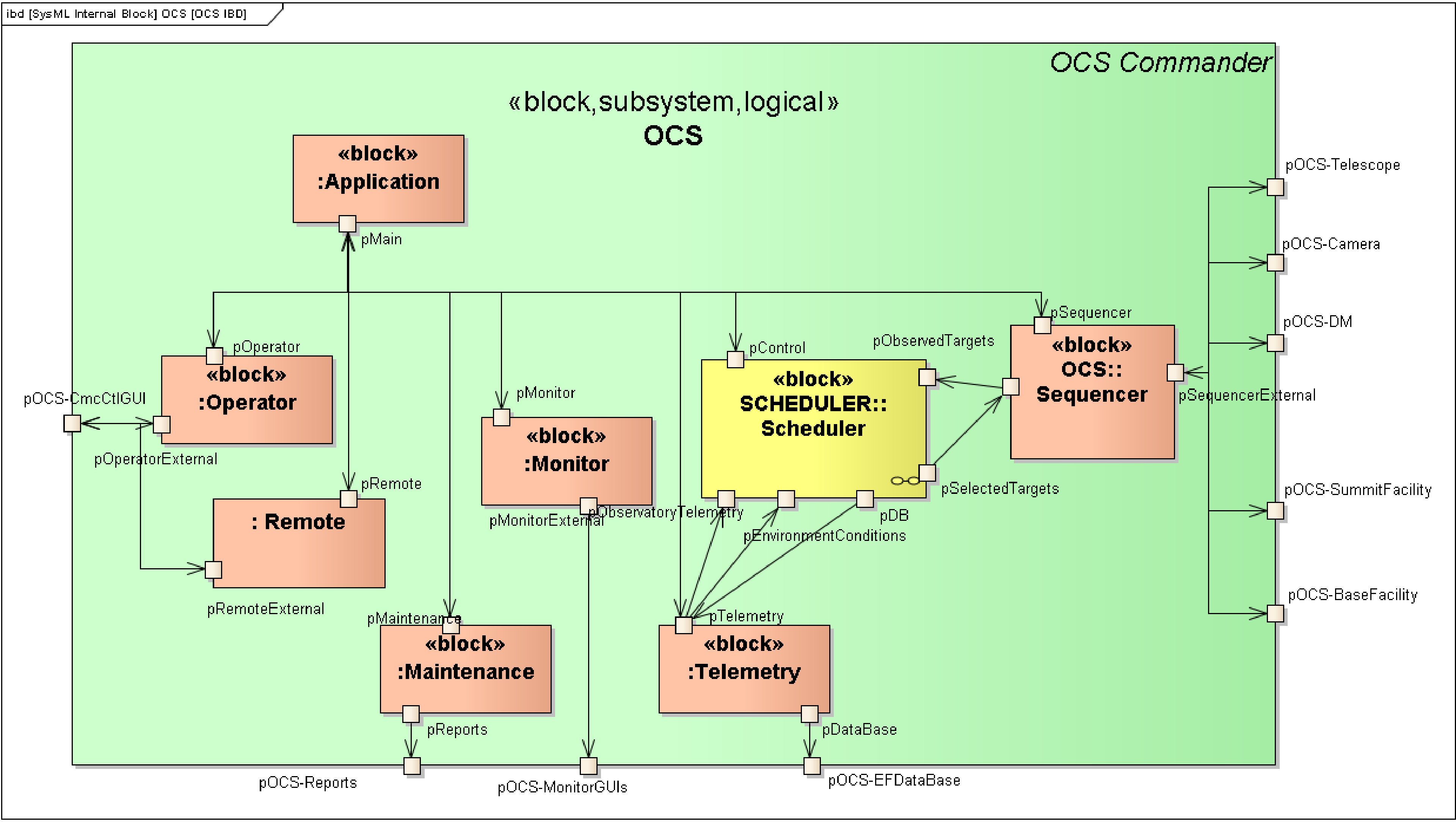
# Scheduler Requirements Traceability



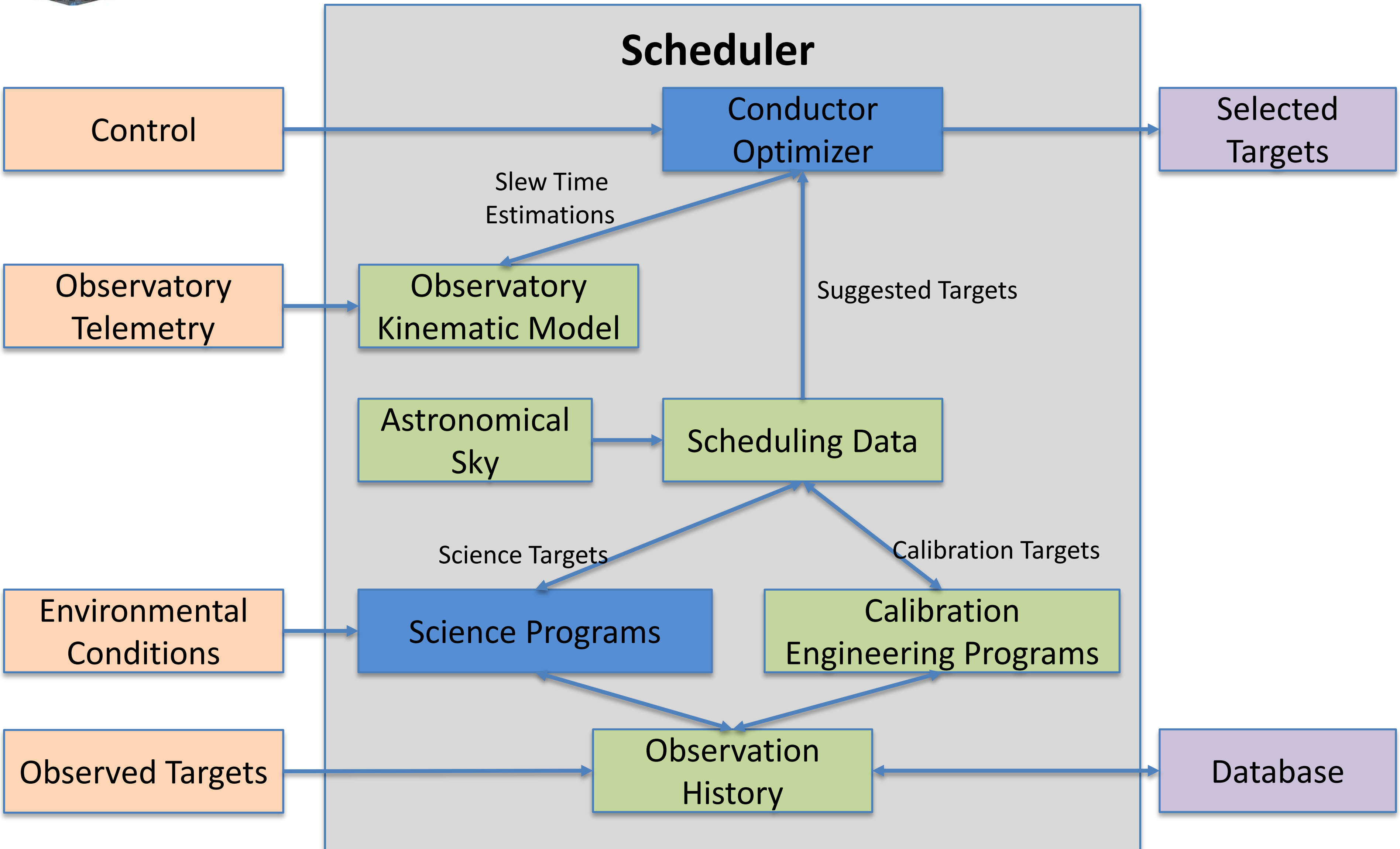
# Scheduler concepts

- Sky field map, tiling regions, a target is a field/filter combination.
- Fully configurable set of concurrent competing science programs.
- Sky brightness dynamically modeled for each sky field with look-ahead window.
- Comprehensive observatory kinematic model for slew time optimizations.
- Target score balances science value and time cost

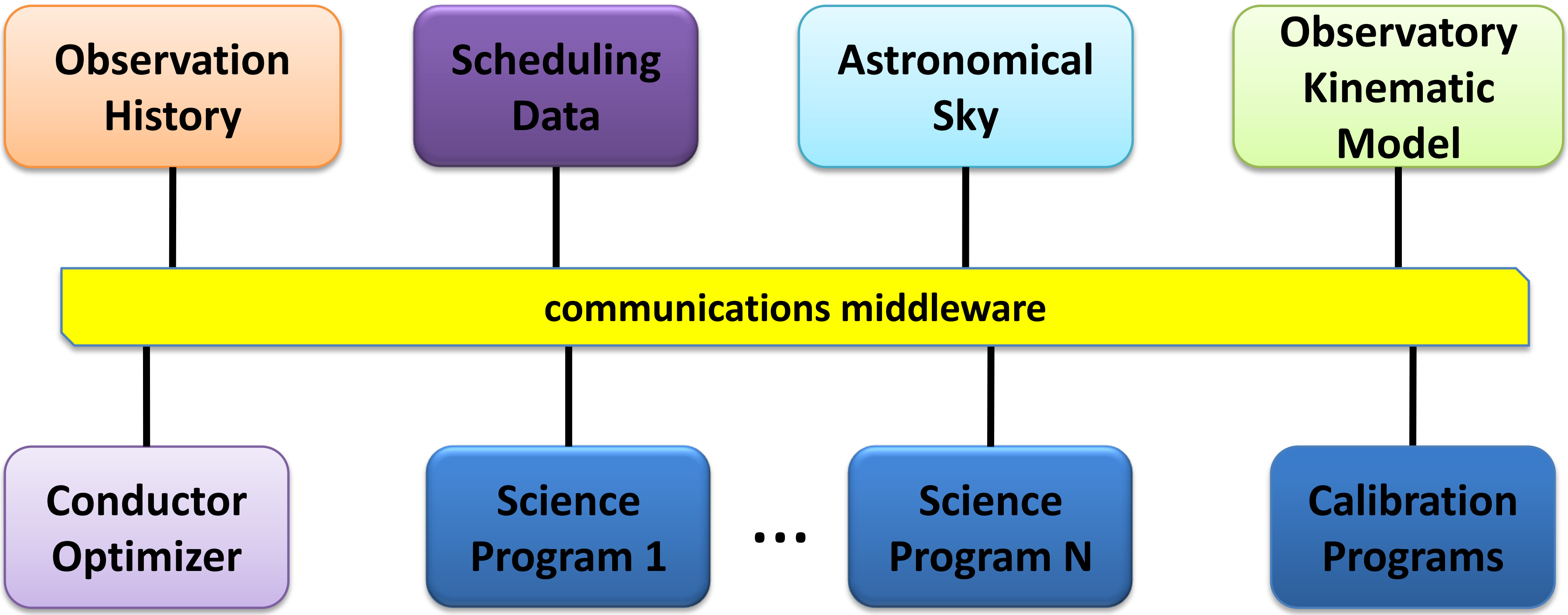
# Observatory Control System



# Scheduler Internal Block Diagram



# Scheduler internal communications



# Science Programs parameters

- Sky region.
- Number of visits per field in each filter.
- Cadence constraints for revisits or sequences.
- Airmass limits.
- Sky brightness constraints.
- Seeing requirements.
- Activation times.

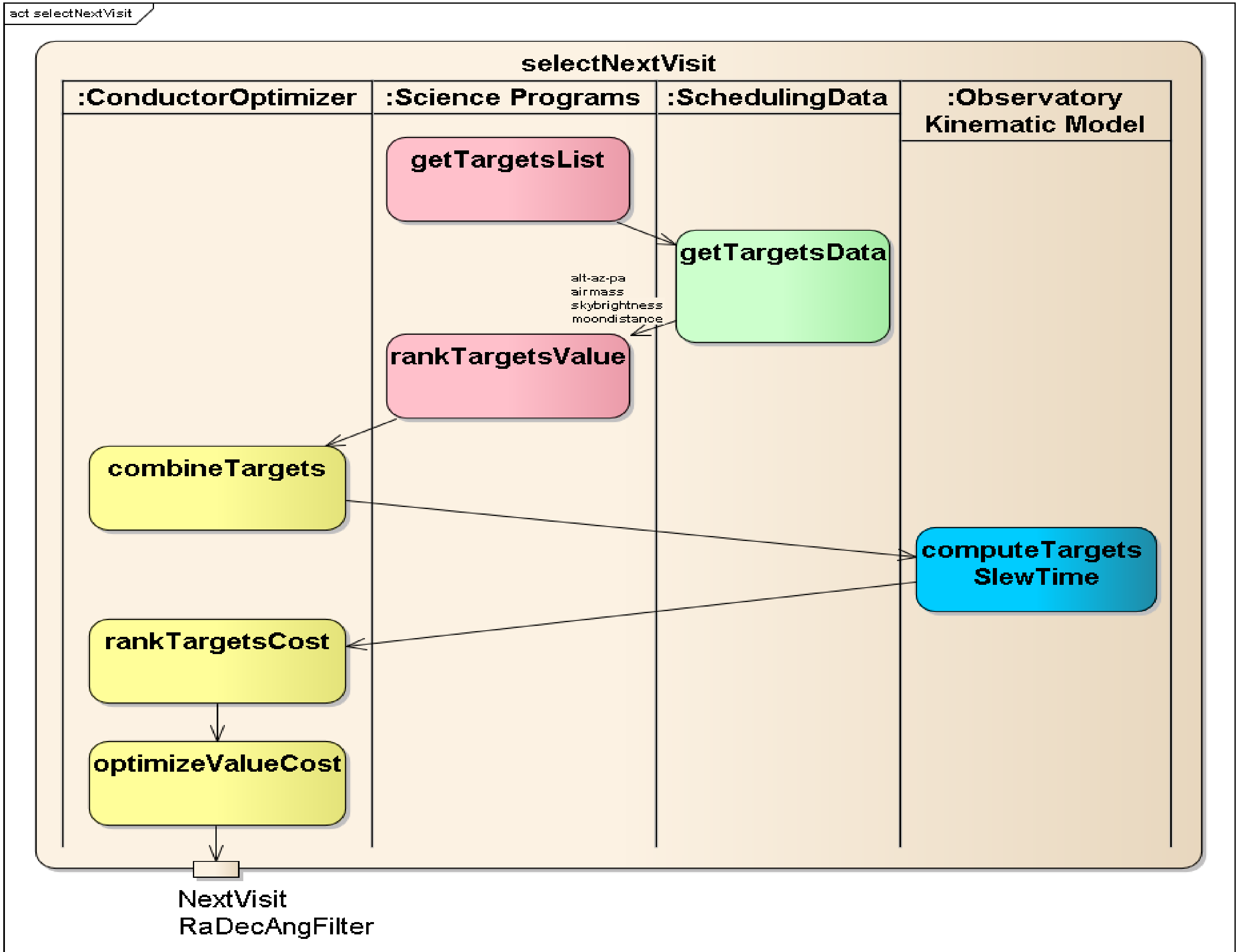
# Science Programs classes

- Area distribution programs
  - Designed to obtain uniform distribution
  - Basic parameter: goal visits per filter
  - Look-ahead info: future available time for the targets
- Time distribution programs
  - Designed to obtain specified intervals in sequences
  - Basic parameter: time window for visits interval
  - Look-ahead info: visibility for next intervals

# Selecting the next visit

- Dynamic and adaptive process for each visit:
  - Each science program:
    - analyzes its assigned sky region and selects the candidate targets that comply with its requirements.
    - computes the **science merit** for each selected target according to its own distribution and cadence constraints.
  - The conductor optimizer combines the targets and their science merit  from all the science programs.
  - The observatory model computes the **slew time cost** for each target from the current position.
  - The target with the highest overall rank is selected.

# Select Next Visit



# Look-ahead

- A time window is defined for a number of nights to the future.
- For each target from the candidates list:
  - Airmass and sky-brightness are pre-calculated.
  - Visibility is determined from each science program constraints.
- Science programs have this look-ahead information for improving time distribution and efficiency in sequences.

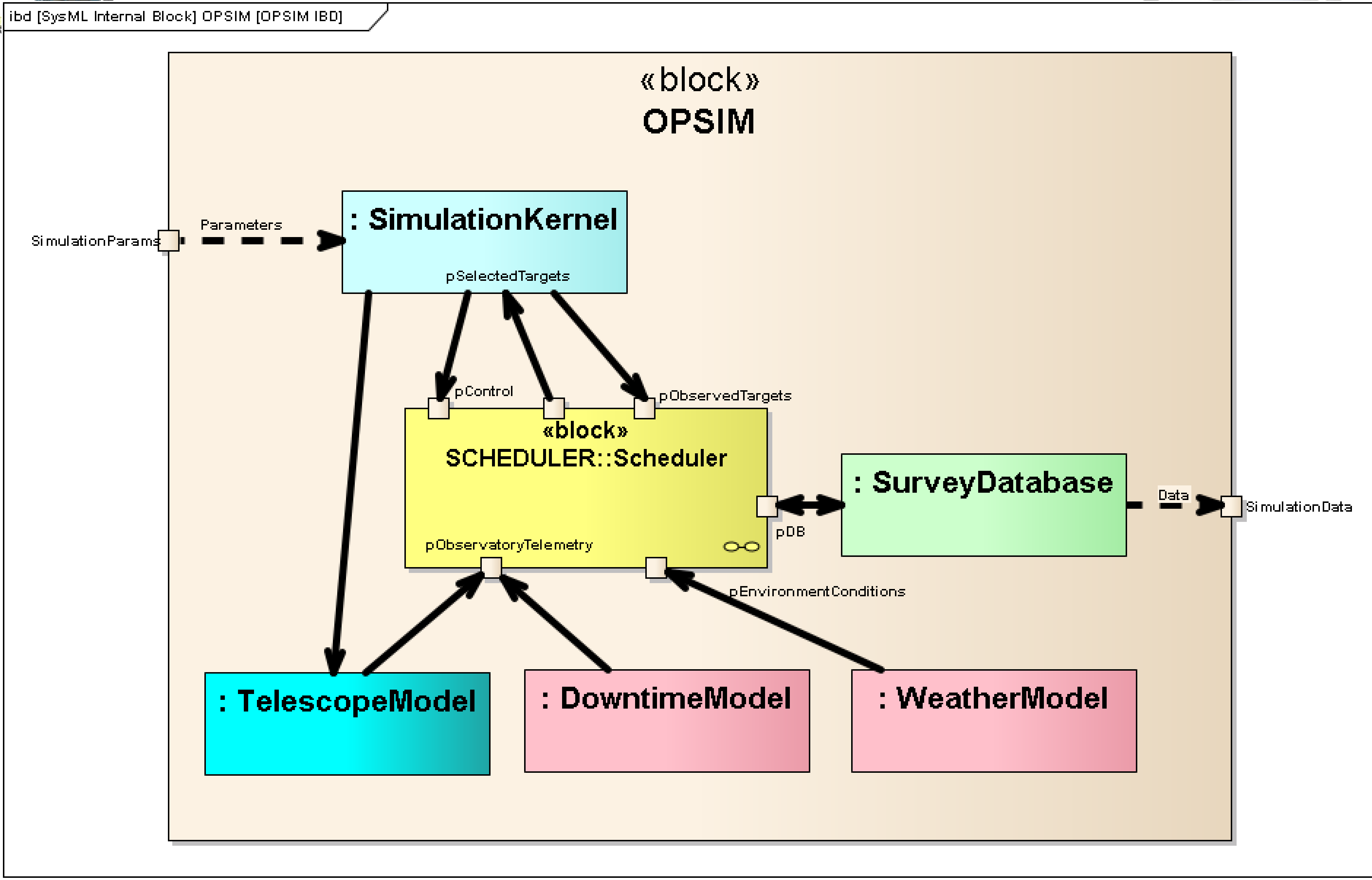
# Operations Simulator

- System simulation and prototype for the Scheduler
- Validate observatory design
- Design science programs to achieve SRD
- Develop an efficient LSST scheduling strategy
- Systems engineering trade off studies
- Support Commissioning and Operations

# OpsSim requirements

- Simulate Operations visit by visit for 10 years
- Simulate Observatory (Telescope & Camera kinematics, slew & track)
- Simulate Environment (clouds, seeing, sky brightness)
- Prototype Scheduler (targets generation and scheduling algorithms)
- Set of proposals, SRD defined universal plus auxiliary projects
- Flexibility for algorithm experimentation

# OpSim Architecture

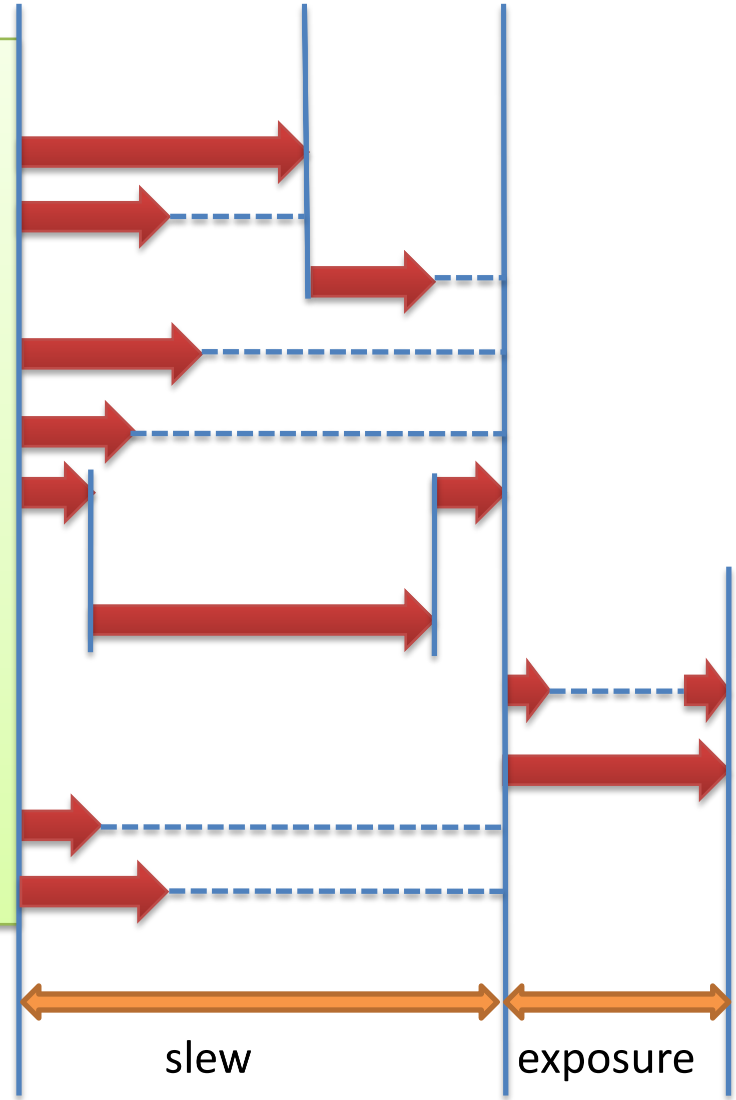


# Environment Models

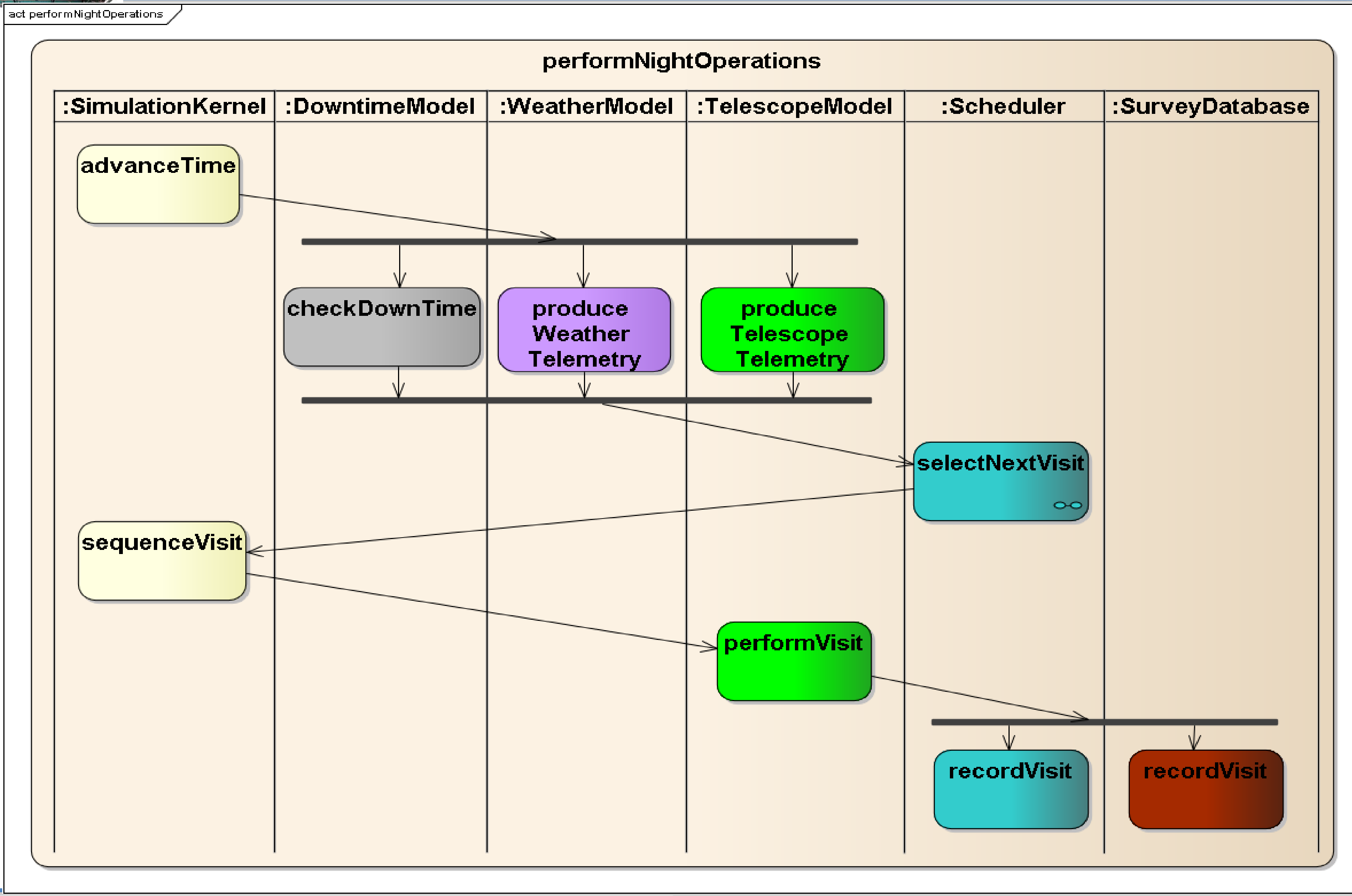
- slalib for sun & moon
- Sophisticated sky brightness model using the Krisciunas and Schaeffer model with twilight.
- Actual seeing historic measurements from the site.
- Actual clouds historic record from the site.

# Observatory Model

- **Second order kinematic model for the slew activities**
  - ❖ Mount Azimuth with cable wrap……………………
  - ❖ Mount Altitude……………………………………………
  - ❖ Mount Settle time………………………………………
  - ❖ Dome Azimuth……………………………………………
  - ❖ Dome Altitude……………………………………………
  - ❖ Rotator Angle……………………………………………
- Delay model for Camera
  - ❖ filter change……………………………………………
  - ❖ Shutter time……………………………………………
  - ❖ exposure time…………………………………………
  - ❖ Readout time……………………………………………
- Active Optics correction…………………………………



# OpSim activity diagram of a visit



# OpSim implementation

- Python language for the logic and data handling

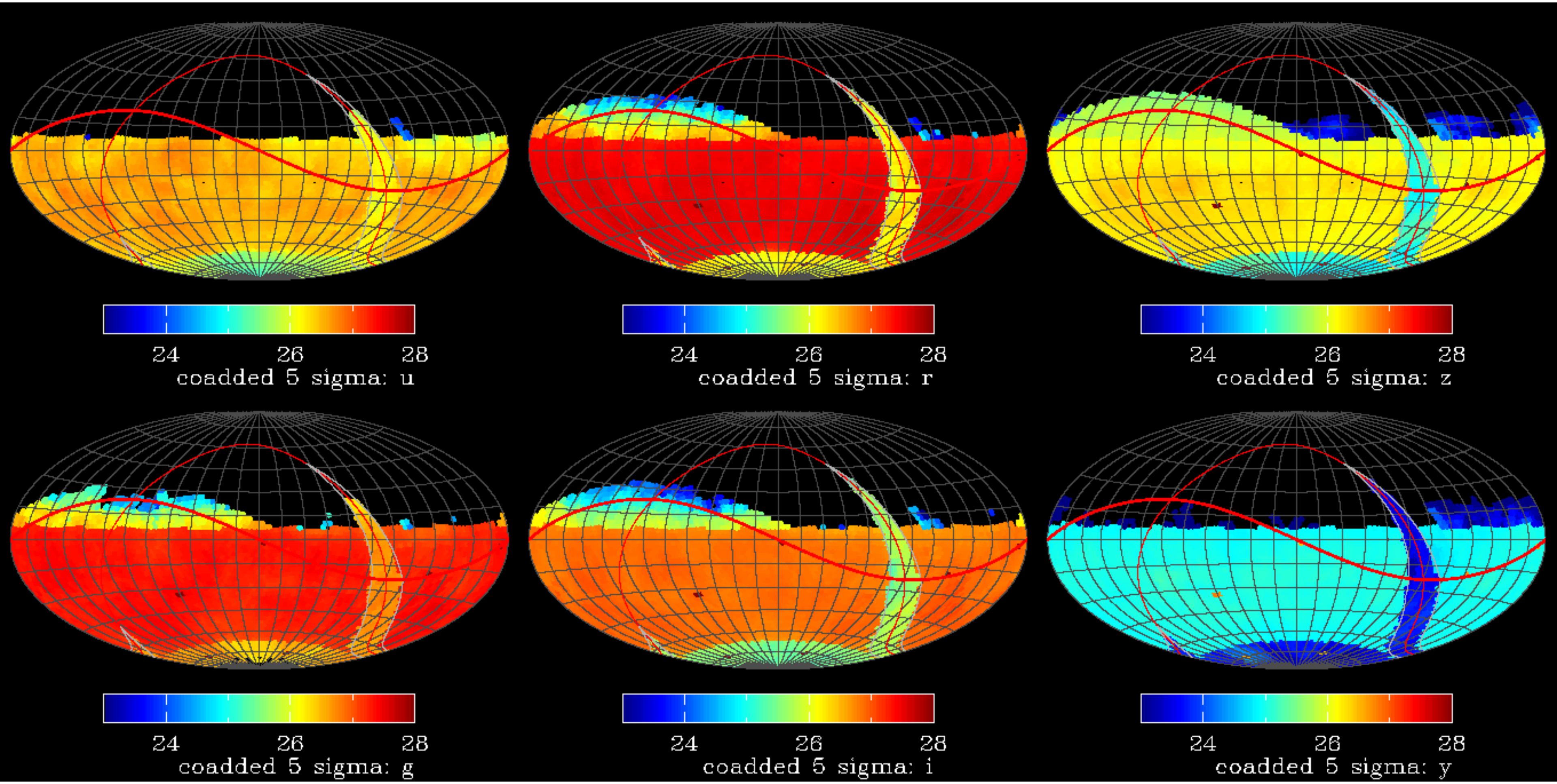
  C++ for libraries, such as slalib

  20k lines of code approx.

- Typical 10 year run takes 50 hours in personal computers

- MySQL database with 22 tables for the history of visits, slews and sequences, sky conditions, etc.

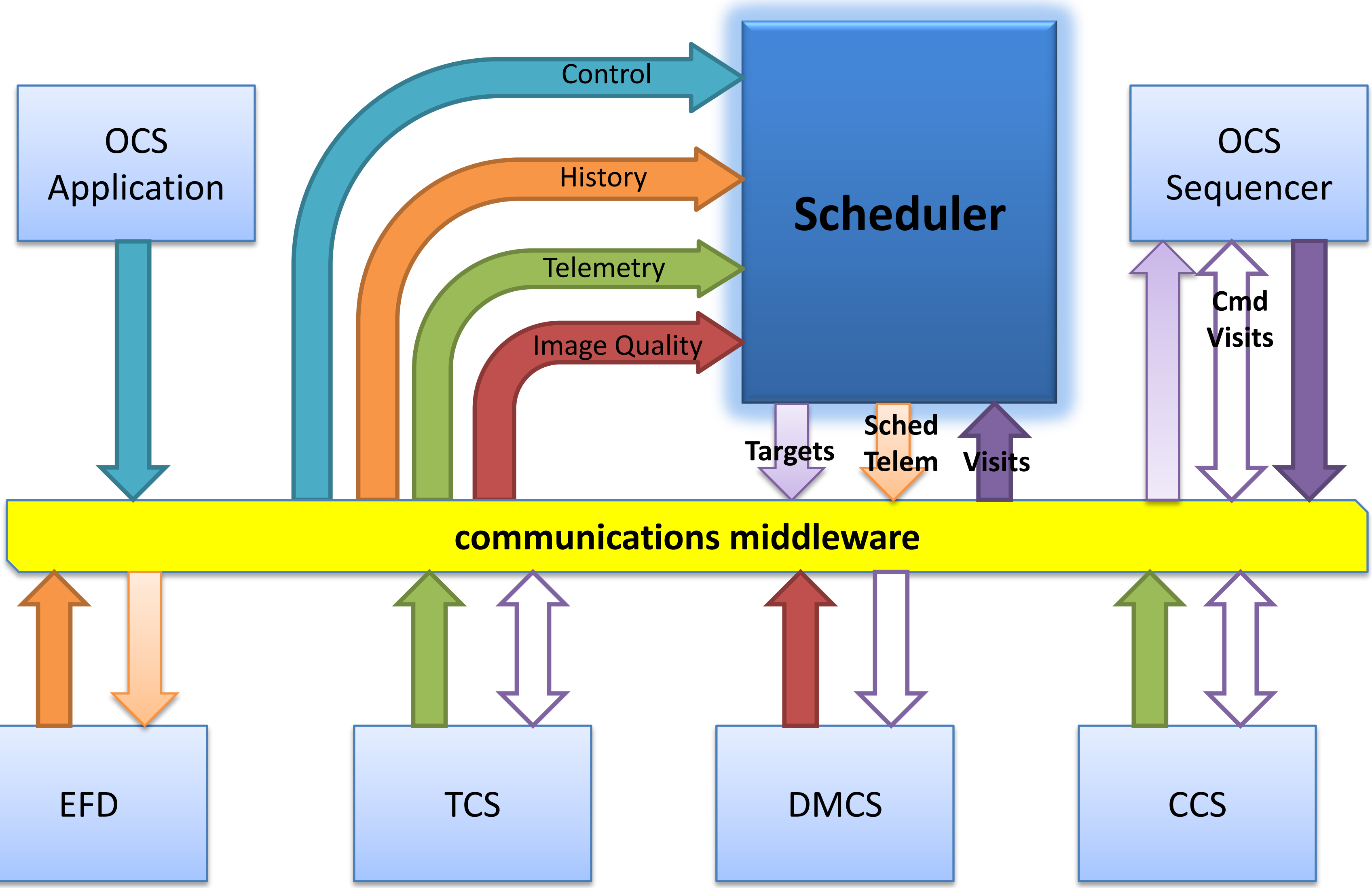
# Sky coverage per filter



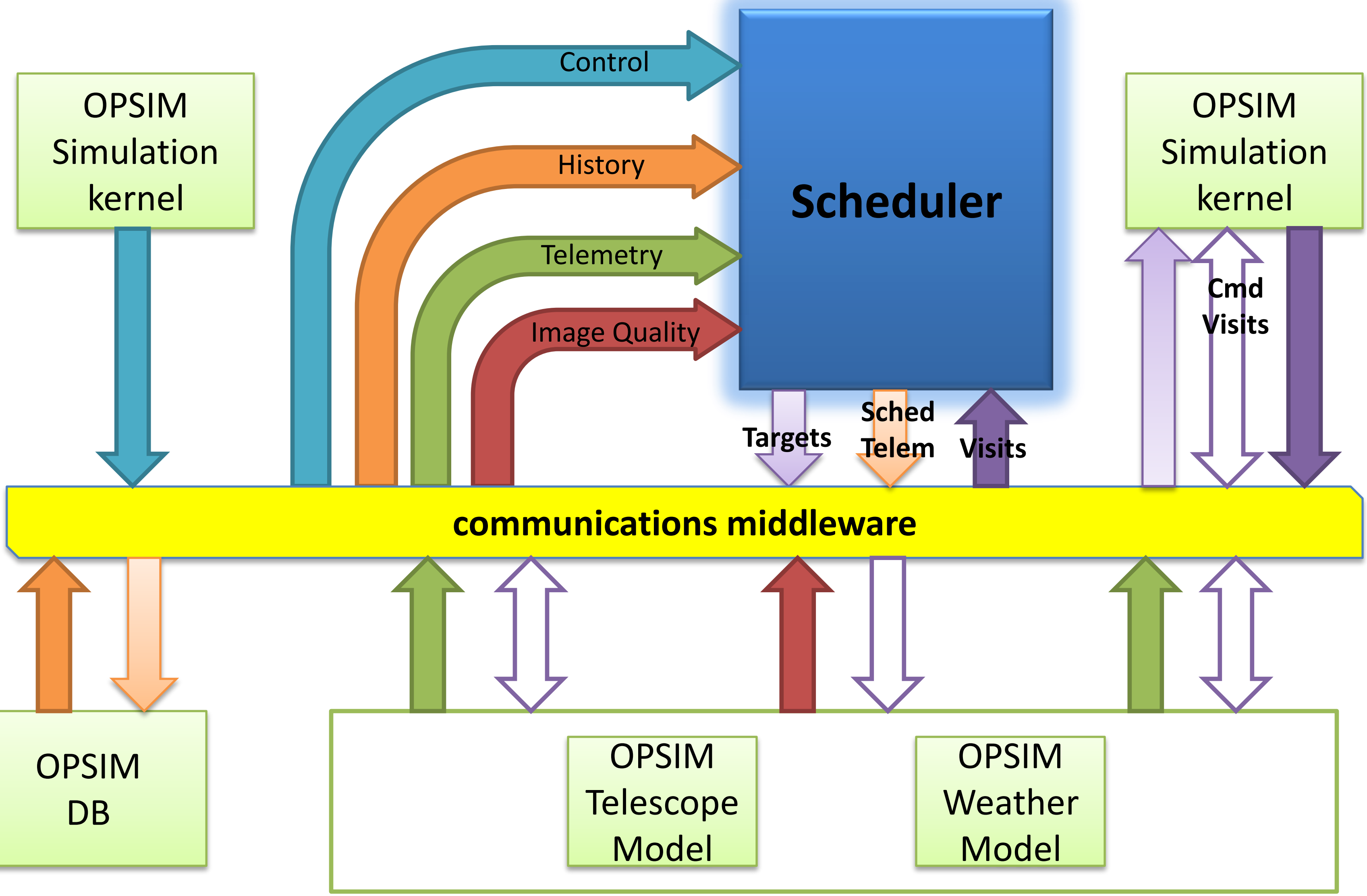
# OpsSim & Scheduler configuration

- **System** 117 parameters, including the site, sky model and the kinematic model
- **Scheduler** 11 parameters for controlling the algorithms
- **Survey** 130 approx. parameters for each the science programs
  - Typical set of 5 programs
  - 3600 sky fields
  - Parameters for depth per color
  - Parameters for sequence cadences
  - Sky brightness limits
  - Airmass limits
  - Seeing limits

# Scheduler Interfaces in OCS



# Scheduler Interfaces in OPSIM



# Scheduler Inputs/Outputs

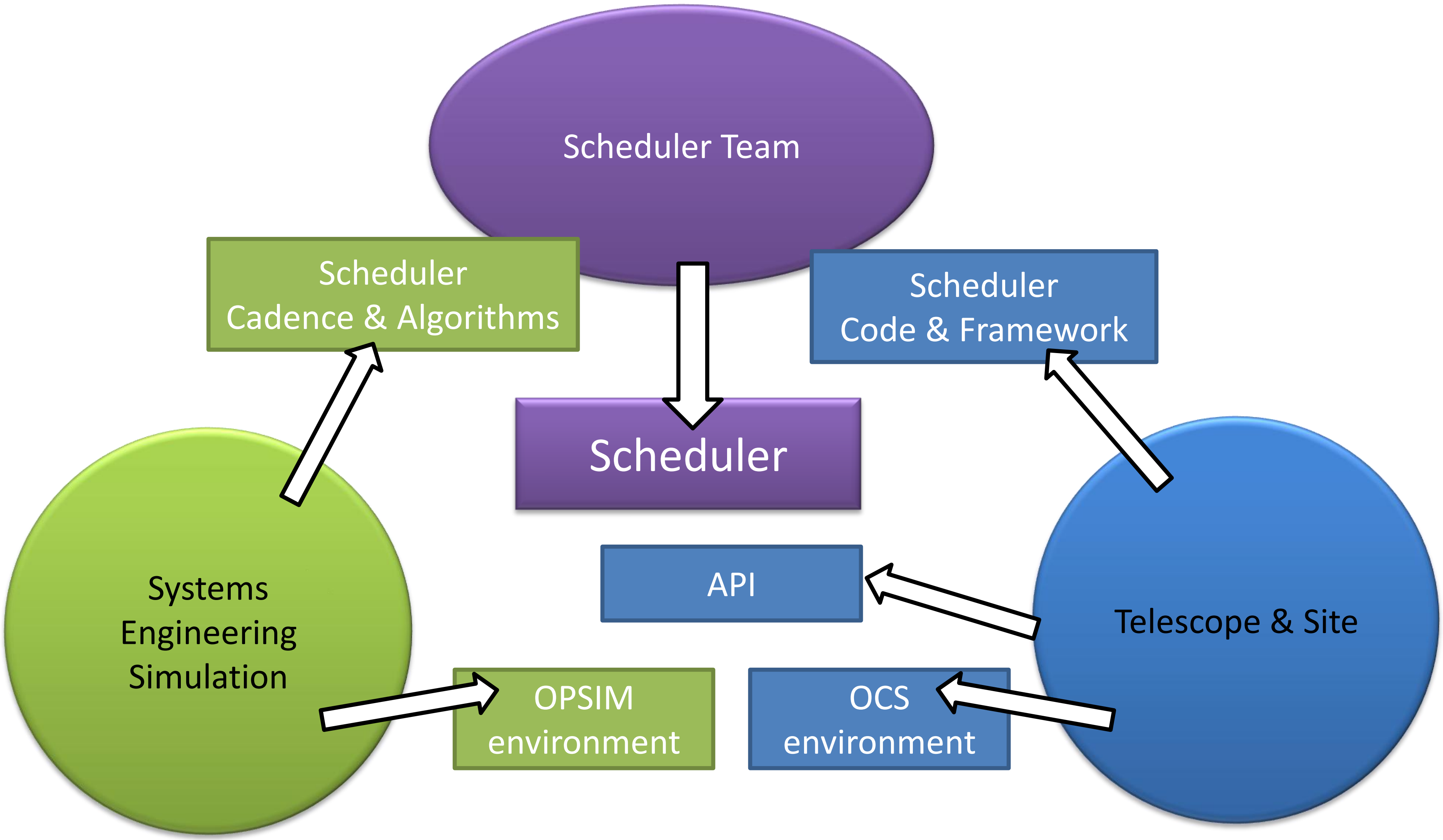
- Inputs
  - Control
    - Mode
    - Downtime
    - Degraded
  - Telemetry
    - Observatory conditions
    - Environment conditions
    - Forecast
  - History
    - Past observations
  - Visits
    - Current observation
  - Image Quality
    - Quality parameters
- Outputs
  - Targets
  - Scheduling telemetry

# Scheduler Development Partition

- Design & Implementation (T&S)
  - API
  - Architecture
  - Coding
  - System parameters
    - Conductor/Optimizer
    - Scheduling Data
    - Generic Science Program
    - Calibration Engineering Programs

- Cadence & Algorithms (SE Simulation)
  - Science cases
  - Algorithms
  - Survey and Scheduling  parameters
  - Coding
    - Observatory Kinematic Model
    - Astronomical Sky
    - Specific Science Programs
    - Observations History

# Deliverables



# Summary

- Scheduler design integrated with OCS architecture.
- OCS telemetry architecture enables the use of any variable for scheduling purposes.
- Partition and architecture makes for a flexible implementation.
- Designed to allow a distributed deployment.
- Scheduling strategies have been extensively tested in OpSim.
- Simple scheduling algorithms applied to thousands of competing targets produce emerging behavior to solve a complex problem.

# End of Presentation