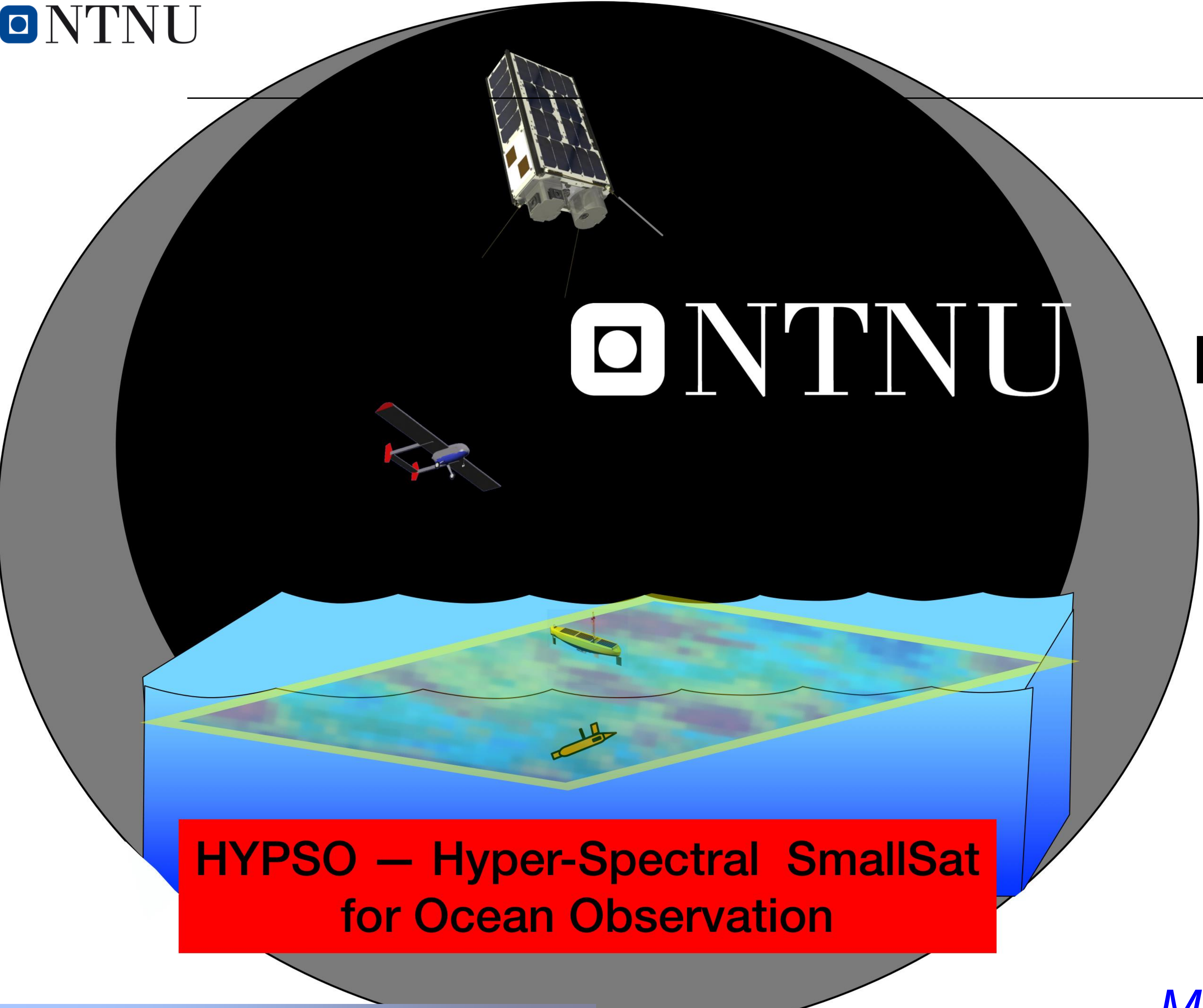


HYPSO Ground Data System (GDS) Preview

Kanna.Rajan@ntnu.no

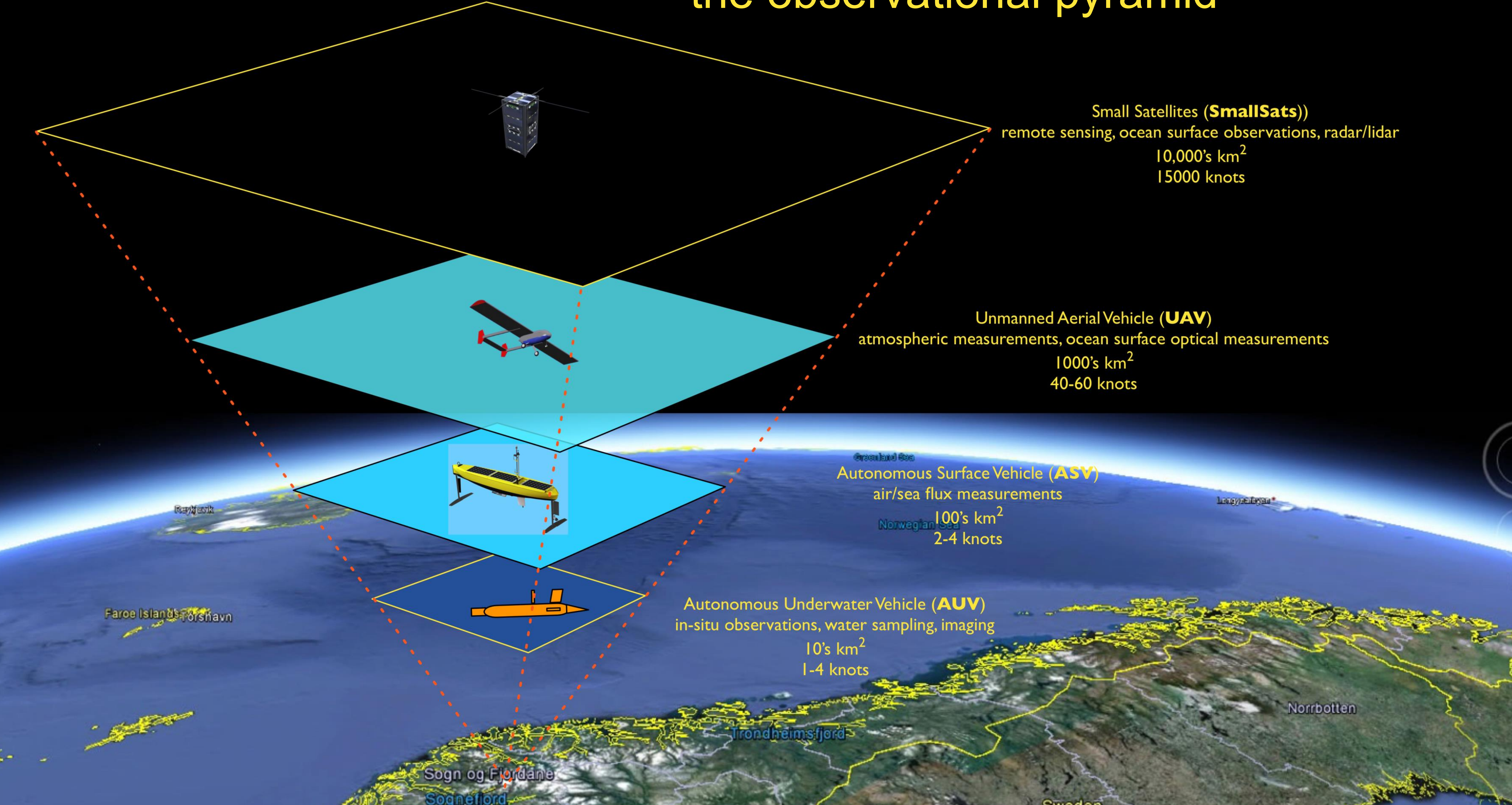
&

Mariusz.Eivind.Grotte@ntnu.n

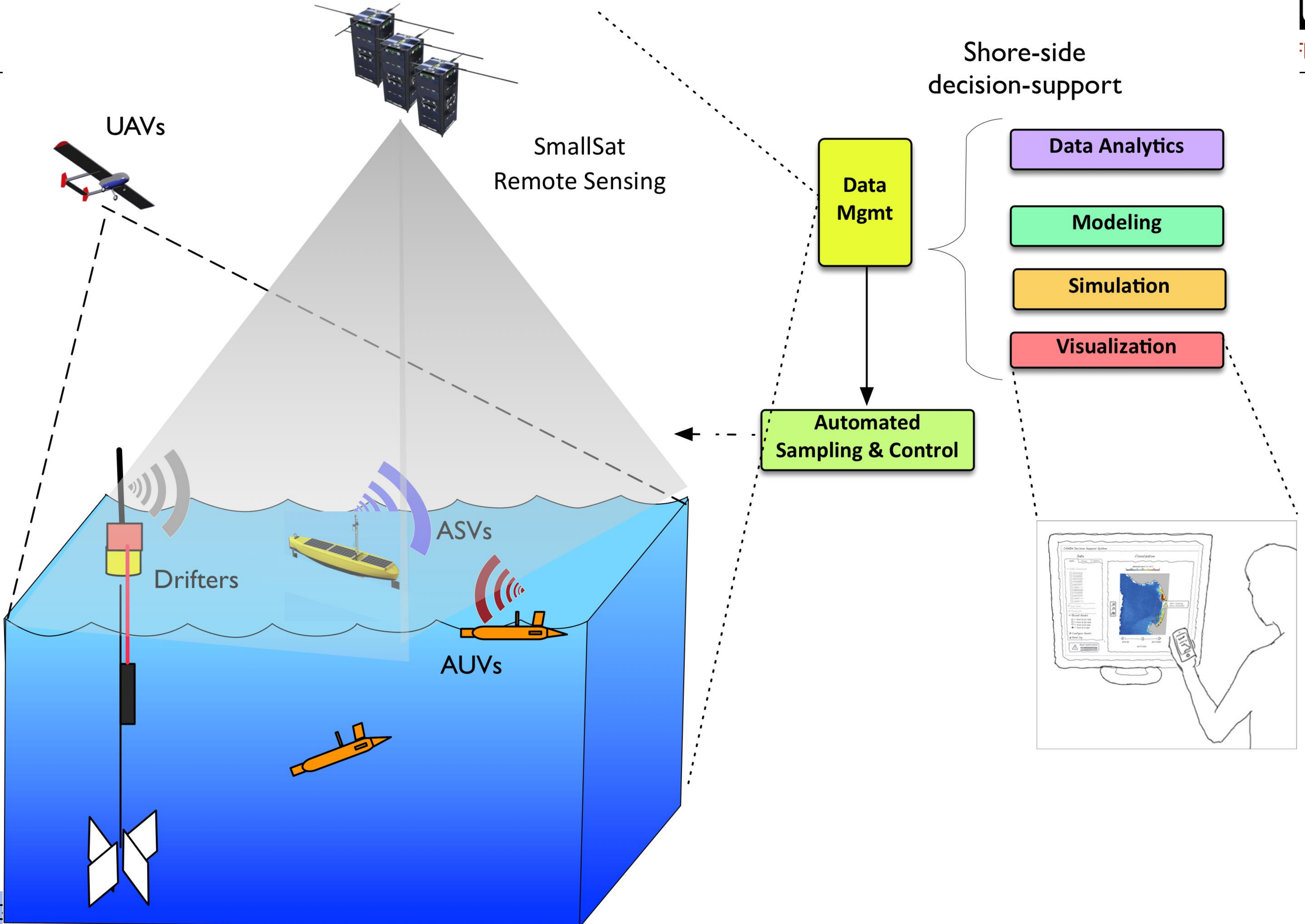
goal

- An attempt to sketch out a conceptual Ground Data System primarily for HYPSO, but also for future missions
- As generic as possible, with some dive into specific detail
- A draft of the plan with some suggestions on how to proceed
- More detail on the notion of Automated Planning — what/how/where

the observational pyramid



the larger picture

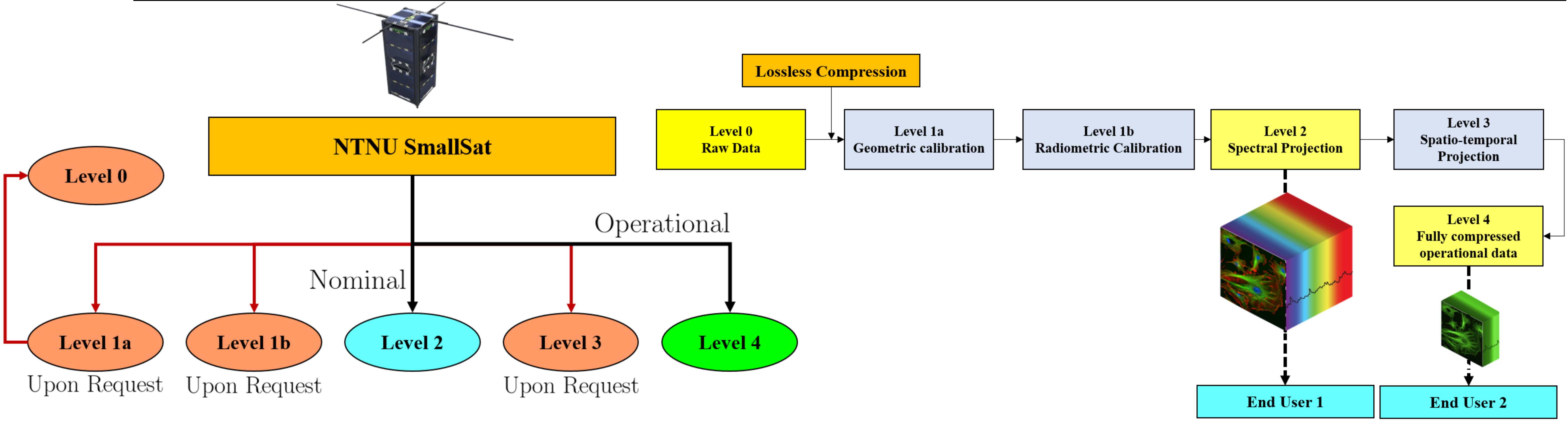


HYPSO — Mission Overview

- Provide ocean color mapping & monitoring with a hyperspectral Imager (HSI) payload
- Concert of robotic agents
  - augment architecture with UAVs, USVs, AUVs and buoys
  - remote sensing linked to database from other agents
  - in-situ validation necessary
- Launch scheduled in Q4 2019
  - tight constraints on payload development and project flow
  - individual subsystems and integrated system to be tested
  - follow up by second mission in Q3 2020
- Distributing scientific and operational datasets
  - nominal are Level 4 and Level 2 types
  - Level 0, 1a, 1b and 3 upon request from end user
- S/C = 6U platform
  - high battery capacity
  - high-performance ADCS
  - S-band downlink radio
- NTNU = Mission Control Center
  - data are both uplinked and downlinked and mission operations are monitored and commanded
  - ideally a Svalbard Ground Station would be necessary to downlink in the same pass consequently after imaging

General	**Definition**
Objective	Ocean color
Subject	Coast of Norway
Target Location (baseline)	Lat: 63.867608 °, Lon: 8.663644 ° (Frøya, Norway)
Target area (in-track × cross-track)	50 km × 70 km
Orbit	
Type	10:00 AM/10:00 PM LTAN SSO
Altitude	500 km
Revisits to target	3
Launch	Q4 2019
Launch (successor)	Q3 2020
Payload	
Type	Pushbroom Hyperspectral Imager
Spectral Range	400-800 nm
Spectral Resolution	5-10 nm
Operating Modes	High Res; Medium Res
Instantaneous optical resolution	250 m (Nadir)
Swath Width	70.32 km
Ground Sampling Distance	39 m
Raw SNR per frame @ 500 nm	$\approx$ 306 (1 mg/m^3 Chl-a)
S/C Bus	
Size	6U
Energy	54 Wh
Mass	$\approx$7 kg
Autonomy	
Data Processing	Onboard geometric, radiometric, spectral and spatial processing
Downlinked Data Products	Level 2 and 4 (Levels 0, 1a, 1b and 3 upon request)
Operations	Uplink and downlink to Ground Station, tasks determined by mission control
Communications	
Bands	S-band (downlink); UHF (uplink)
Ground Stations	NTNU (Trondheim), Longyearbyen (Svalbard)

HYPSO — Mission Overview

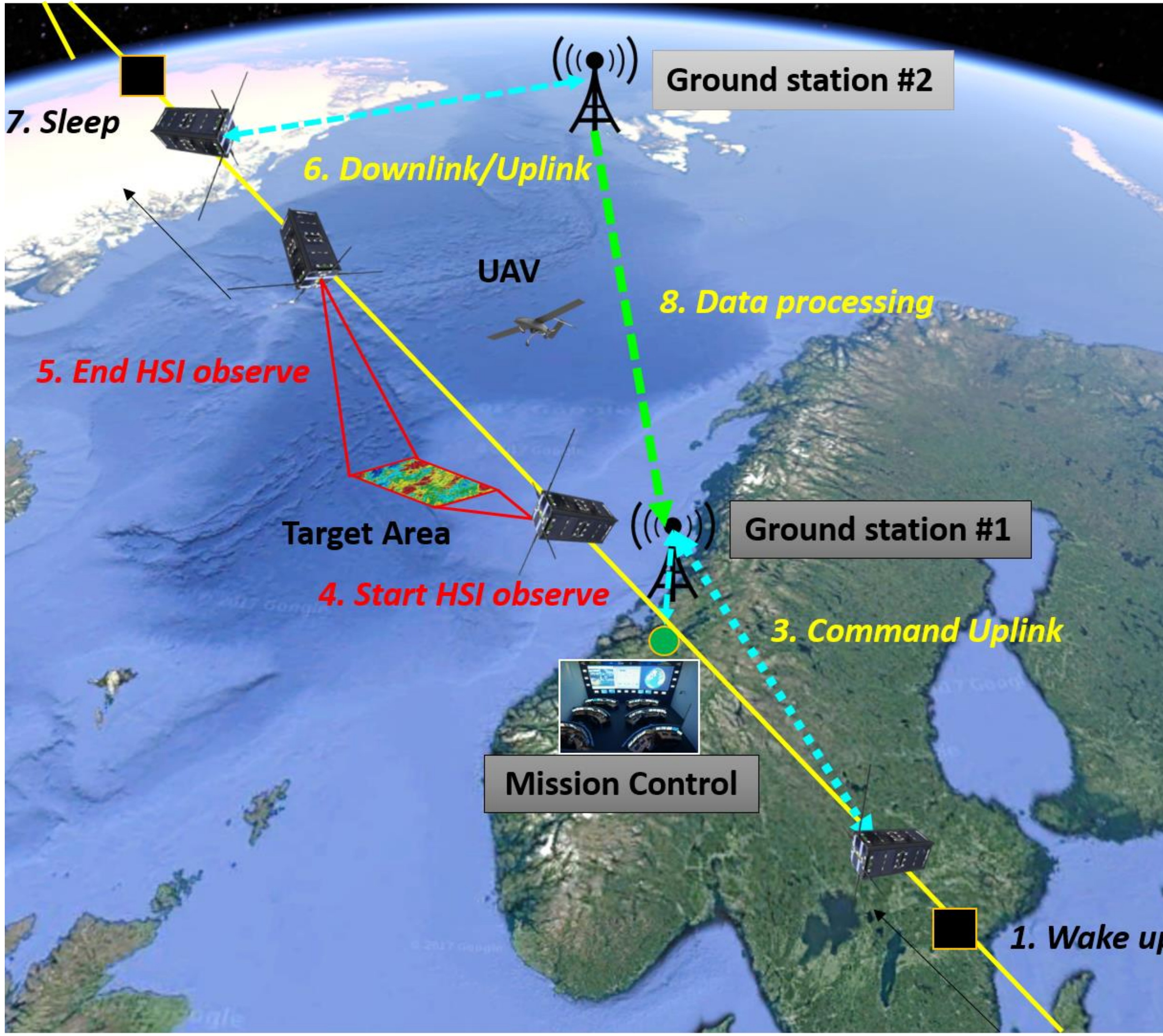


- L0: Raw data
- L1a: L0 + geometric variables + attitude data
- L1b: L1a processed to sensor units
- L2: L1b with overlapping pixels fused, geometrically and radiometrically processed
- L3: Image with variables mapped on uniform space-time grid scales
- L4: Radiometrically, geometrically, spectrally and spatially compressed data

https://science.nasa.gov/earth-science/earth-science-data/data-processing-levels-for-eosdis-data-products

HYPSO — mission phases

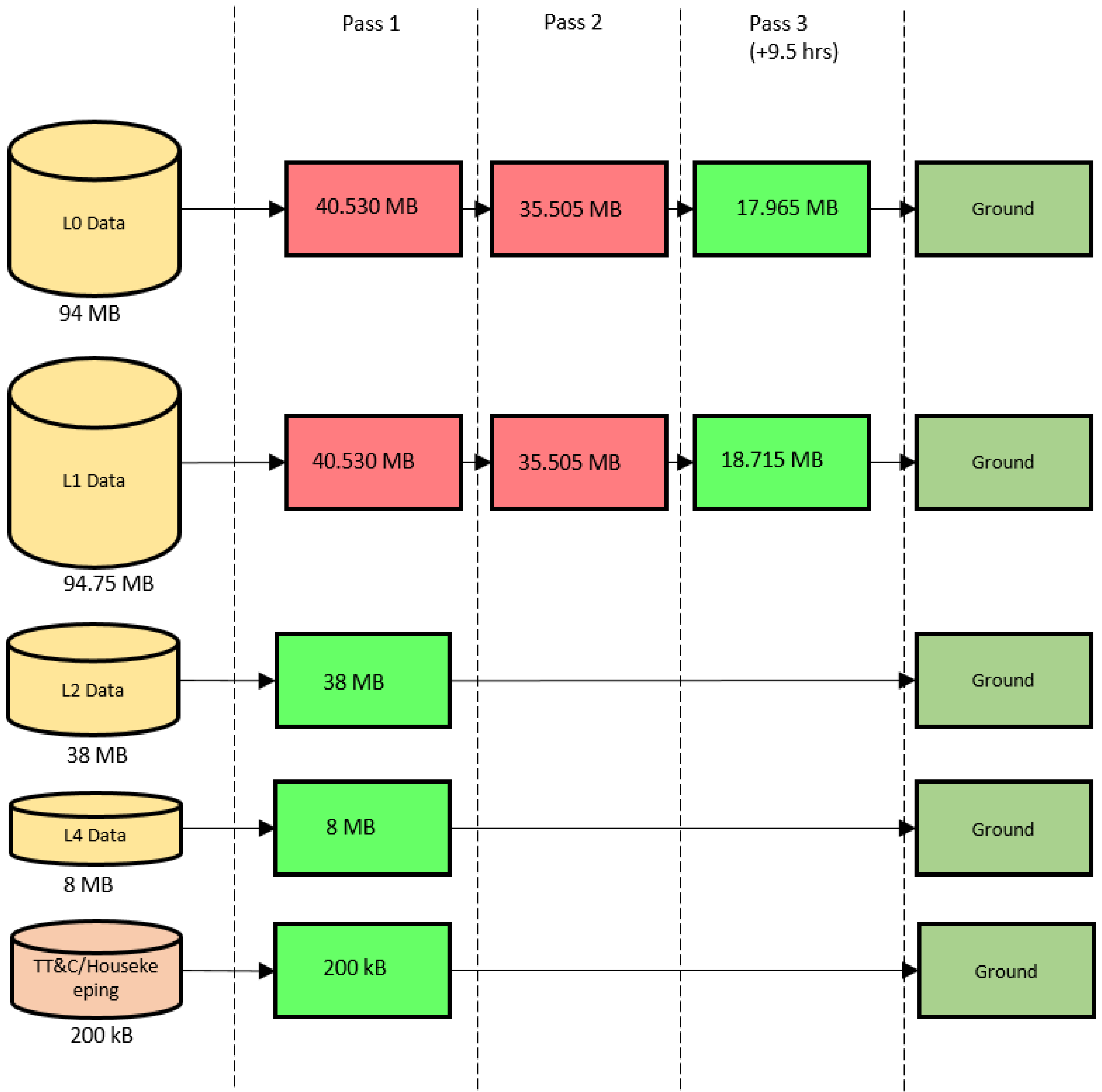
1. Initializing/waking up: HYPSO ready to receive data
2. Uplink: HYPSO receives mission from NTNU
3. Slew Maneuver: HYPSO prepares slew maneuver
4. Imaging: HYPSO images the target area
5. Data processing: HYPSO performs data processing
   - radiometric, spectral, spatial and geometric processing
   - reduced to spectral signatures (L4 data) or images correlated with in-situ measurements (L2 data)
6. Downlink: Data sent to Svalbard
   - distributed to Mission Control in Trondheim → further analysis and mission planning for next passes.
   - housekeeping data and TT&C will also be downlinked
7. Sleep/Idle: HYPSO goes to “sleep” and harvests solar energy



HYPSO — data budgets

- Assumptions:
  - Ground station antennas at NTNU and Svalbard have elevation angle of 10 degrees
  - S-band for downlink and UHF for uplink
  - Uplink, imaging, data processing and downlink shall happen in one pass over Norway
- Scheduling of downlink and uplink is crucial for mission to succeed in terms of near real-time data handling and response
- Uplink of mission planning data prior to imaging is feasible
- Uplink of FPGA programming logic (6 MB) requires two passes
- Downlinking TT&C, L4 data, L2 data subsequently after imaging (in one pass) is feasible
- L1 and L0 data can be downloaded after 3 passes
  - L0 data would only be necessary for optical characterization
  - L1 data would be nice-to-have
  - Urgent response time not required for these data types as can be analyzed in "normal" fashion

Downlink

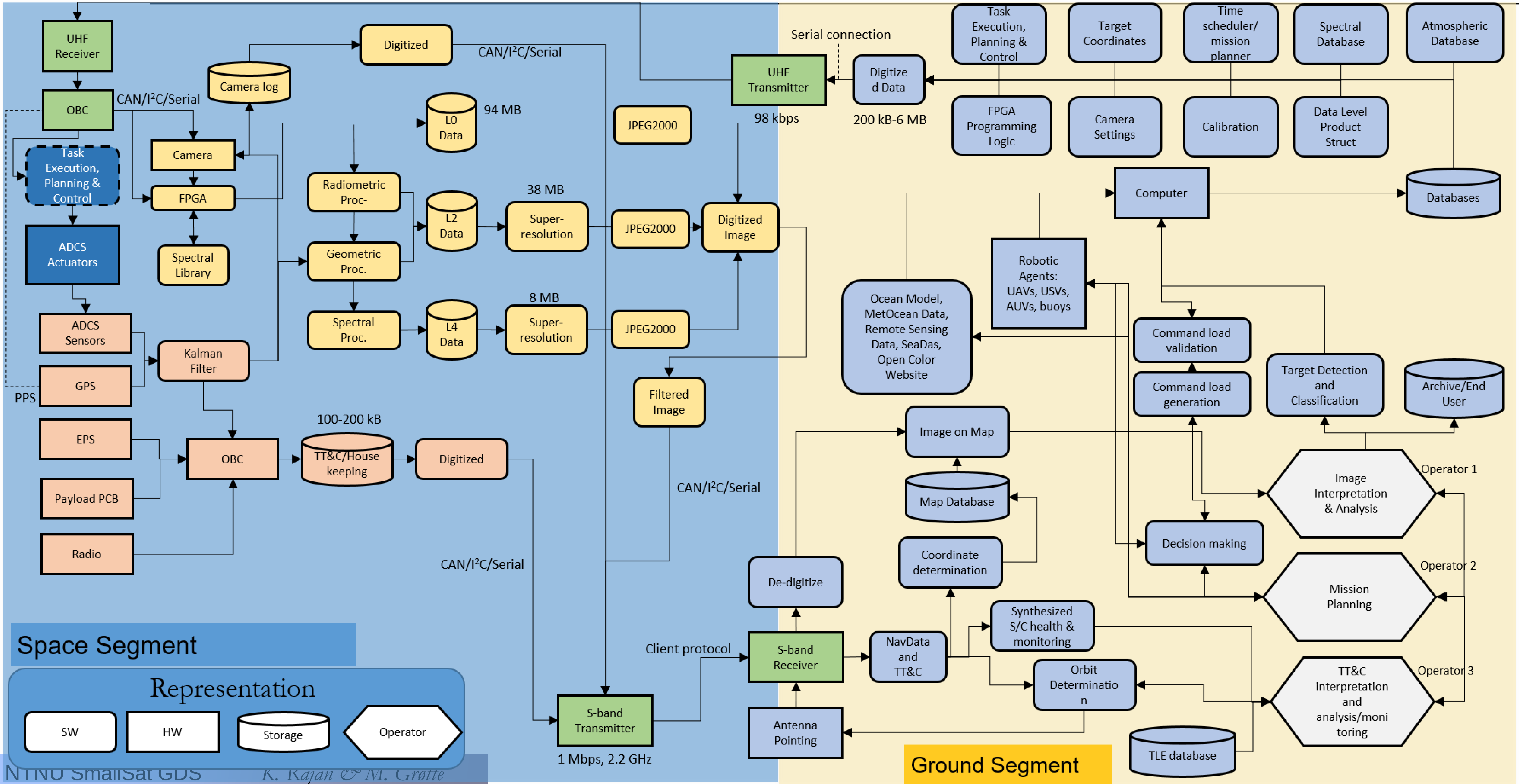


HYPSO — data budgets

- Assumptions:
  - Ground station antennas have elev. angle = 10 deg
  - S-band for downlink+UHF for uplink
  - Uplink, imaging, data processing and downlink shall all happen in one pass over Norway
- Scheduling of downlink and uplink is crucial for mission
- Uplink of mission planning data is feasible prior to imaging
- Uplink of FPGA programming logic requires two passes
- Downlinking TT&C, L4 data, L2 data subsequently after imaging is feasible
- L1 and L0 data can be downloaded after 3 passes
  - L0 data only necessary for optical characterization
  - L1 data is nice-to-have
  - Urgent response time not required for these types → can be analyzed in "normal" fashion

Target area to image	50×70 km^2
Imaging operations duration	1.94 min
1st access time to NTNU	5.129 min
2nd access time to NTNU	7.408 min
3rd access time to NTNU	3.374 min
1st access time to Svalbard	6.755 min
2nd access time to Svalbard	4.743 min
3rd access time to Svalbard (9.5 hrs after 2nd)	5.226 min
Uplink data rate	98 kbps
Downlink data rate	1 Mbps
L0 size	94 MB
L1 size	94.75 MB
L2 size	38 MB
L4 size	8 MB
TT&C size	200 kB
Uplink FPGA logic size	6 MB
Uplink mission plan size	200 kB
Downlink time L0	12.492 min
Downlink time L1	12.64 min
Downlink time L2	5.04 min
Downlink time L4	1.0584 min
Downlink time TT&C	1.6 s
Uplink FPGA logic time	8.16 min
Uplink mission plan time	16.33 s
# passes to downlink L0, L1	3
# passes to downlink L2	1
# passes to downlink L4	1
# passes to downlink TT&C	1
# passes to uplink FPGA logic	2
# passes to uplink mission plan	1

HYPSO GDS- big picture view



HYPSO GDS- big picture view

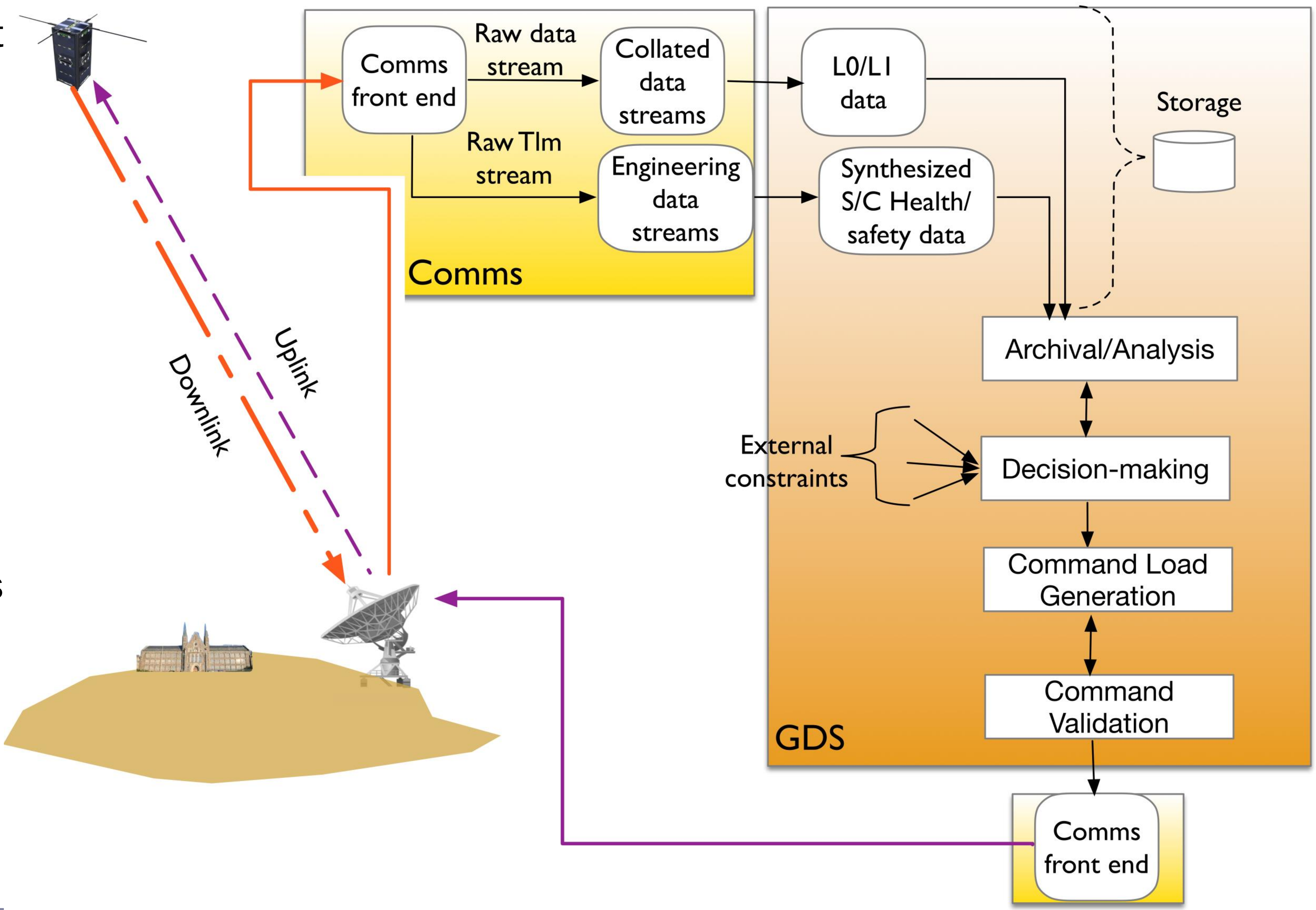
- Should be student led and operated — h/w, s/w and processes to operate/use both
- Simple processes, to ensure focus on continuous operation — ideally event driven
- Clarity in off-nominal conditions will require operator training to understand s/c health/status
- Use a mix of bus vendor provided s/w with APIs for building our own tools
  - Start with modest tools and build incrementally; leverage open source tools as much as possible
  - Tools should focus on cross-cutting functionality not just for HYPSO, but future (non optical) missions (e.g. for orbit visualization and pointing)
  - CLI is ok, but vendor/STK provided visualization will be critical
  - Automation is a desire, and should be limited to making sure health/safety of s/c is ensured
- Coordination with in-situ (oceanographic) assets should be with automated planning/scheduling methods already well established

Minimum tool needs

1. Tlm monitoring (health/safety)
2. Payload data processing/visualization
3. Planning tools
   - commanding for coordination
   - long-term Vs short term plans Orbital dynamics and pointing for playback (what happened) and command (what will likely happen)
4. Command load V&V

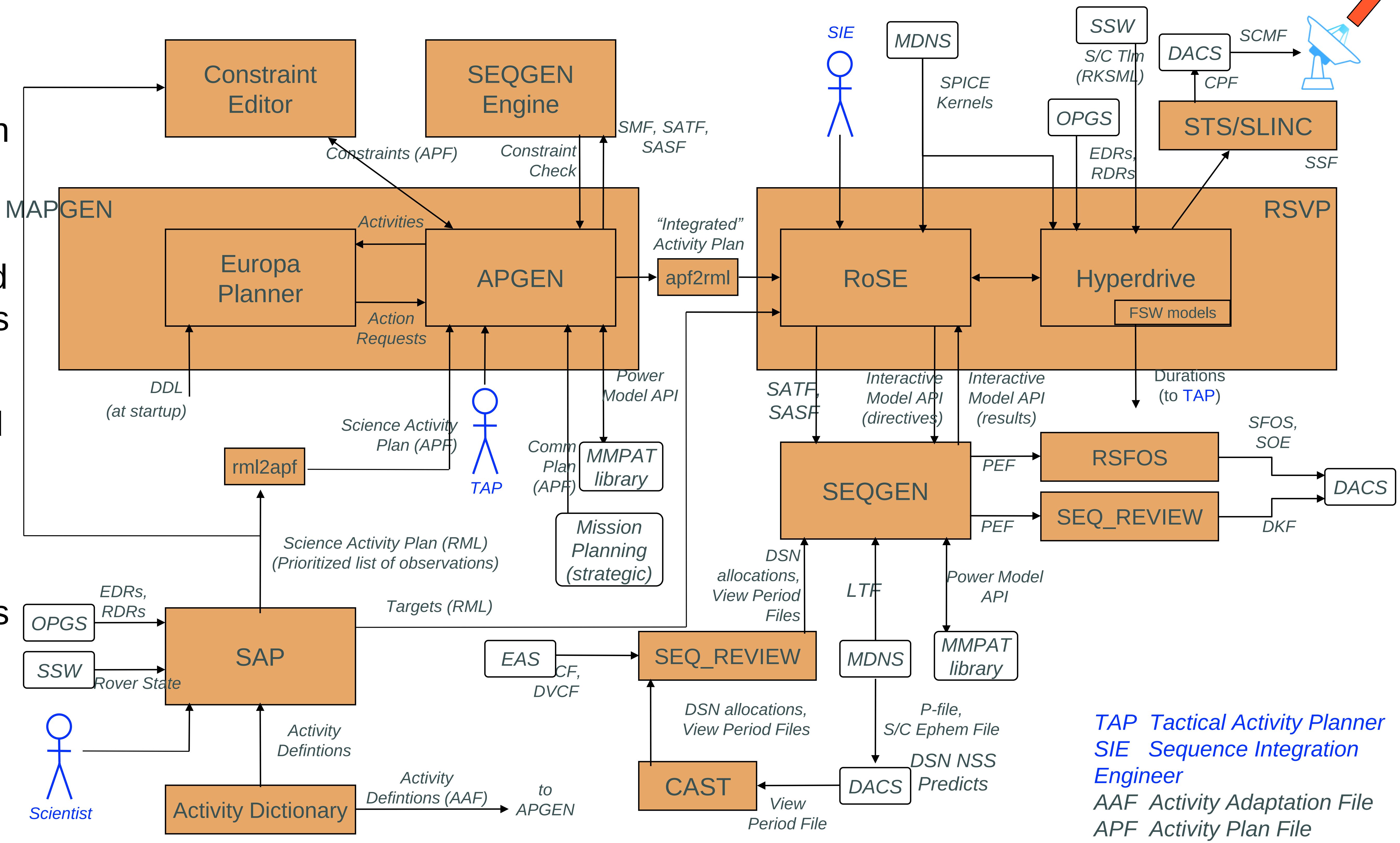
what is a GDS?

- GDS is not just h/w or s/w but the entire "process":
  - downlink+store
  - data (eng/science) retrieval
  - analysis
  - interaction for decision-making
  - uplink
- Complexity is both in analysis and the process of deciding "*what to do next*"
- This GDS is not a conventional definition, but good for MASSIVE
  - small size, student-led, simplicity



NASA Mars Exploration Rovers (MER) 2003 GDS

- How not-to design a GDS!
- Mix of processes with h/w+s/w explicitly detailed
- Note the process and sequence of activities are hidden
- Downlink is assumed
- Note also, that this is “Band-aided” and hacked together to make legacy systems work together.
- MAPGEN was “new” with an AI Planner hidden behind a clunky UI (APGEN)



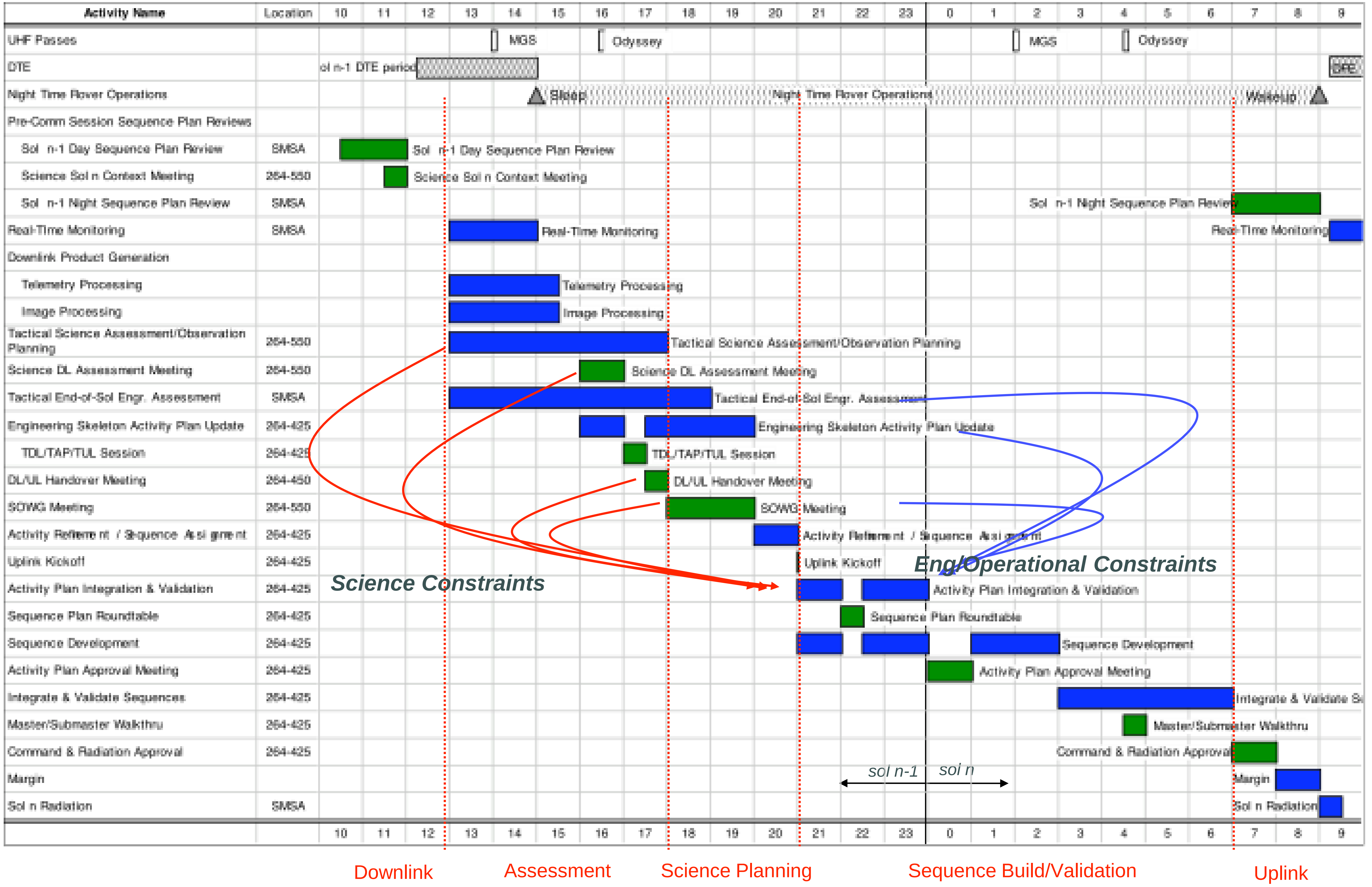
TAP Tactical Activity Planner
SIE Sequence Integration Engineer
AAF Activity Adaptation File
APF Activity Plan File
PEF Predicted Events File
RML Rover Markup Language

NTNU

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

NASA MER mission process with GDS

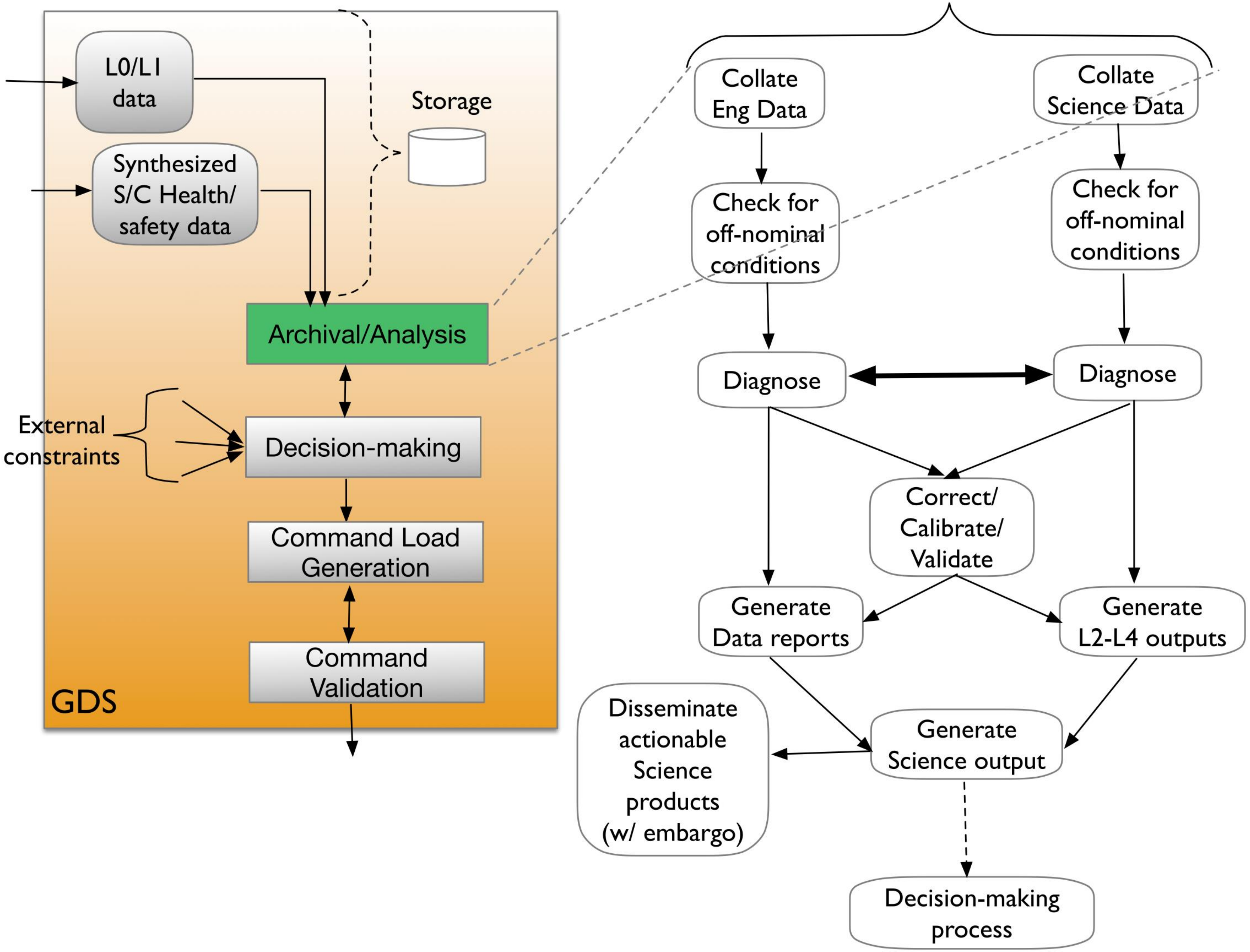
- Mix of processes with h/w+s/w hidden away
- Note the process and sequence of activities including meetings
- Note also the rover would “wake up” at 0900 and “sleep” 5 (Mars) hours later
  - So it was crucial to squeeze in as much time for planning in between



Courtesy: Jim Erickson, MER Mission Manager

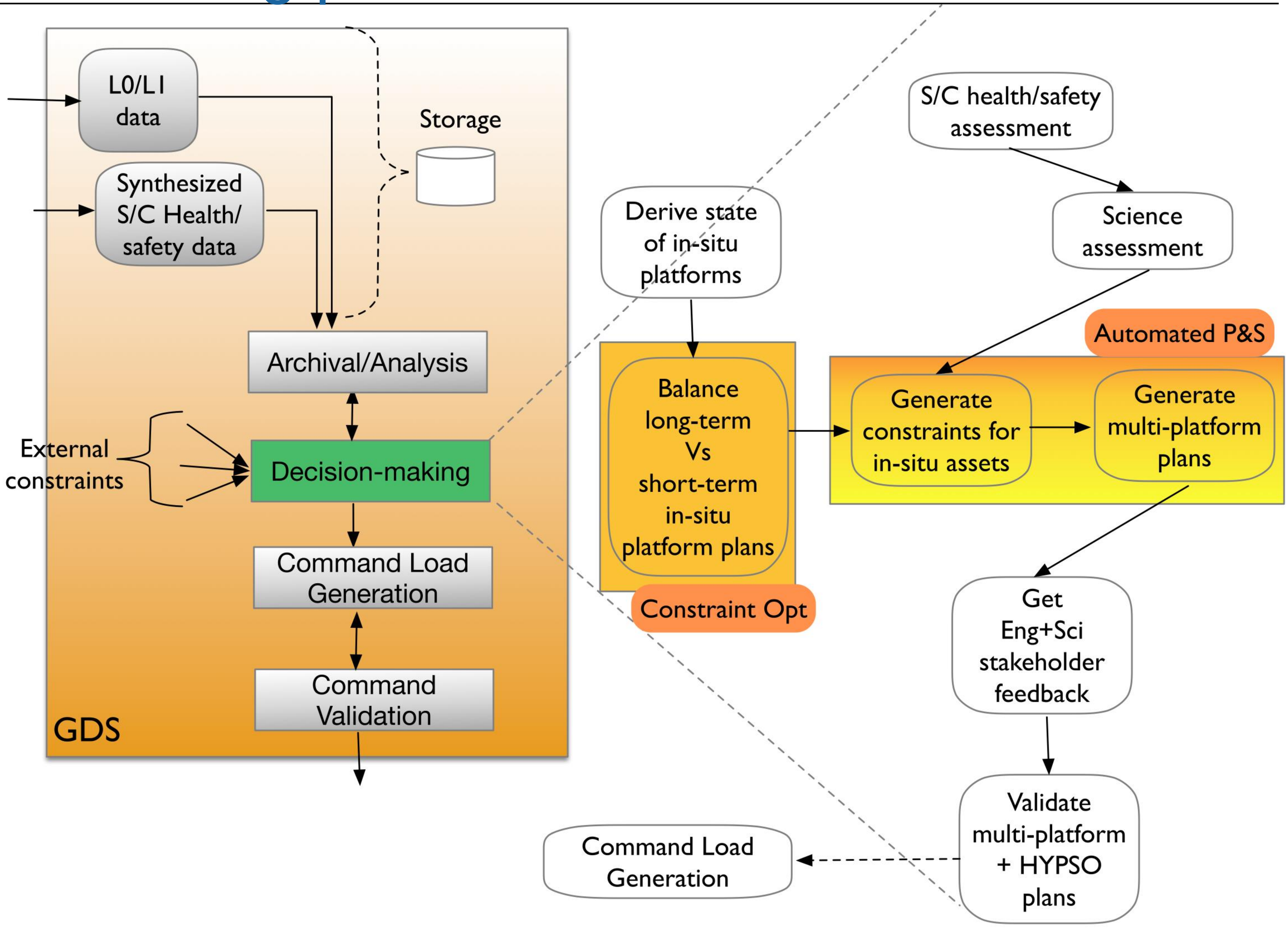
HYPSO archival/analysis events

- Eng data (telemetry) will likely come from different channels than science data
  - But will be needed to provide meta-data for science
- Comms related failures possible — leading to diagnosis and data QC and validation
- Science products as actionable items — either with manual targeting of vehicles, or for informational/policy management



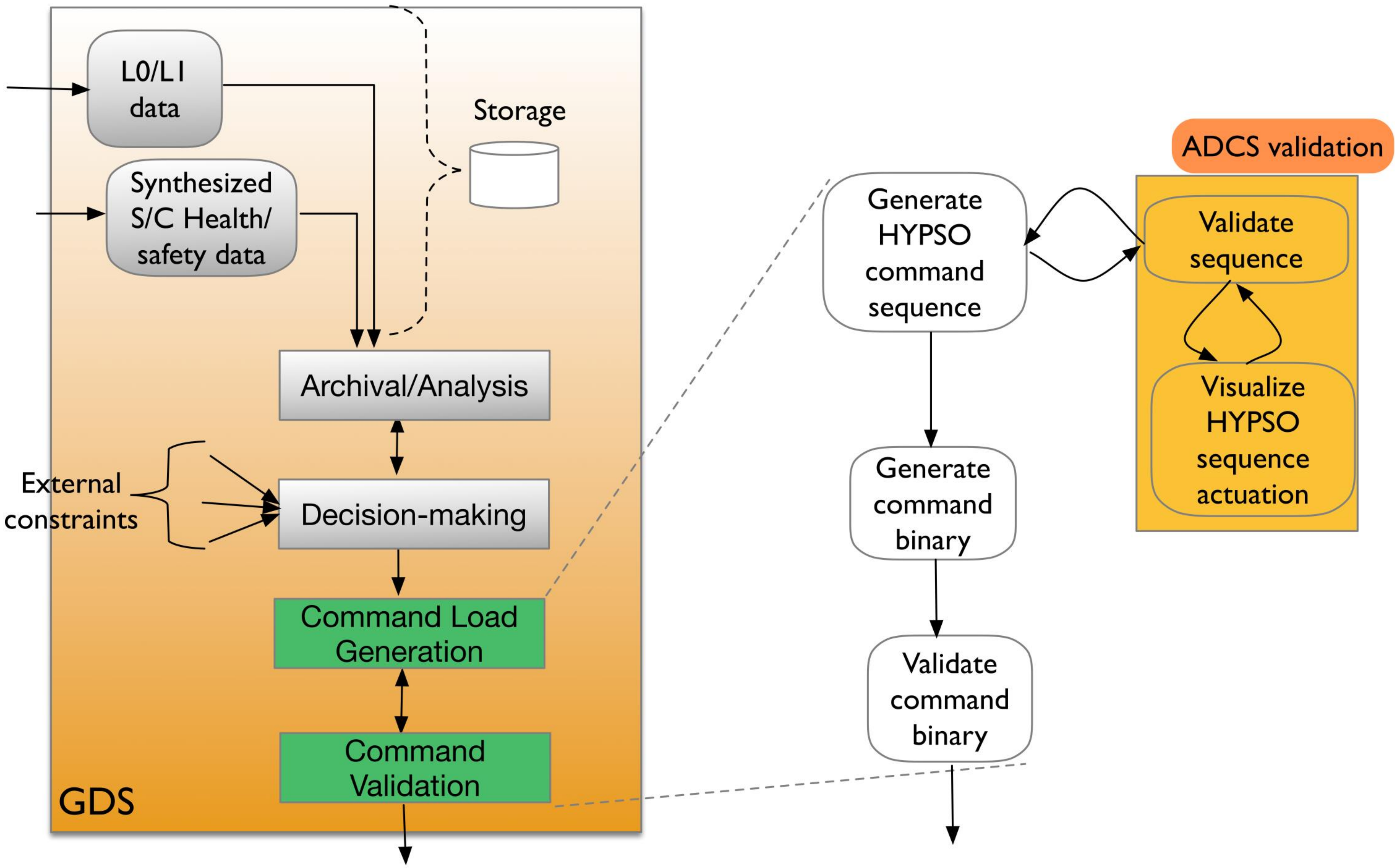
HYPSO decision-making process

- Monitored state of in-situ assets needs to be assessed
  - Needs and constraints of those in-situ assets quantified (e.g. related to operational constraints)
  - Trade between long-term objectives and short-term 'event response' (e.g. bloom detection)
- Use automated mixed-initiative approach to plan multi-platform plans for tasks
  - Factor resource constraints (e.g. onboard energy/data for all vehicles, proximity to hazards)



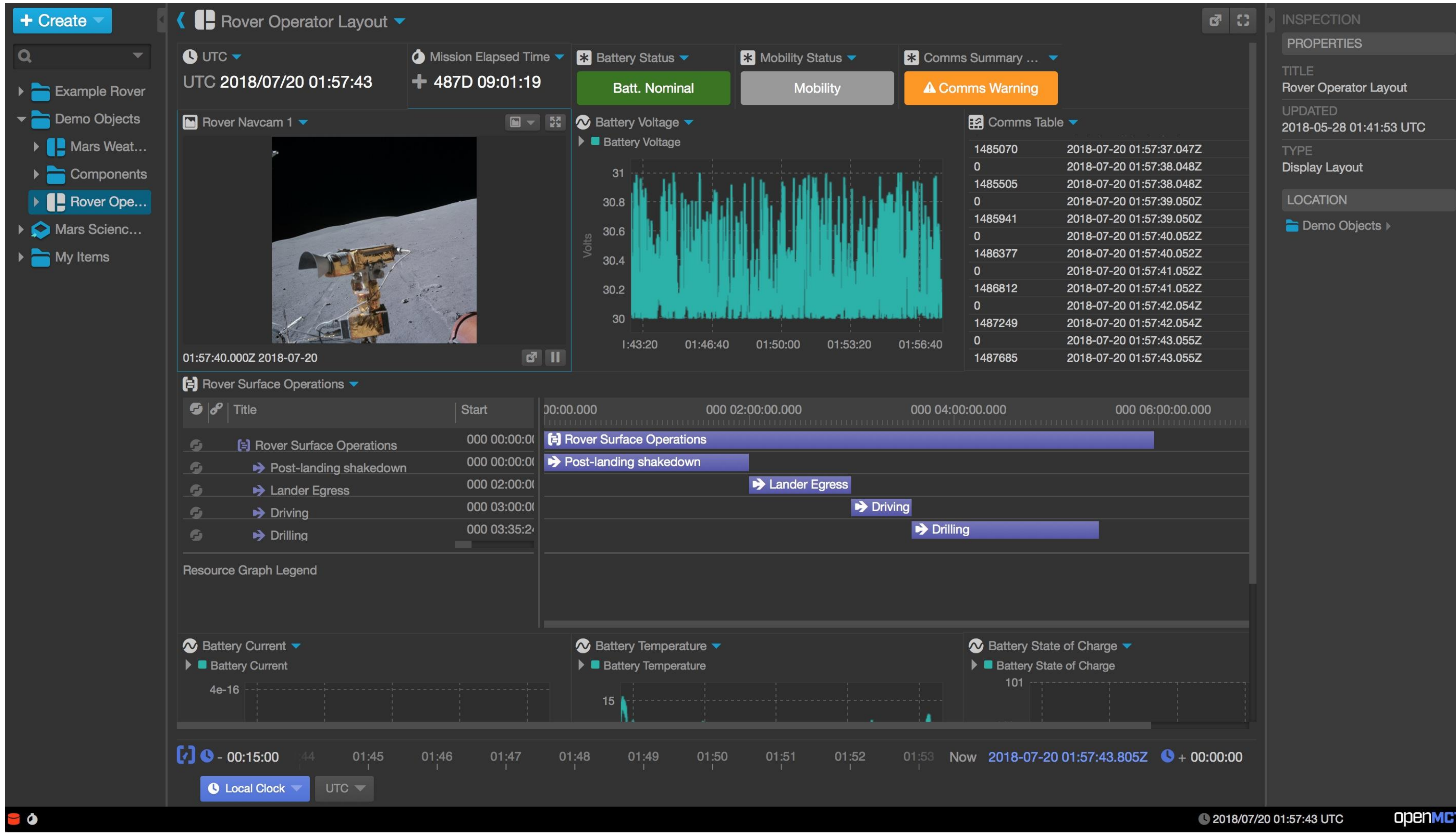
HYPSO command/load generation

- Visualization of HYPSO s/c orbital dynamics will be crucial
- V&V of orbital dynamics and s/c ops could require full expansion of command binary



HYPSO GDS- Tlm monitoring [OpenMCT]

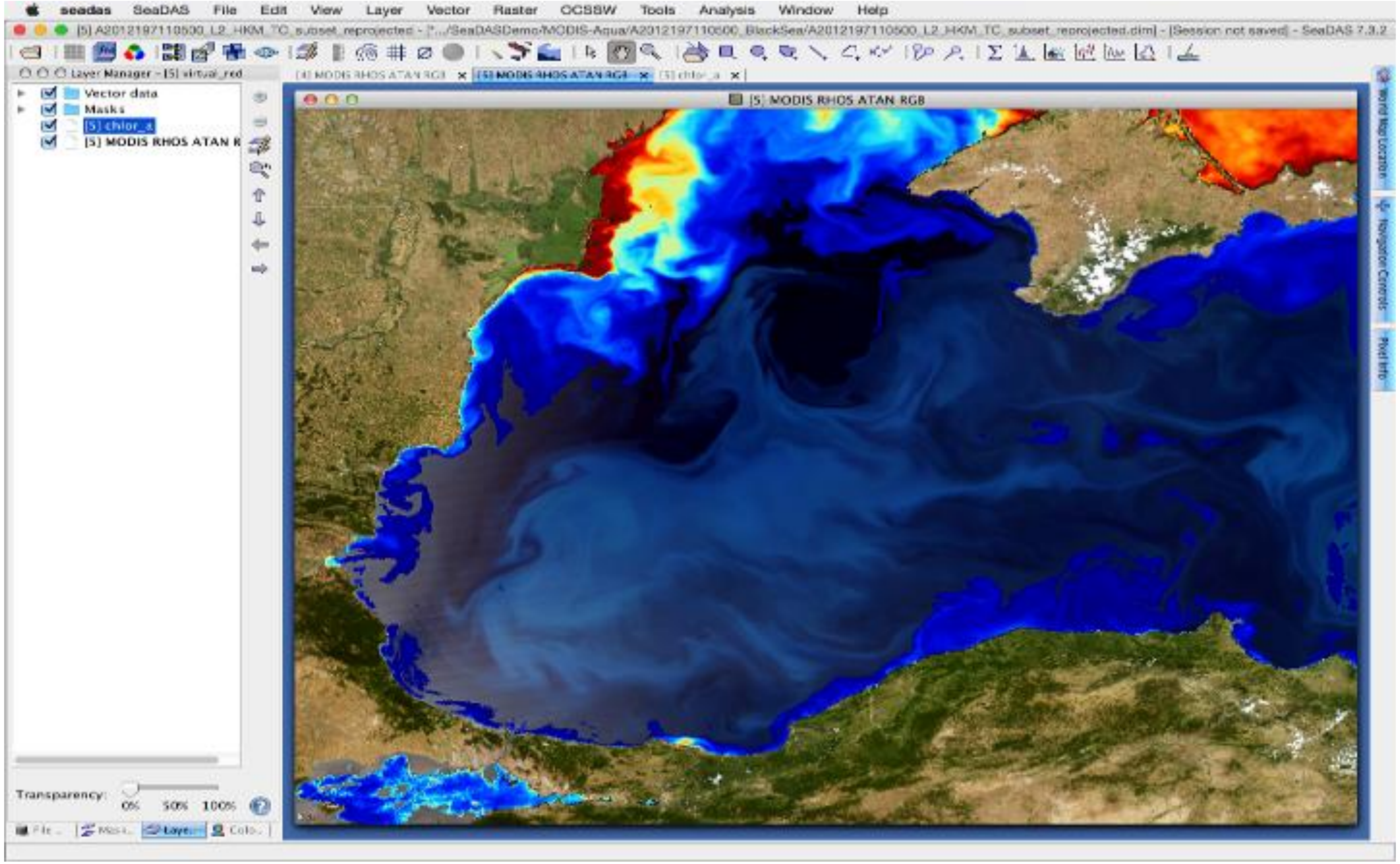
- *Only* for downlinked Tlm
- MCT should be able to take the Tlm stream from NA's bus and display
  - See NA Sec 4.1 of Software ICD for Tlm output data
- We can write meta-level code for anomalies in TLM for alarm generation, in addition to limits MCT can set
- Can couple s/c dynamics to a simulation of what the s/c did — playback of s/c activit(ies)



https://nasa.github.io/openmct/

HYPSO GDS- HSI data pipeline + visualization[SeaDAS]

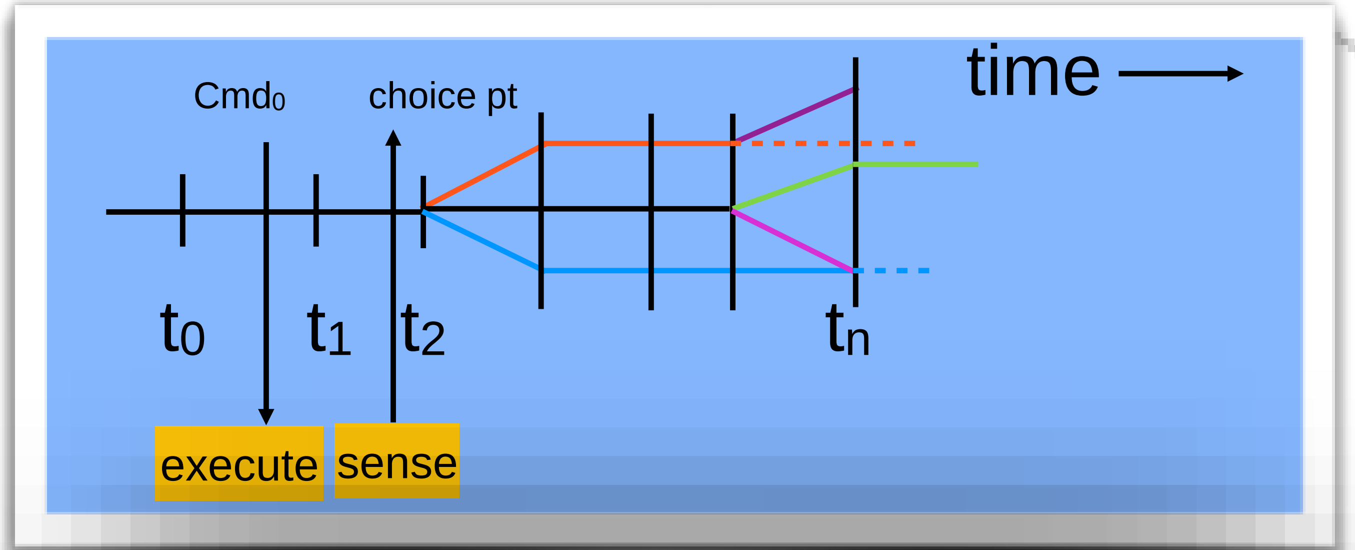
- SeaDAS is the default Ocean Color pipeline for NASA. Used for all agency OC missions.
- Both a processing pipeline + visualization/manipulation tool
- Connects directly to world-wide OC community
- Mariusz/Ajit in contact with primary SeaDAS person at GSFC



https://seadas.gsfc.nasa.gov/docs/SeaDAS_Walk_Through.pdf

Planning/Scheduling — what/why

- Command/control — making informed decisions about a robot's actions, given existing and anticipated environmental conditions and goals
- In static and simple environments, this is a reasonable way to execute commands
  - In dynamic environments within a complex set of interacting variables, this approach breaks down
  - Often dependent on human input/vagaries of commanding — i.e. non systematic
  - Brittle in the context of failure in real-world conditions
- "Planning" allows for:
  - projection of state while simultaneously ensuring response(s) to dynamism in the environment via sensing
  - anticipatory view and resource projection
  - systematicity in autonomous control and decision-making
  - ability to remove human input/errors



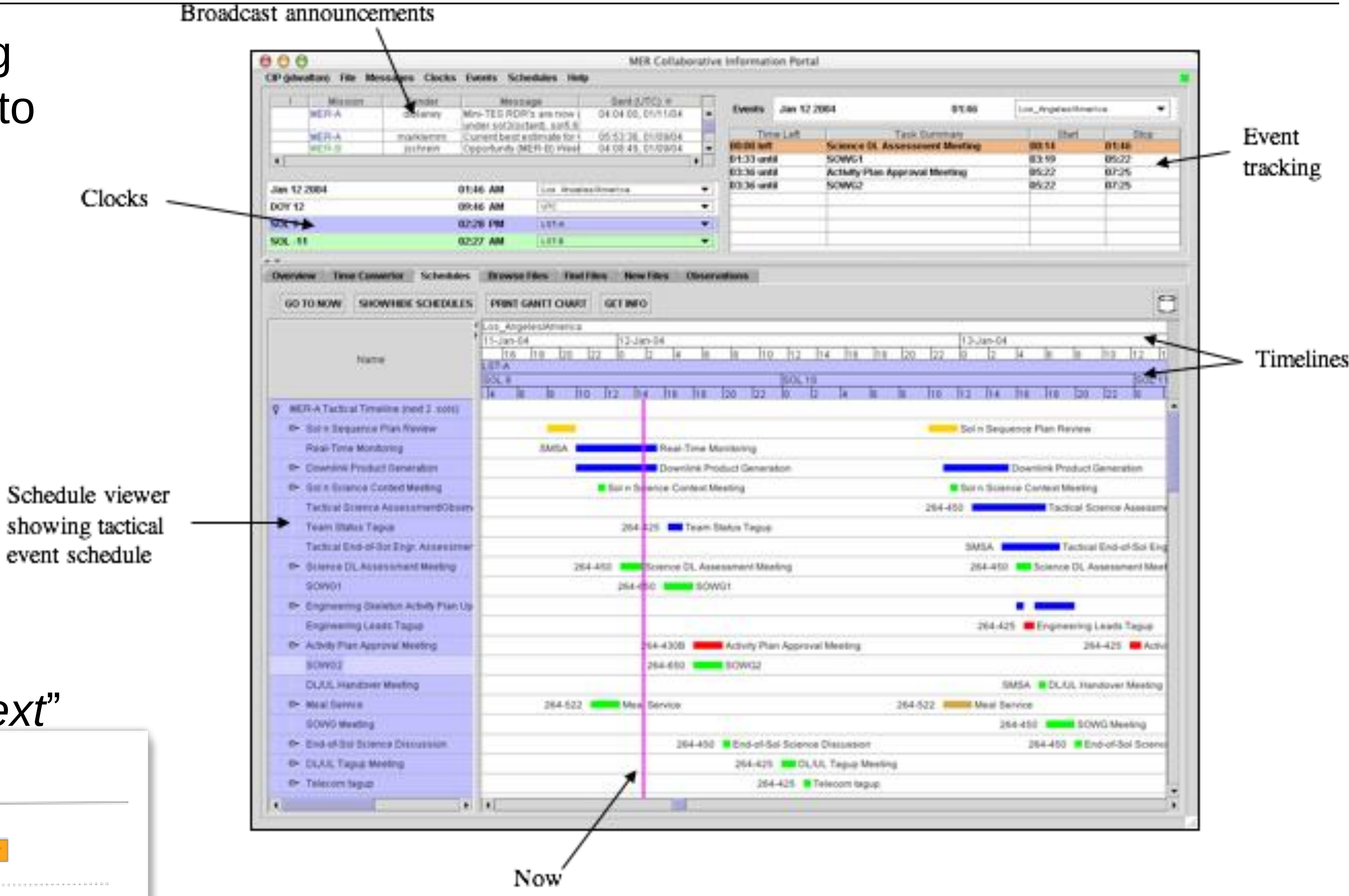
- At fixed time(s) execute a command/script
- Decide based on sensing (or state) input
- Pursue conditional execution branches based on this input
- Repeat

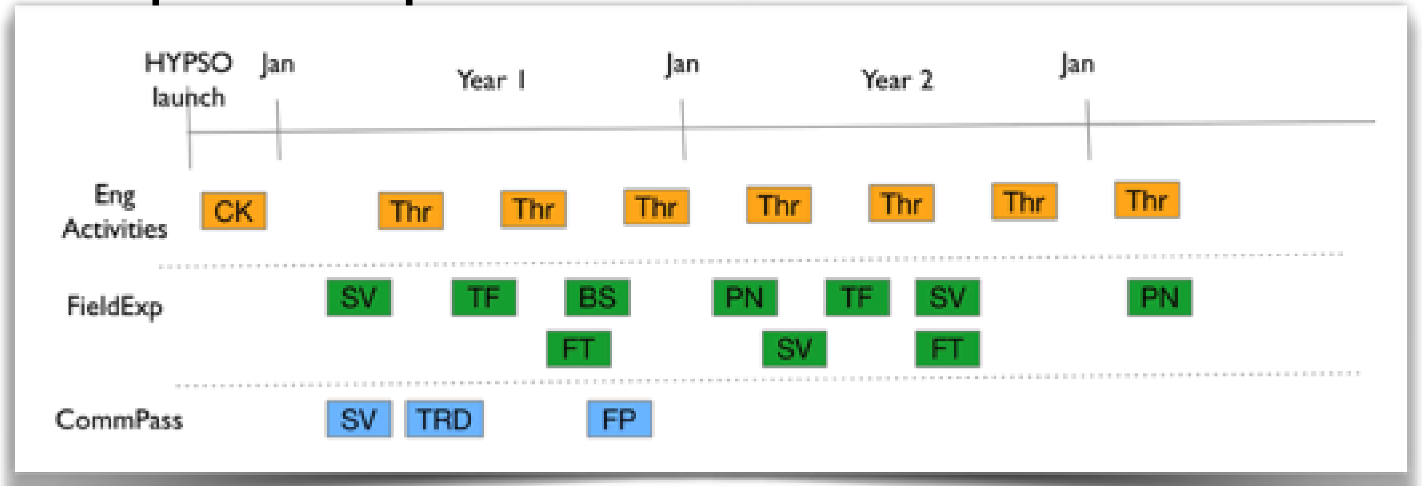
Traditional View of commanding

- "Scheduling" is a way to measure effectiveness of a plan given available resources (e.g. battery SOC, data volume, bandwidth etc)
- Together they can:
  - execute commands in a flexible environment
  - optimize time/resources
  - be robust to failures
  - systematically execute repetitive tasks

HYPSO GDS- Planning/Scheduling

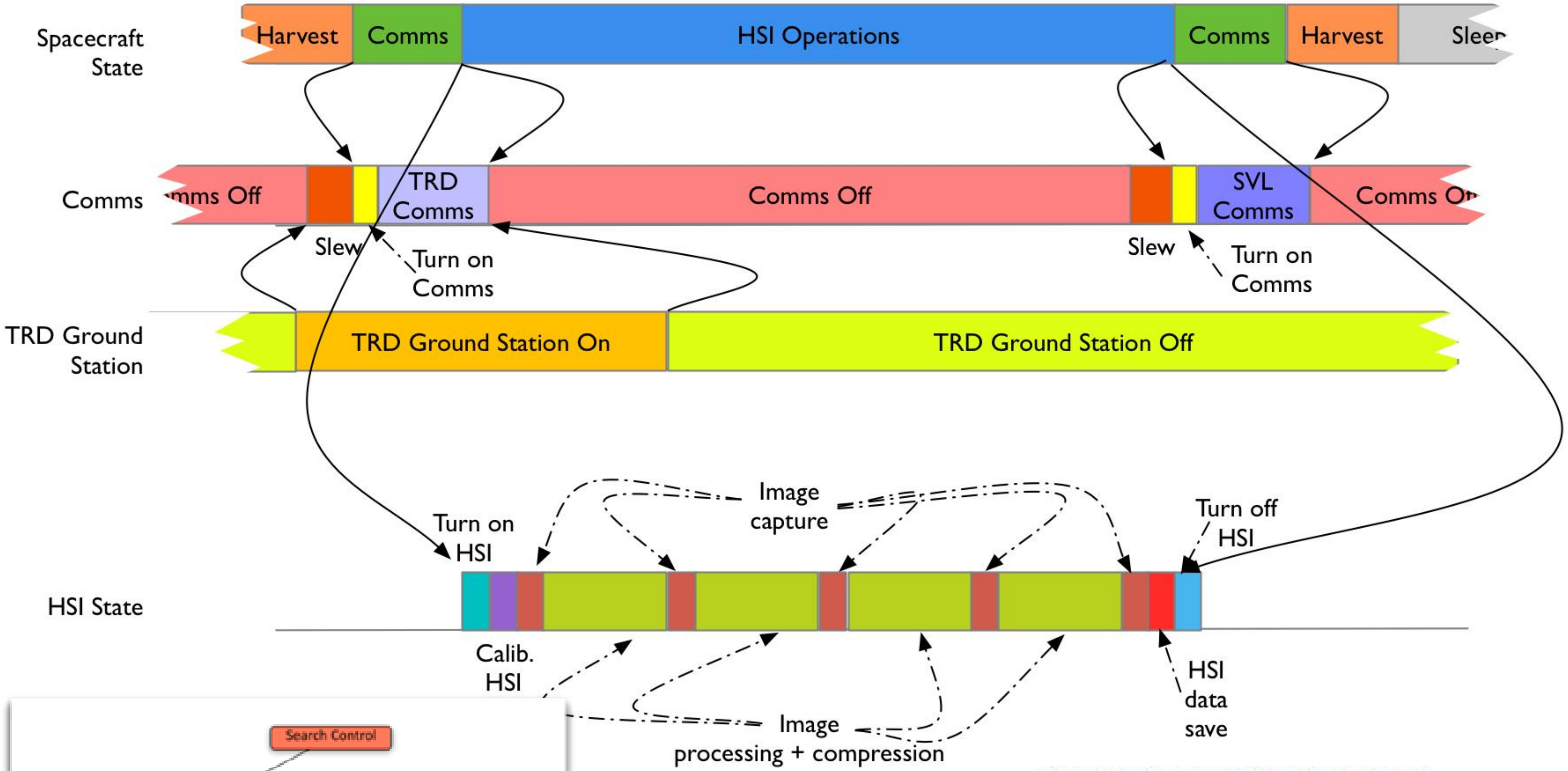
- Long-term Vs Short term planning can be simpler — a spreadsheet to start with!
- Tools such MER Collaborative Information Portal (MER CIP) provided cross-platform event information
- Simple time-line representation (below) might be adequate
  - CIP has this visual
- In part responds to "*what to do next*"

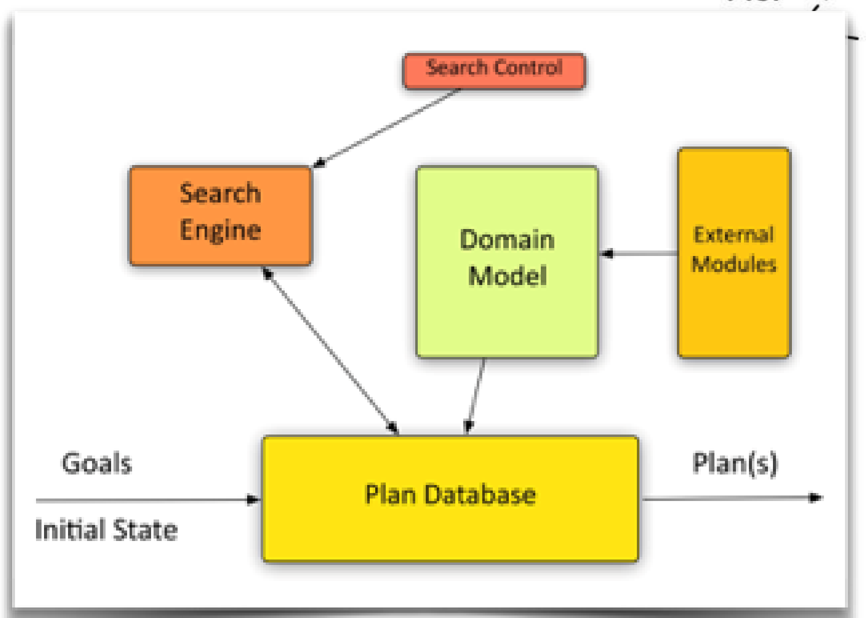




HYPSO GDS — Planning/Scheduling principles

- Partial plan of onboard activities using constraints over known state variables
- Allows flexibility in tasking the s/c with goals *from the ground*
  - GDS Planner interacts with onboard planner
- Planning onboard allows the s/c to be “self driven” and responsive to unanticipated events
- Temporal constraints remove notion of rigid script based tasking
  - key is the ‘domain model’ which not only has an ‘activity dictionary’, but also codified inter-relationships





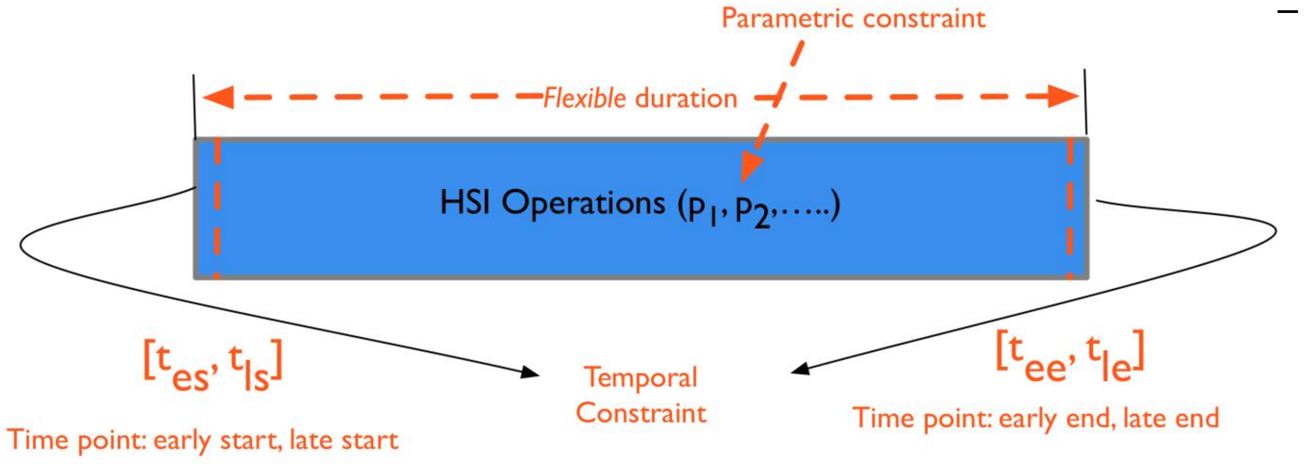
MISSION PHASES IN ORBIT 1 CONCEPT

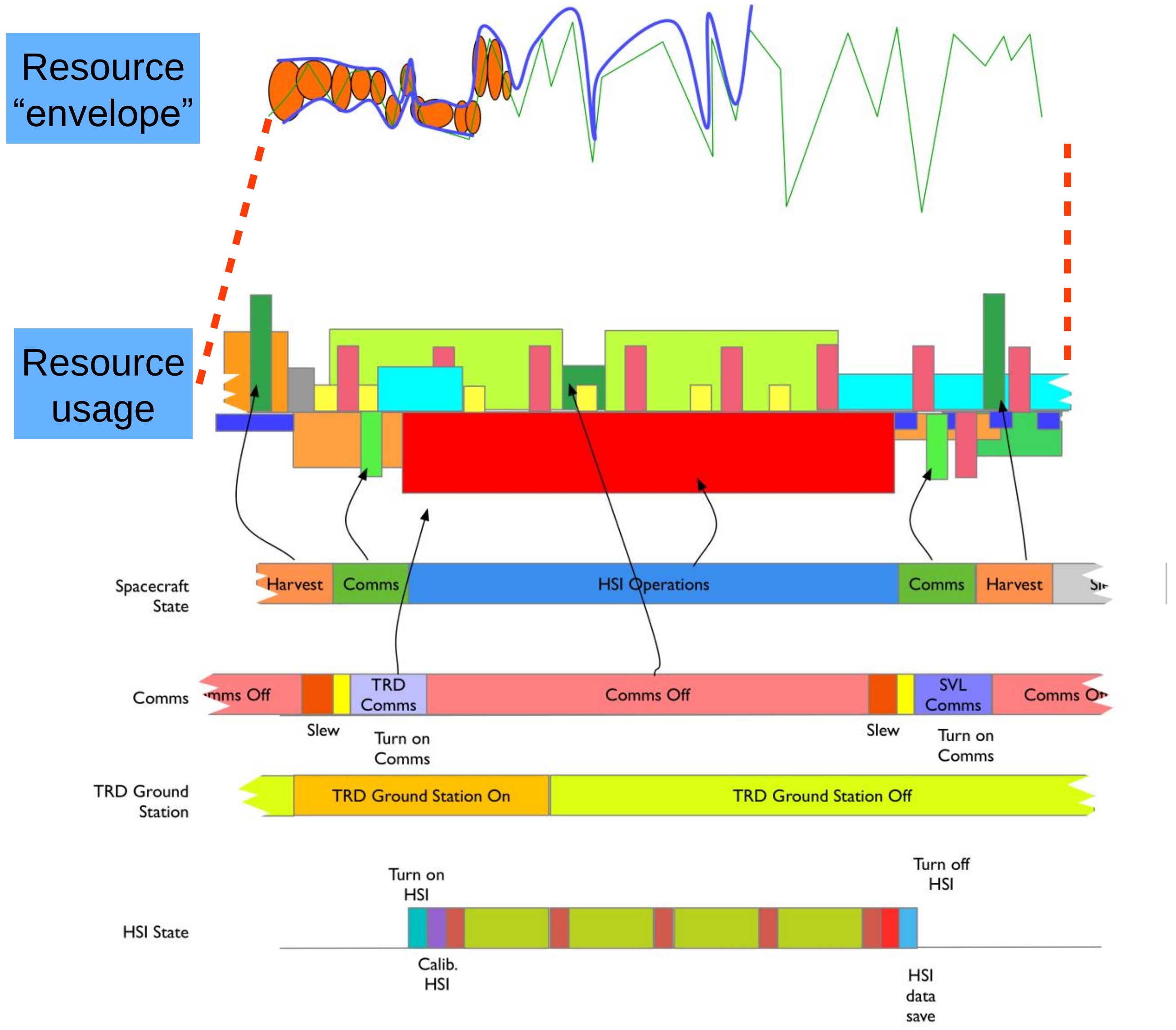
Segment	Description	Start (UTC)	Duration (s)
Phase 1	Harvest	09:37:10	5
Phase 2	Comms. Trondheim	09:37:15	125
Phase 3	Prepare slewing	09:39:20	115
Phase 4	HSI operations	09:41:15	54
Phase 5	Data processing	09:42:09	74
Phase 6	Point to Svalbard	09:43:25	20
Phase 7	Comms. Svalbard	09:43:45	270
Phase 8	Harvest	09:48:15	605
Phase 9	Sleep	09:59:20	2207
Phase 10	Harvest	10:36:07	2255
N+1	Next target	11:13:42	373

example based on Mariusz’s orbital concept

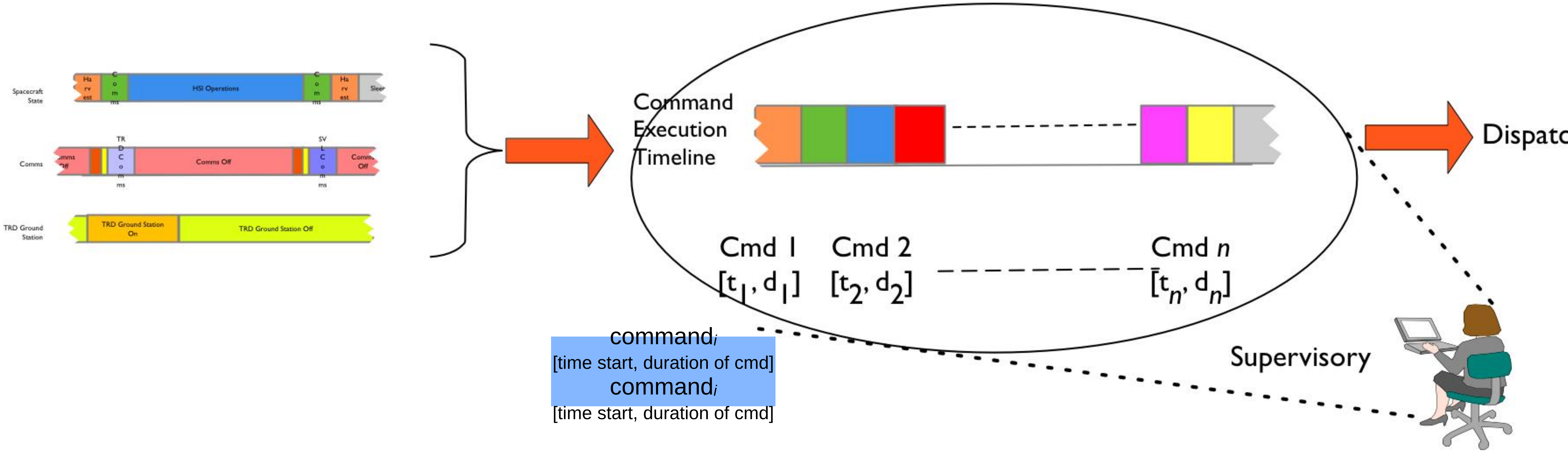
HYPSO GDS — Planning/Scheduling principles

- Associated with every flexible temporal plan is a ‘token’ or a block where a specific script driven by an activity is executed.
  - Flexibility allows for discrepancies between expected and actual execution times including diagnosis and recovery from off-nominal conditions
- Flexibility in plans makes it more robust to perturbations between plan time and execution time, but complicates the math for propagation of current state — therefore use *Allen* (interval) *Algebra*
- Tasks contribute (+) or subtract (-) to available resource (e.g. battery SOC) — flow computation over time, allows summing up non-linearities in how resources are consumed providing an “envelope” of upper and lower bound
  - useful to understand the fragility/robustness of a partial plan
  - therefore allows insertion of activities to mitigate situations of concern instead of leading to “fail nominally” rather than “fail safe”



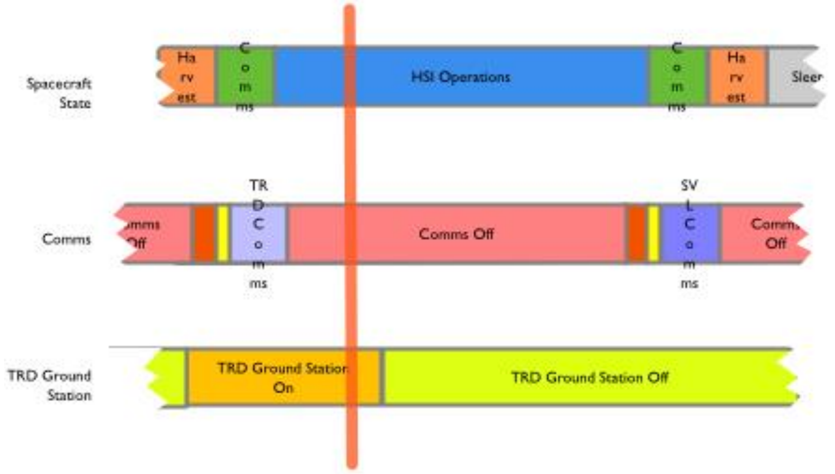


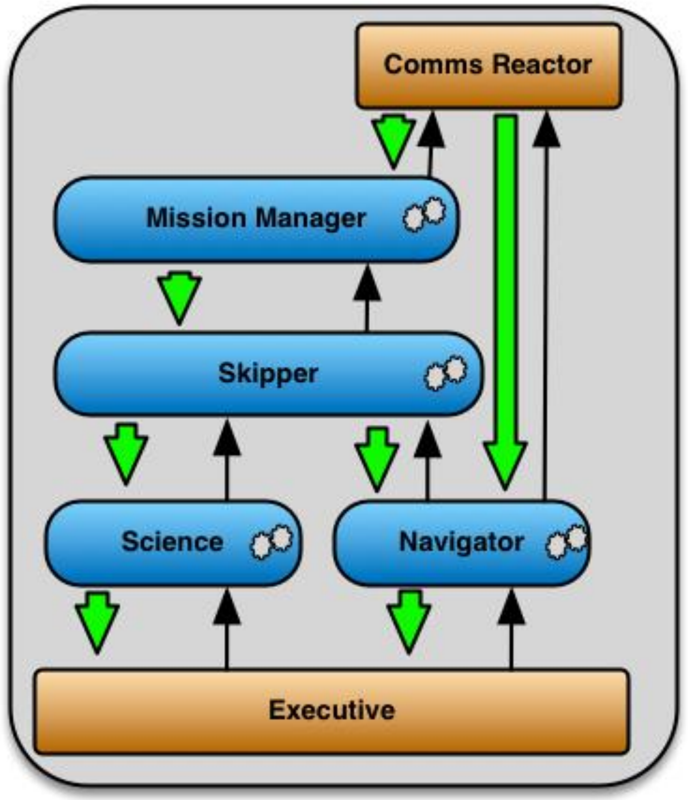
HYPSO GDS — Planning/Scheduling methods -I



- Command load generated on the ground by a simple script + human-in-the-loop → Mixed-initiative
- Execution is a simple sequencer on board, which executes Command *n* at time t_n for duration d_n
- Standard approach to all s/c commanding
  - Simple, verifiable with fixed expectation of s/c behavior
  - No feedback, not responsive to dynamic environment — ok for HYPSO

HYPSO GDS — Planning/Scheduling methods -II





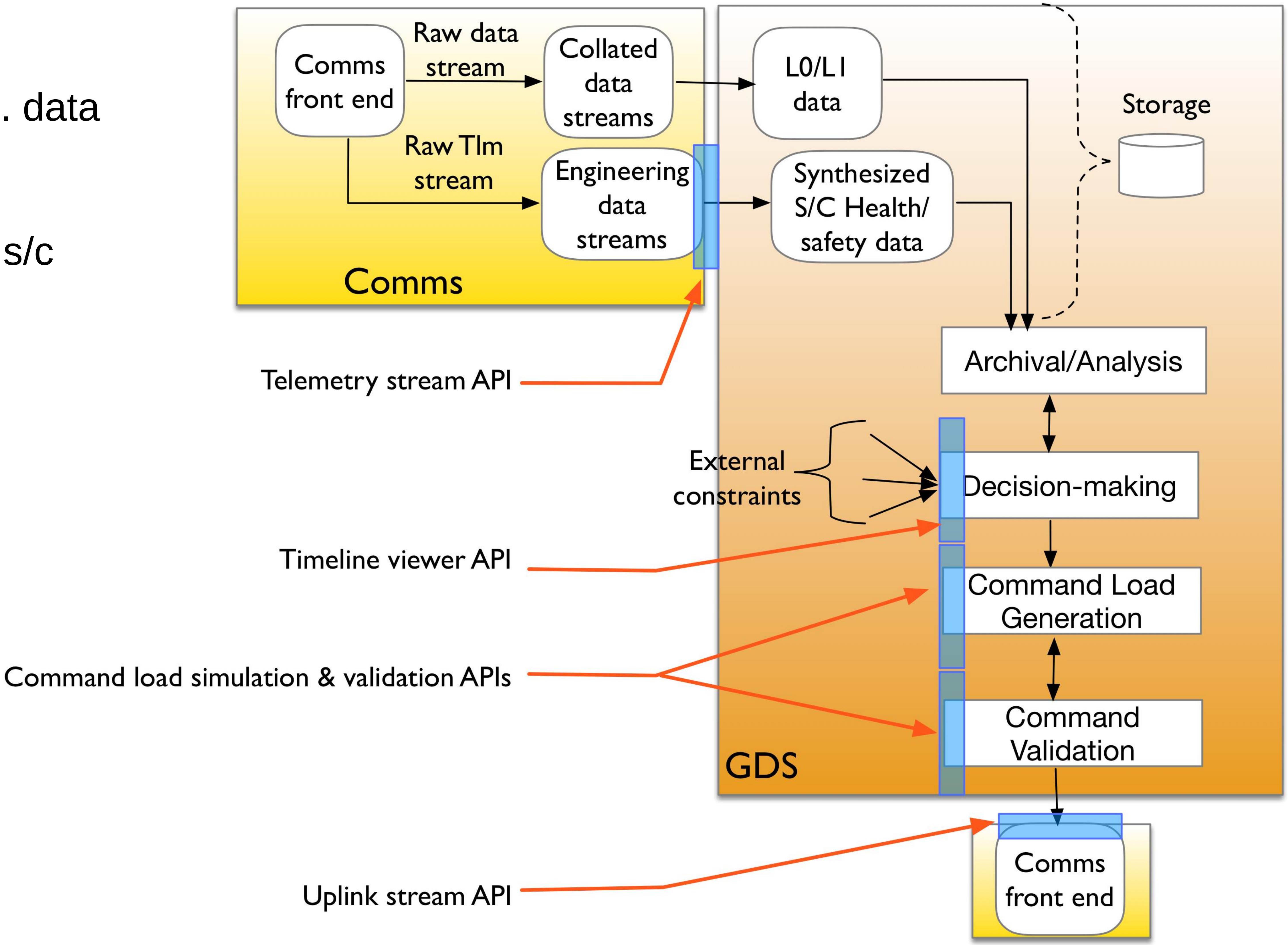
Automated execution engine

Dispatch

- More advanced and automated with an onboard “smart” executive (e.g. T-REX)
- Closes control-loops onboard with goals driven possibly from GDS and/or self-generated
  - Suitable for more dynamic or complex interactions with the environment
  - More complexity in model design, since *execution* is factored into *planning*
  - Unnecessary for HYPSO — possible as an “experiment” for extended mission

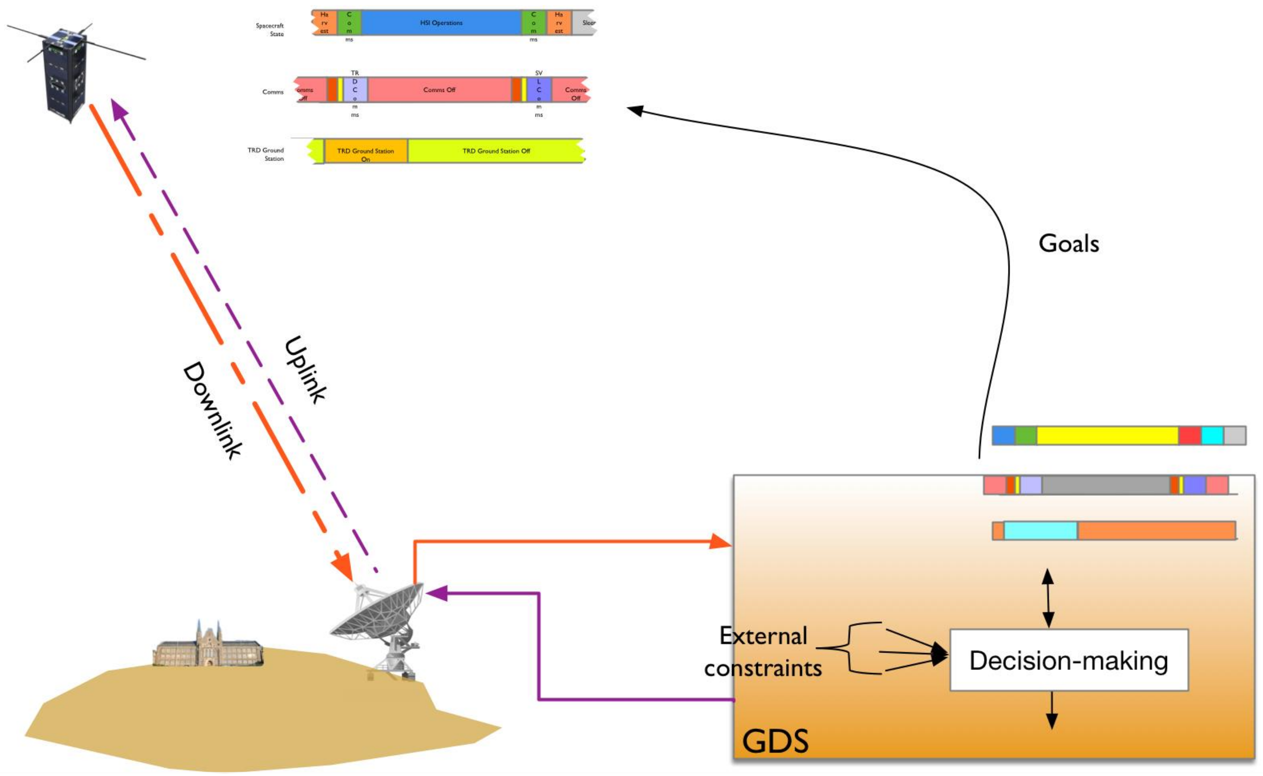
HYPSO interfaces to vendor

- API's needed to vendor s/w + h/w
  - downlink Tlm stream for s/c eng. data
  - uplink command stream
  - potential use of timeline view of s/c commands
  - any tools for command load simulation & V&V

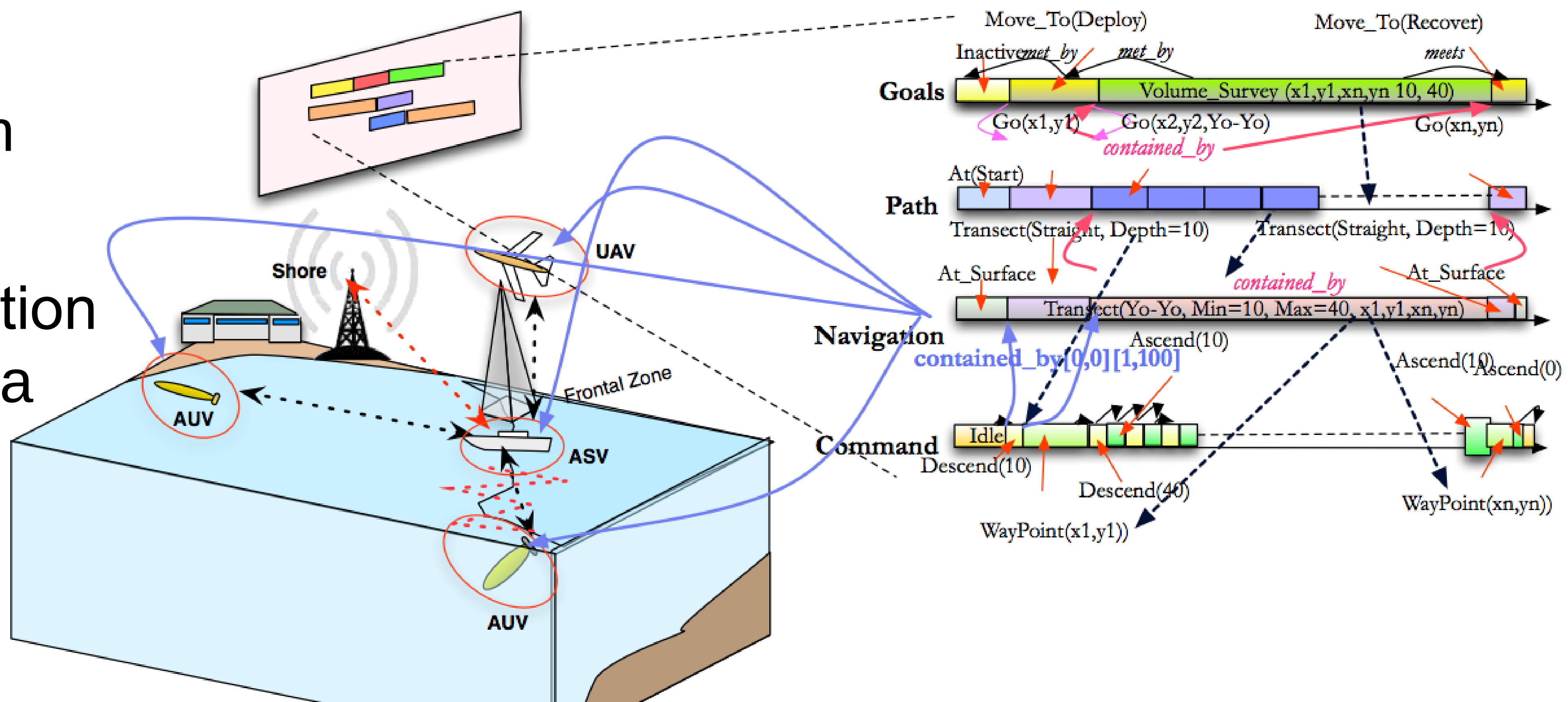


HYPSO GDS/Onboard- Planning/Scheduling

- Coupling GDS planning with onboard HYPSO planning/scheduling a <u>*stretch*</u> goal
- However with limited resources and not enough time/in-house capability, safe to use only for GDS
- Concept in images in this viewgraph
- Adds complexity
    - Nice to have? Yes
    - Do we need it? No
        - Only when a large number of s/c in orbit
- Alberto's focus will be on this coordination planning in GDS with simple tasking via LSTS toolchain

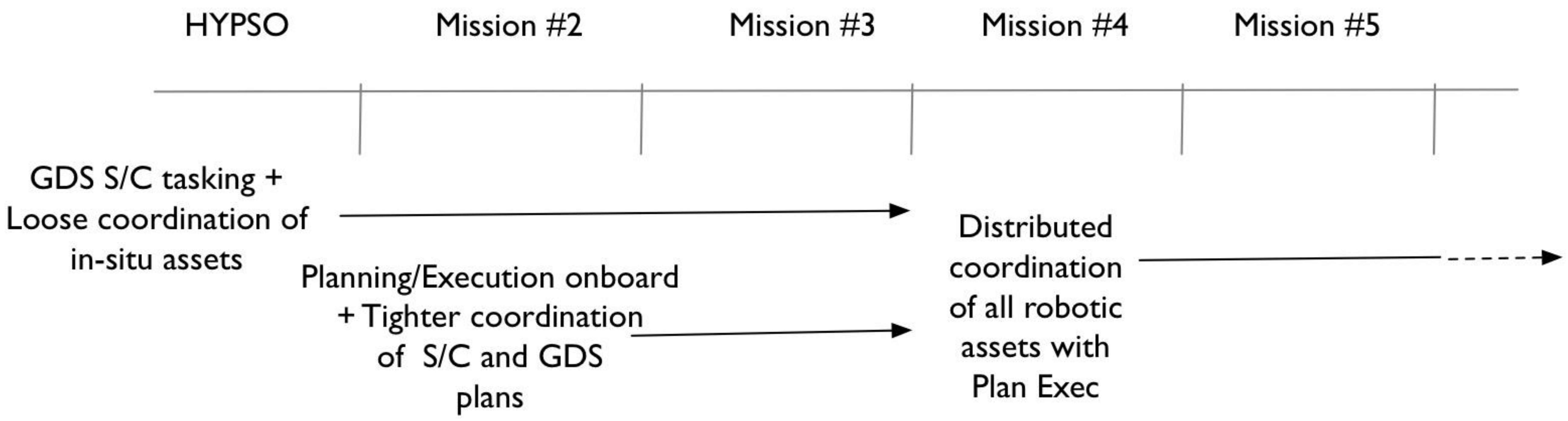


Constraint-based Multi-vehicle Planning



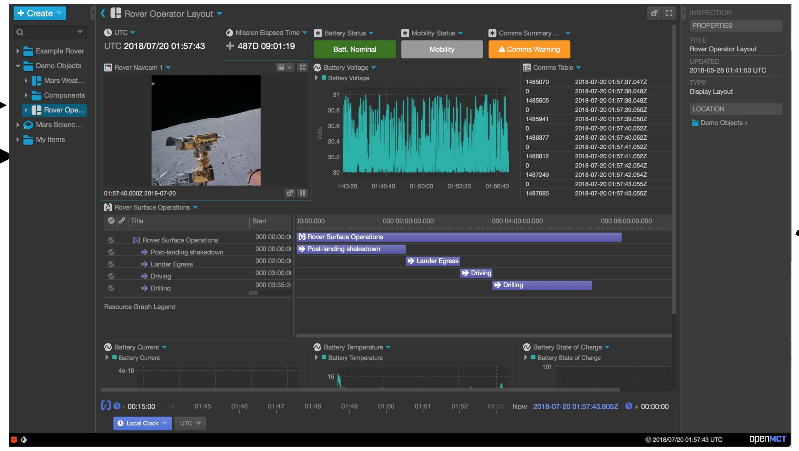
HYPSO Planning/Scheduling

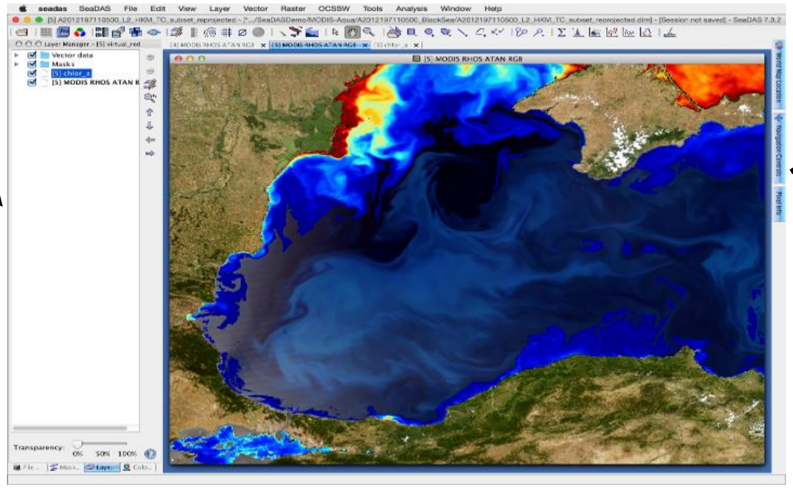
- Variations of NASA's open-source EUROPA2 planner with its constraint engine + resource computational engine flown onboard and used for MER (MAPGEN)
- Viable for use with some front end (TBD) which can visualize plans — MER tools can potentially be leveraged
- CBP (Constraint-based Planning) is the most practical way to plan and schedule
- Scheduling is critical to ensure viable resource usage onboard and on the ground
  - For HYPSO, recommend we focus planning only in GDS — subsequent missions we move towards onboard s/c



GDS tools — a simplistic view

Ground Station h/w





Planning/Scheduling tools

Decision-making user-interface

Command V&V tools

- We focus on just the essentials and integrate with what is available
- Automated P&S should play a role — user-interface will need to be leveraged
- Use STK and others for V&V of command-load validation, unless vendor provisioned