National Aeronautics and Space Administration

NASA

VIPER

VIPER Lunar Rover and the Exploration of the Moon

Izmir

Jay Trimble

MARCH 14, 2023

The Mission

The New Moon….

Not that long ago, we understood the Moon very differently…

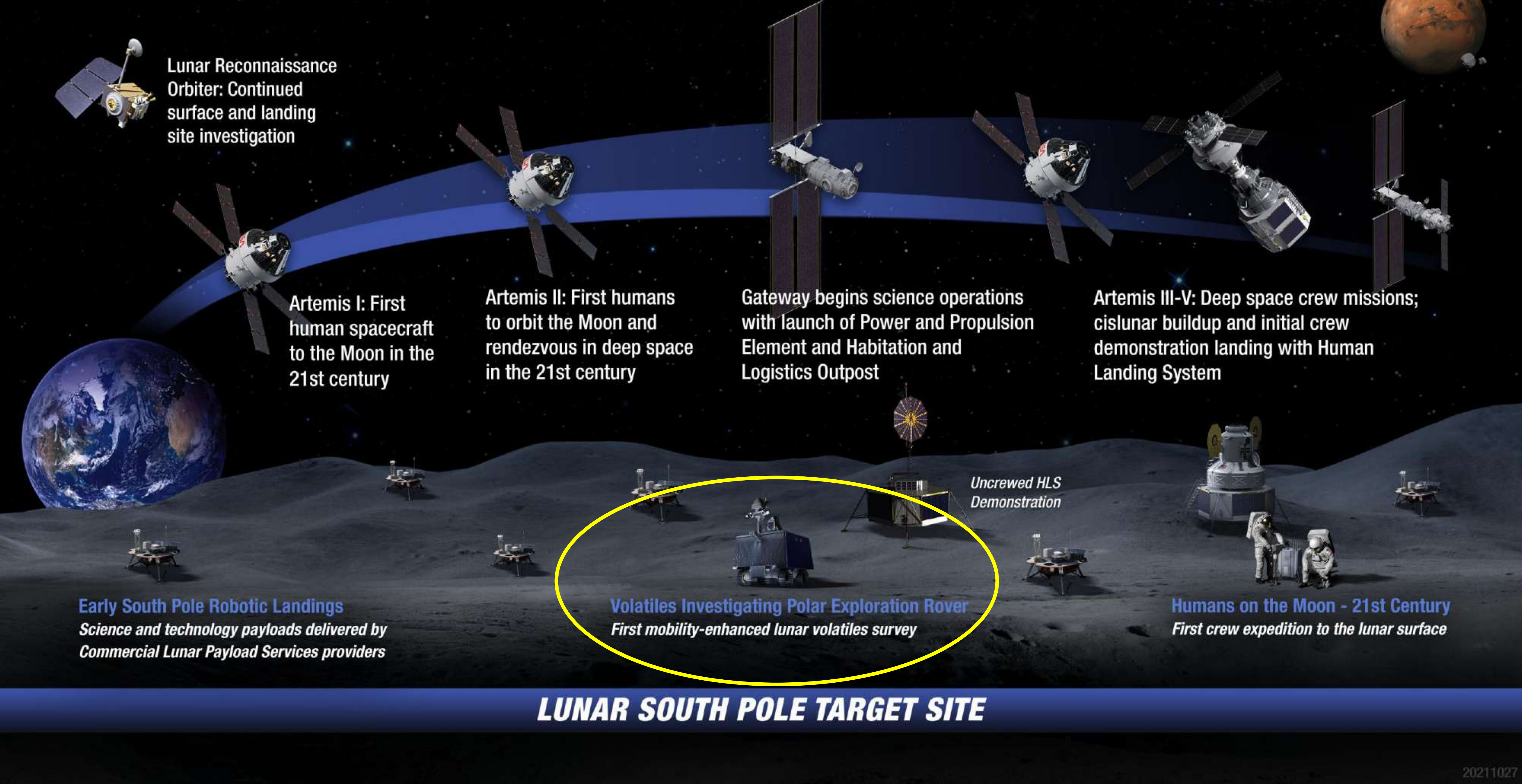
We studied from the Earth, from the Moon’s surface, and had returned samples to Earth.

General conclusion was:

- Surface was relative constant
- Essentially no atmosphere
- Bone dry

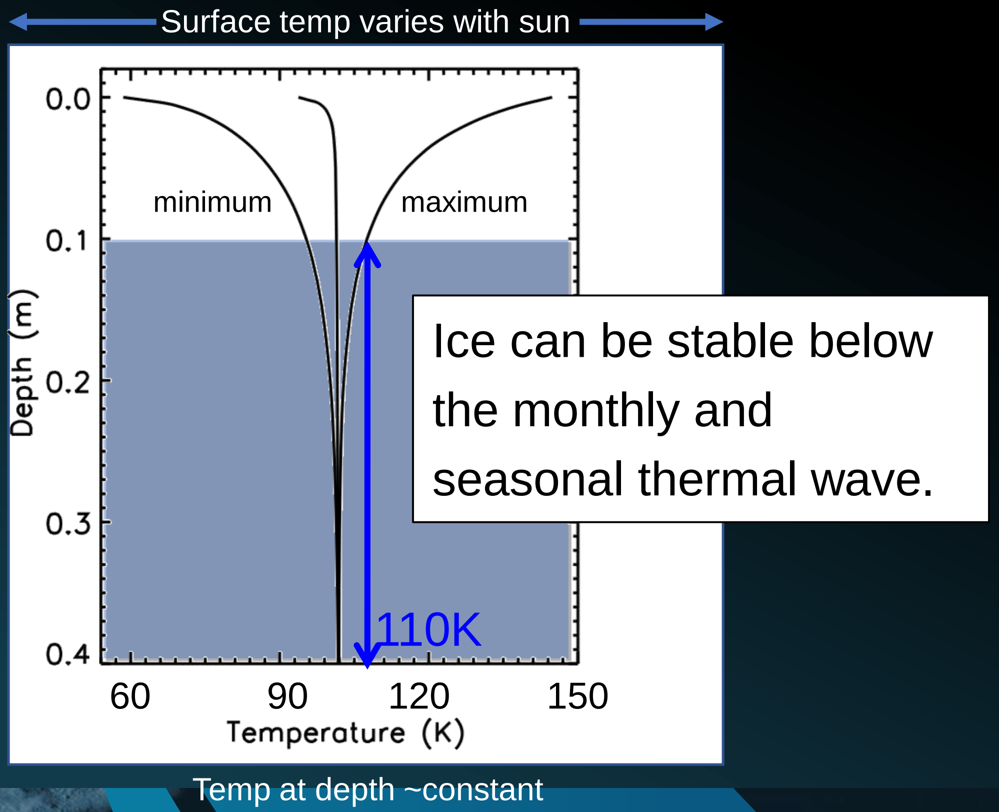
Recent Lunar Missions like LCROSS, LRO, LADEE, and others changed all that…

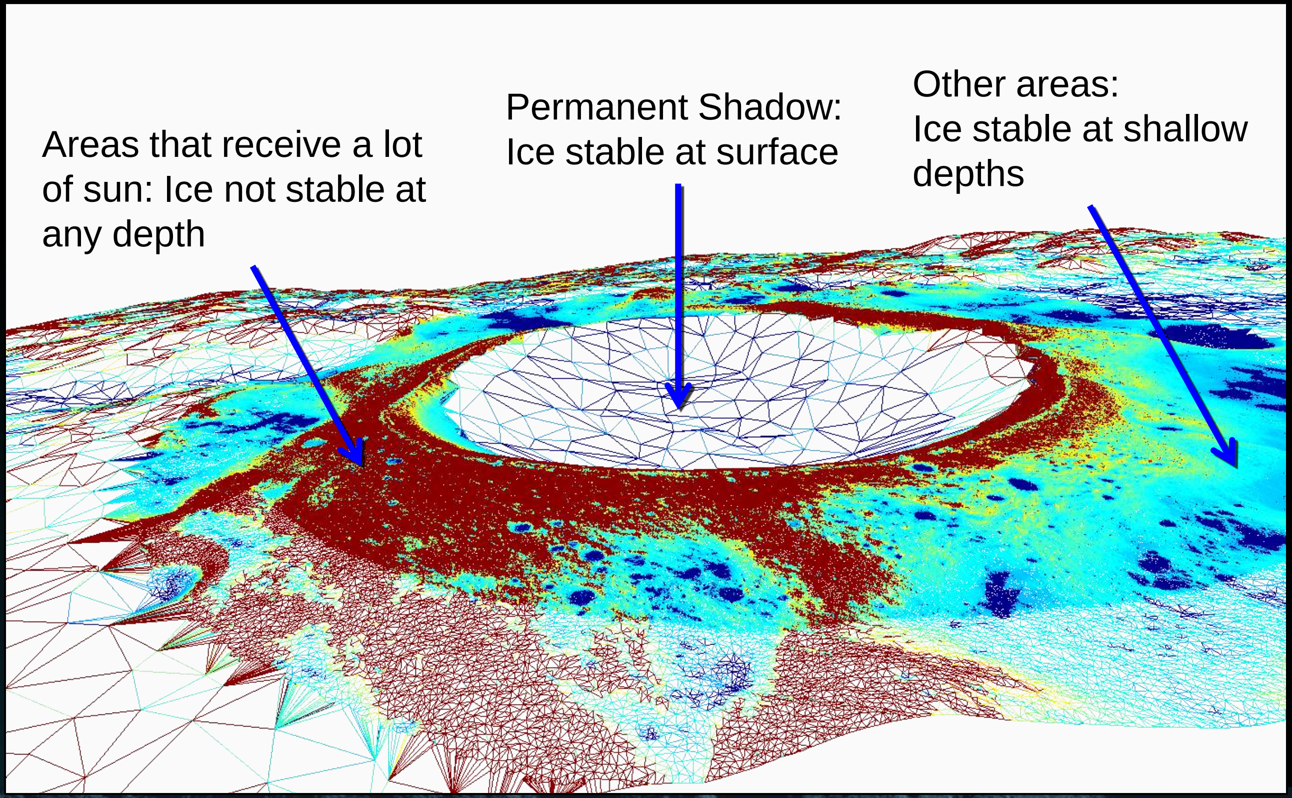
Enter Artemis: Landing Robots & Humans On the Moon



If Ice were present, where could it last?

- Temperatures that remain below 110 K are necessary for long-term ice sequestration
- Where does it stay cold enough over long periods of time?
  - On the surface in permanent shadow
  - Below the surface … how deep depends on transient surface heating by the sun.
- Summarized as 'Ice Stability Depth'



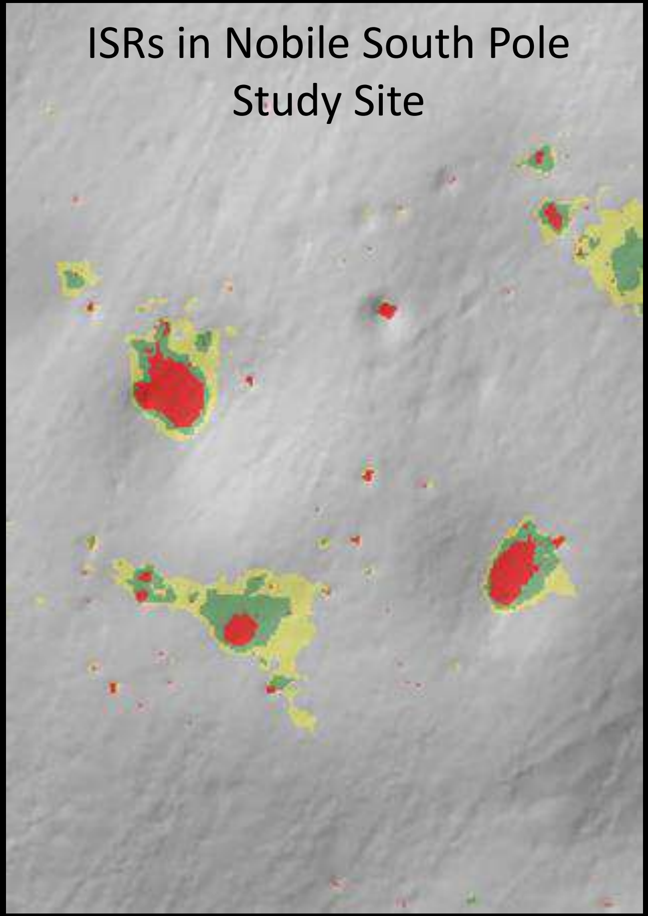


Maps generated by Matt Siegler (PSI) and Dave Paige (UCLA)

Where will VIPER explore?

VIPER will explore four polar **"Ice Stability Regions"** (ISRs)*:

- "**Surface**" - Ice expected stable on the surface (PSRs – Permanently Shadowed Regions)
- "**Shallow**" - Ice expected stable between 0-50cm of the surface
- "**Deep**" - Ice expected stable between 50-100 cm of the surface
- "**Dry**" - Ice *not* expected stable (0-100cm *too warm* to be stable)

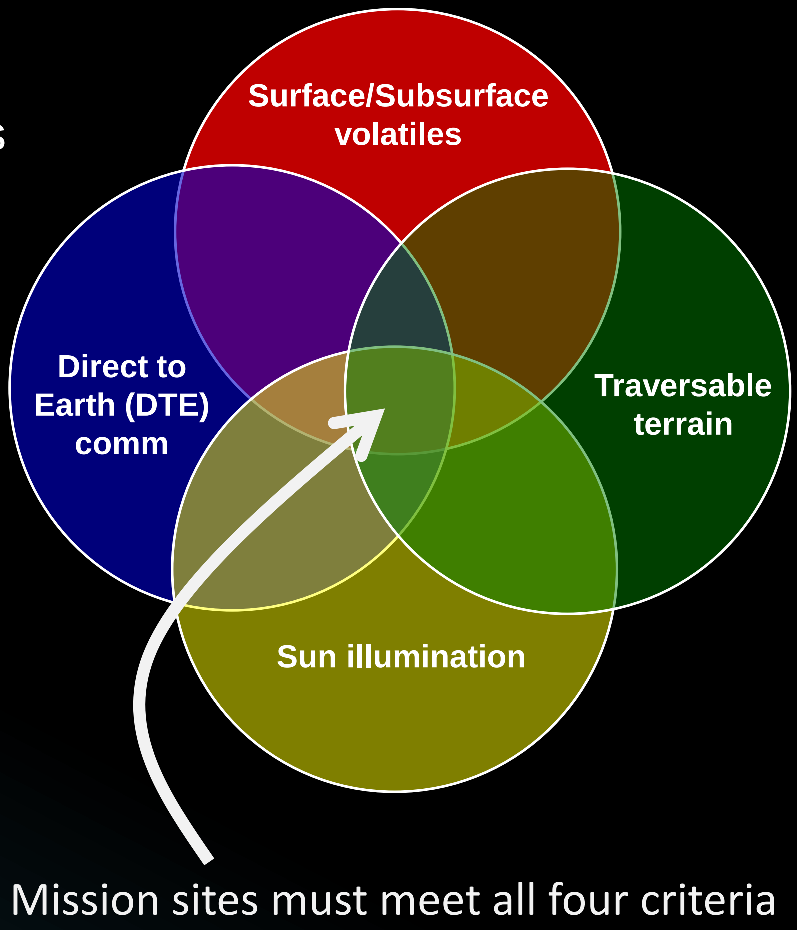


* *ISR's are based on the predicted thermal stability of ice with depth*

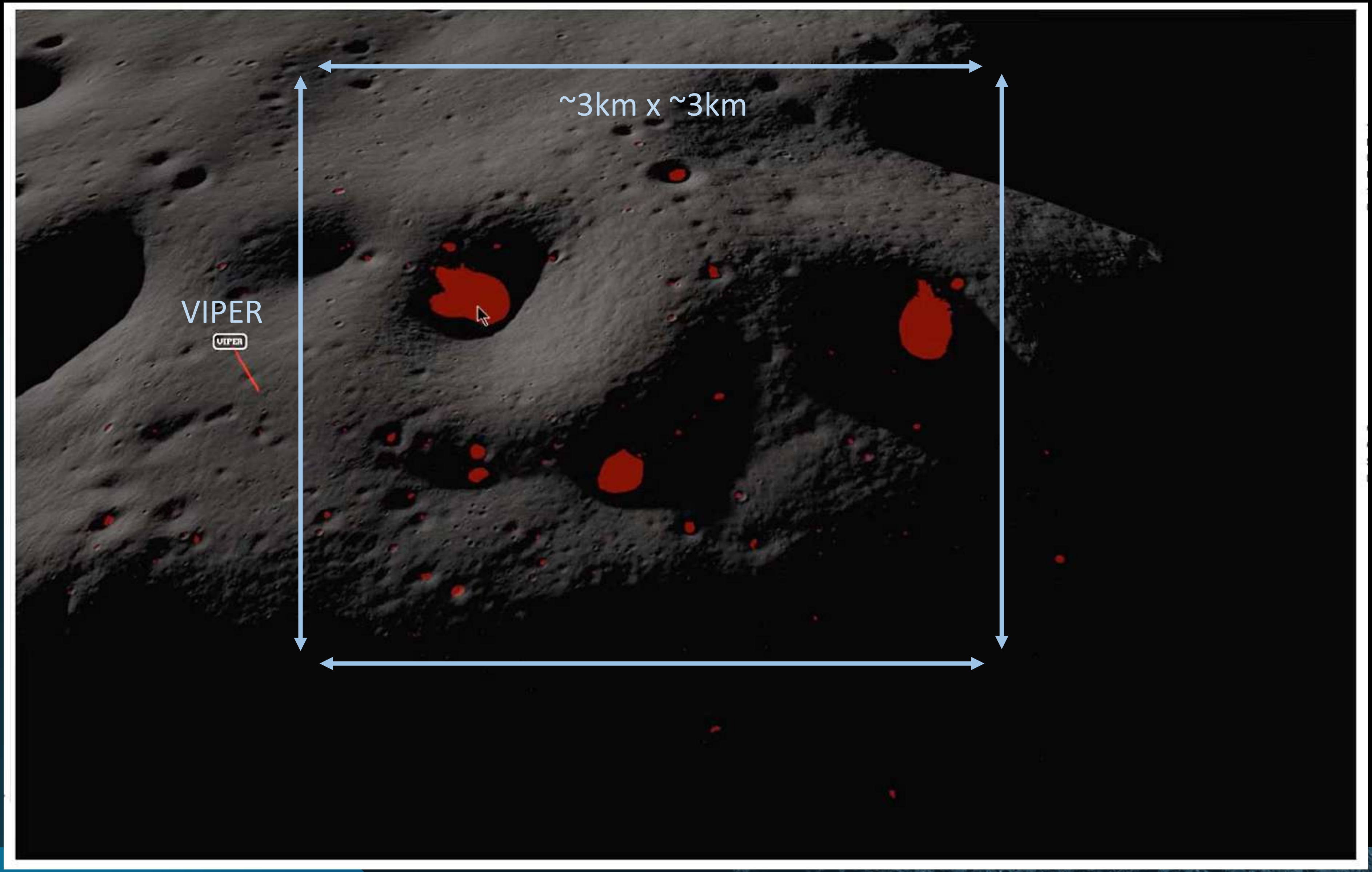
Key Requirements Paraphrased

The Mission Site must meet these four criteria:

- Contain locations that plausibly contain surface & subsurface volatiles
- Provide sunlight for power (including when out-of-comms)
- Provide sufficient line-of-sight to Earth for radio communication
- Contain reasonable slopes for landing and traverse

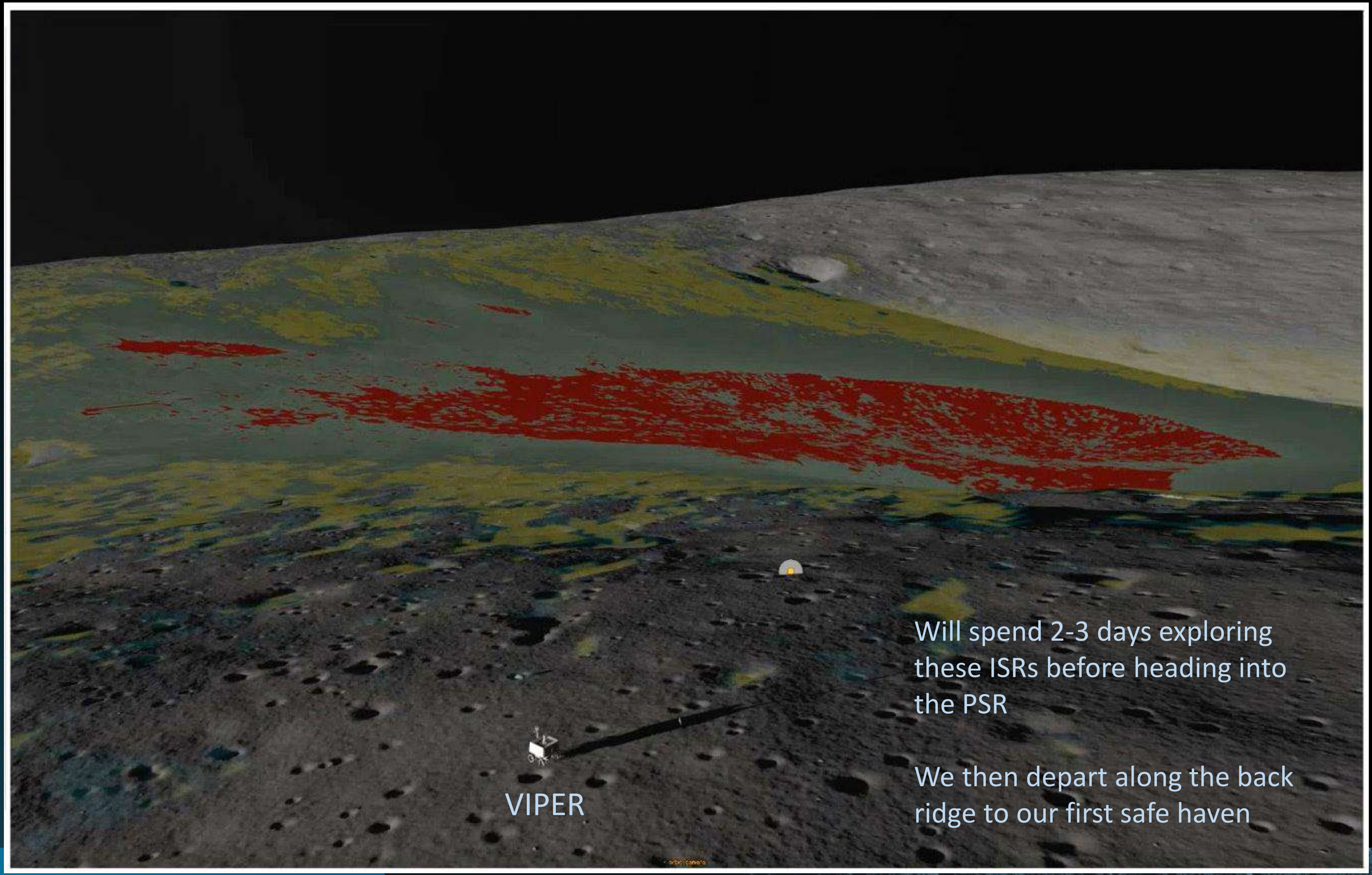


VIPER Nobile mission region on the Lunar South Pole



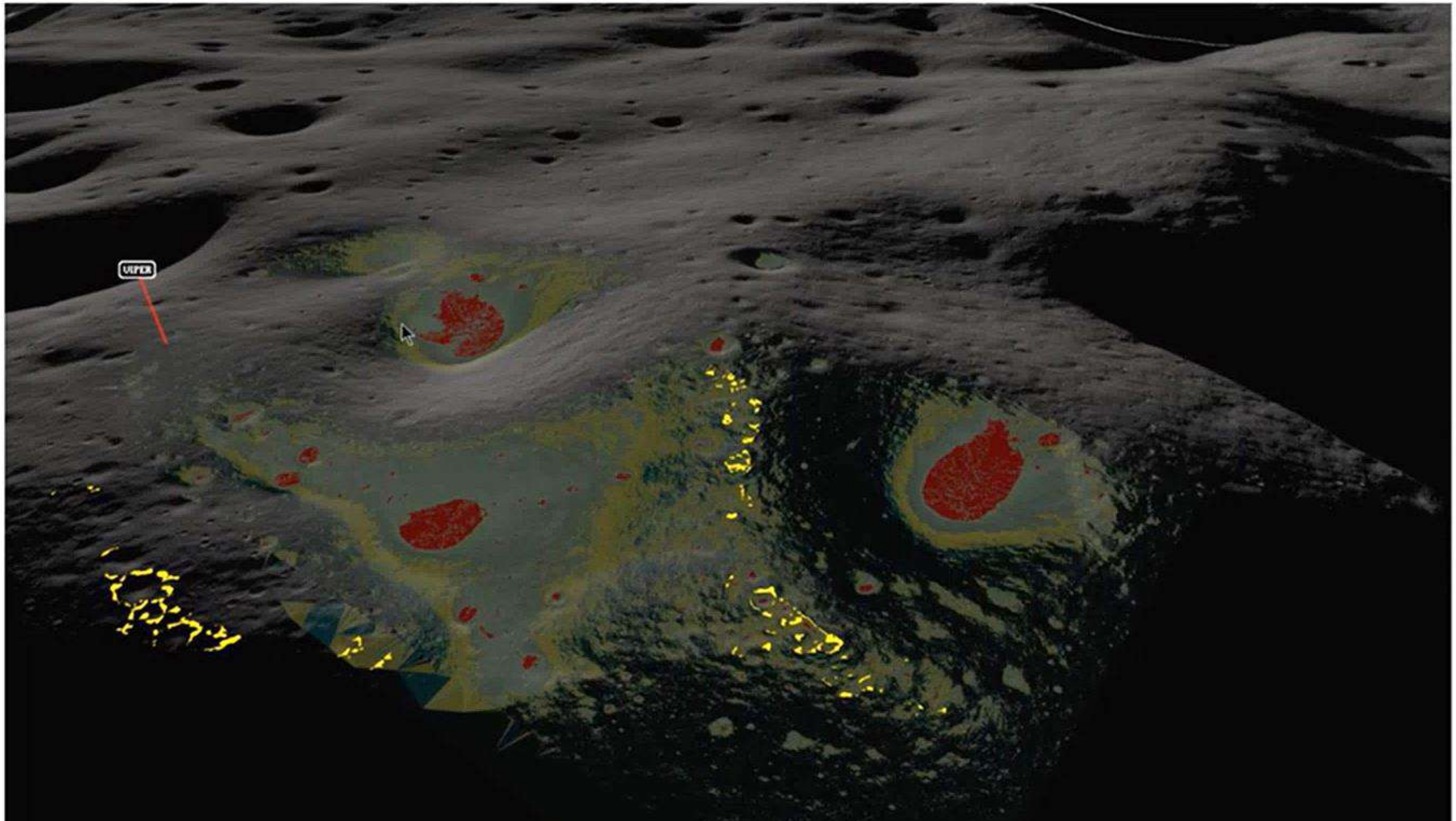
Screenshot from the VIPER Traverse Planning tool /M. Shirley

VIPER and the four Ice Stability Regions (ISRs)



Screenshot from the VIPER Traverse Planning tool /M. Shirley

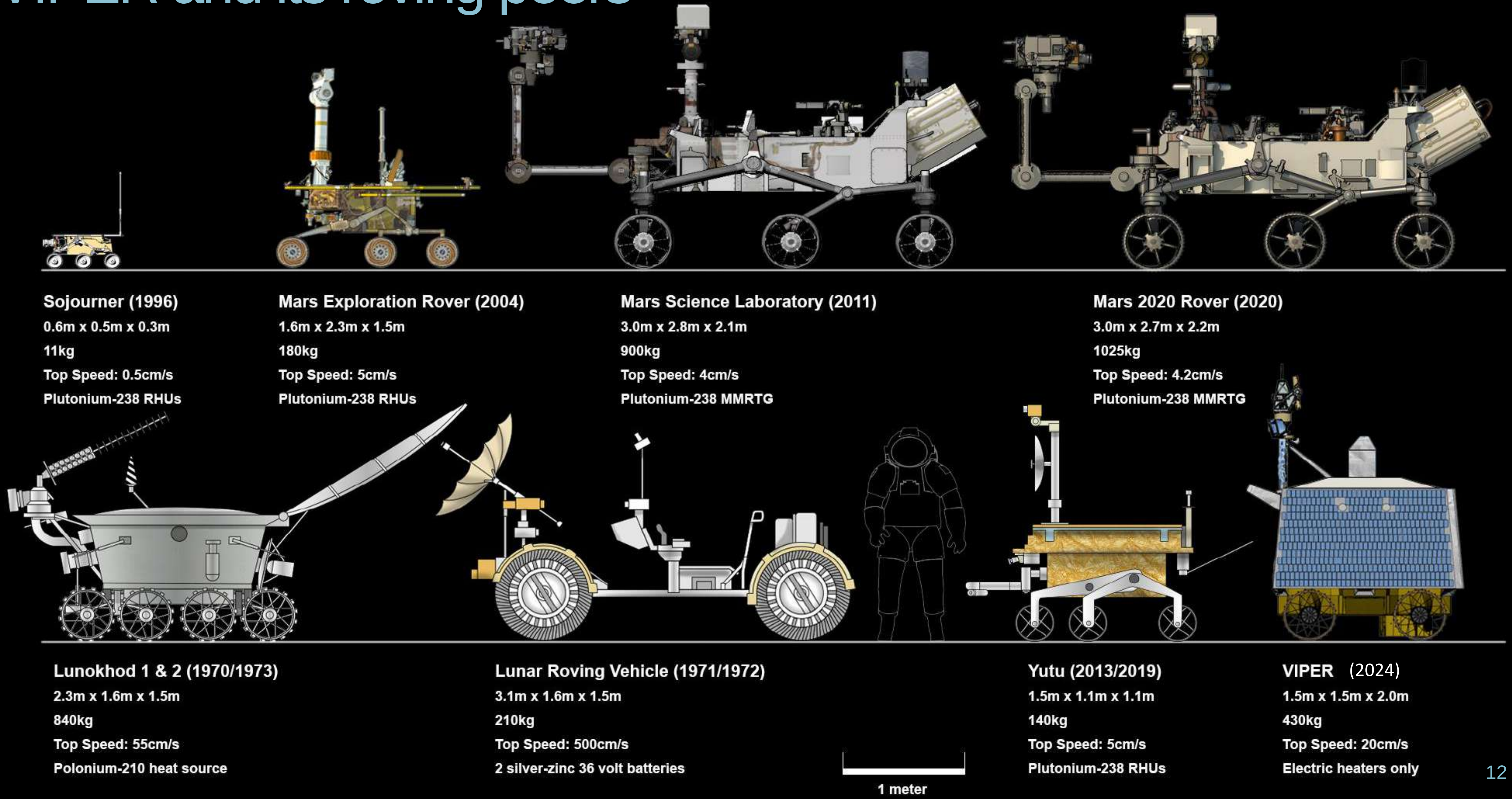
Nobile region with Safe Havens



Yellow Safe Havens experience less than 50hrs of contiguous shadow during the 2 weeks VIPER is out of view of Earth

Screenshot from the VIPER Traverse Planning tool /M. Shirley

VIPER and its roving peers



The NASA CLPS program has selected Astrobotic Technology, (Pittsburgh, PA, USA) for delivery of VIPER to the lunar pole in 2024 aboard their Griffin Lander

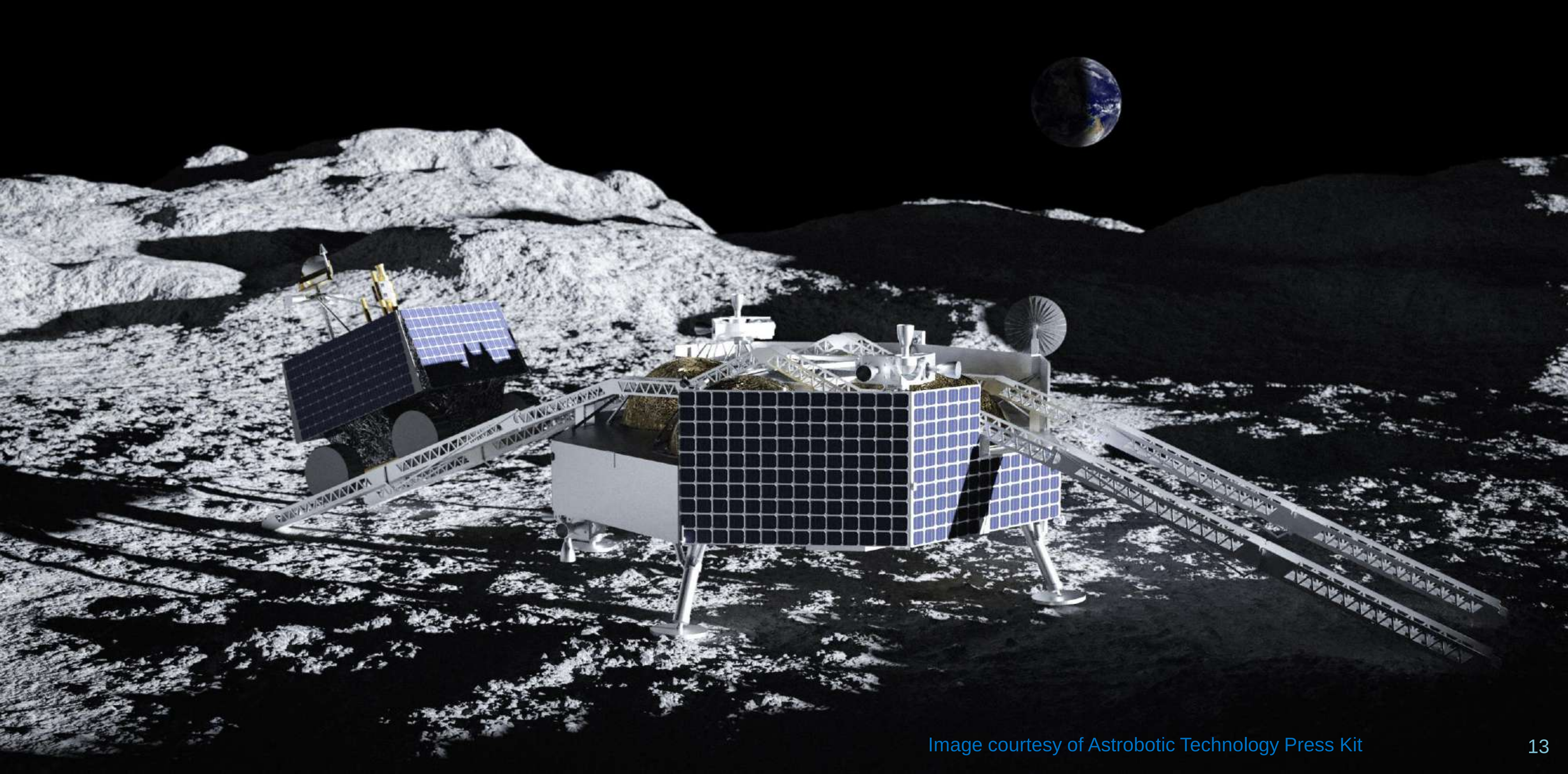
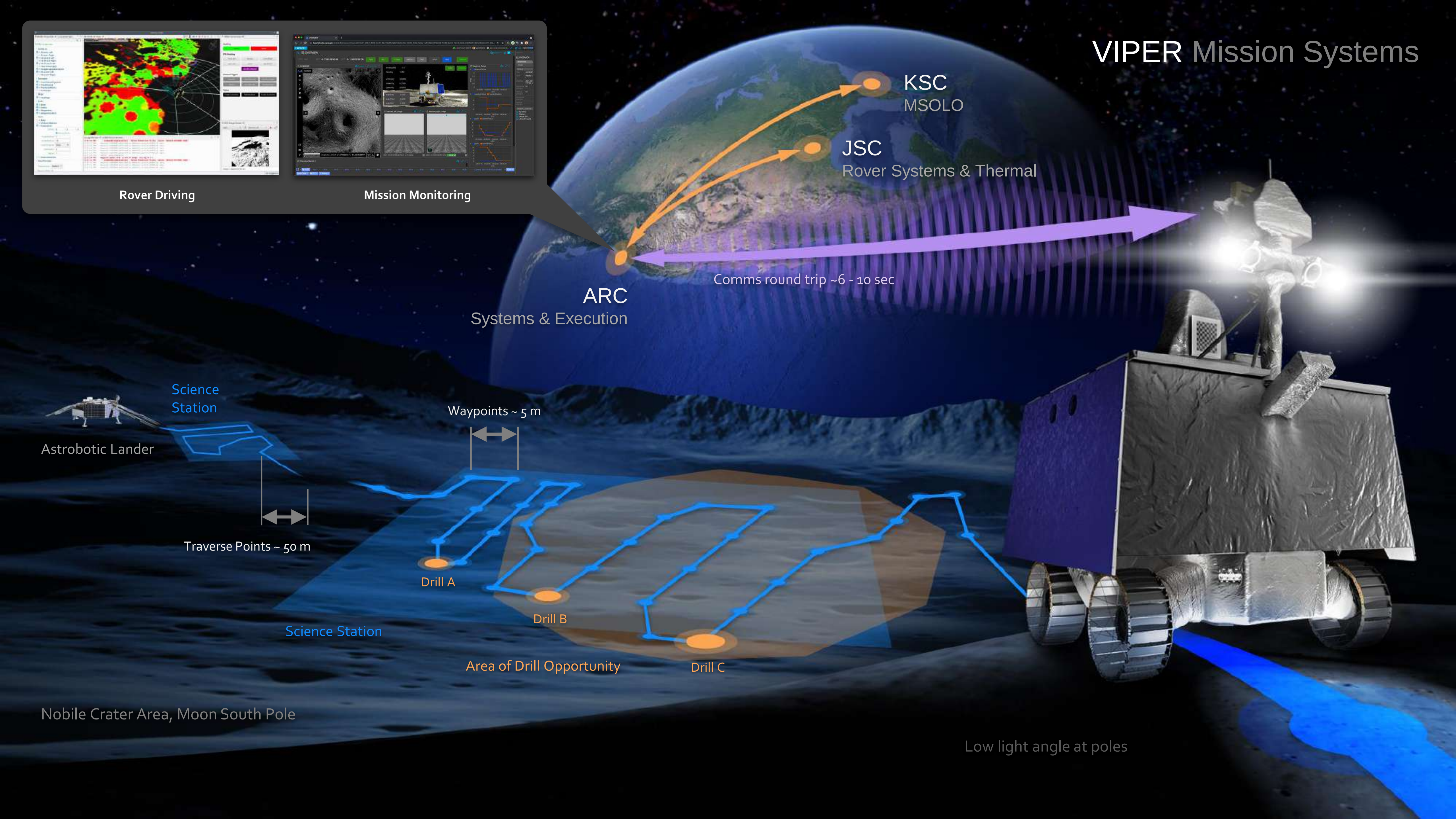


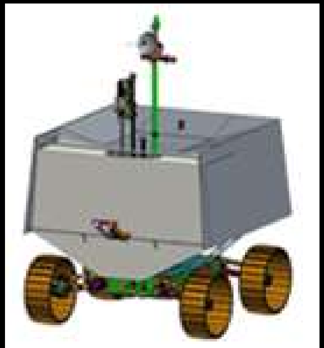
Image courtesy of Astrobotic Technology Press Kit

Operations



VIPER Mission Systems
Rover Driving
Mission Monitoring
KSC
MSOLO
JSC
Rover Systems & Thermal
ARC
Systems & Execution
Comms round trip ~6 - 10 sec
Science Station
Astrobotic Lander
Waypoints ~ 5 m
Traverse Points ~ 50 m
Drill A
Drill B
Drill C
Science Station
Area of Drill Opportunity
Nobile Crater Area, Moon South Pole
Low light angle at poles

Ops Approach: VIPER vs. Mars Rovers vs. ISS



	VIPER	MER/MSL	ISS EVA Robotics
Comms	continuous	~once per Sol	continuous
Latency	6-25 sec	20-40 minutes	<=2 sec
Environment	unstructured	unstructured	engineered
Approach	waypoint commands	command sequences	direct teleoperation

Surface Environment Key Operational Drivers

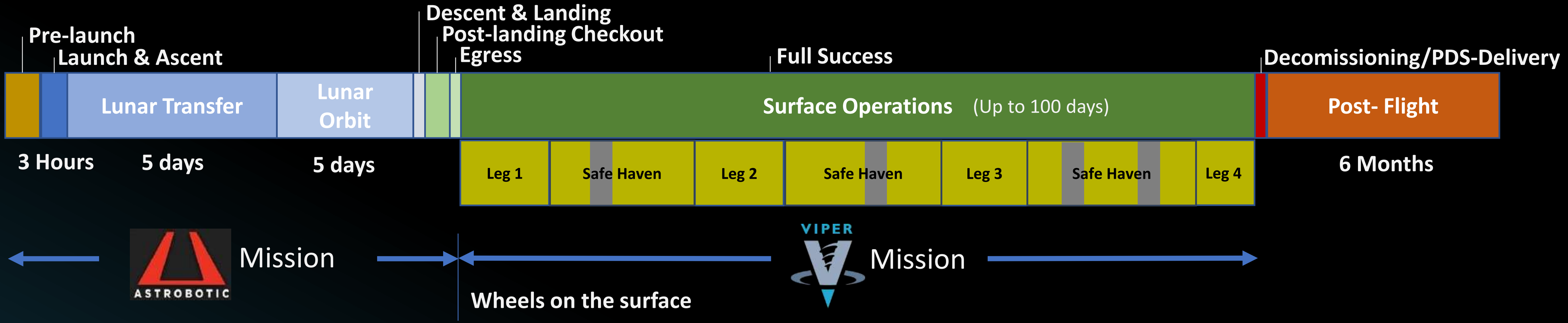
- Regolith/Rocks/Craters
  - Vehicle design to environmental spec
  - Drivers responsible to stay within vehicle mobility limits
  - Real-time science inputs

- Lighting
- Line of Site Comm
- Temperature

Lighting & Comm at the S. Pole v Apollo

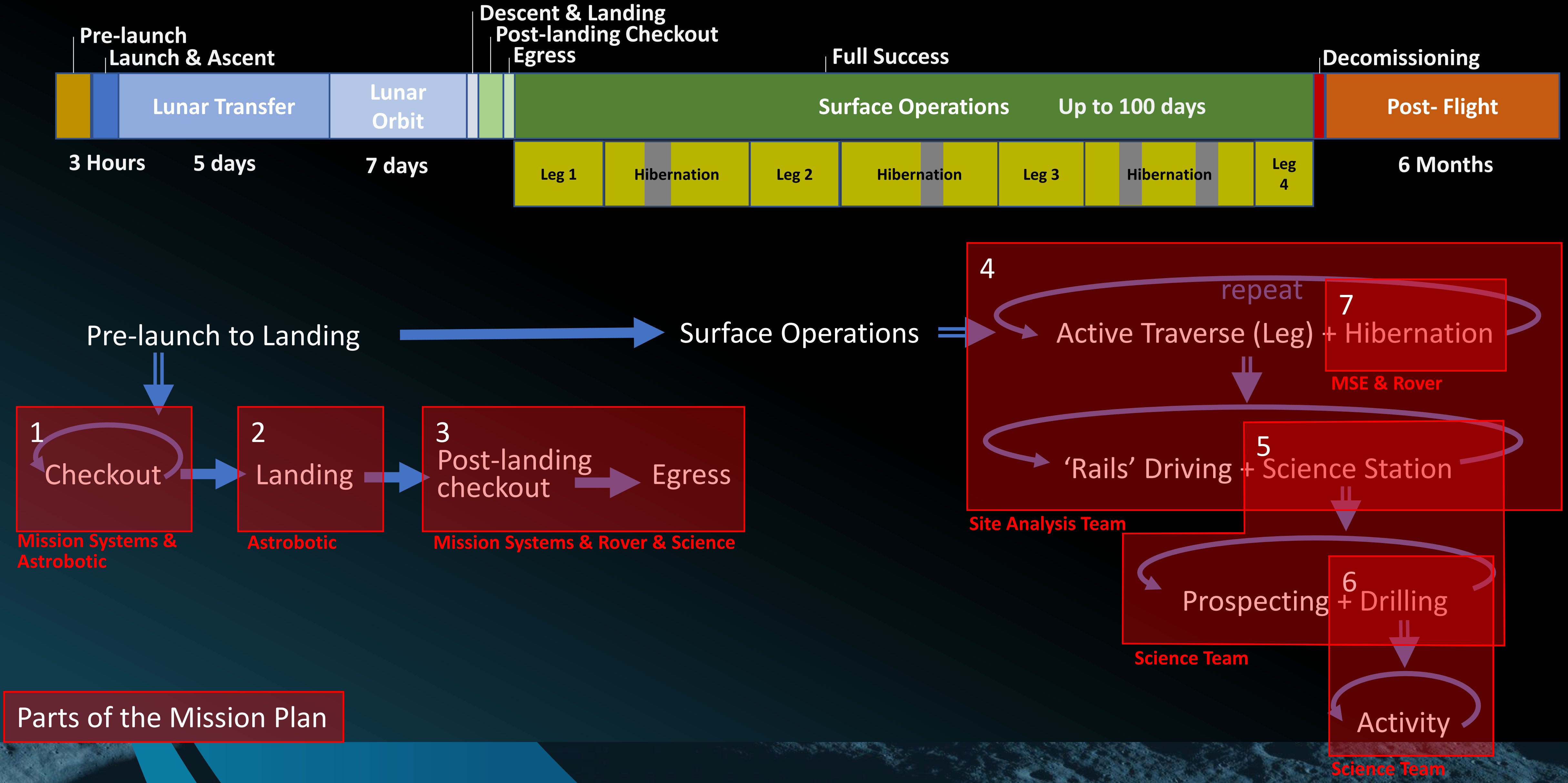

Courtesy ESA

VIPER Mission Phases

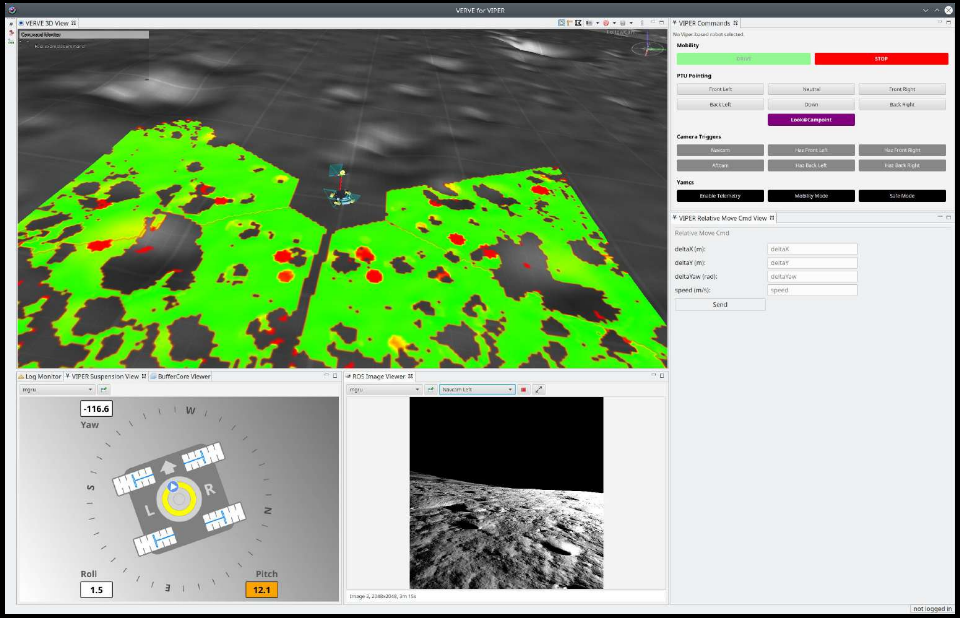


- Surface Ops consist of periods of activity (traverse “Legs”) and periods of inactivity (“Safe Haven”)
  - Traverse Legs are in view of Earth and sun (except for planned shadow (PSR) ops (<8 hours))
  - Safe Havens are NOT in view of Earth, but in view of sun, with periods of sun shadow (<70hrs hibernation)
  - Lunar Day = one Traverse Leg + one Safe Haven
- Mission Success:
  - Minimum Mission Success planned by end of Lunar Day 1
  - Full Mission Success planned by end of Lunar Day 2
  - Lunar Days 3 (and 4 if possible) offer either contingency time, or improved science data

Mission Plan Structure



Driving



- Continuous communication with Earth
  - Low bandwidth: 256 kbps
  - Latency: 6-10 sec nominal round-trip
- Limited computing, smarts on the rover
- Waypoint Driving

  Command rover to drive to a waypoint ~4.5m away, based on lookahead distance and predictability of mobility (e.g. wheel slip)

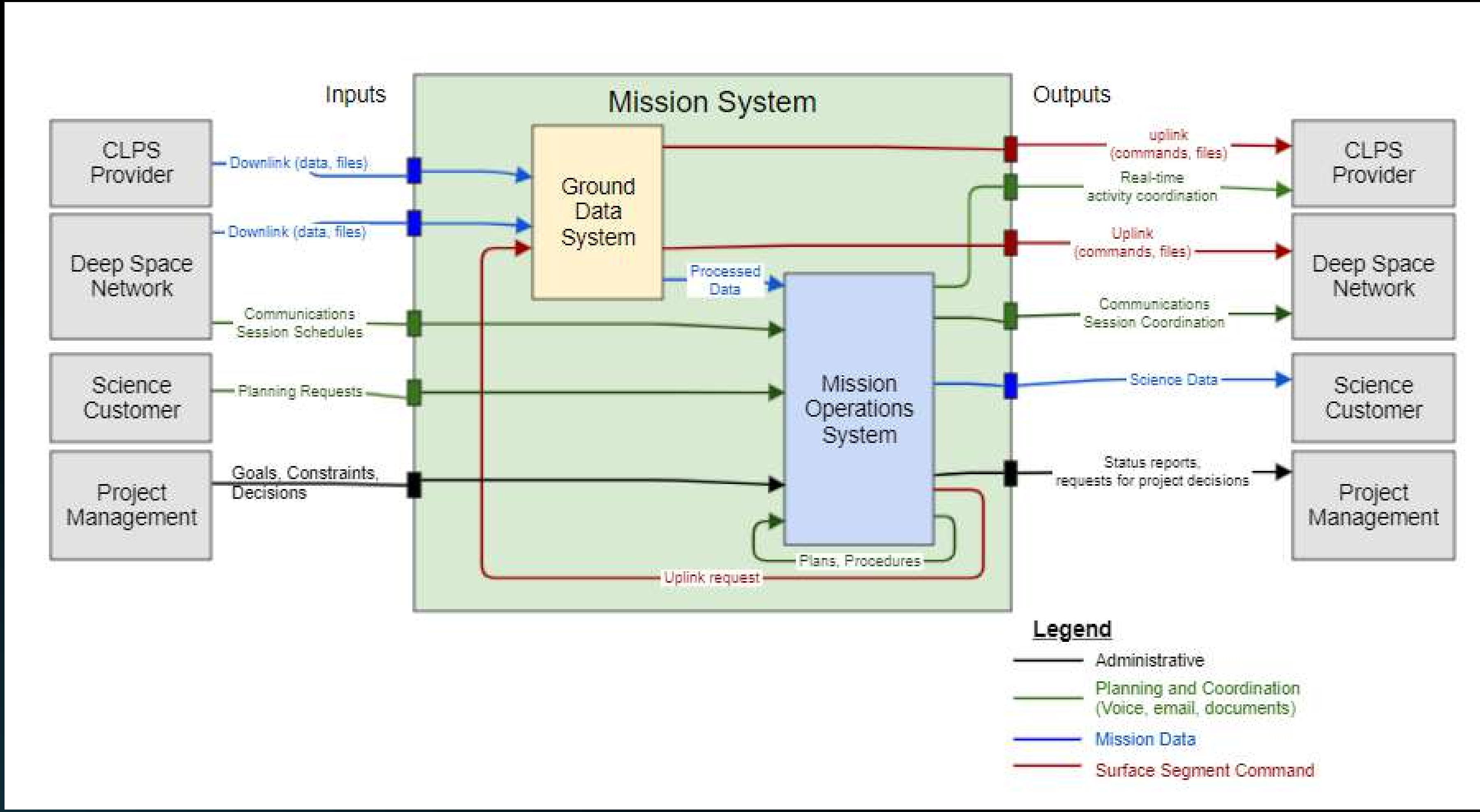
  Rover onboard control achieves the waypoint, minus wheel slip or other errors
- Need to drive using data from the rover
  - Driver/co-driver responsible for safe driving
  - Real-Time Science to interpret terrain when needed (Science Station)

Not continuous “rate control” input, e.g. joystick or gas pedal

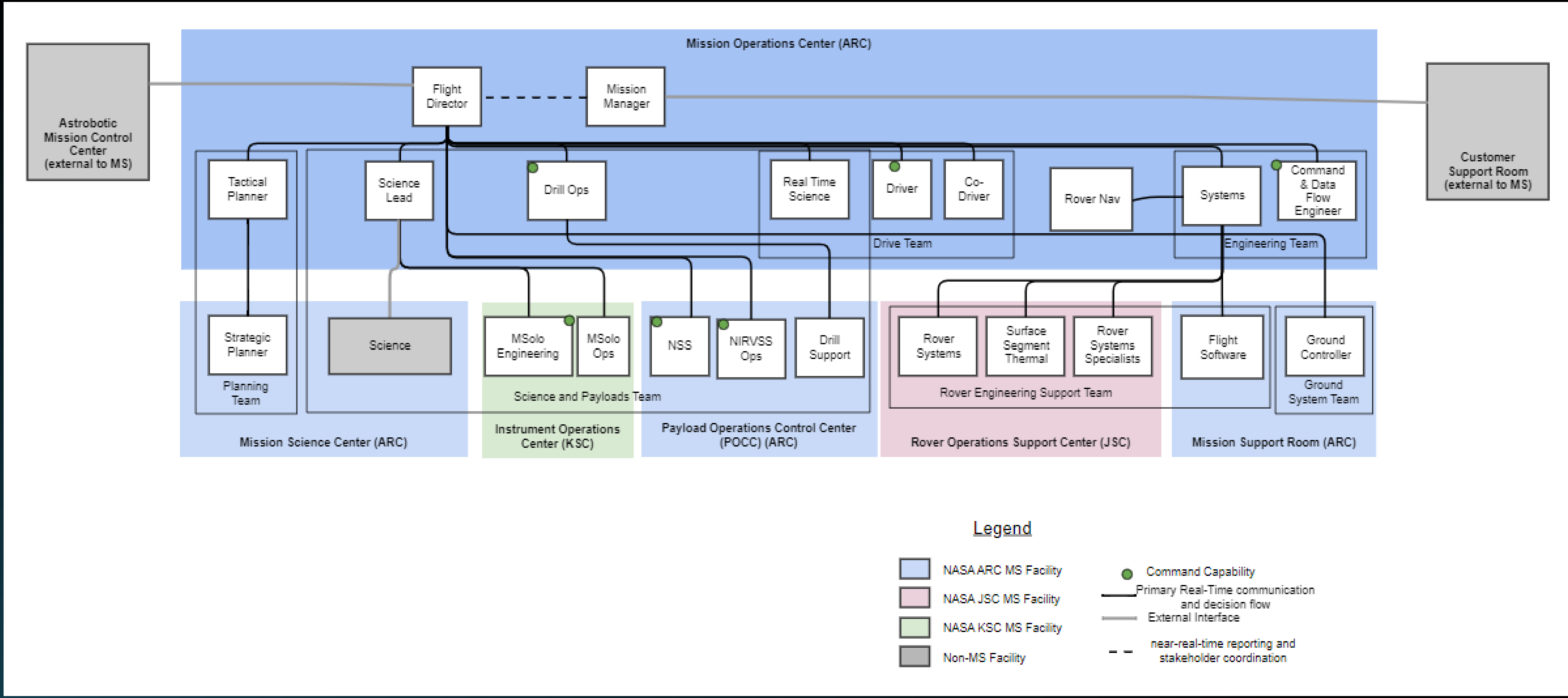
Not sol command sequences, e.g. MER/MSL rovers

The Mission System

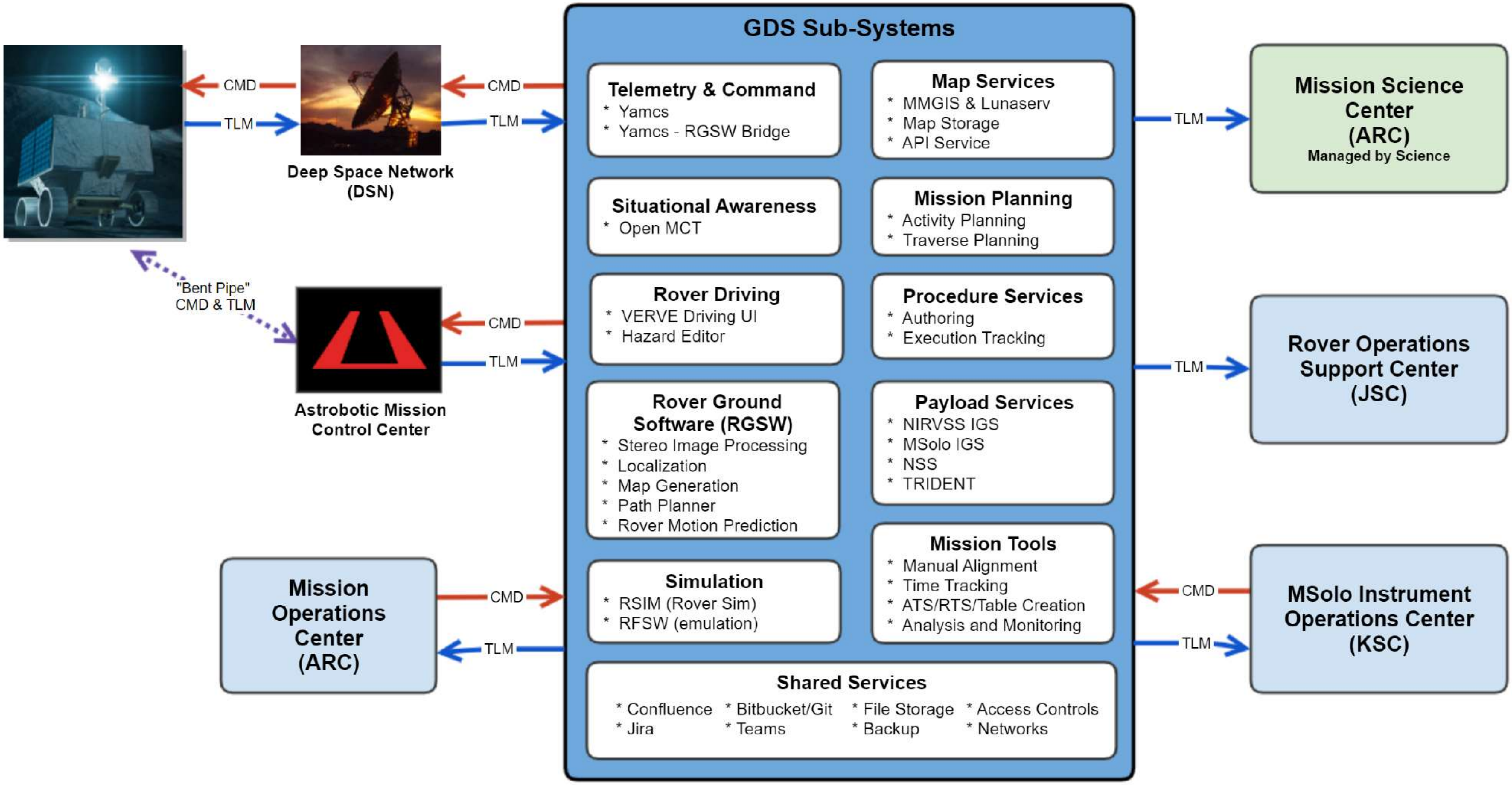
MS Architecture



Mission Operations Team



GDS Overview Diagram



Test & Training

Thread Tests

- Verify/validate selected parts of the MOS/GDS, initial process run throughs, does not require a full flight team

Integrated Simulations

- Demonstrates ability to execute activities
- Mission simulations of portions of the nominal timeline with contingencies introduced in some circumstances

End to End Testing

- Test end-to-end command and data flow, execute test as you fly activities

Operational Readiness Testing

- Demonstrates ability to perform the mission
- Time realistic, final MOS training and cert, operational readiness

Mission Rehearsals

- Maintain and enhance operational proficiency

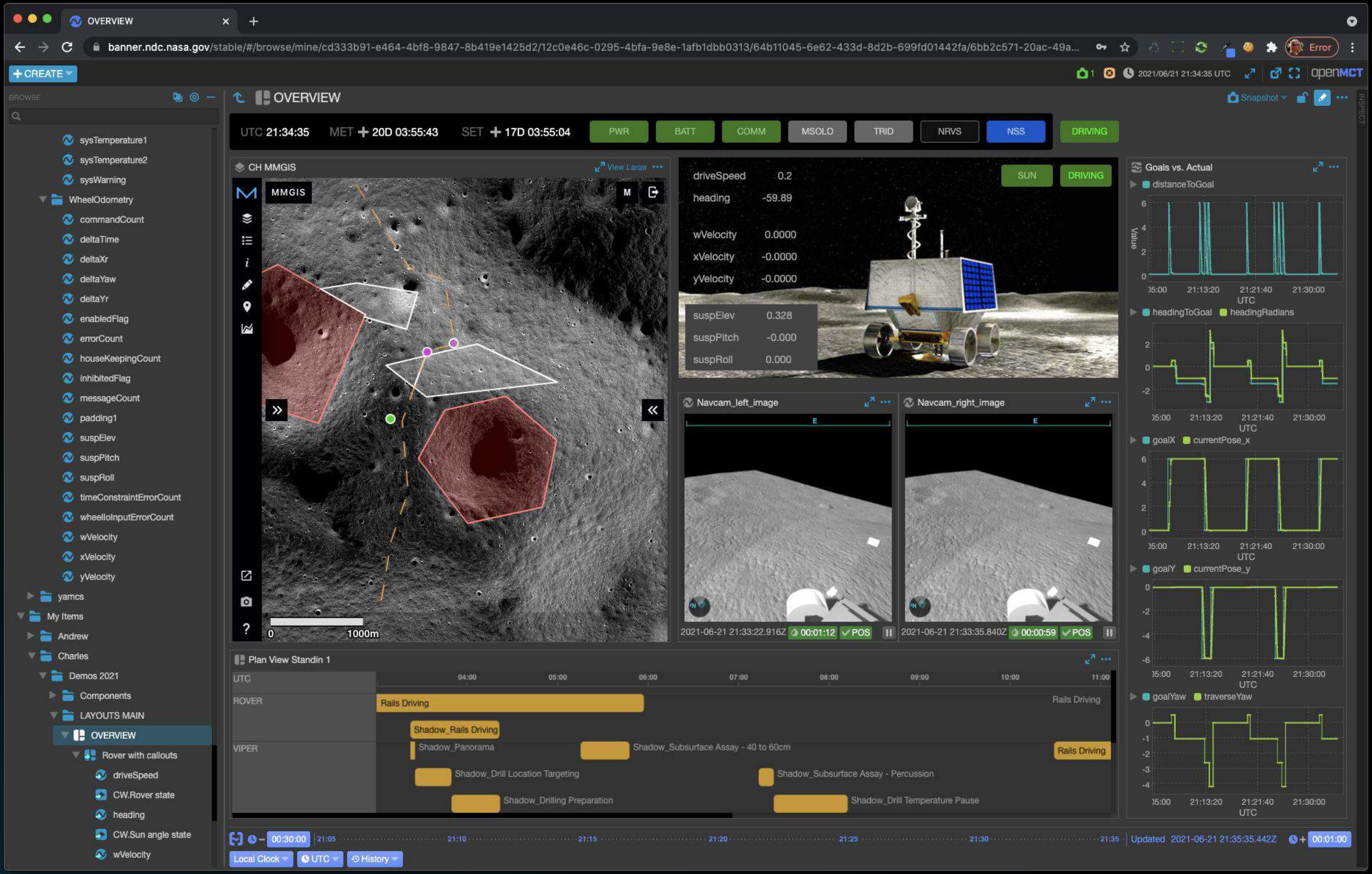
Observing Ops

- Console time as an observer at JSC and JPL

Software

Open MCT Integrated Visualization

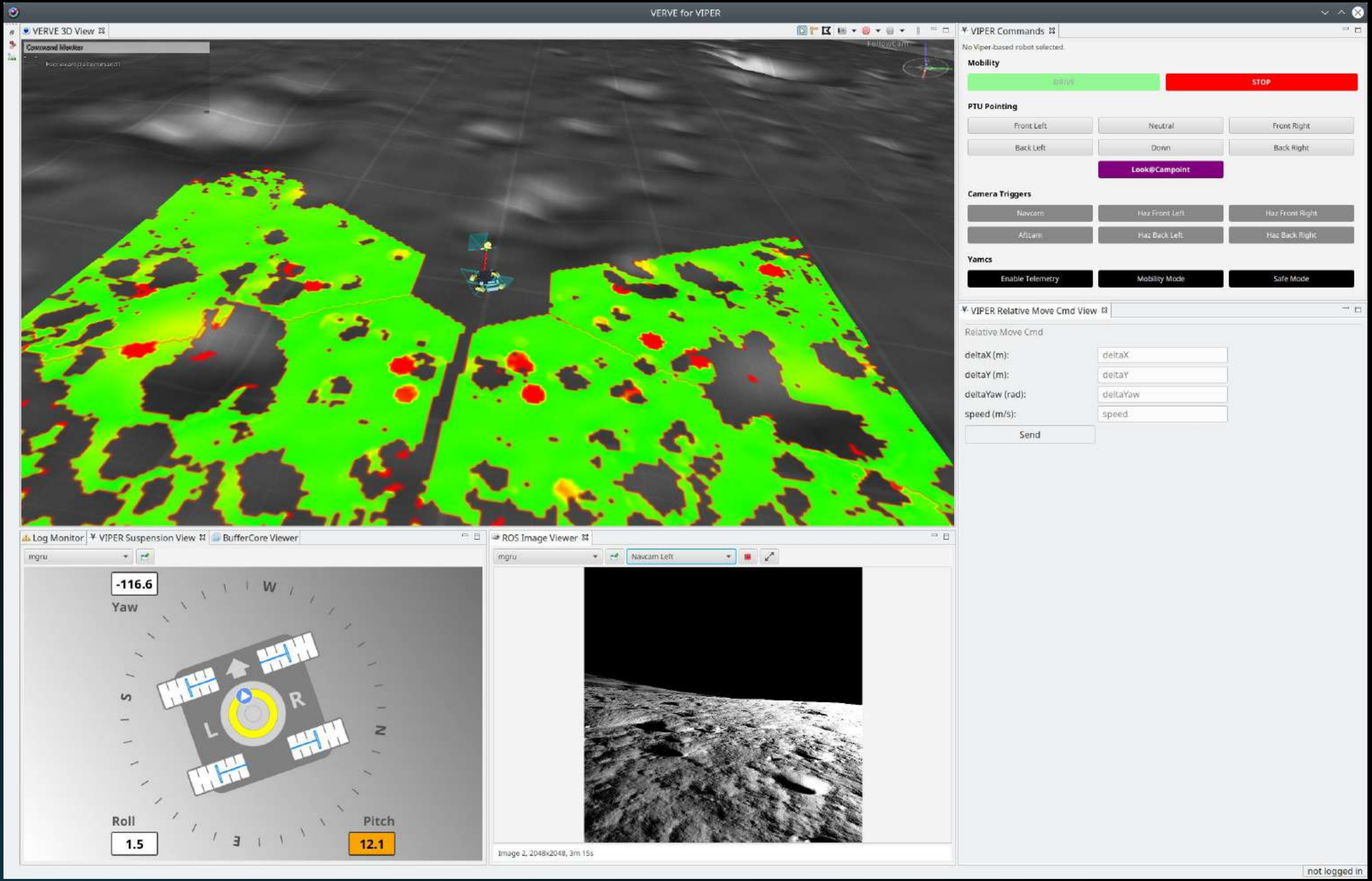
- Realtime and Historical Plots/Tables
- Rover Image Display
- 2D and 3D Map Layer Display
- Instrument Data Heatmaps
- Rover Traverse Plans
- Activity Timelines
- Events and Limit Violations
- Fault Manager
- Procedure Library
- Shared Procedure View
- User Annotations
- Console and User Notebooks
- Customized Layouts



VERVE

- Orient Pan Tilt Unit
- Nav/Aft/Haz image commands
- 3D environment visualization
- Hazards (w/manual editor)
- Open MCT (embedded)
- View desired traverse
- Set waypoints
- View predicted path
- Command rover motion

Visual Environment for Remote Virtual Exploration



Open Source

Open Mission Control Technologies - Open MCT

Info

- https://nasa.github.io/openmct/
- From info site, click on Try it

Code

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

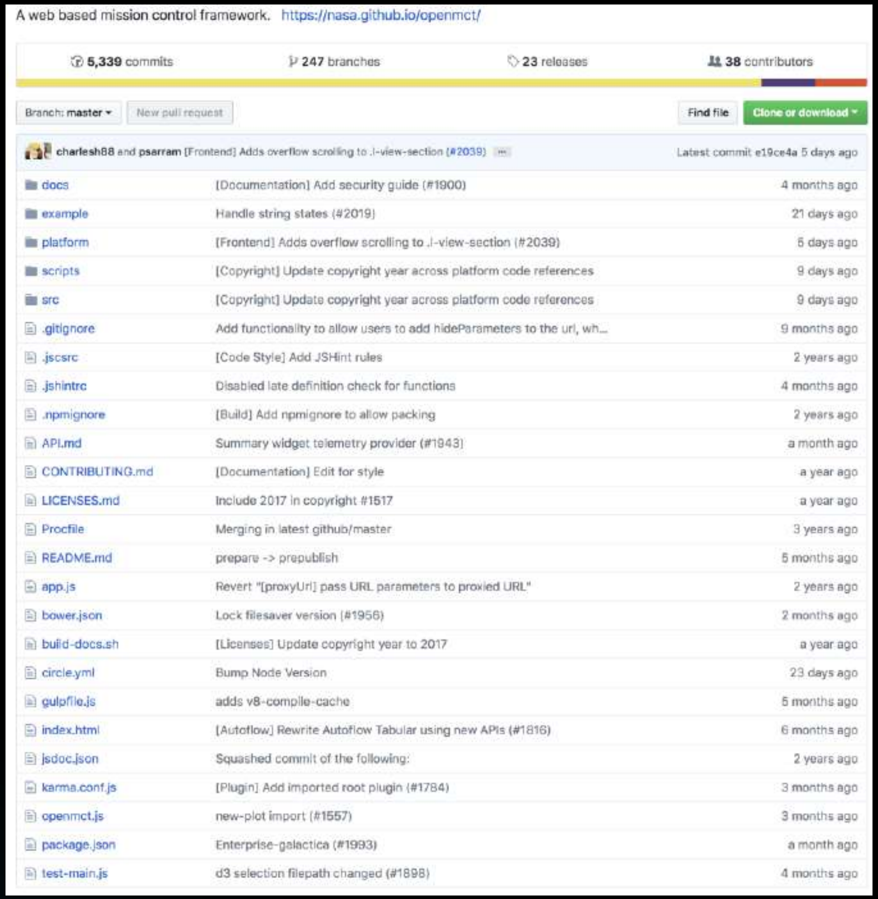
Open Source Benefits

Collaboration that works

Use, adopt, make it your own, contribute

No ownership issues

Instant access



https://github.com/nasa/openmct

Agile

VIPER: A new approach to planetary missions

- VIPER follows **NPR 7120.8 with augmentation**
  - Apply ARC and NASA engineering processes and standards with **tailoring** to improve efficiency while reducing cost and schedule
  - VIPER Review Team (VRT) provides streamlined approach to continuously review the project
  - VIPER does not have a formal risk classification, but is “like” Class D
- Adapt and use **NPR 7120.5 “constructs” as needed**
  - Control plans should be baselined and used when needed
  - Reviews should focus on content, not on slide preparation and travel
  - Focus on what provides value & rigor, not just satisfying process
- Rapid design and development
  - Apply agile software development practices as much as possible
  - Use iterative design and test rather than lengthy “waterfall” process
  - Extensive use of collaboration tools

In the Beginning

Software, years before VIPER

Delivery every 6 months

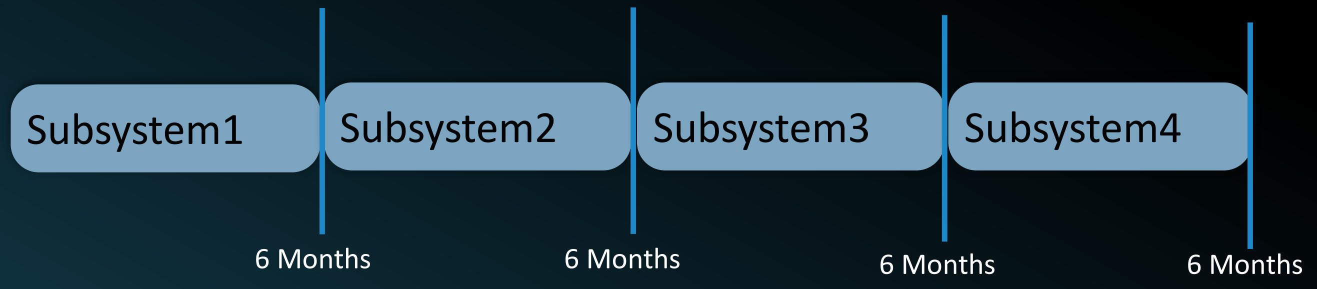
The 6 month delivery cycle created too much time for customer expectations to diverge from what we were building

Customers needed to see the

Progress difficult to measure

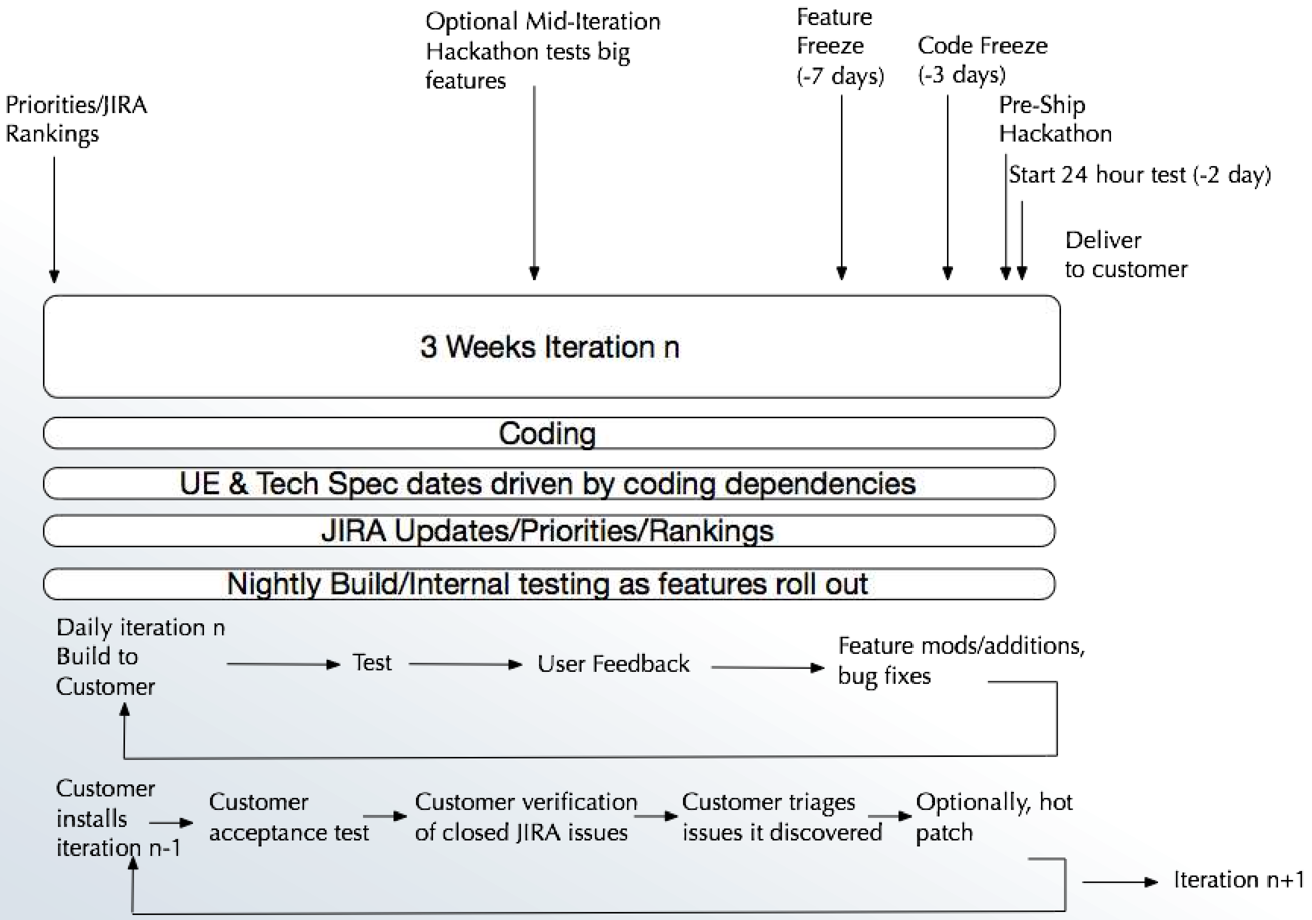
Long and formal design specs

Too much time talking, having meetings, writing documents, not enough time doing



Agile Software Sprint example, external customer

- Agile Tailored for our team
- Deliver to customer every 3 weeks
- Nightly build
- Release every 3 months
- Emphasis on constant interaction and use



jay.p.trimble@nasa.gov

Key attributes of our tailored agile process for software

- The measure of progress is working code
- Demonstrations, not presentations
- Constant use of the software in a relevant environment, such as mission simulations
- Visible progress - nightly or continuous builds
- Ship on time, features that are not ready go into the next sprint or release
- Validation using both QA and customer use in context

Agile Mission Operations

Agile for the Mission System

Mission System (MS)

- Mission Operations System (MOS) – people and processes
- Ground Data System (GDS) – Software, hardware, infrastructure, control center

Agile well established in GDS software

Are there potential benefits from agile in extending it beyond the GDS, to the MOS?

- Emphasis on doing rather than documenting
- Continuous integration and interaction
- Visibility of the system, knowing where you are
- Fail early so you can succeed sooner

Tailored Agile

MOS

Assessment of capability through demonstrated execution of mission capability

Maturation, evaluation and iterative development of the system through continuous use

Early and frequent builds and tests

Software

The measure of progress is working code

Demonstrations, not presentations

Interactions and use over meetings

Continuous visibility of progress (continuous builds)

Sprints ship on time, features that are not ready go into the next sprint

Words to Remember

”Test as you fly, fly as you test”

“Say it then sim it”

Agile Methods from VIPER: Dev Sims

Dev Sims = Development Simulations

Location: Lab

Enables use of the latest running software build

Try the software in a controlled usage scenario, such as driving

Get data to help answer targeted questions

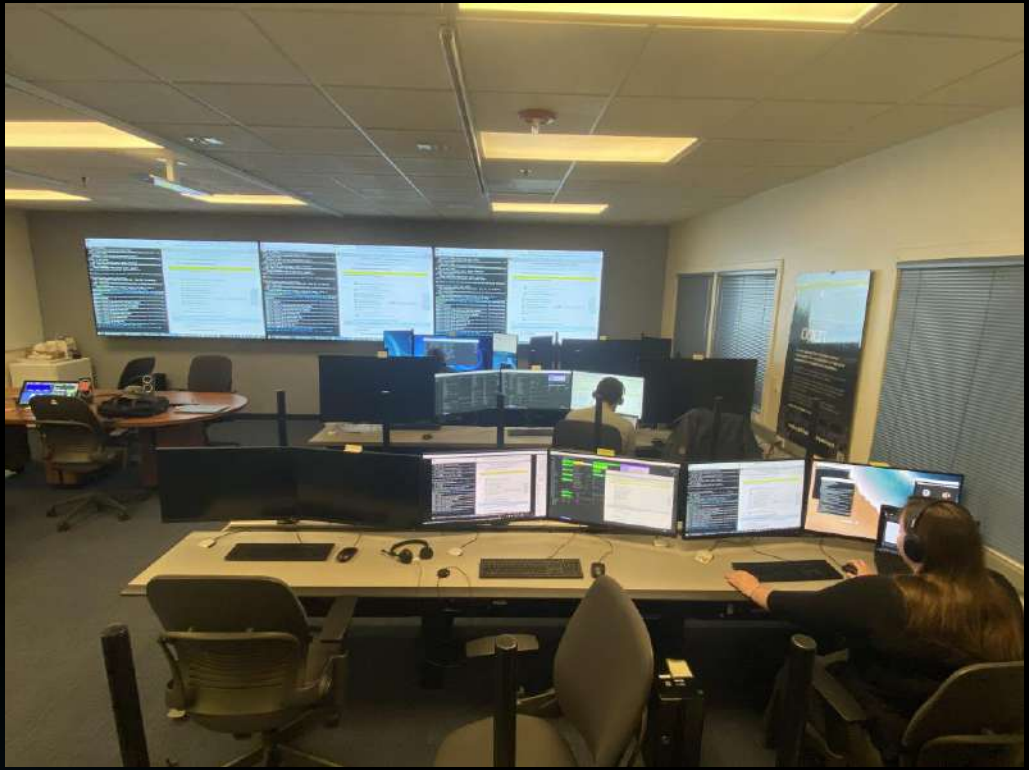
- Example: Driver decision time between command cycles

Find things you may not be looking for

- Example: Aft driving issues

Develop and mature the system by using it

Note: This requires a simulator and a budget



Agile Methods from VIPER: Ops Products

A traditional waterfall ops product cycle might consist of a small number of major releases

- Draft
- Engineering Release
- Test & Training Release
- Mission Release

These releases would each come with a release review, which would involve document review, signoff and likely a set of presentations

Ops products would then be updated during test and training, culminating in a mission release

Agile Methods from VIPER: Ops Products

We develop ops products based on dependencies from test & training activities

For each activity, such as a simulation, or an integration and test activity, develop the ops products required to support it

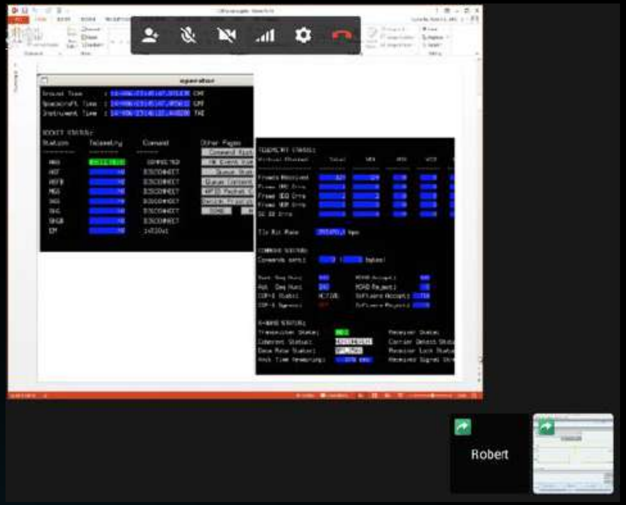
- Rather than a large procedure drop, the team focuses on a limited number of products and matures those products in a structure similar to a software sprint
  - Draft → walkthrough in lab → walkthrough in control center → simulation → iterate and repeat

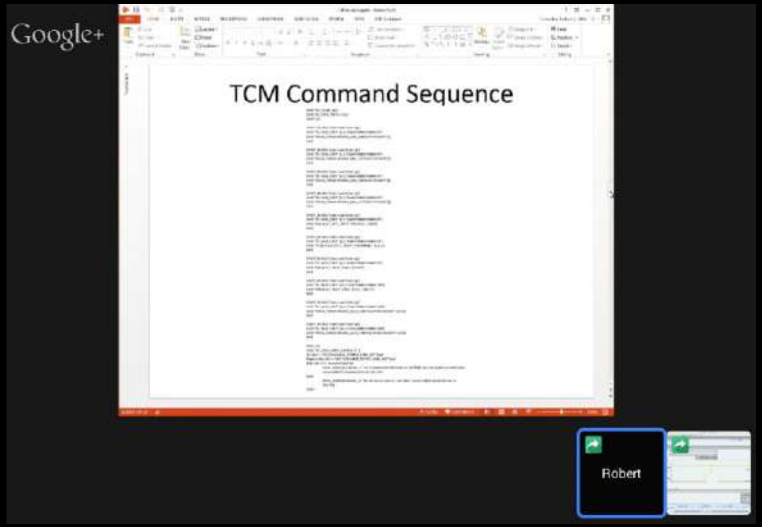
Procedure walkthrough

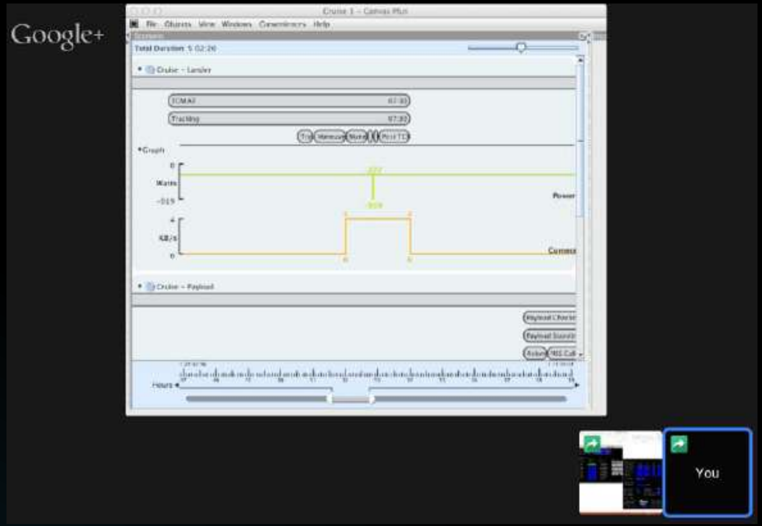
You don't have to wait for mature procedures and software to design your ops products

This is an example from the Resource Prospector Project of an early walkthrough

We used Powerpoint, Google Hangouts and prototype timeline software







Test & Training

Train the team, test the system

Location: Control Center

Get ready for the mission

- Train and certify the operations team
- Verify and validate the Ground Data System (GDS)

Agile elements of test & training

- Structure, iterations
- Rapid response to change in process, products and software (dependent on build and installation cycle)

Parting Thoughts

Create a culture of doing, rather than a culture of documents and meetings

Agile development, for both software and mission operations processes, is compatible with mandated system engineering processes