Cooperative Autonomous Distributed Robotic Exploration (CADRE)

Multi-Agent Autonomy for Space Exploration

Federico Rossi, Ph.D.
Jean-Pierre de la Croix, Ph.D., (Principal Investigator)
Jet Propulsion Laboratory, California Institute of Technology

Presented at the
International Workshop on Autonomous Agents and Multi-Agent Systems for Space Applications (MASSpace ’24)
Auckland, New Zealand
May 7, 2024

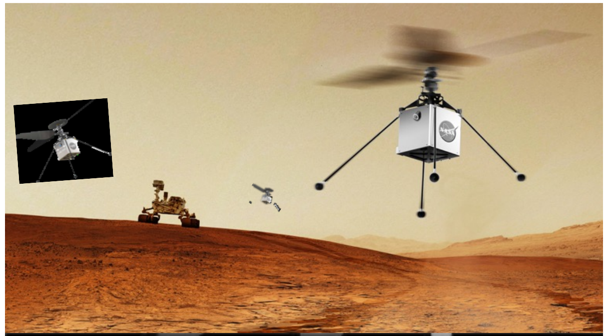
Jet Propulsion Laboratory
California Institute of Technology

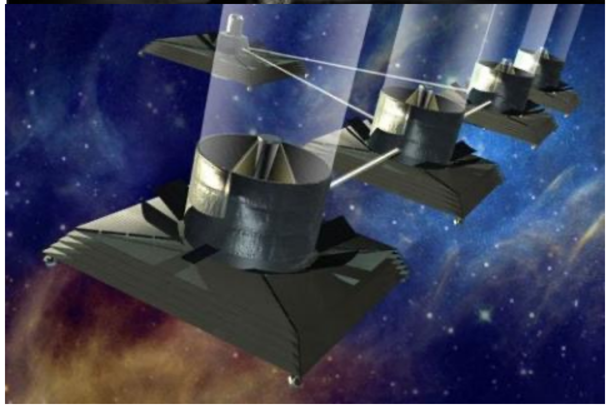
Autonomous Robot Teams for Planetary Exploration

Robot teams are uniquely well-positioned to:

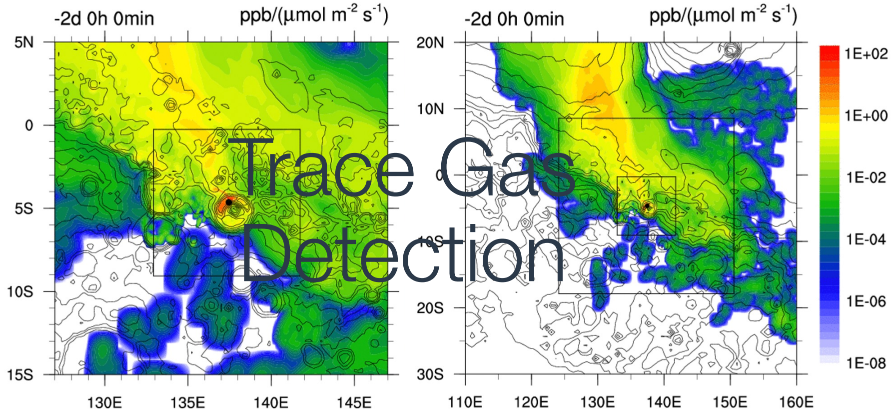
- Collect distributed measurements
  - Seismology
  - Weather and climate
  - Ground-penetrating radar
  - Distributed apertures (in orbit)
- Perform exploration and mapping
- Provide system-level resilience

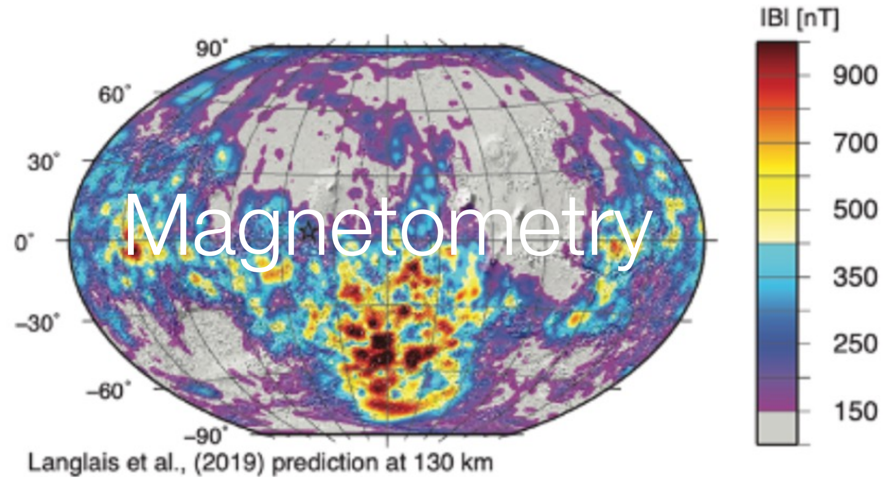
Current operations paradigms do not scale: autonomy is enabling

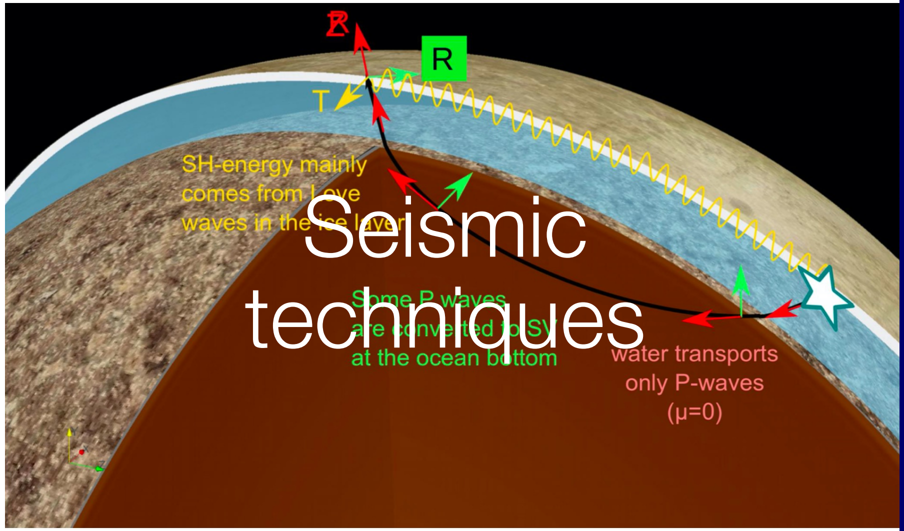


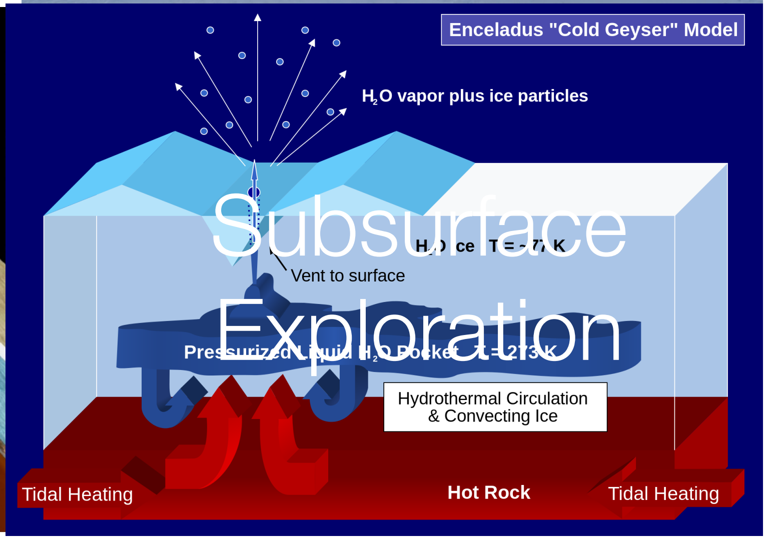


Multi-agent systems enable high-priority science



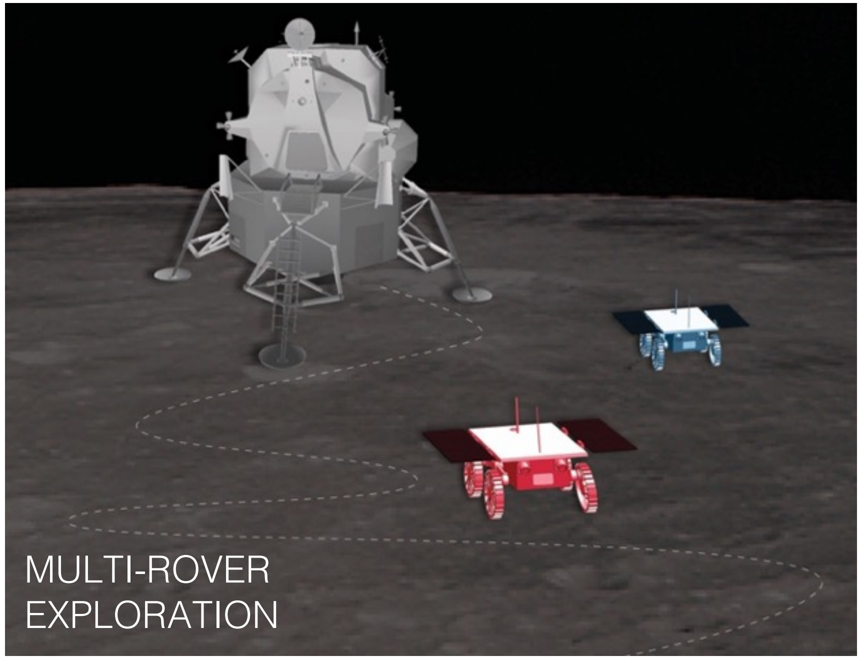


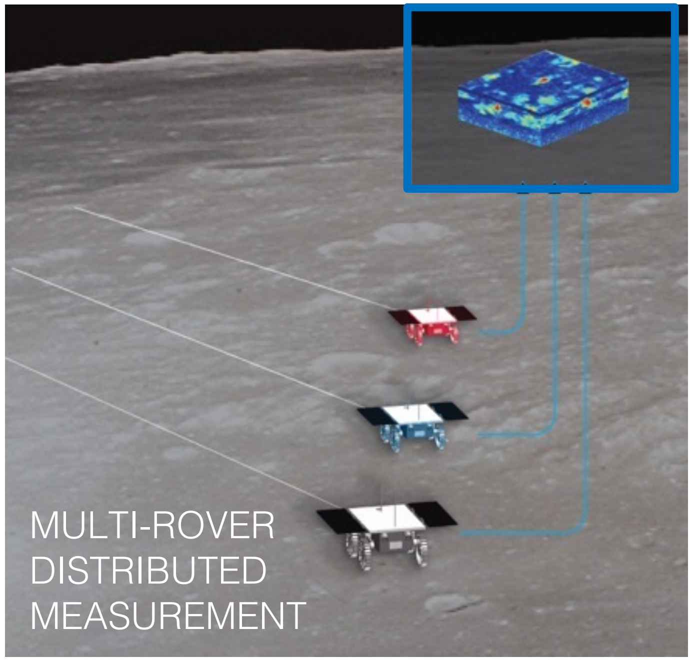


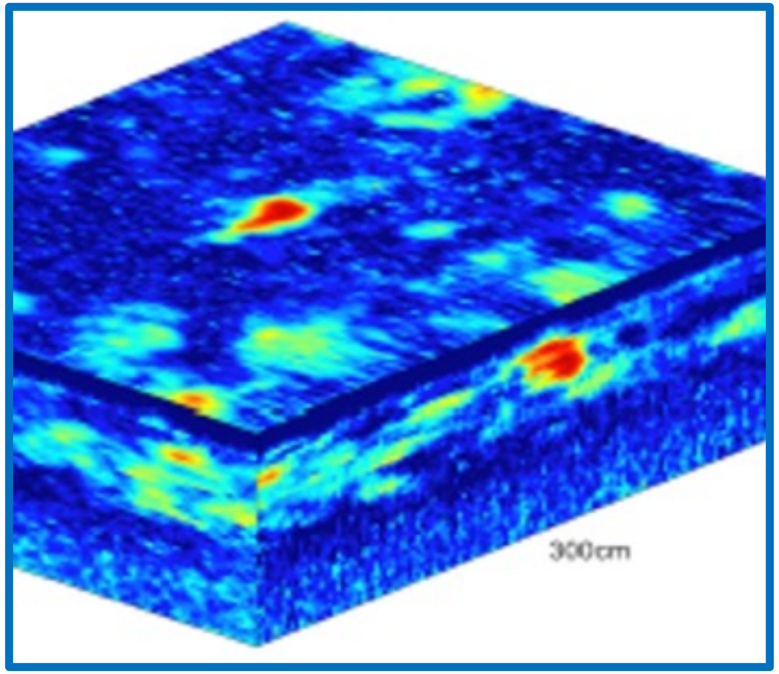


CADRE's Goal

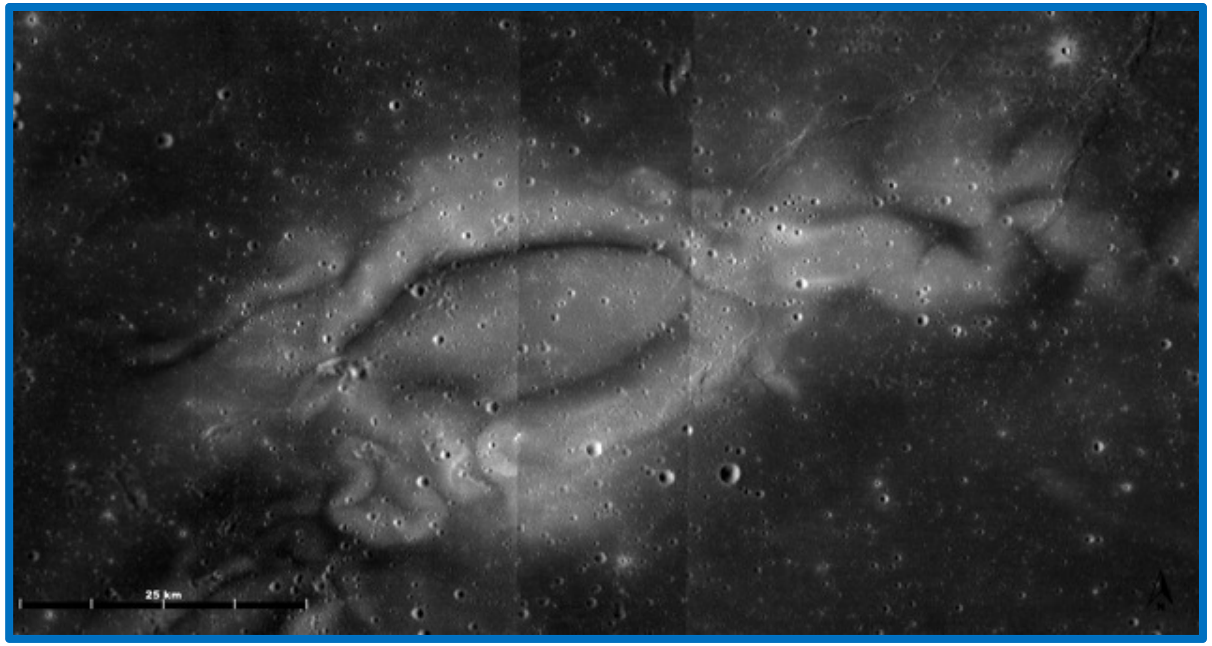
DEMONSTRATE FIRST AUTONOMOUS EXPLORATION AND DISTRIBUTED MEASUREMENT WITH A TEAM OF ROVERS ON ANOTHER PLANETARY BODY







Lunar Technology Demonstration



- CADRE is a flight technology demonstration manifested as a payload on CP11 (CLPS)/ Intuitive Machines (IM-3) mission, targeting **launch in the next year** on Falcon-9.

- CADRE is funded from Space Technology Mission Directorate (STMD) under Game Changing Development (**GCD**).

- Destination: Reiner Gamma is known for its mysterious lunar swirls, where dark and light regolith mix.

- Three rovers will work together to explore the surface nearby during a single Lunar day (about **10 Earth days**).

From concept to flight project

https://www.granttremblay.com/blog/trls

Raising the TRL

NASA Technology Readiness Levels

TRL 1: What if there were Unicorns

TRL 2: We have drawn a Unicorn

TRL 3: `unicorn_v8_final_final.cad`

TRL 4: We have placed a horn on a horse in our lab

TRL 5: We took the horse outside

TRL 6: We're now calling the horse a Unicorn

TRL 7: We're pretty sure the Unicorn might survive if we launch it into space

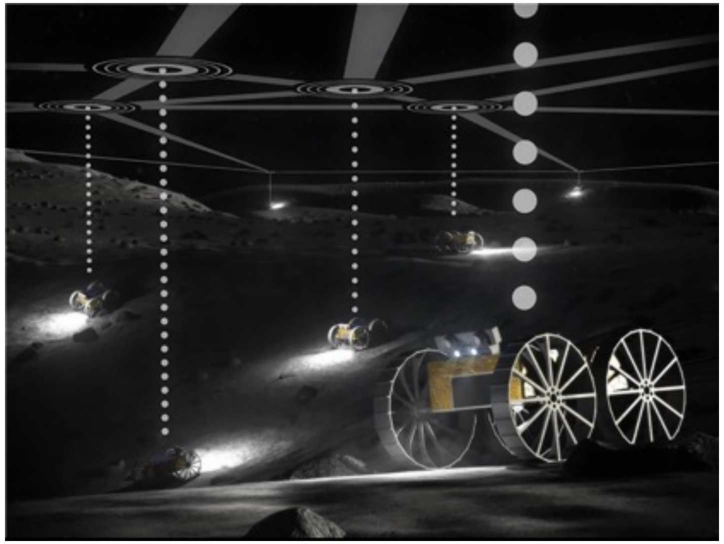
TRL 8: omg it survived

TRL 9: Our reference design incorporates high-heritage Space Unicorns

Technology Development

Development of a small rover with unique mobility capabilities (PUFFER)

Development of multi-agent autonomy for cooperative exploration in unmapped environments (A-PUFFER)



Demonstration of multi-agent autonomy on the Moon (CADRE)

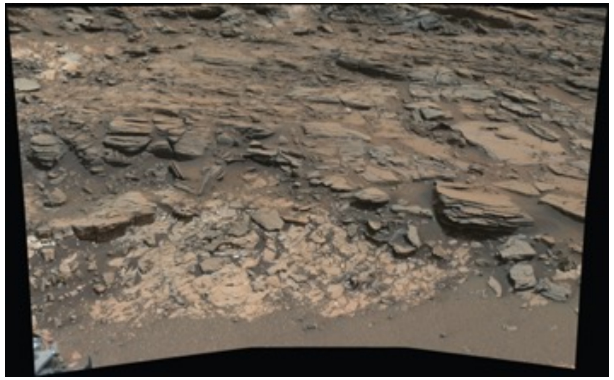
Game Changer: Access to Science-Rich Extreme Terrains

NASA/JPL-Caltech/MSSS:
https://mars.nasa.gov/resources/7507/curiosity-self-portrait-at-big-sky-drilling-site/

"Apollo 14 Cone Crater Boulders",
https://commons.wikimedia.org/wiki/File:Apollo_14_cone_crater_boulders.jpg

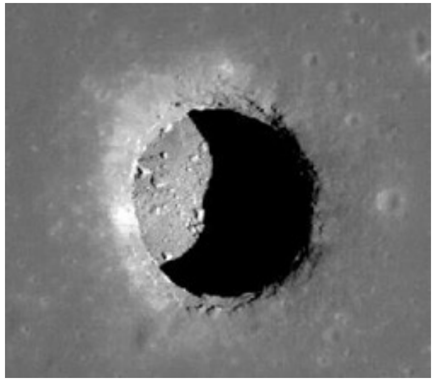
Eroded Rock Features

Surface and Subsurface Features



Steep Stratified Slopes

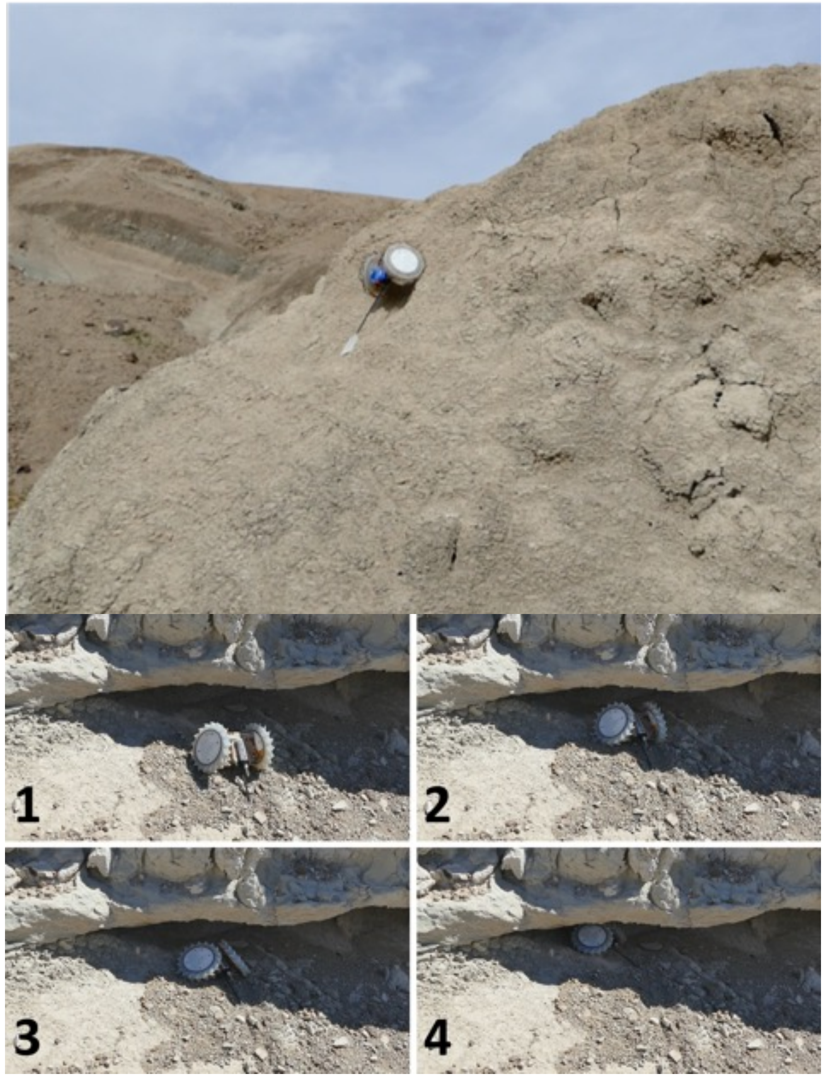
NASA/JPL-Caltech/MSSS: https://mars.nasa.gov/resources/7312/geological-contact-zone-near-marias-pass-on-mars/



Lunar Pit

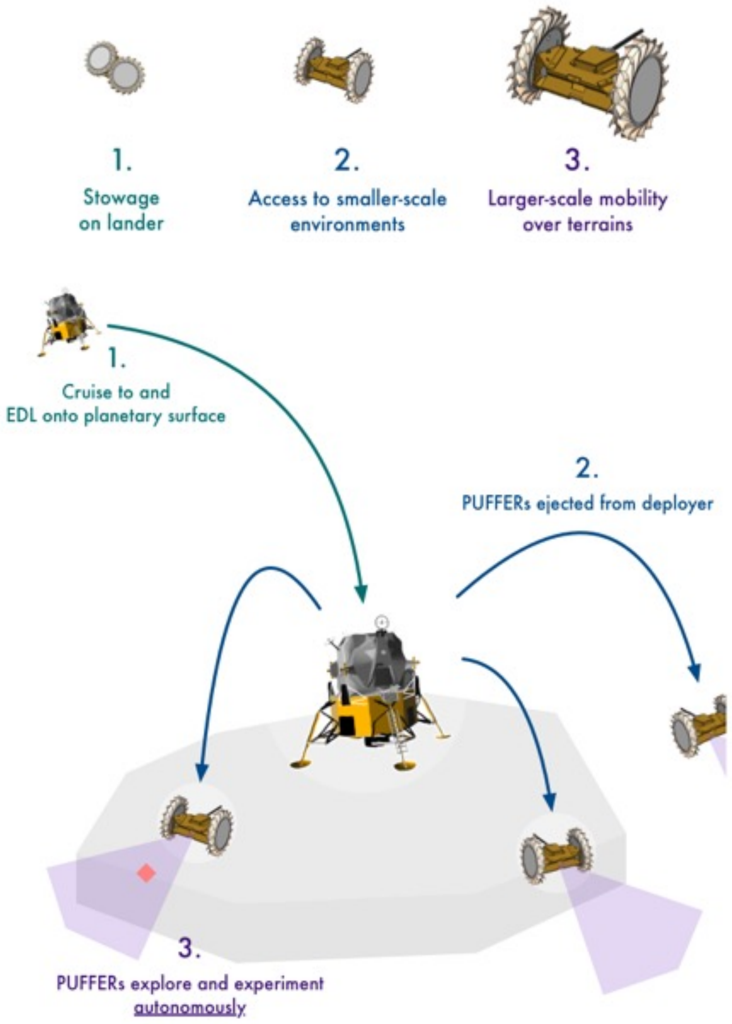
NASA/GSFC/Arizona State University,
http://lroc.sese.asu.edu/images

PUFFER Capabilities



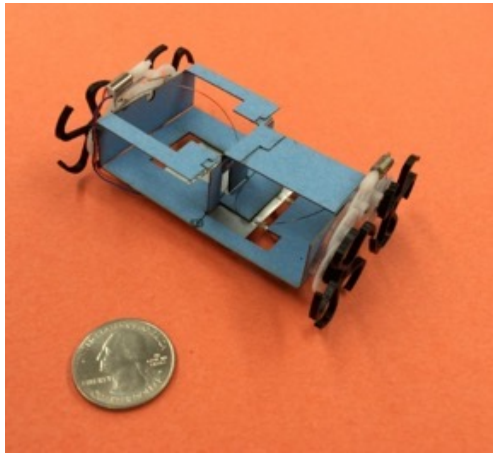
- Small, foldable robotic platforms for accessing extreme terrain (steep slopes, low clearance overhangs).
- Low- mass, volume, cost to enable deployment of larger number of rovers per mission.
- New multi-rover autonomy to achieve scalability, ”strength in numbers”

Importance



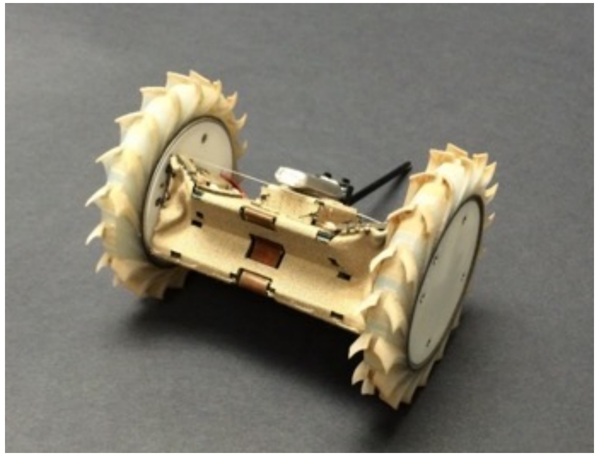
- Foldable robotic platforms with high stowed-to-deployed ratios, but scalable on-demand
- Success in high-risk (high-uncertainty) extreme environments with limited comms via hands-off autonomy
- Provides quicker-to-develop, lower-cost COTS solutions to flight (à la MarCOs, Mars Heli)

Progression of PUFFER Prototypes



v0 - 2014

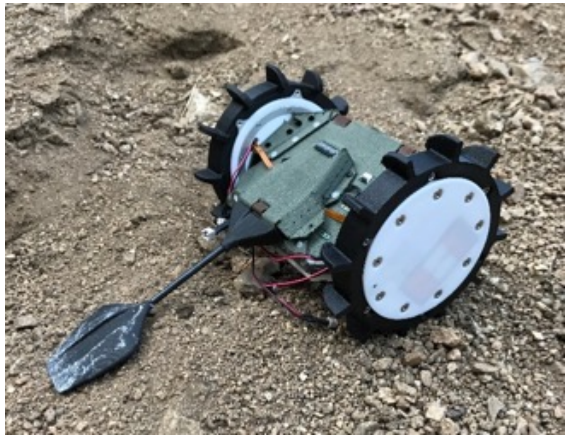
2015



2015

2016

“Snow PUFFER” - 2017

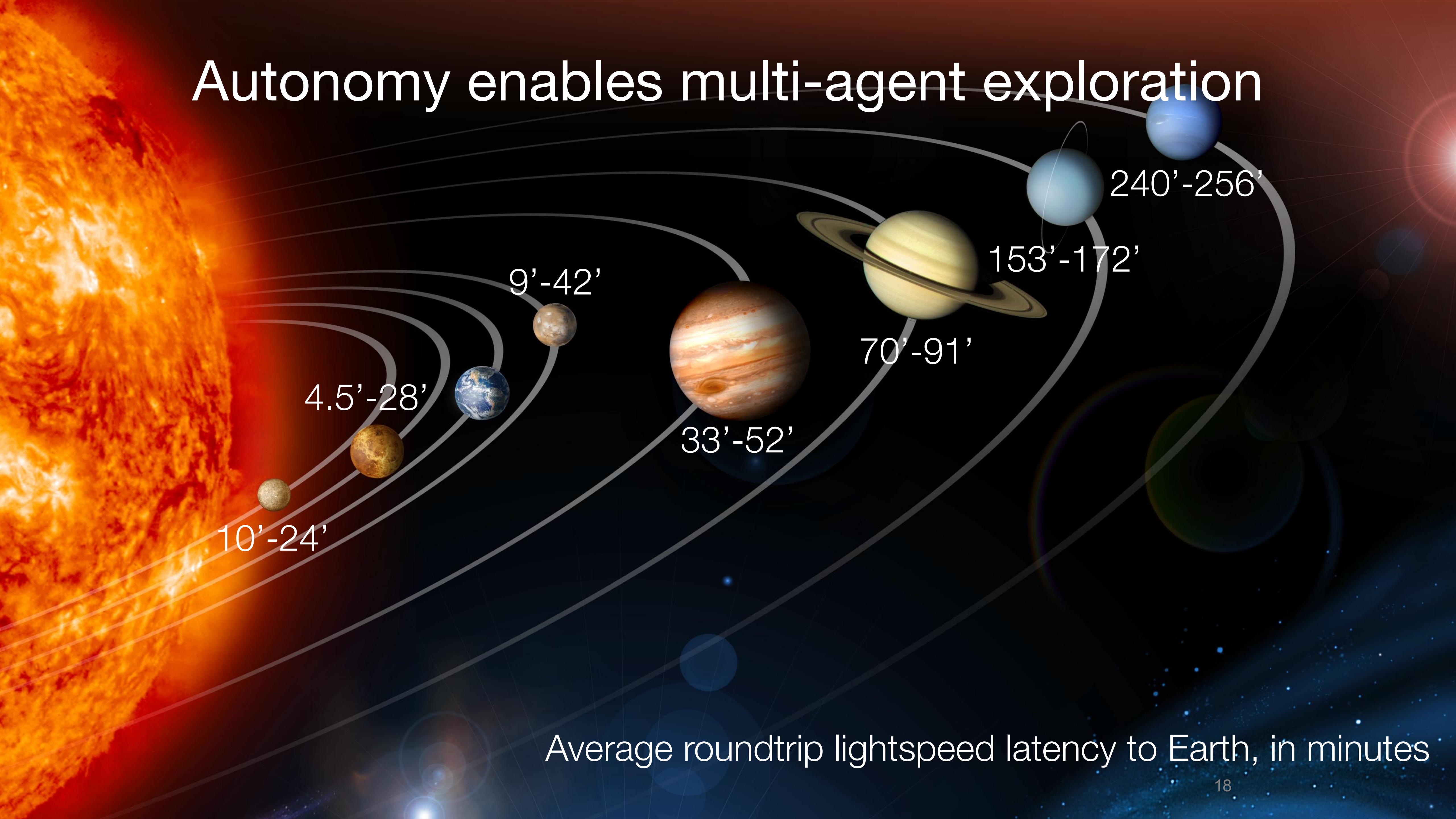


2018

v3.5 - 2019

v4.0 - 2020

Autonomy enables multi-agent exploration

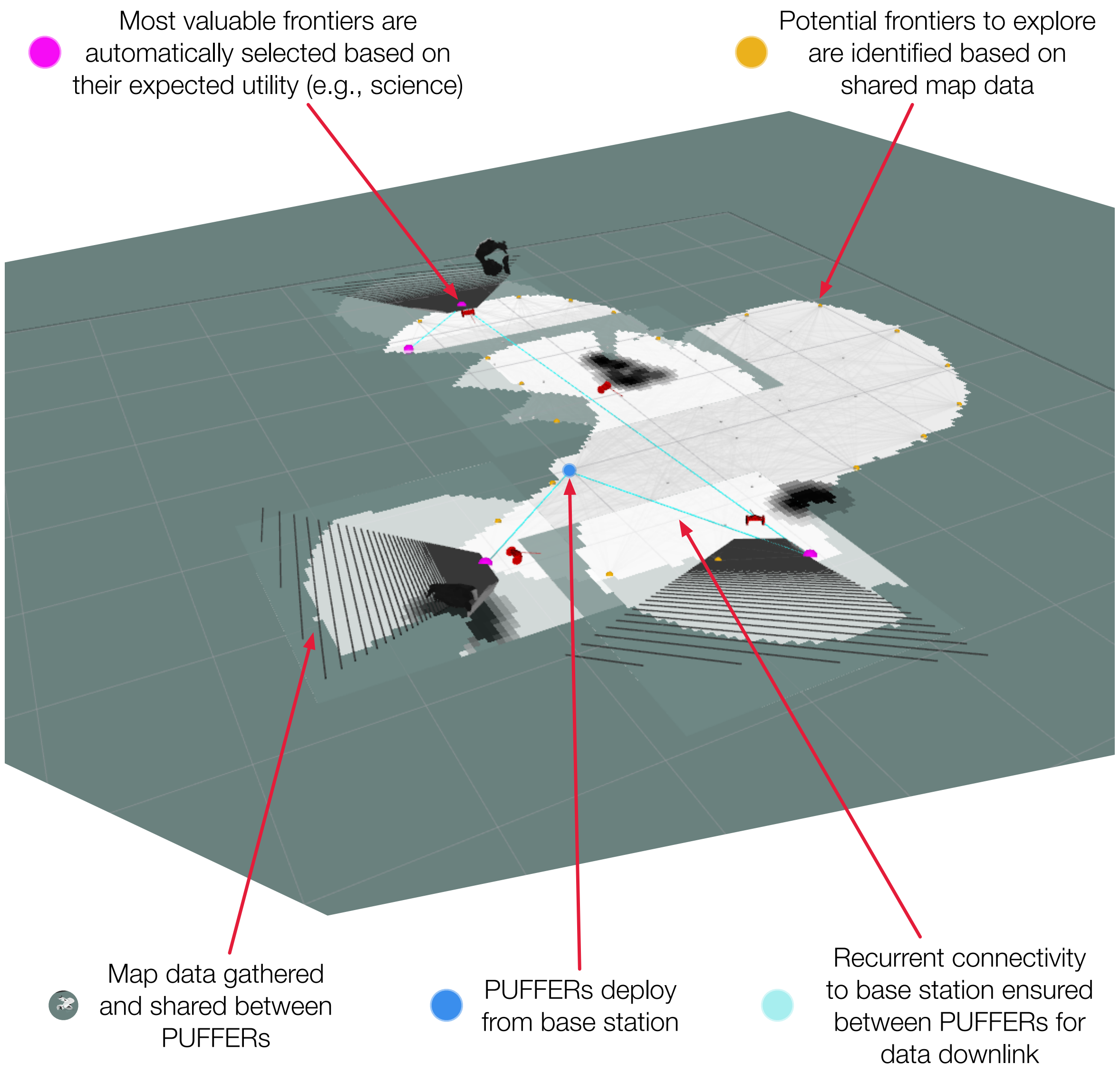


Average roundtrip lightspeed latency to Earth, in minutes

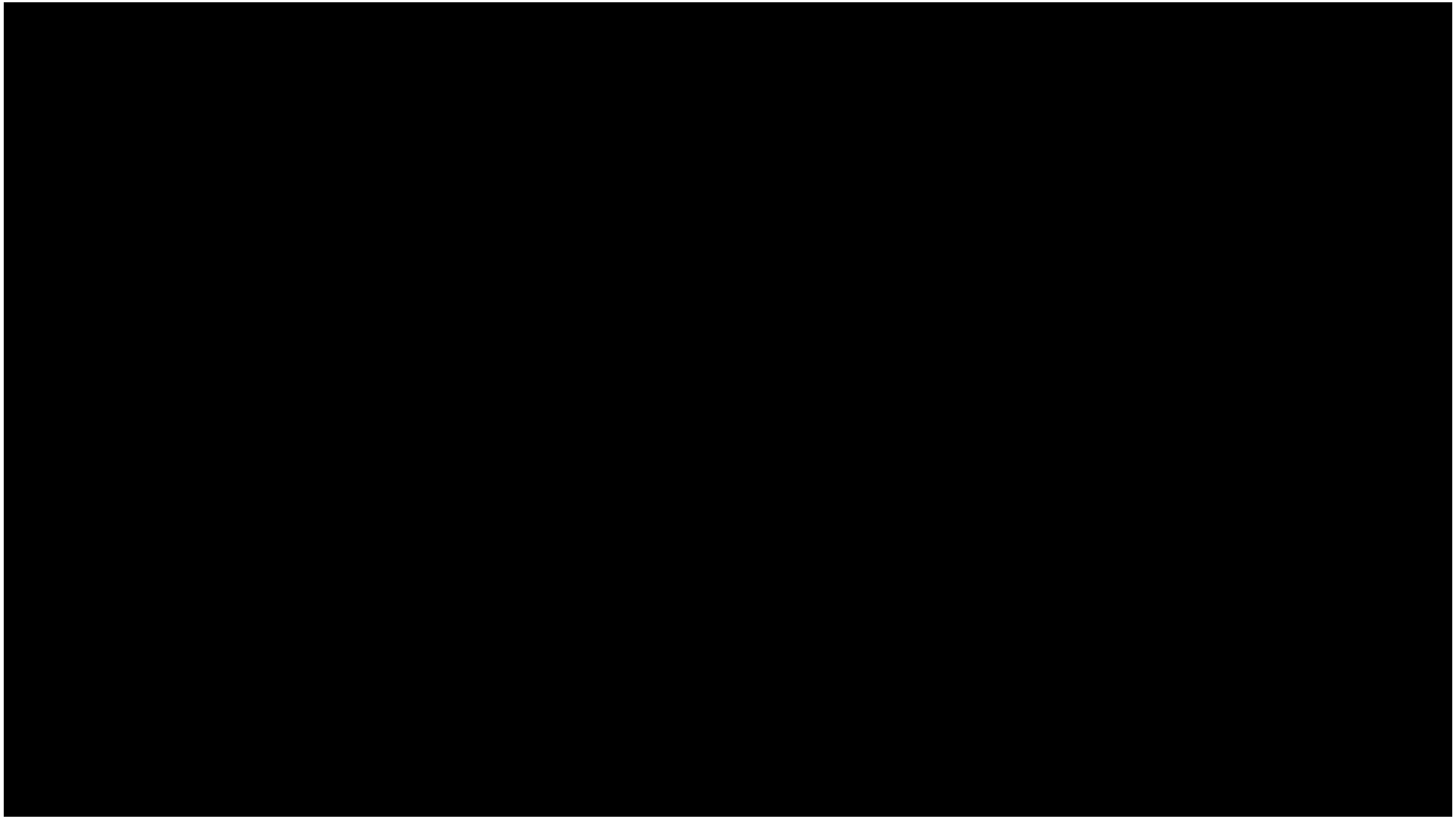
Autonomy enables multi-agent exploration

Exploration

1. Multiple PUFFERs map an unstructured area
2. PUFFERs operate autonomously and share local maps and other data with base station when communication is available
3. Base station constructs global map by merging all robots' local maps
4. Base station sets individual robot targets for exploration or science measurements
5. PUFFERs use local information and current global map (if available) to navigate to those targets

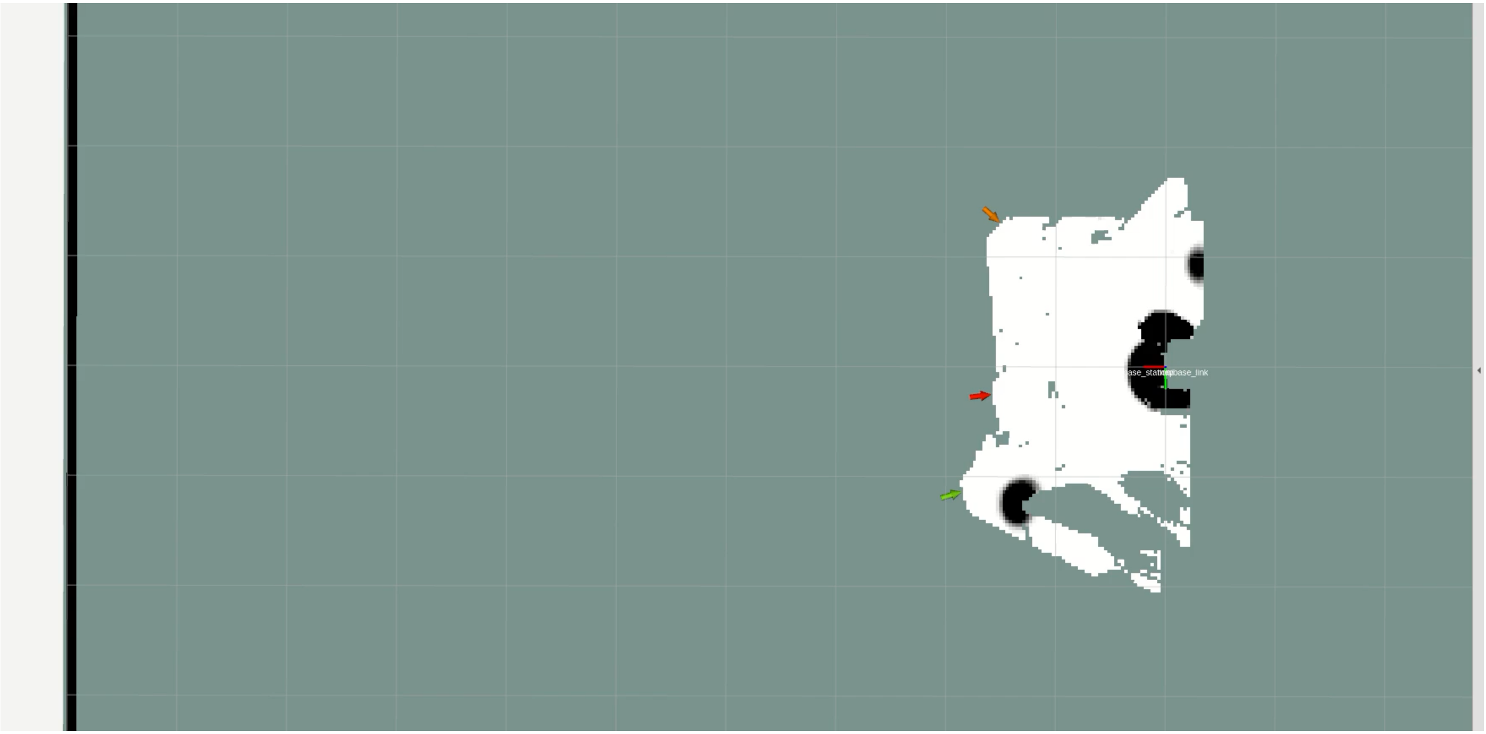


Solitary autonomy at first…



…, but then with more PUFFERs!

Cooperative Exploration in Action

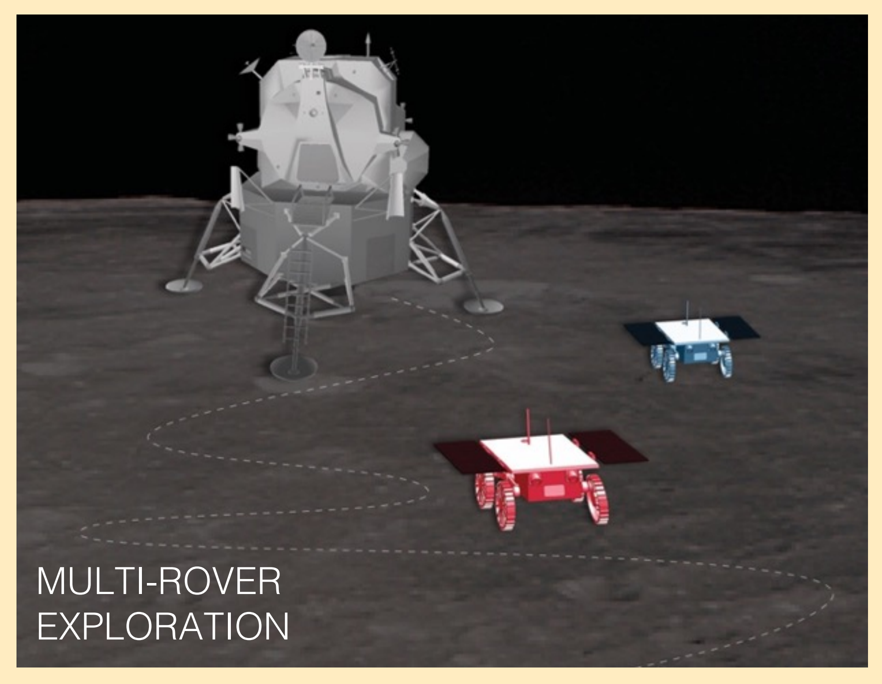


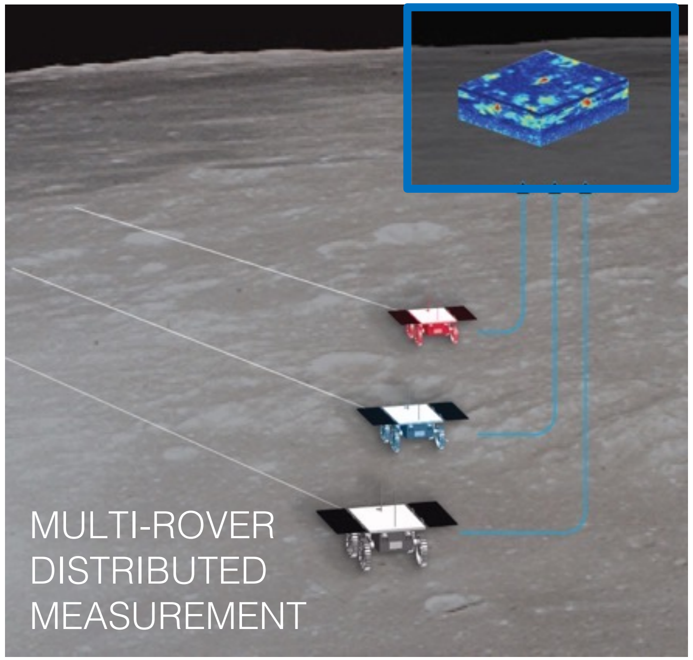
Going to space

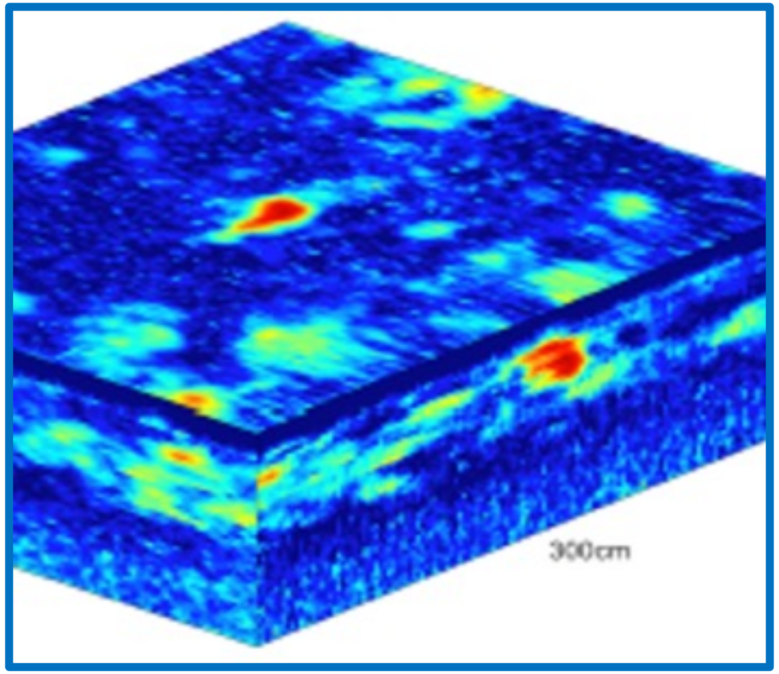
Mission Overview

- CADRE (Cooperative Autonomous Distributed Robotic Exploration) is a NASA STMD Game Changing Development (GCD) project to advance multi-agent autonomy and demonstrate it on the Moon.
- CADRE is manifested as a payload on Intuitive Machine's IM-3 mission, and is headed to Reiner Gamma (Moon's equatorial region)
- CADRE's technology demonstration will focus on cooperative exploration and distributed measurements using multiple ground penetrating radars (GPRs).

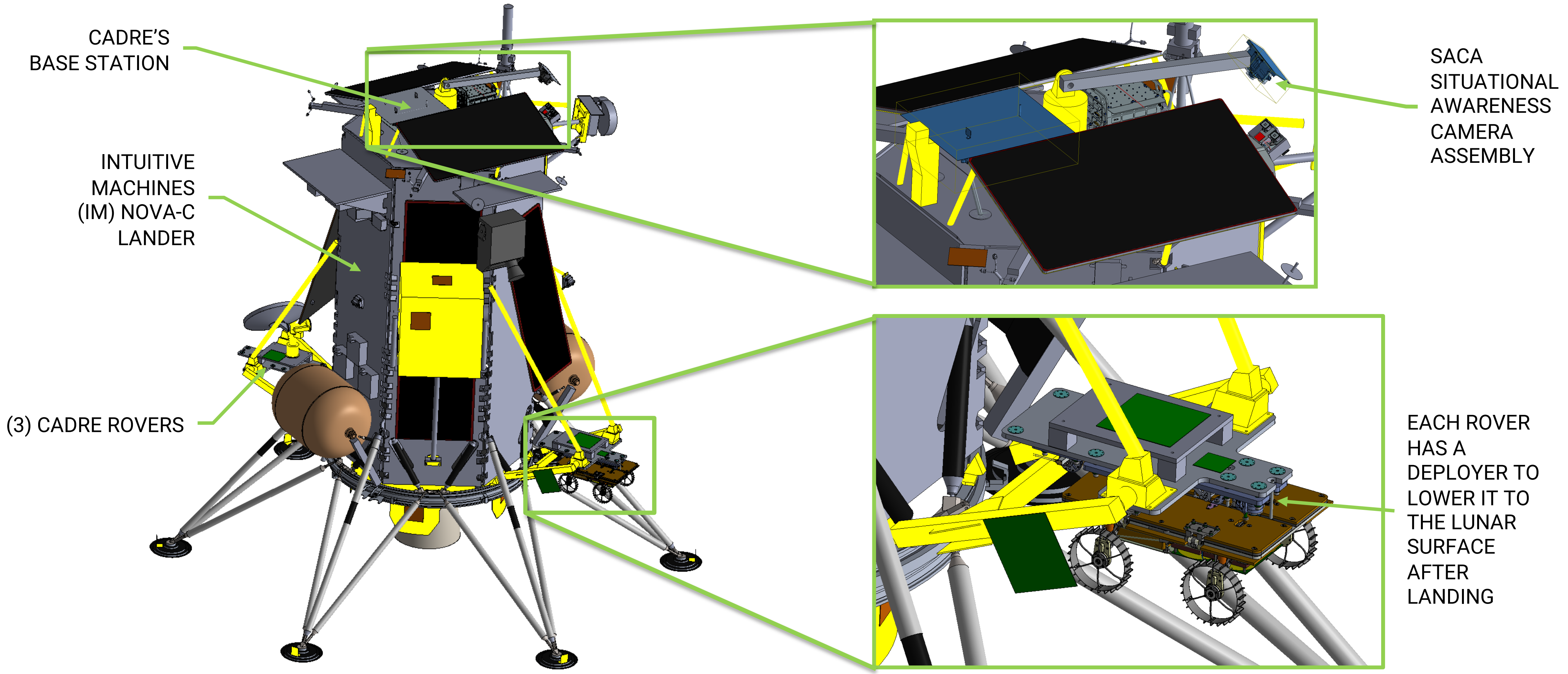
Building this for space!



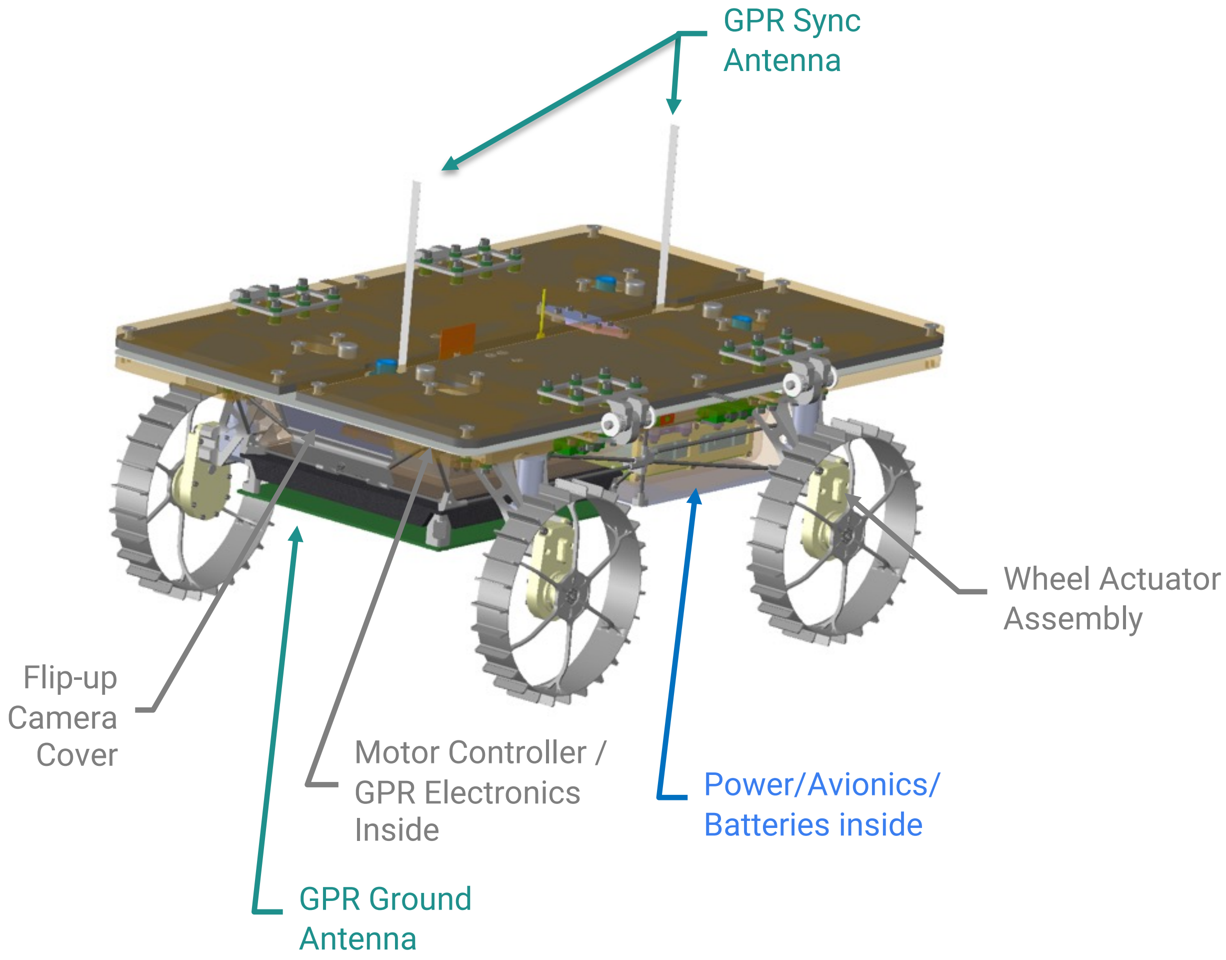




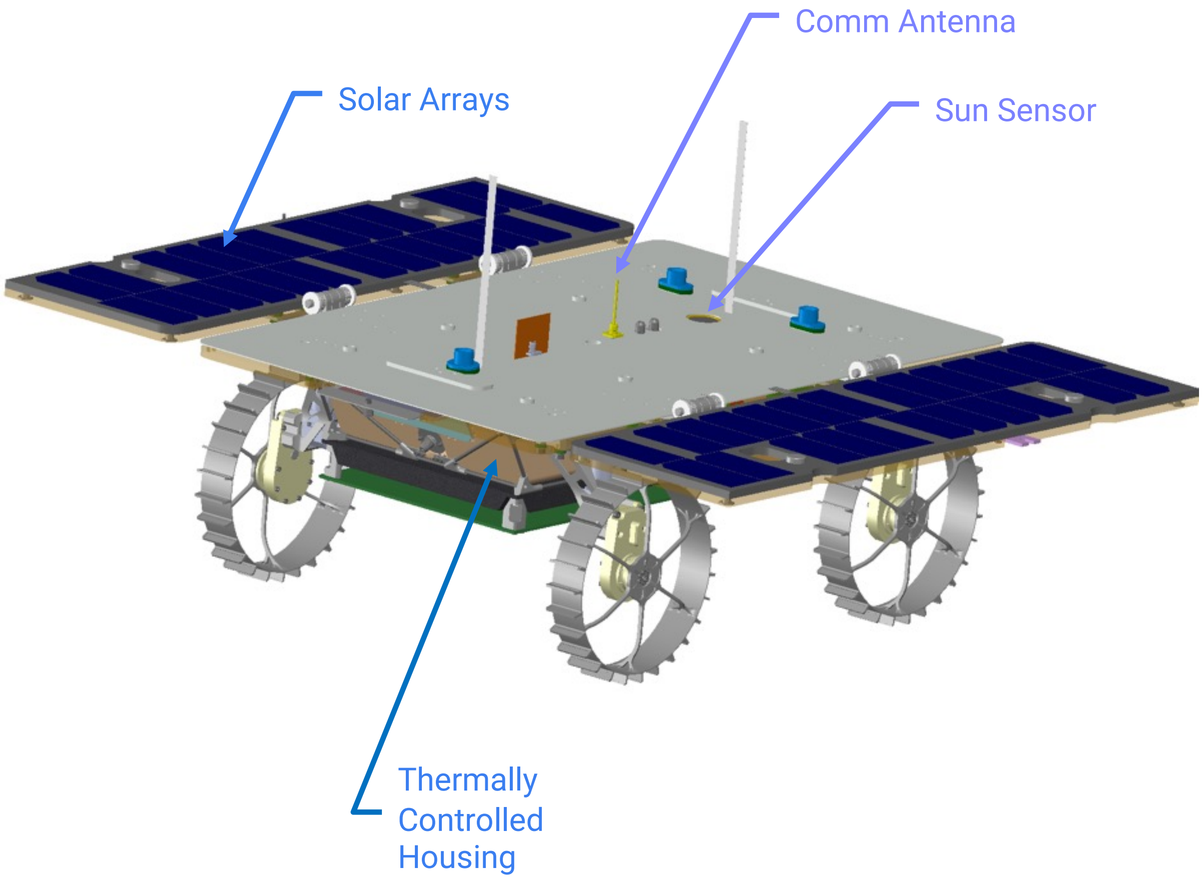
System Overview



Rover Overview



STOWED

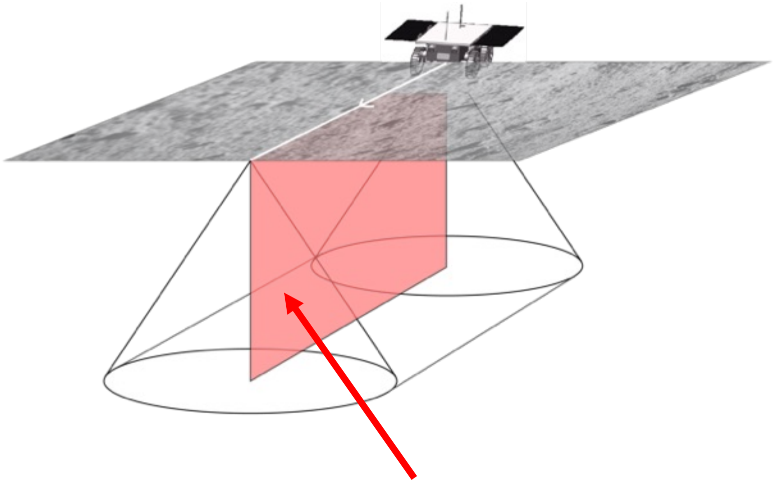


DEPLOYED

Cooperative Distributed Measurements

Today.

Single Monostatic Measurements



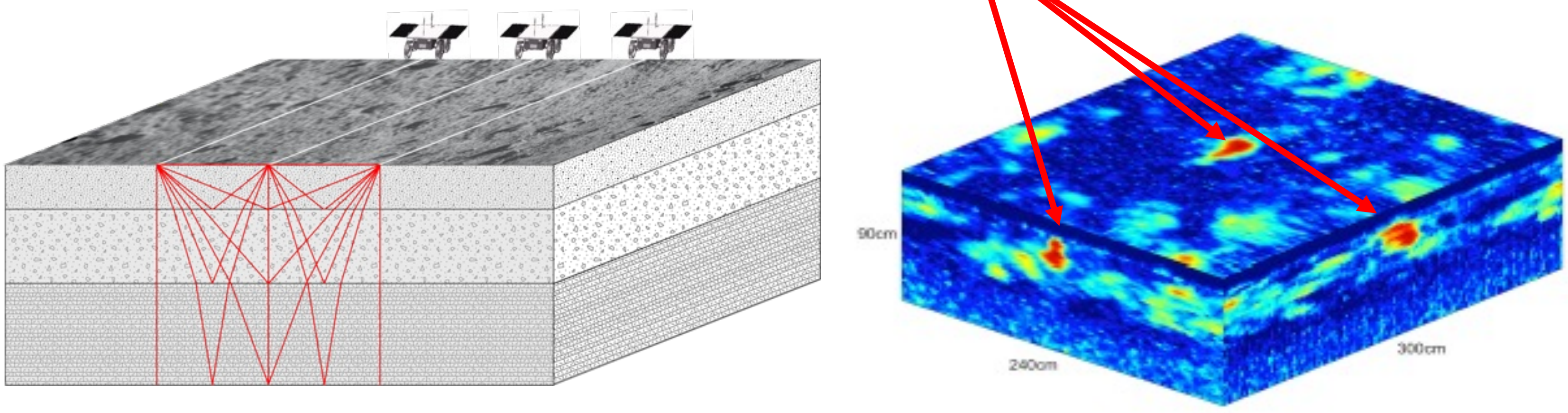
2-DIMENSIONAL DATA ONLY

STATE OF THE ART:
Chang'e 3 and 4 ***Lunar Penetrating Radar*** *on the Moon*
Perseverance ***RIMFAX*** *on Mars*
Zhurong ***Lunar Penetrating Radar*** *on Mars*

Objective.

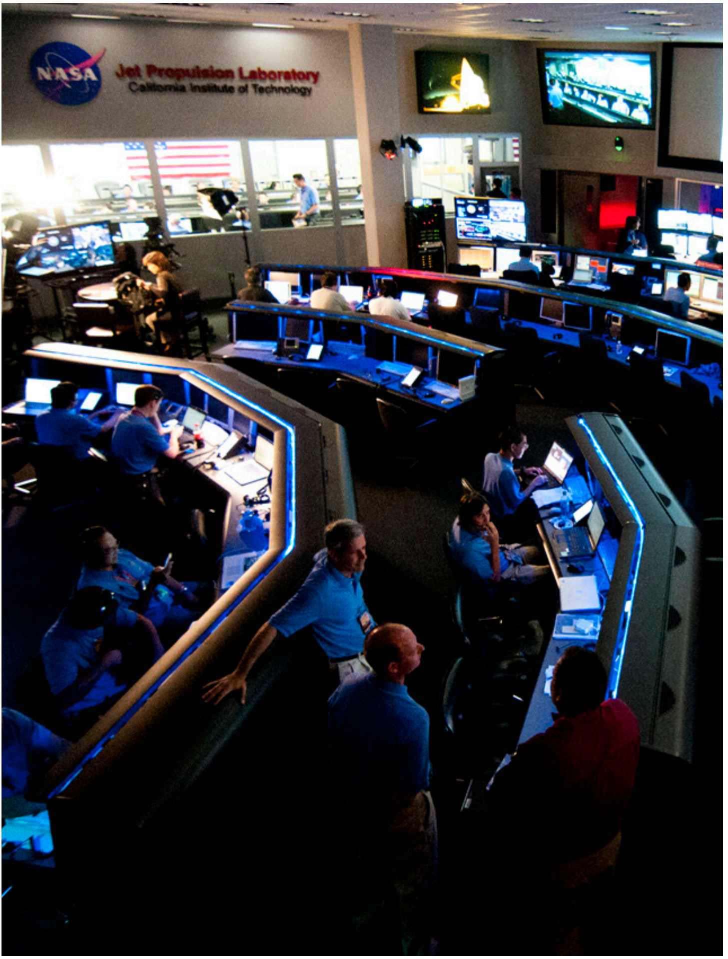
Distributed & Adaptable Multi-Static Measurements

3D SUBSURFACE IMAGERY FROM MULTIPLE GPRs WORKING TOGETHER

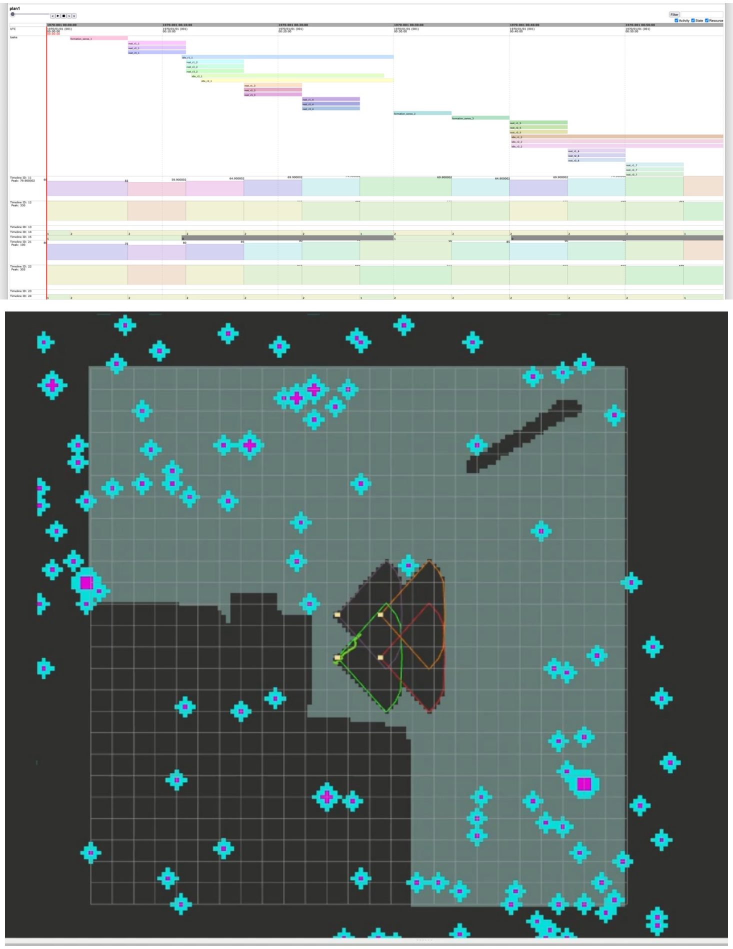


- Rovers have to navigate across the lunar surface in a specific formation (separation dictates measuring depth) and maintain this formation within a certain threshold (derived from required SNR)

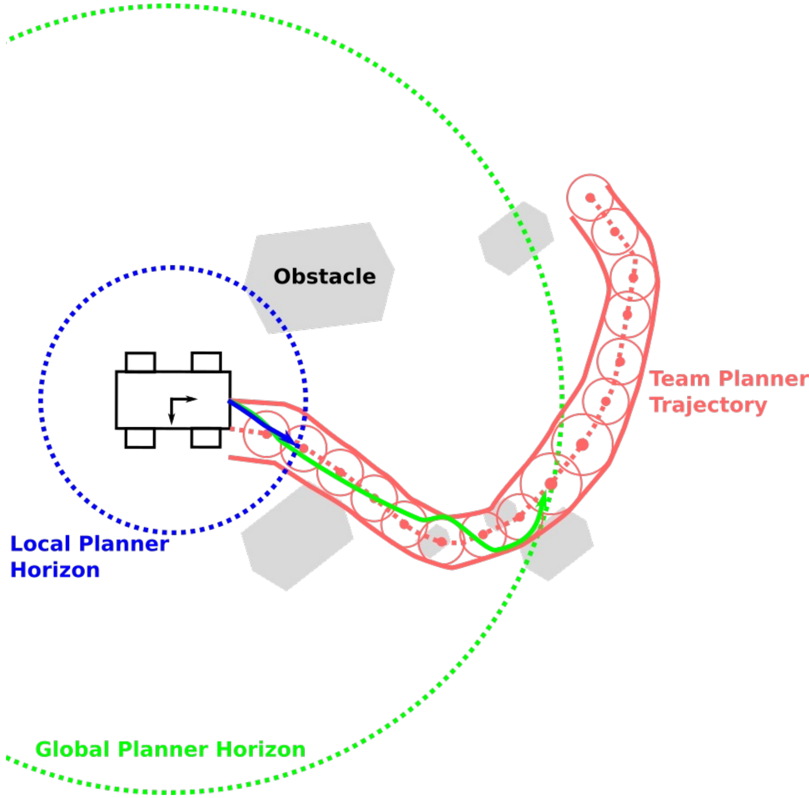
The Autonomy Stack



Ground operators
Which region to explore?

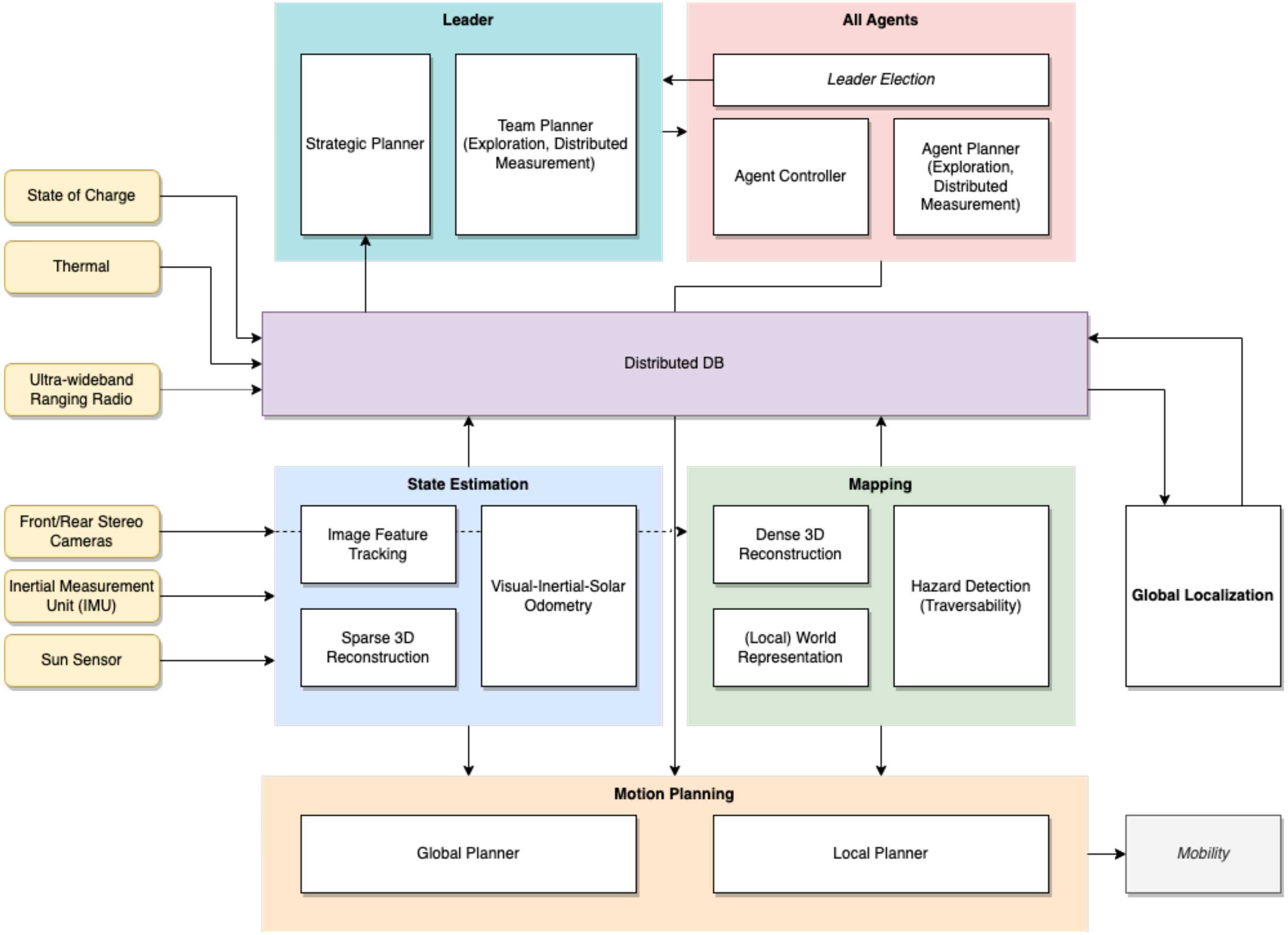


Planning, Scheduling & Execution
Where do we go now?



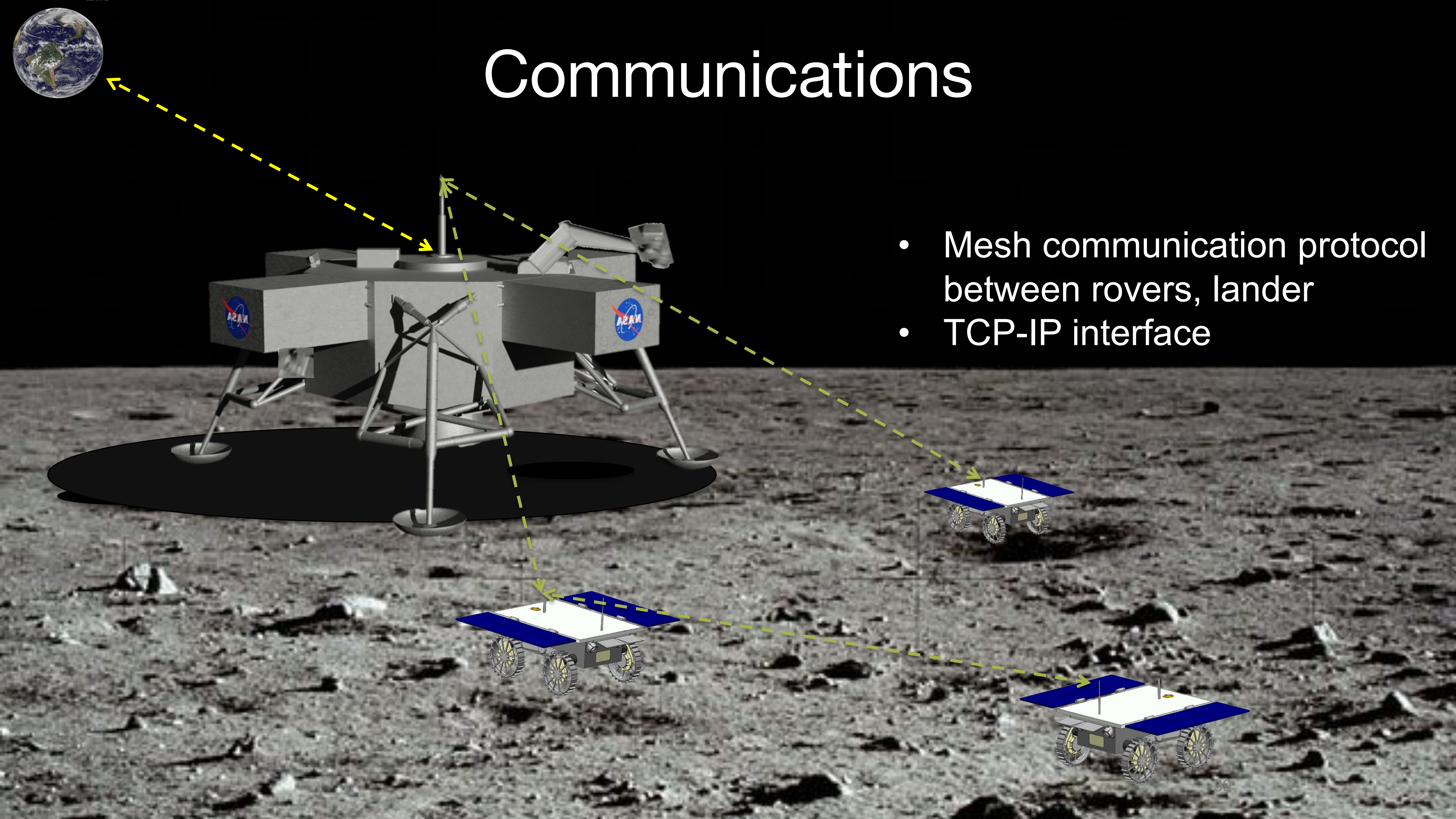
Guidance, Navigation & Control
How do I get there?

Software Architecture



- Sensors in the flight hardware are handled by the flight software (written in F'), as well as, other core capabilities, such as communication
- Autonomy components are implemented in C++, integrated with F', and run on the ModalAI VOXL computer running non-realtime Linux
- Autonomy architecture is hybrid, where planning is centralized on an elected leader, while execution is distributed.

Communications



- Mesh communication protocol between rovers, lander
- TCP-IP interface

Planning, Scheduling, and Execution

Architecture

1. How do we coordinate?
2. Who (if any) is the leader?
3. Where are the decisions made?

Algorithms

4. When do we drive, and when do we sleep?
5. How do we explore a region together?
6. How do we perform a distributed measurement?

Planning, Scheduling, and Execution

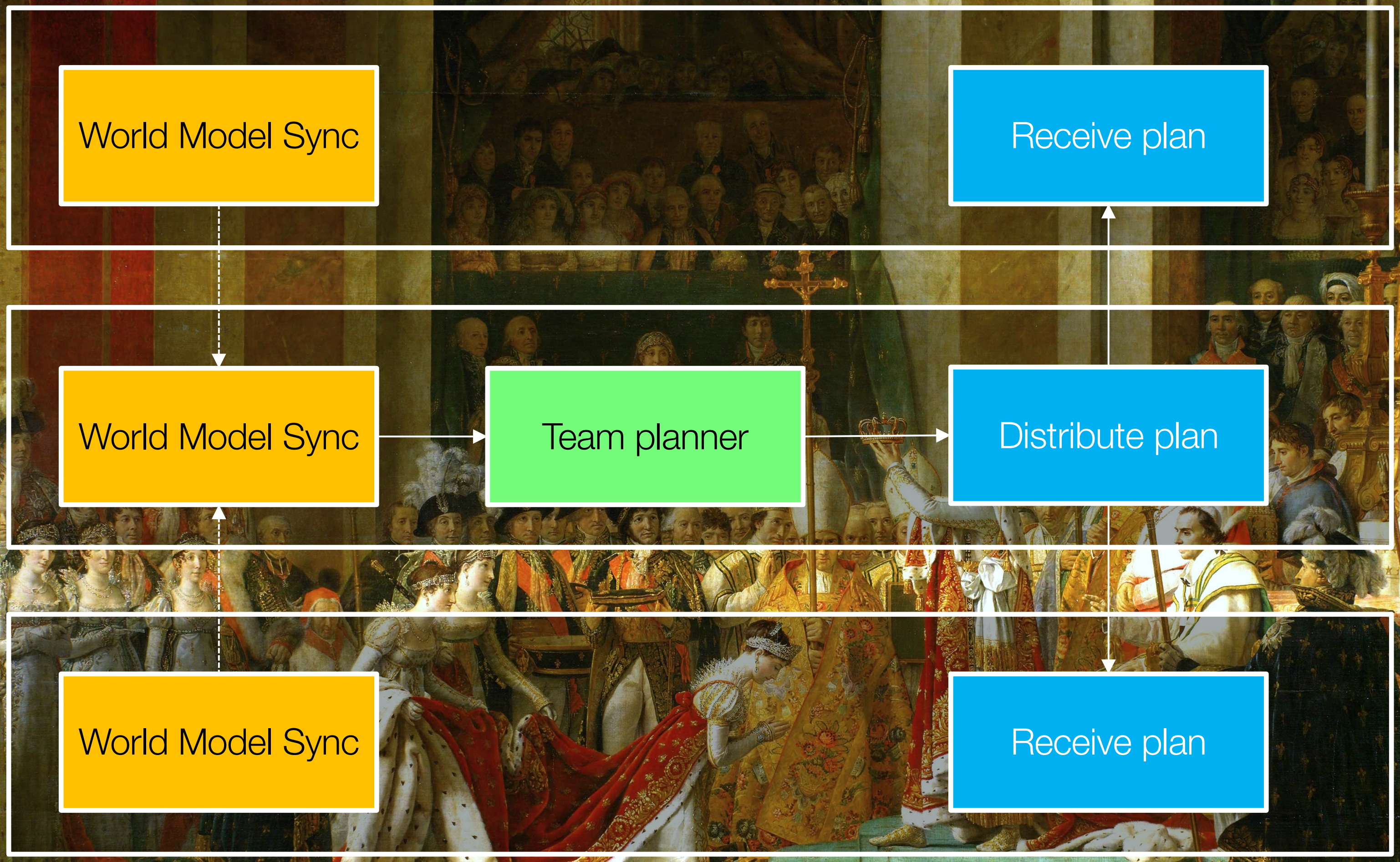
Architecture

1. How do we coordinate?
2. Who (if any) is the leader?
3. Where are the decisions made?

Algorithms

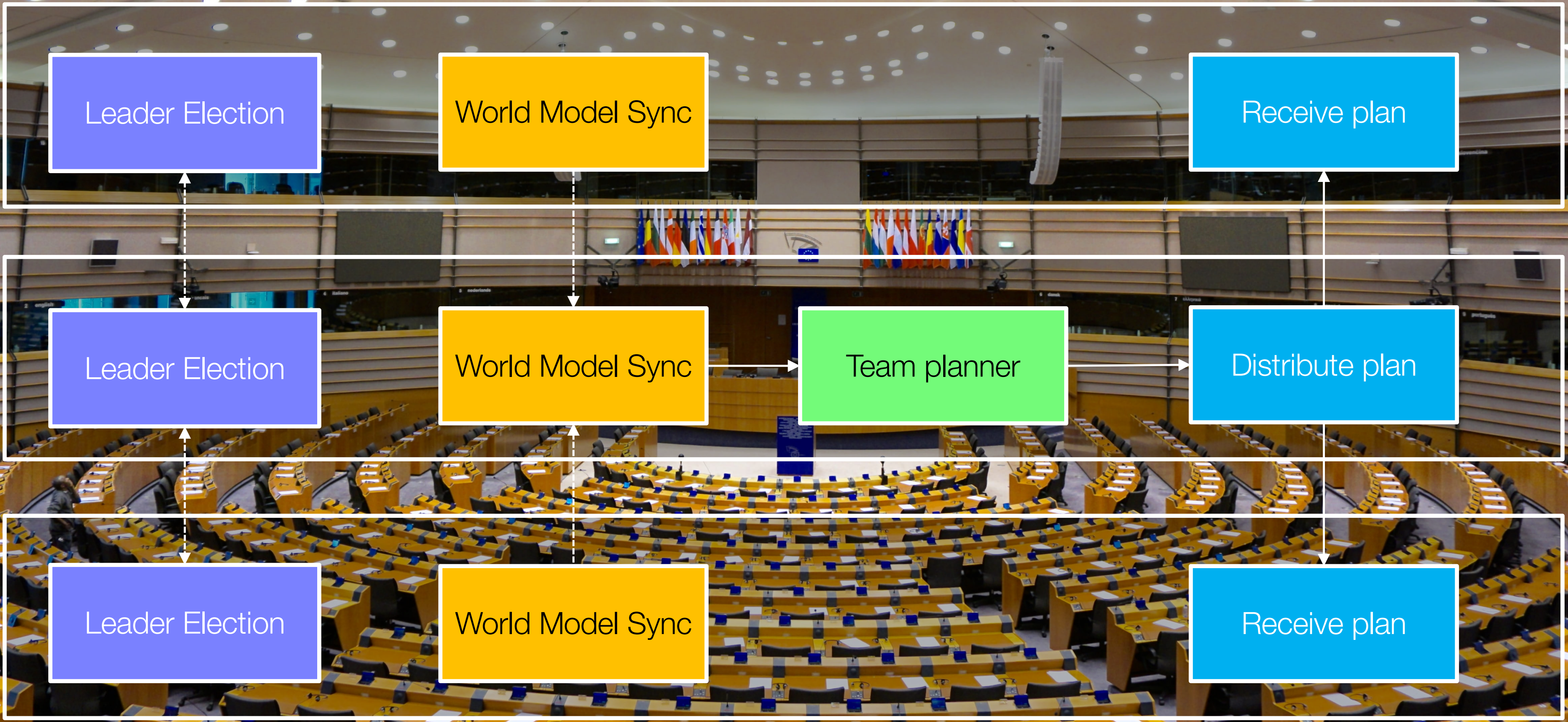
4. When do we drive, and when do we sleep?
5. How do we explore a region together?
6. How do we perform a distributed measurement?

Coordination Architectures : Monarch

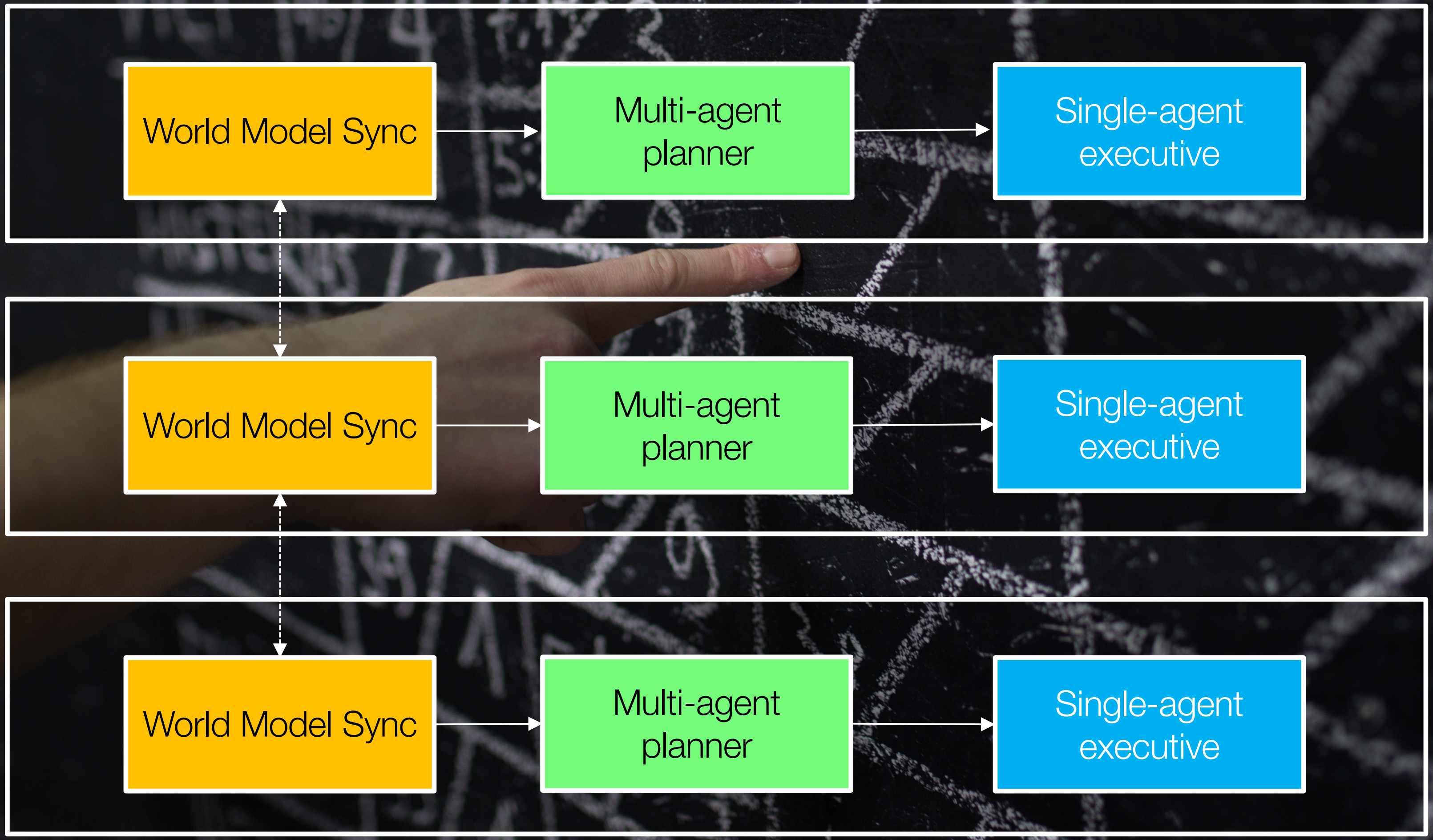


Coordination Architectures : Monarch

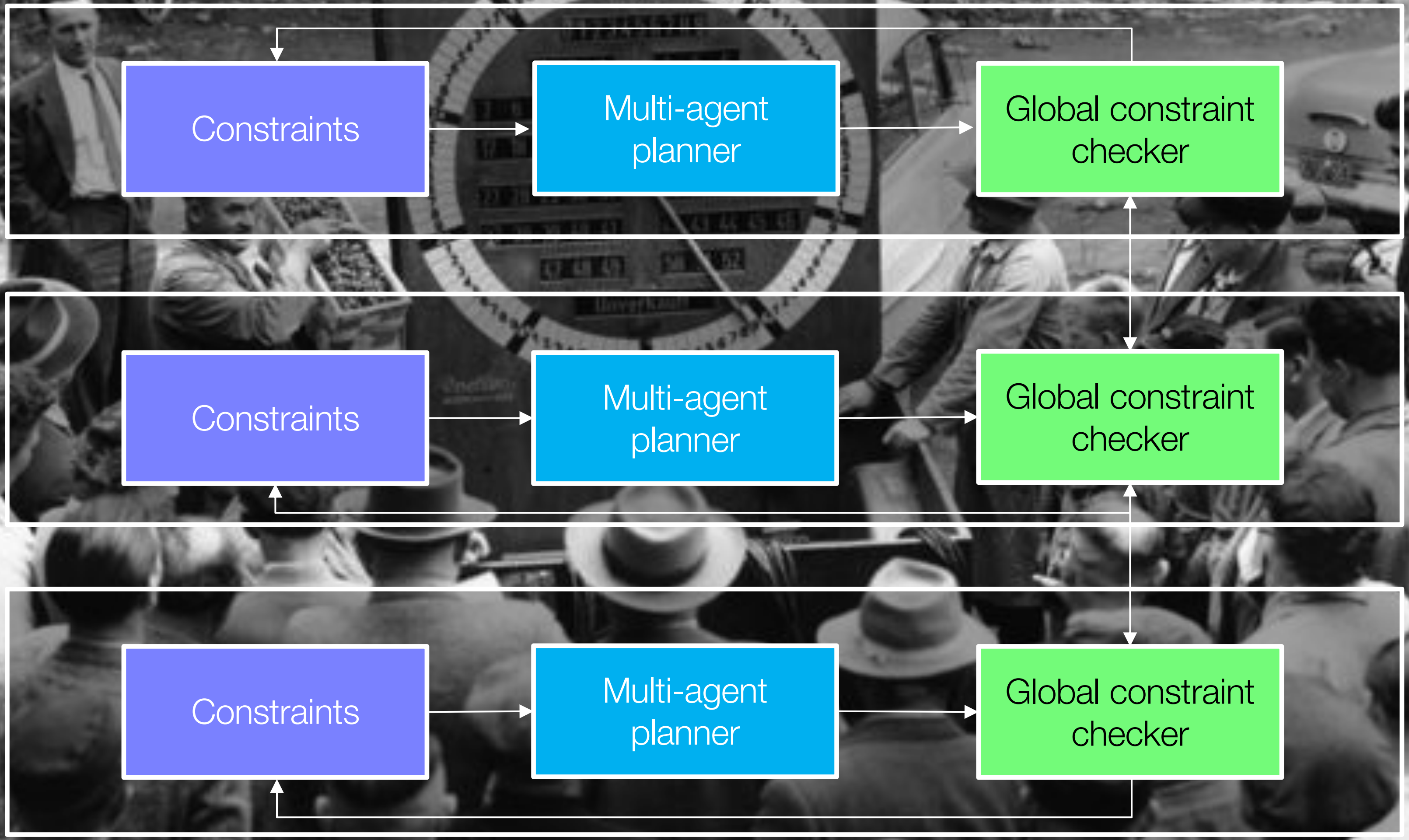
Coordination Architectures : Elected Leader



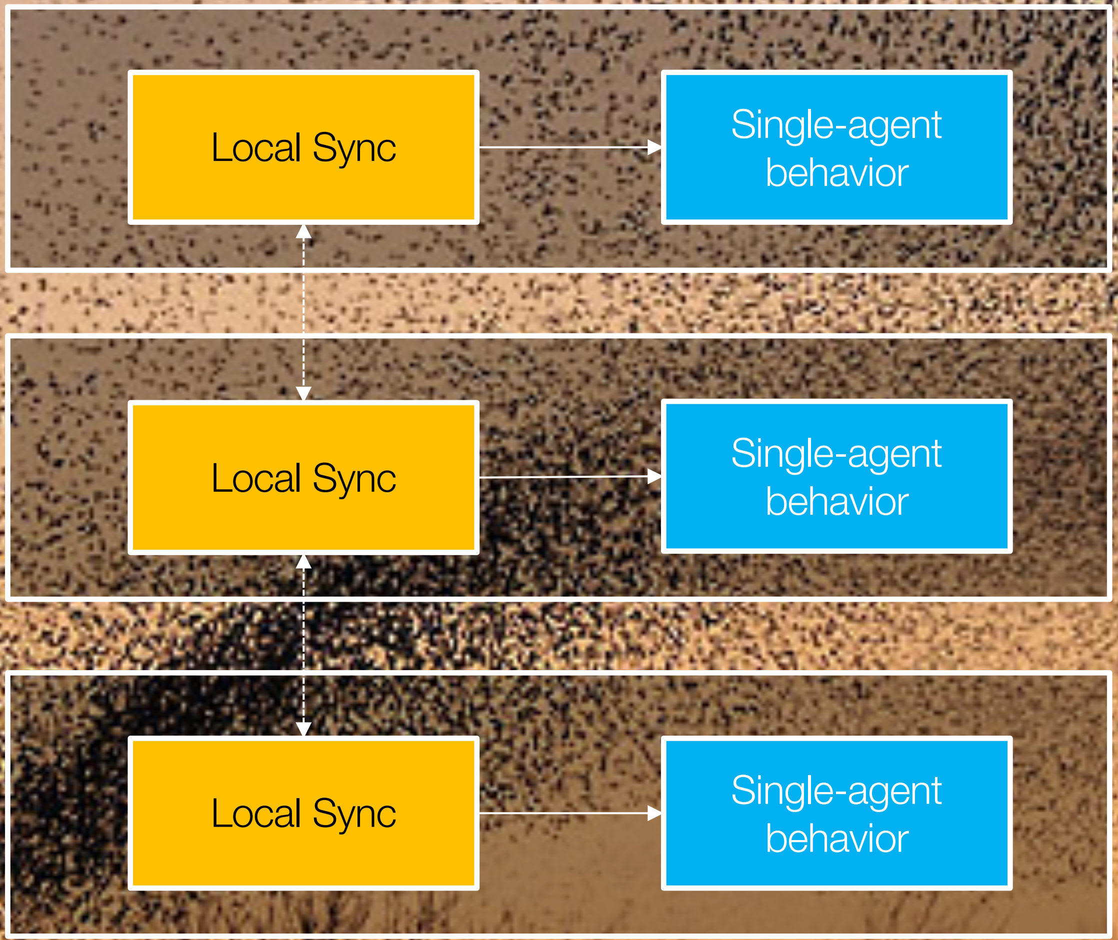
Coordination Architectures : Implicit Coordination



Coordination Architectures : Explicit Coordination

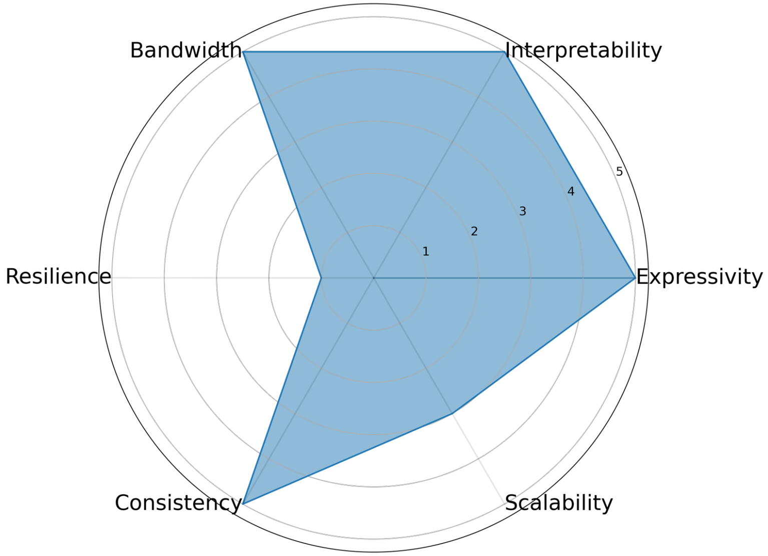


Coordination Architectures : Emerging Behavior

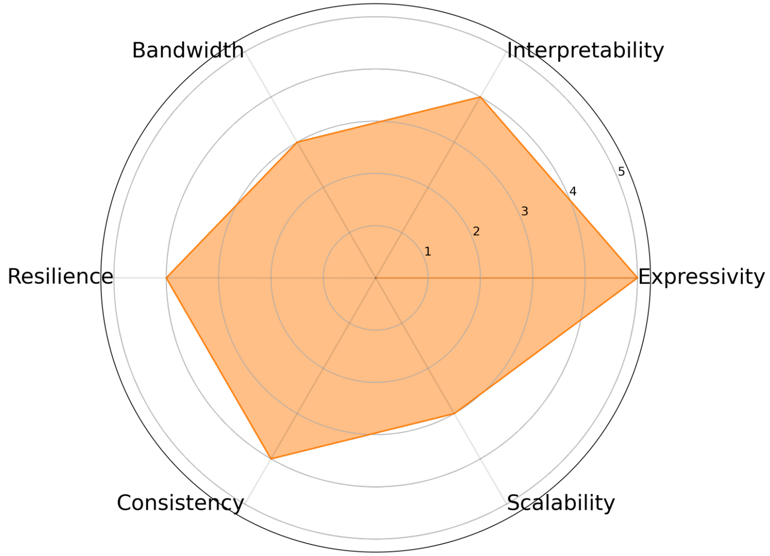


Coordination Architectures

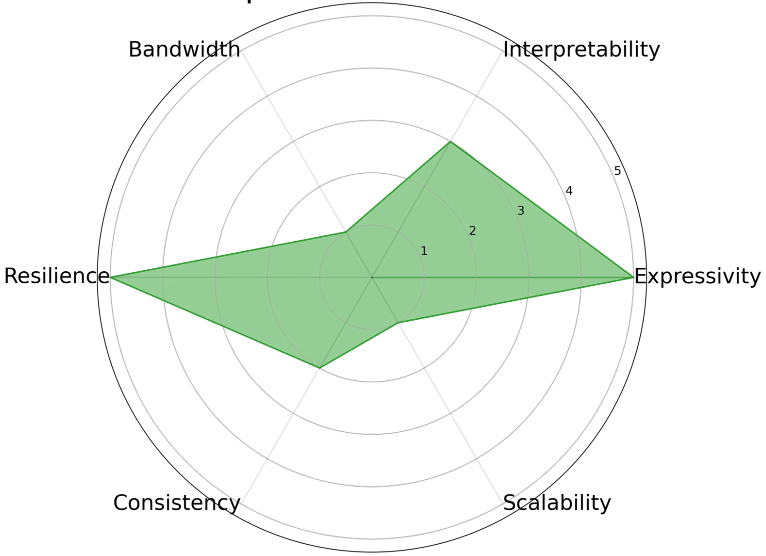
Monarch



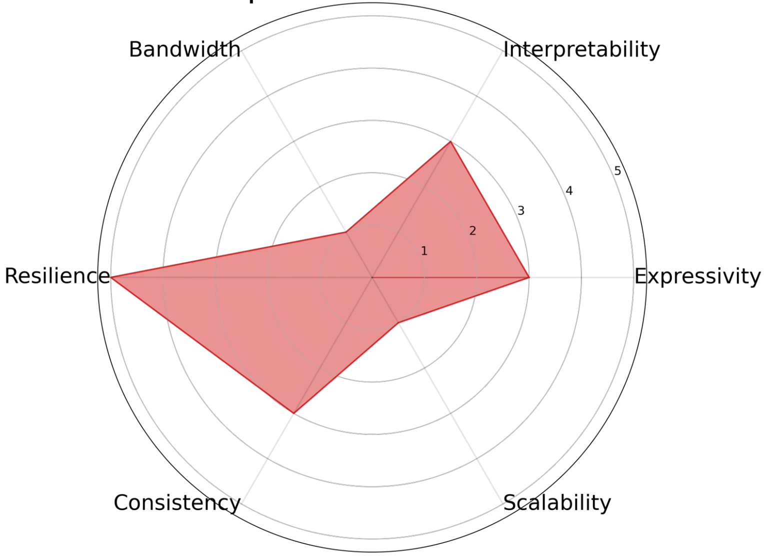
Elected Leader



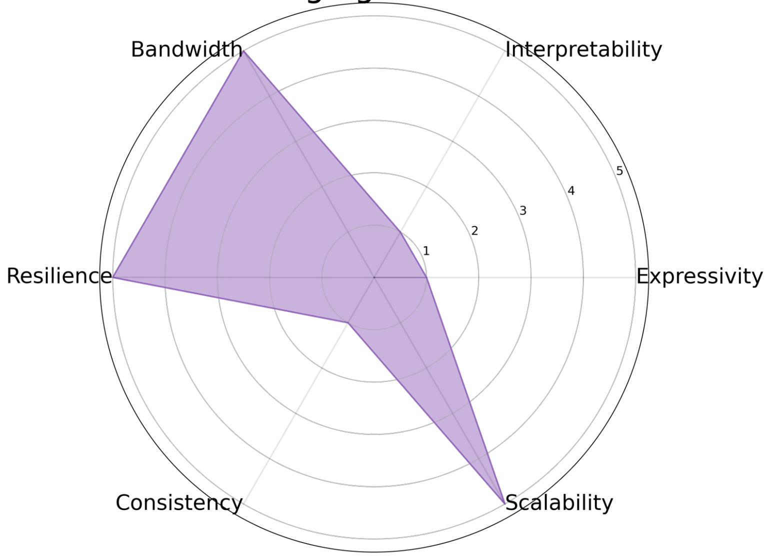
Implicit Coordination



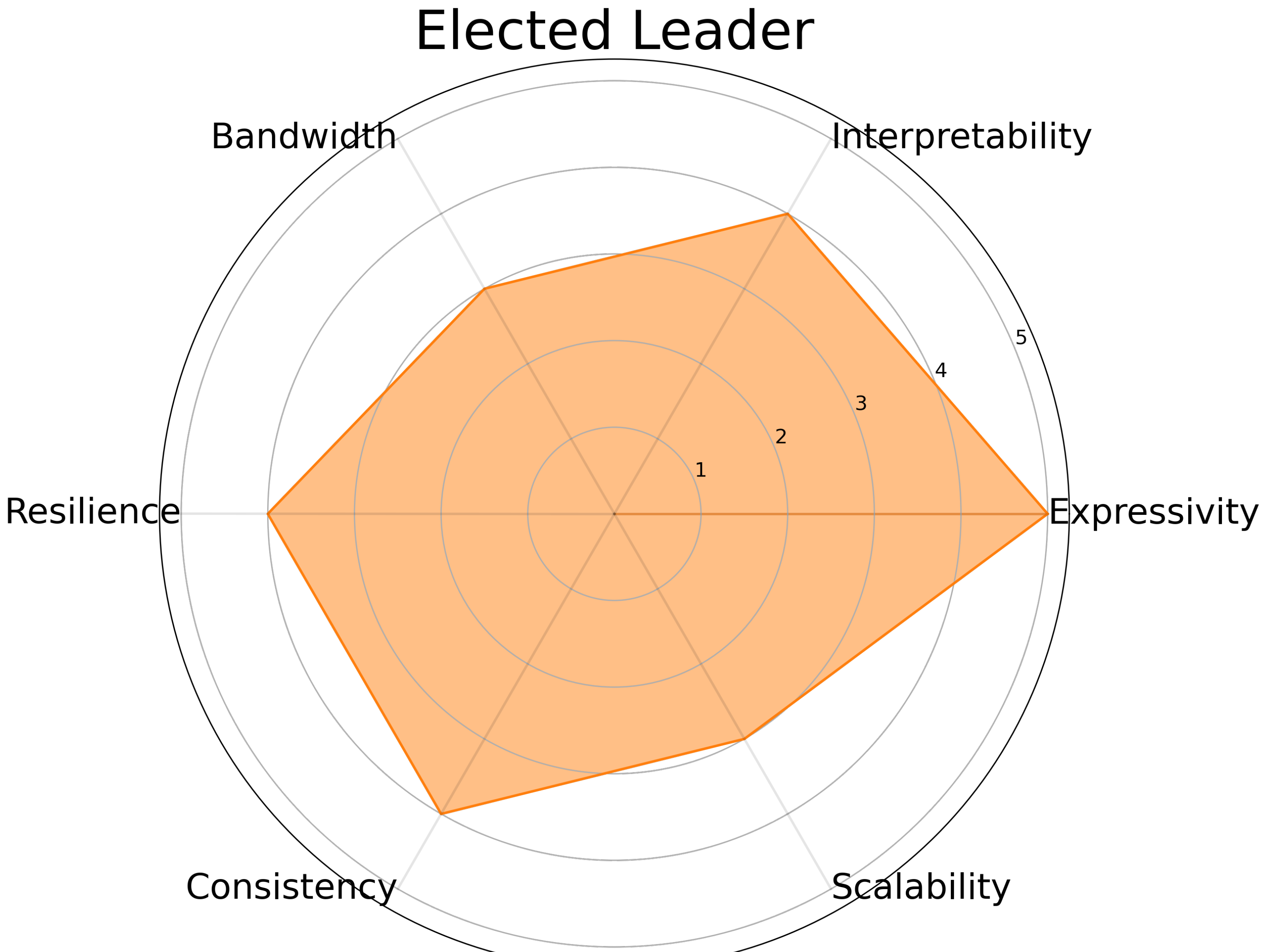
Explitit Coordination



Emerging Behavior



Coordination Architectures

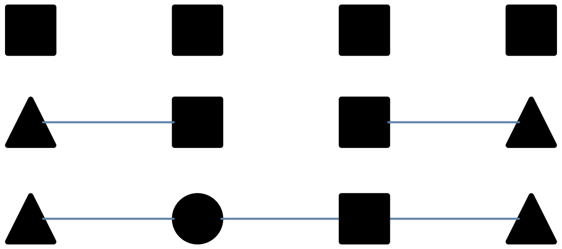


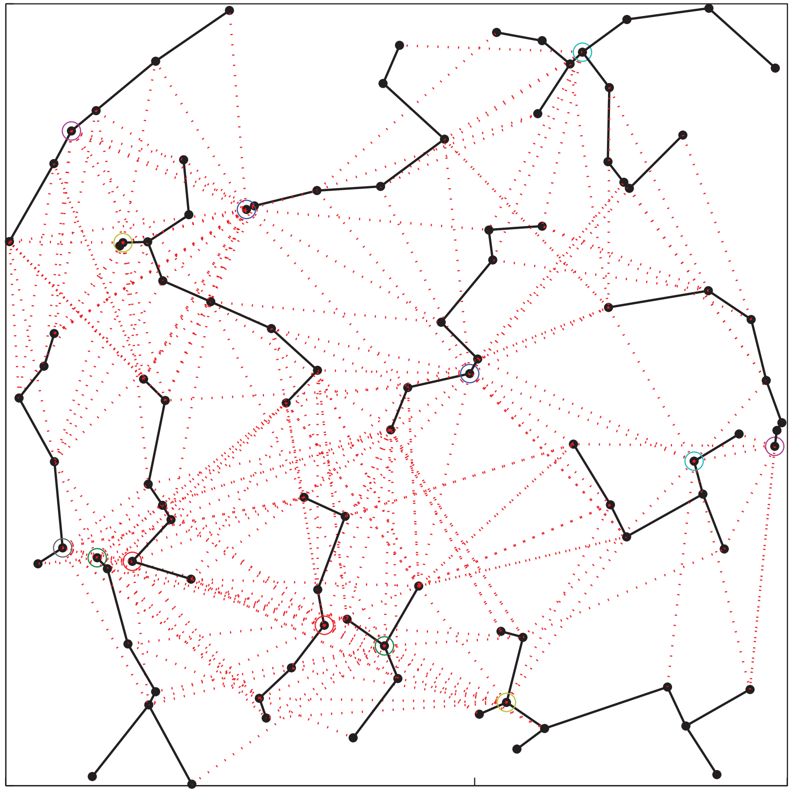
Electing a leader

a) Minimum Spanning Tree using GHS algorithm

Intuition: recursively merge trees log(n) times.

0. Everyone is root (■) and leaf (▲) of own one-node subtree
1. Ping neighbors to find edge cost, send to root along own subtree's MST
2. Nearest* neighbor subtrees merge
   **defined by edge cost*
3. Repeat
   up to log(n) times





b) Tree root monitors leader health, nominates next leader

- *Tree root is not necessarily the leader*
- *Decouple (i) finding a unique agent from (ii) finding the best leader*

Planning, Scheduling, and Execution

Architecture

1. How do we coordinate?
2. Who (if any) is the leader?
3. Where are the decisions made?

Algorithms

4. **When do we drive, and when do we sleep?**
5. How do we explore a region together?
6. How do we perform a distributed measurement?

Resource-aware planning and scheduling

States

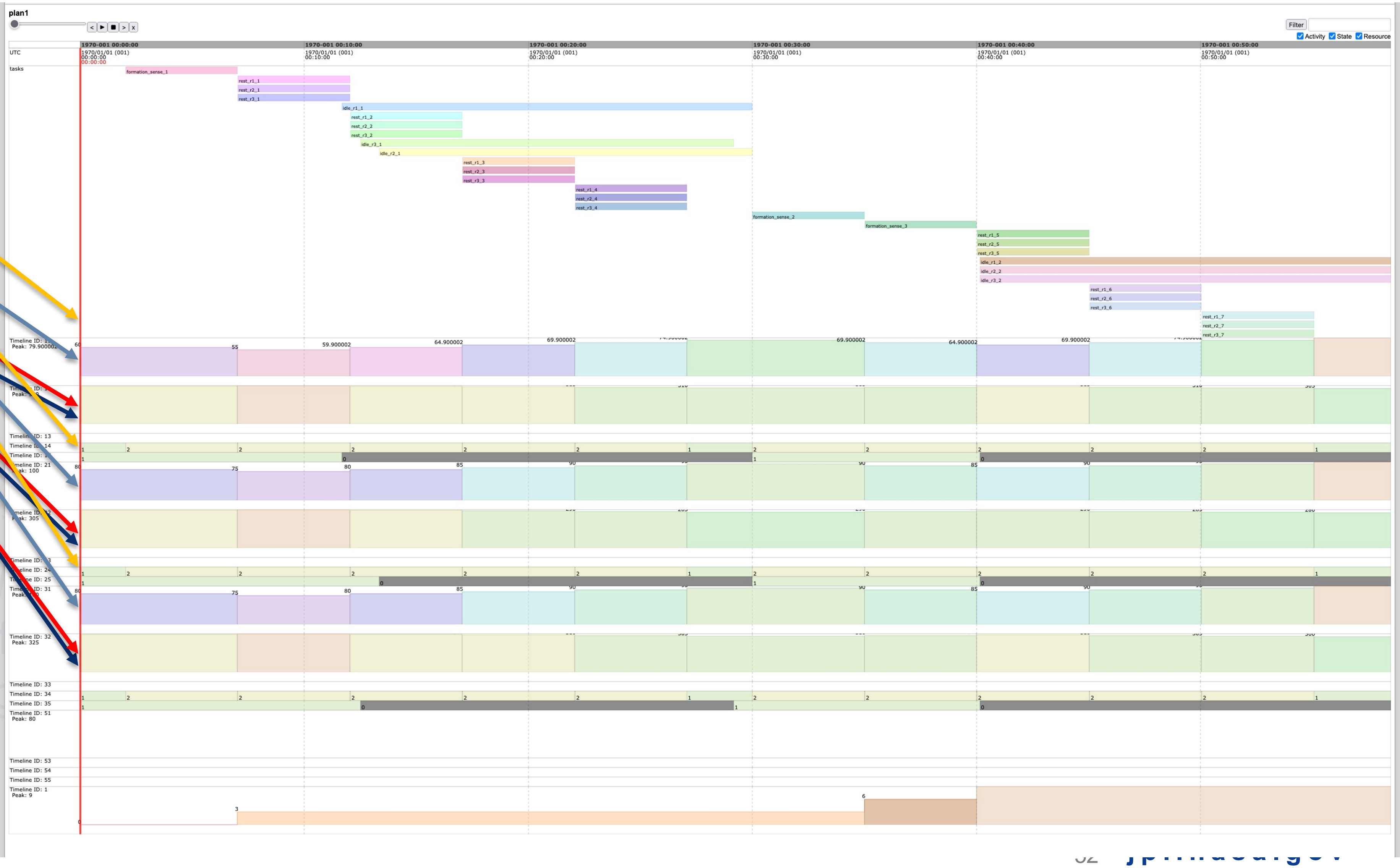
- Power
- Thermal
- Health
- Duty cycle

for every rover, a

- % exploration c
- % formation sensing c

Tasks

- Explore for X minutes (
- Formation sense (for a
- Transfer data to base (
- Cool down and recharg



Resource-aware planning and scheduling

States

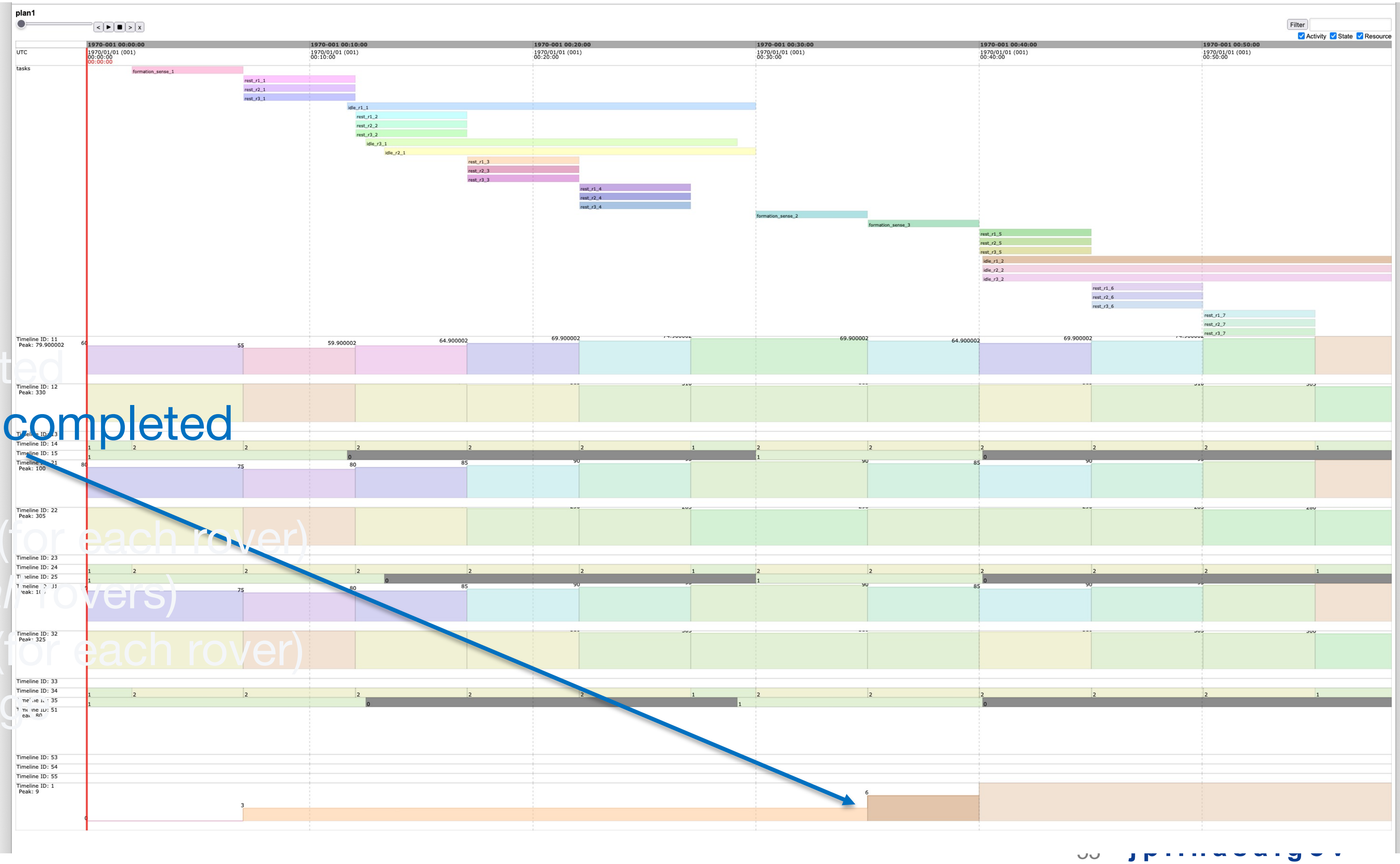
- Power
- Thermal
- Health
- Duty cycle

for every rover, and

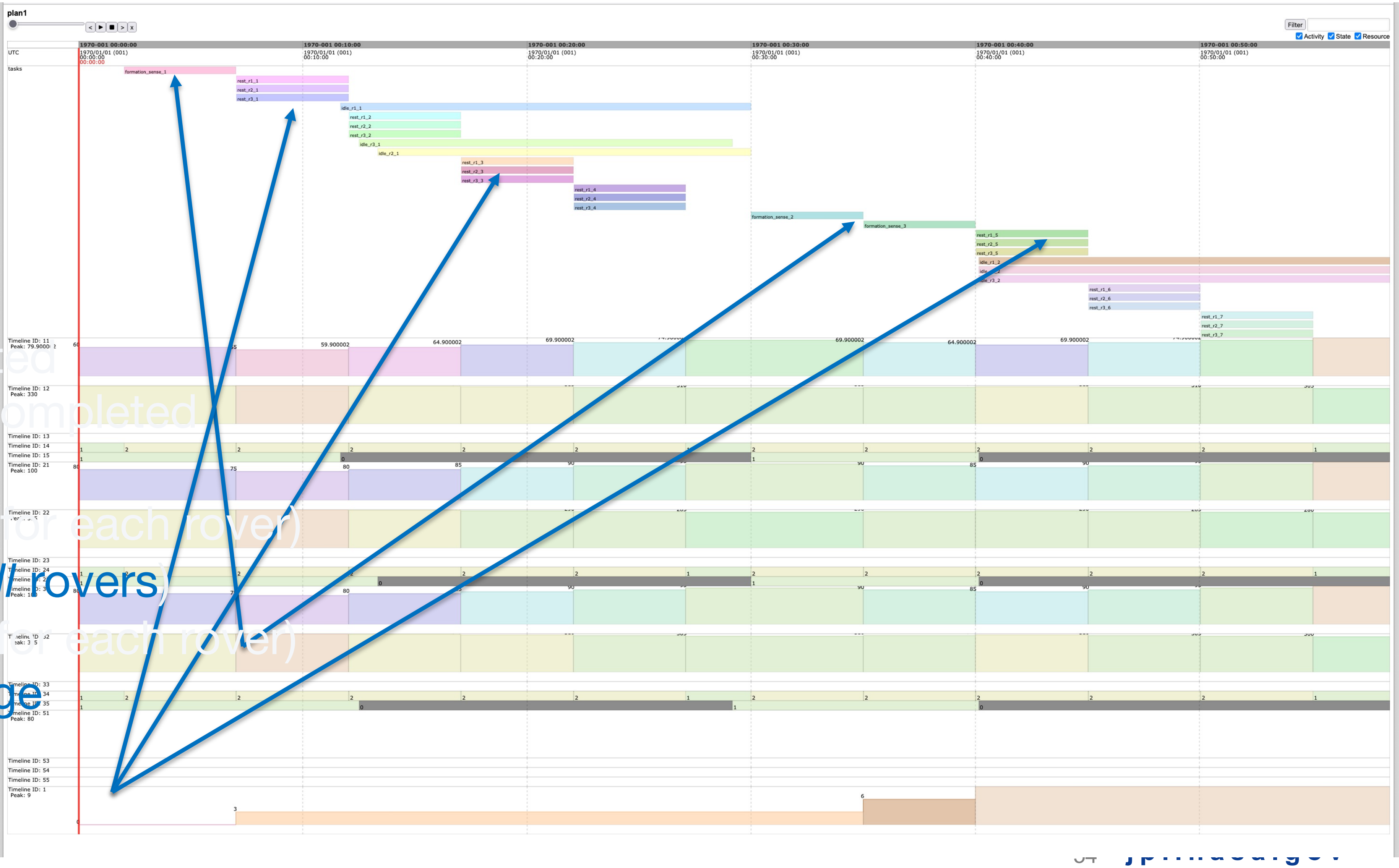
- % exploration completed
- % formation sensing completed

Tasks

- Explore for X minutes (for each rover)
- Formation sense (for all rovers)
- Transfer data to base (for each rover)
- Cool down and recharge



Resource-aware planning and scheduling



States

- Power
- Thermal
- Health
- Duty cycle

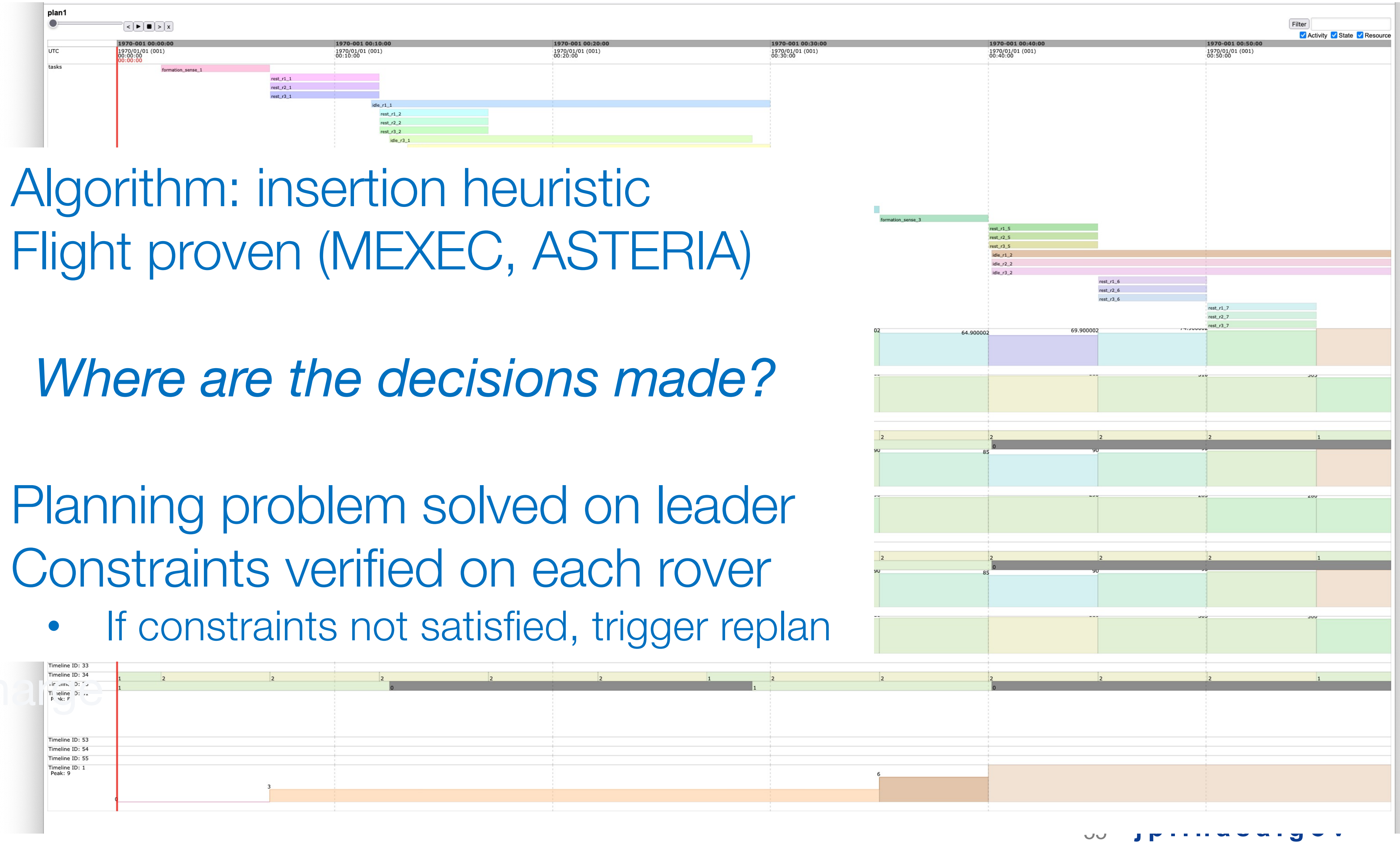
for every rover, and

- % exploration completed
- % formation sensing completed

Tasks

- Explore for X minutes (for each rover)
- Formation sense (for *all* rovers)
- Transfer data to base (for each rover)
- Cool down and recharge

Resource-aware planning and scheduling



- Algorithm: insertion heuristic
- Flight proven (MEXEC, ASTERIA)

Where are the decisions made?

- Planning problem solved on leader
- Constraints verified on each rover
  - If constraints not satisfied, trigger replan

Planning, Scheduling, and Execution

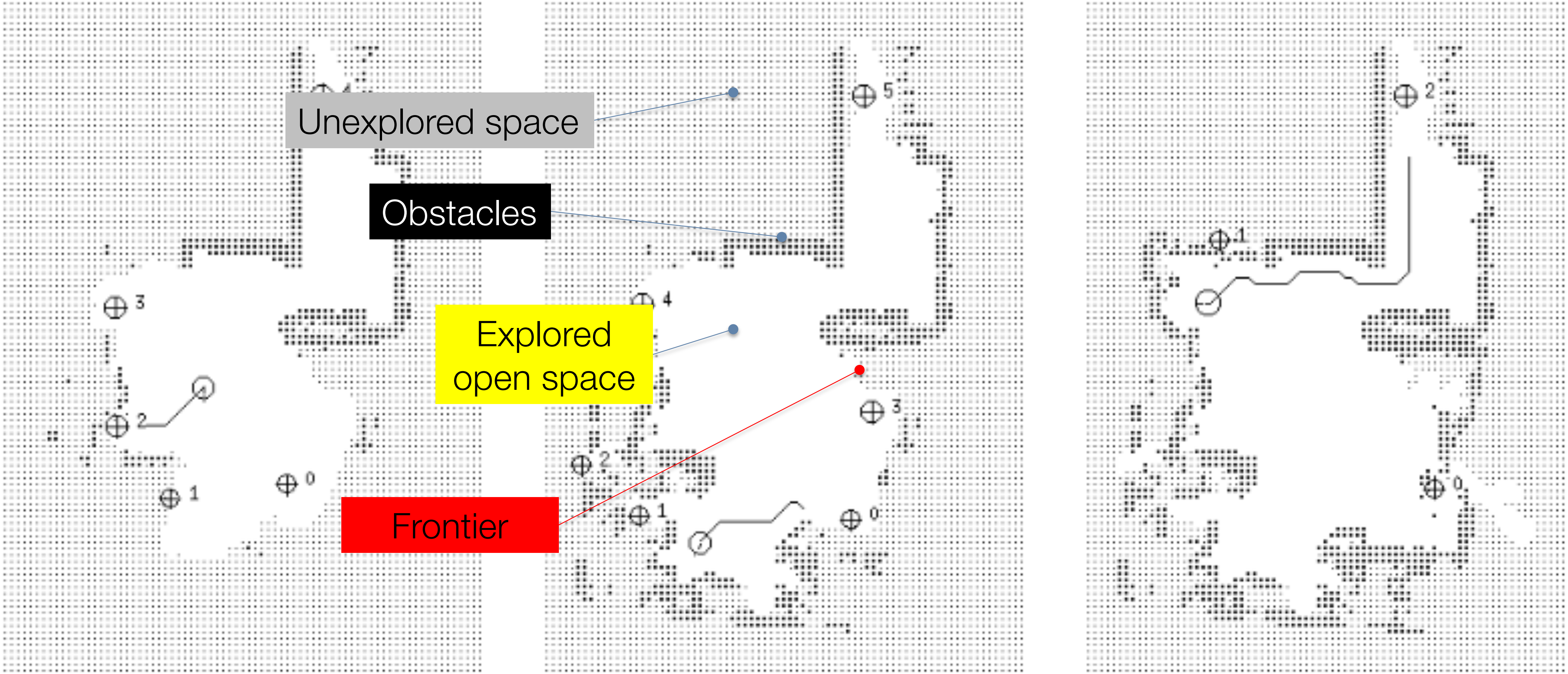
Architecture

1. How do we coordinate?
2. Who (if any) is the leader?
3. Where are the decisions made?

Algorithms

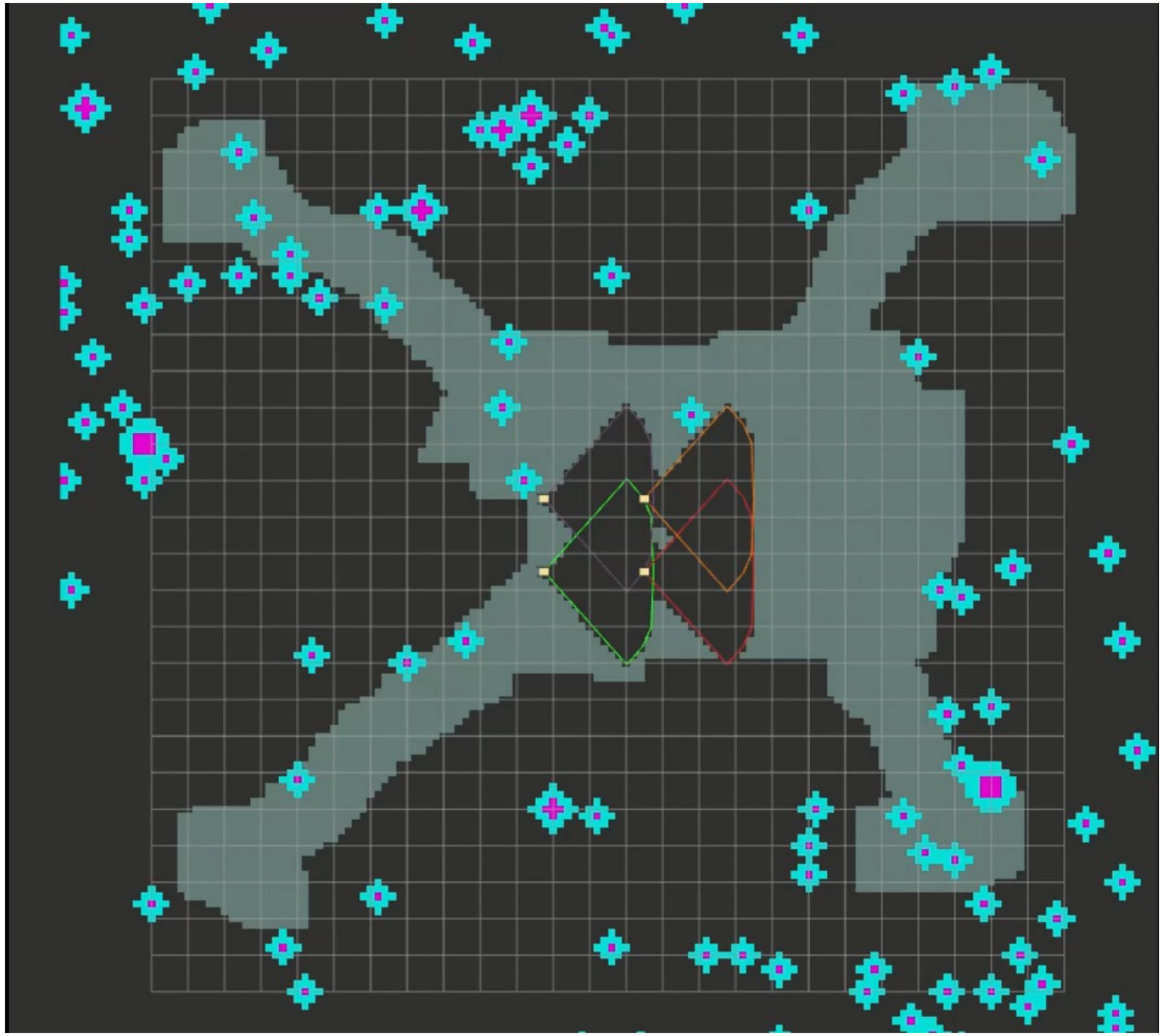
4. When do we drive, and when do we sleep?
5. **How do we explore a region together?**
6. How do we perform a distributed measurement?

Exploration

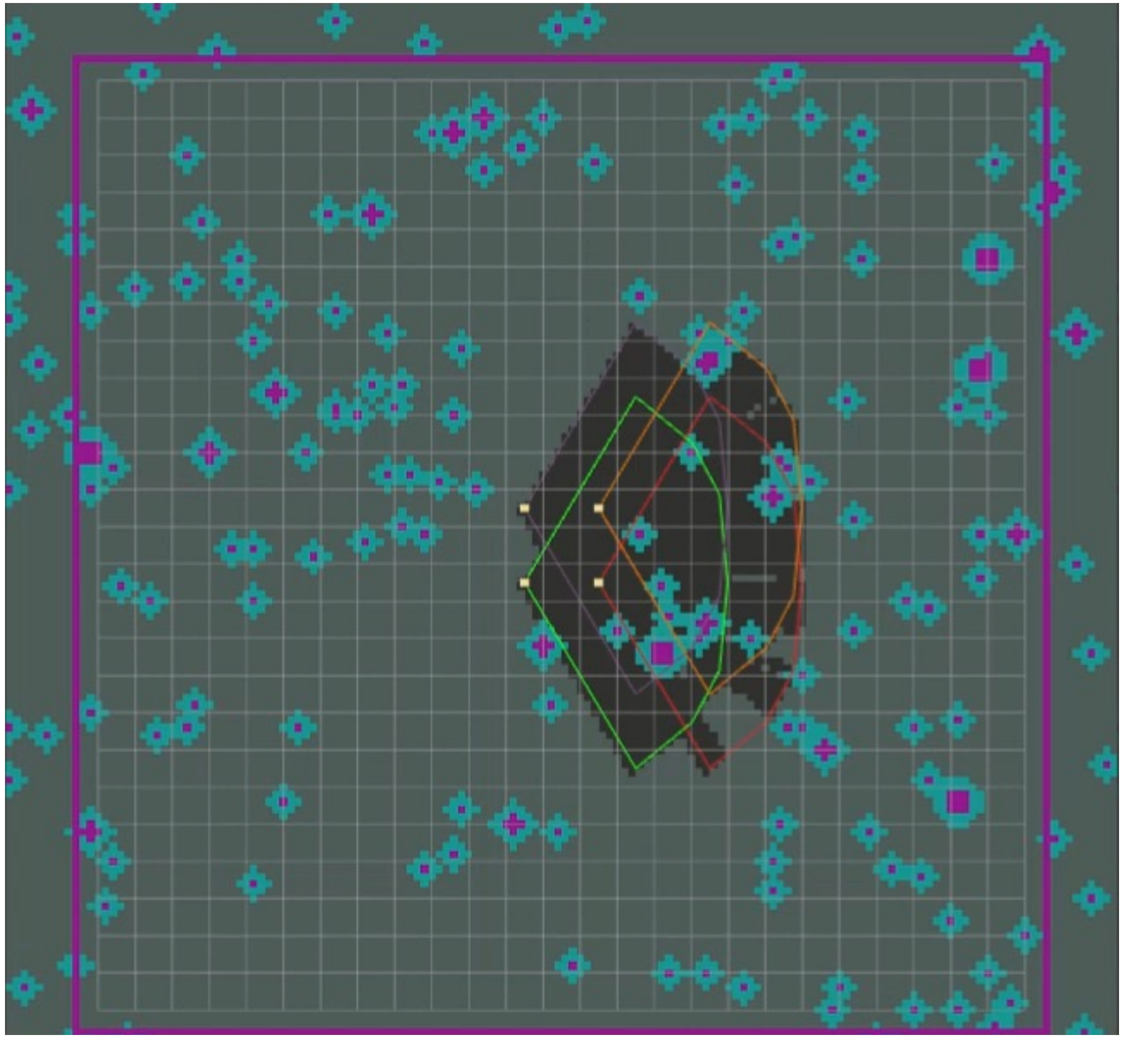


Yamauchi, Brian. "A frontier-based approach for autonomous exploration." Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97) 'Towards New Computational Principles for Robotics and Automation'. IEEE, 1997.

Exploration



Distributed: *"divide and conquer"*



Centralized

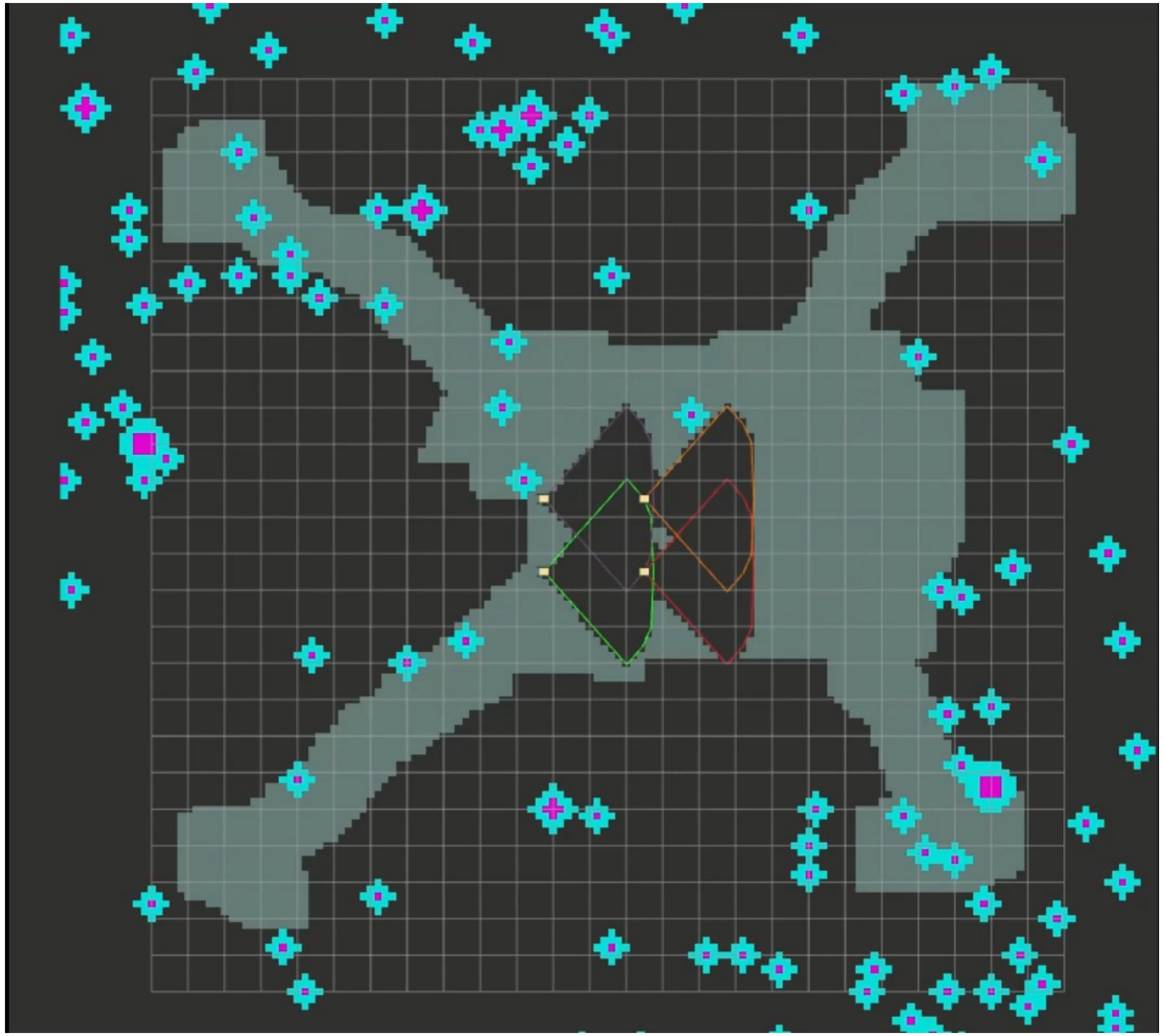
Exploration

	"Divide and Conquer"	"All in one"
Performance	Looser coupling between team and node planner can result in suboptimal performance, No communication relays	Better performance, closer to optimality Can accommodate communication relays
Information Exchange	Significantly less coordination required	Requires more information from agent to team planner, more often
Operability	Easier to interpret • Every node has a clear role	Harder to interpret • "Why are nodes going back and forth across the workspace?"

- Coordination is laborious (latency, network reliability, stale data, etc.), that is where things can go wrong

- Tighter coupling between team and node planner: closer to optimum
- More frequent team-level decisions incorporate new information as it is available

Exploration



Distributed: *"divide and conquer"*

Where are the decisions made?

Ground

- Provides region to explore

Leader

- Computes sub-regions for each robot

Each robot

- Explores its own region

Exploration

Planning, Scheduling, and Execution

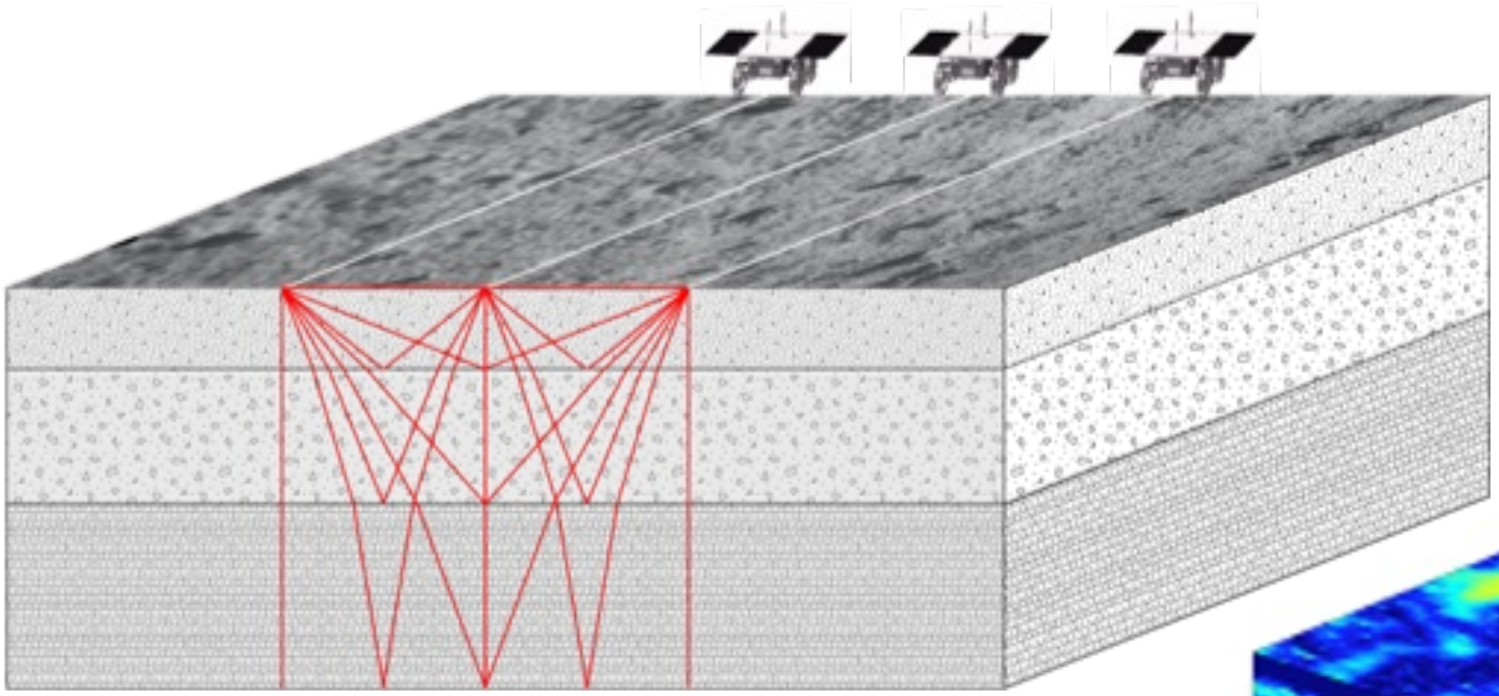
Architecture

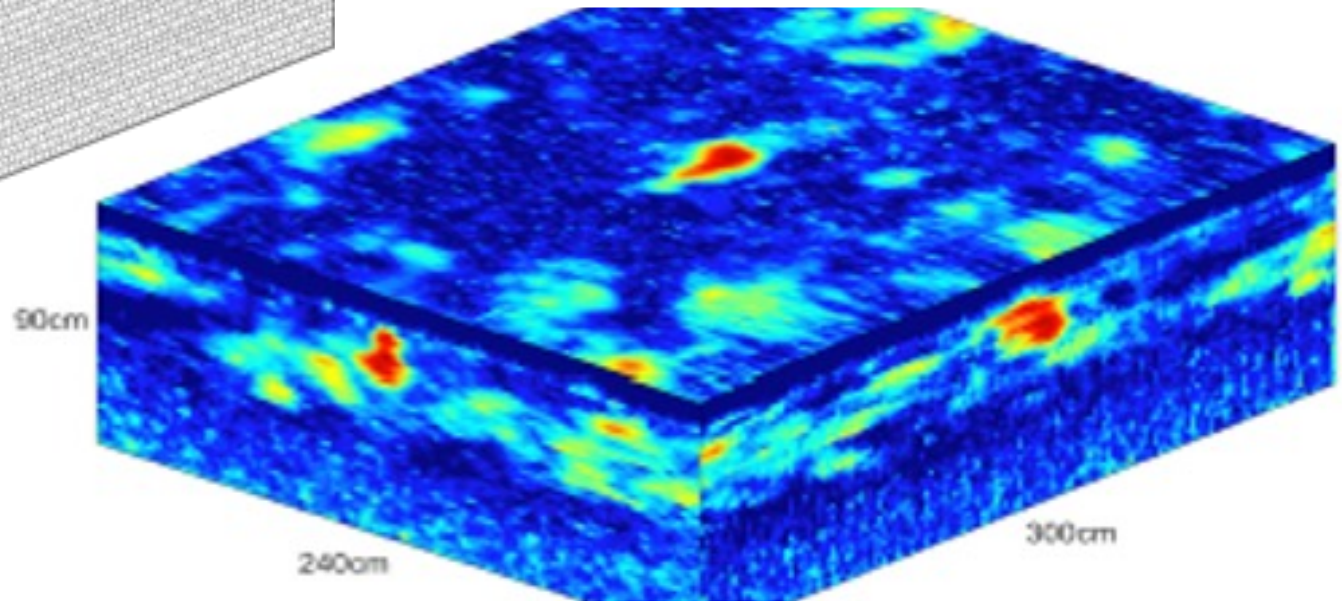
1. How do we coordinate?
2. Who (if any) is the leader?
3. Where are the decisions made?

Algorithms

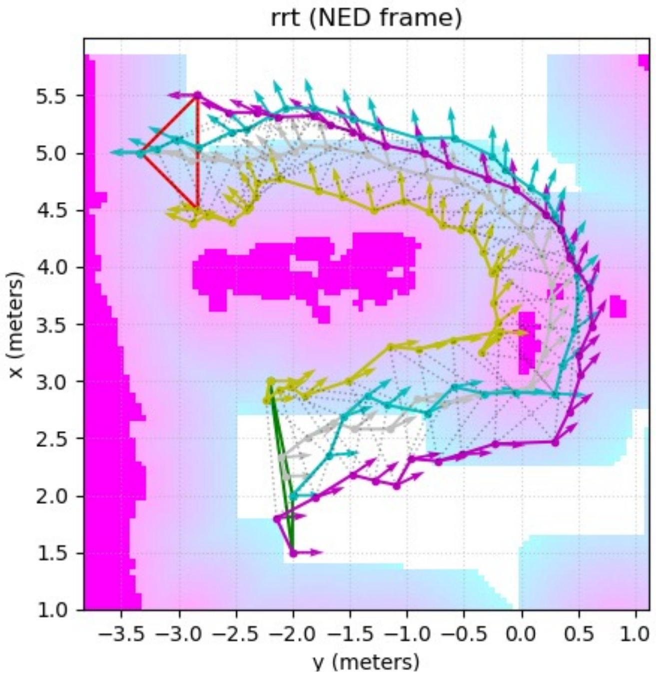
4. When do we drive, and when do we sleep?
5. How do we explore a region together?
6. **How do we perform a distributed measurement?**

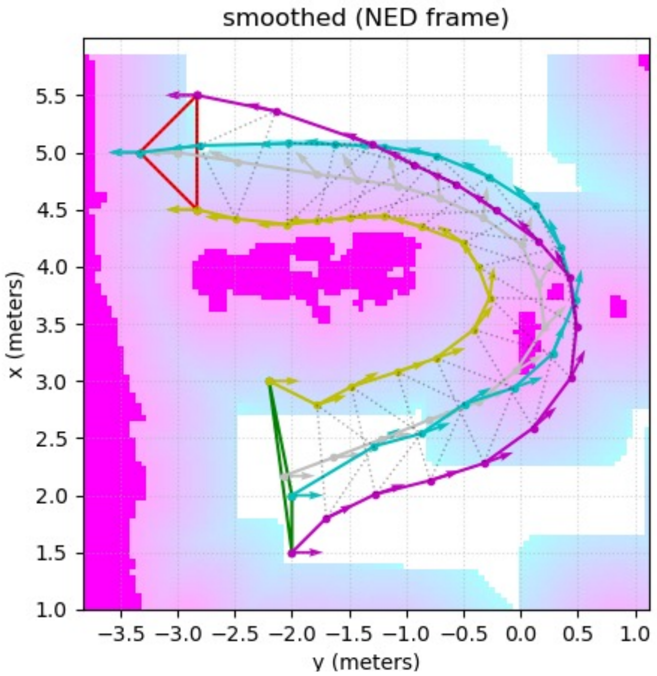
Distributed Measurement and GPR

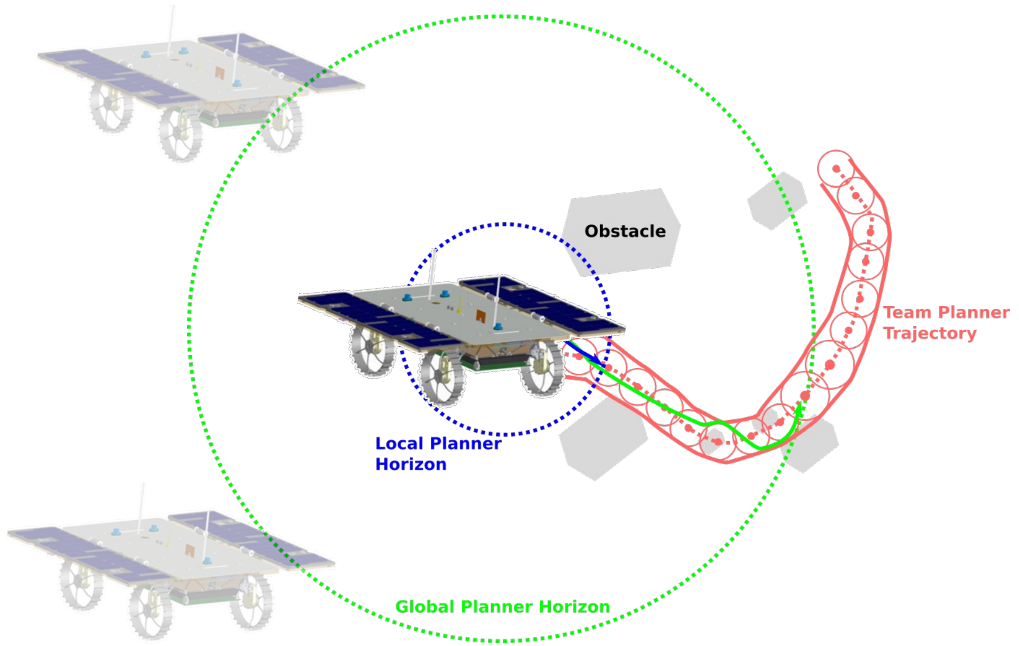




Distributed Measurement







- Sampling-based planner is used to plan formation (with allowable deviation) through map
- RRT* output is adjusted to smooth motion and solution density

- Paths are timestamped to create trajectories and given tolerances (”tubes”) given time-space constraints needed for GPR
- Surface mobility motion planners plan within tubes (and signal fault to leader if unable)

Distributed Measurement

Distributed Measurement

	Deliberative Planning	Formation Control
Performance	Can explicitly enforce formation constraint	Best-effort formation maintenance
Computational Complexity	Higher (planning in the robots' joint space)	Low (closed-loop control)
Information Exchange	Infrequent communication from agents to leader	Requires continuous information exchange between all agents
Operability	Single source of authority for cooperation, no reliance on emergent behavior	Reliant on emergent behavior (e.g. in presence of obstacles)

- Higher computation complexity
- Explicitly enforces formation control constraint
- Lower communication burden

- Tighter control loop
- Relies on emerging behavior to maintain formation while going around obstacles
- Agent-to-agent communication

Putting it all together

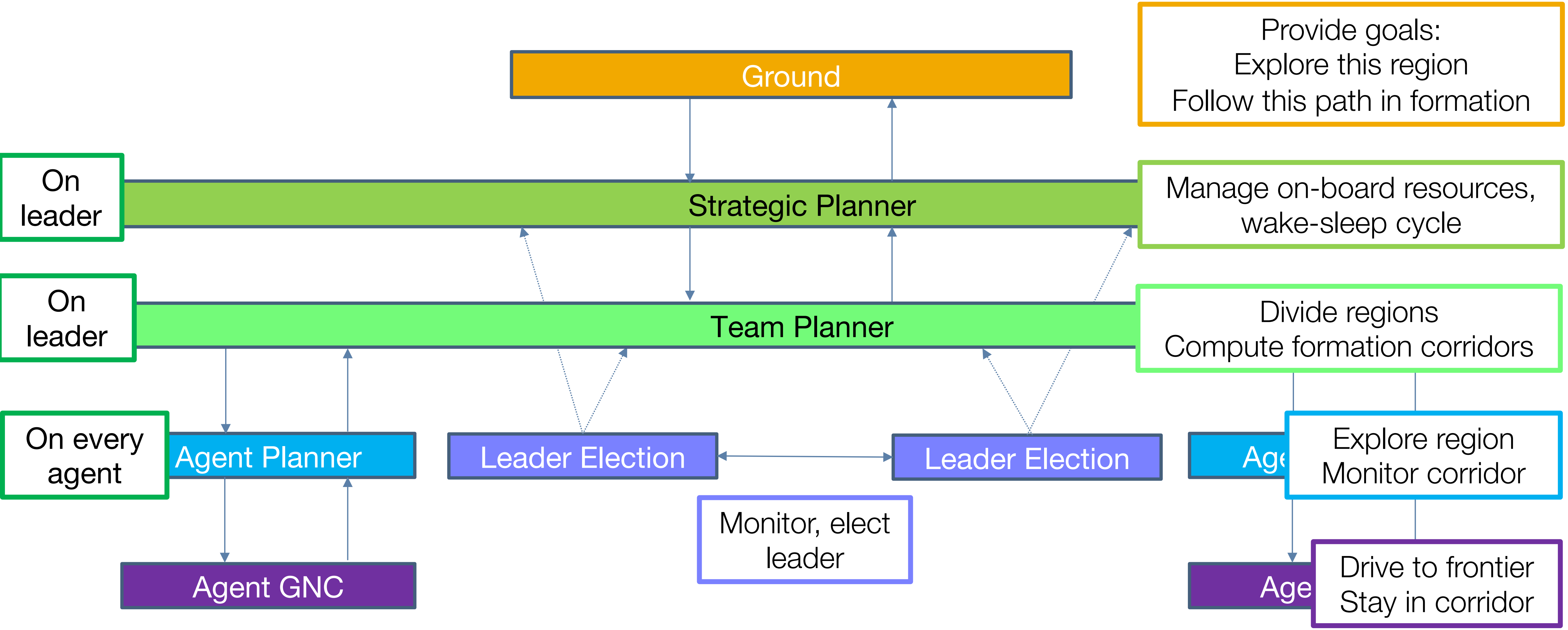
Architecture

1. How do we coordinate?
2. Who (if any) is the leader?
3. **Where are the decisions made?**

Algorithms

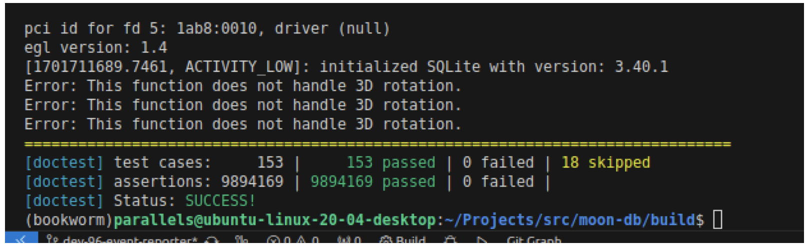
4. When do we drive, and when do we sleep?
5. How do we explore a region together?
6. How do we perform a distributed measurement?

Putting it all together



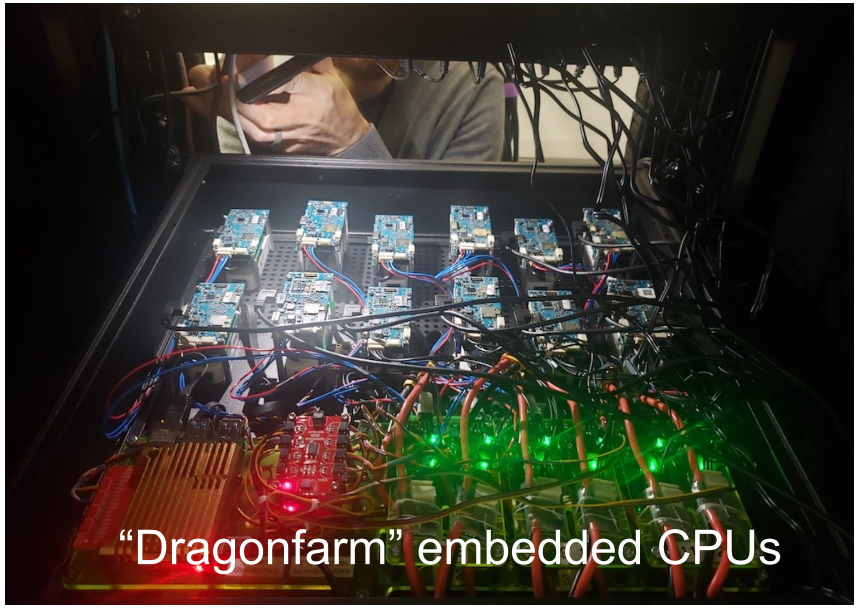
Testing for flight

Testing venues

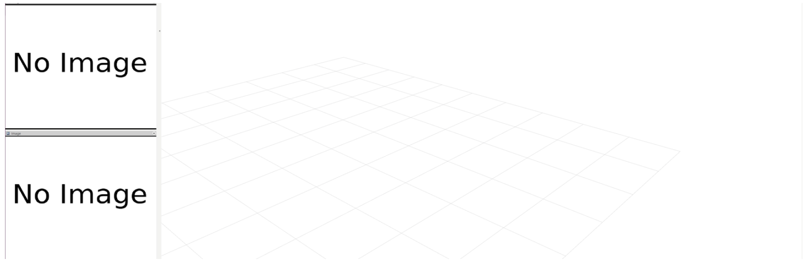


```
pci id for fd 5: 1ab8:0010, driver (null)
egl version: 1.4
[1701711689.7461, ACTIVITY_LOW]: initialized SQLite with version: 3.40.1
Error: This function does not handle 3D rotation.
Error: This function does not handle 3D rotation.
Error: This function does not handle 3D rotation.
===============================================================================
[doctest] test cases:     153 |     153 passed | 0 failed | 18 skipped
[doctest] assertions: 9894169 | 9894169 passed | 0 failed |
[doctest] Status: SUCCESS!
(bookworm)parallels@ubuntu-linux-20-04-desktop:~/Projects/src/moon-db/build$
```

Unit tests

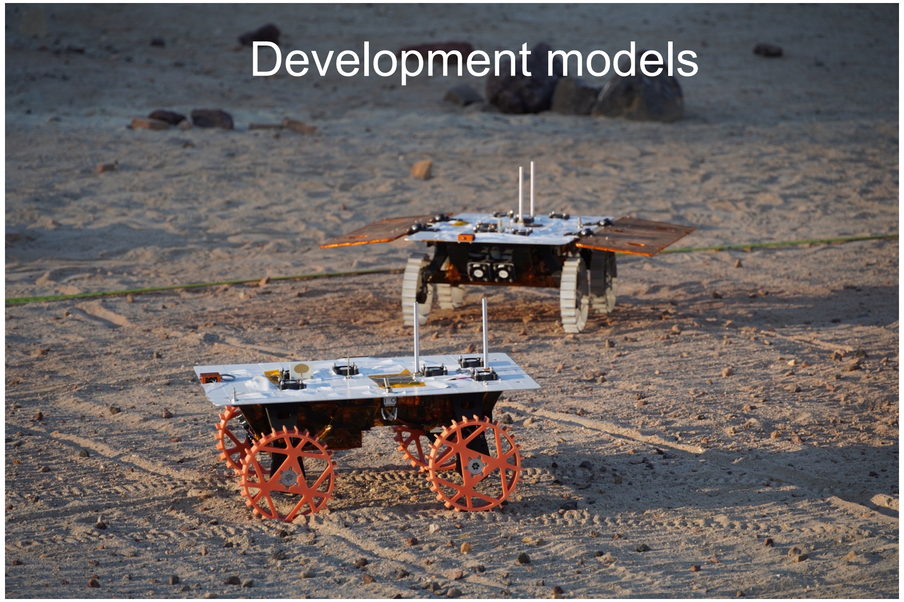


“Dragonfarm” embedded CPUs

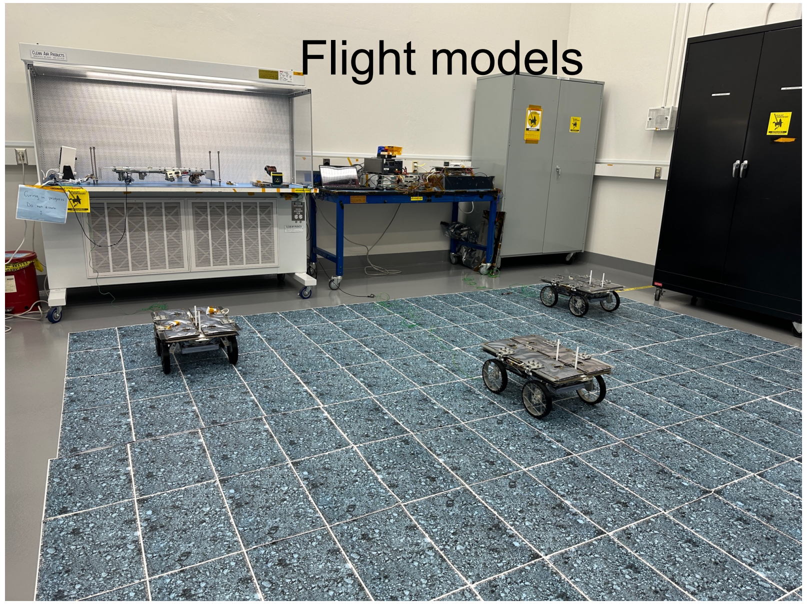


Simulation

“Mercury 7” test mules



Development models



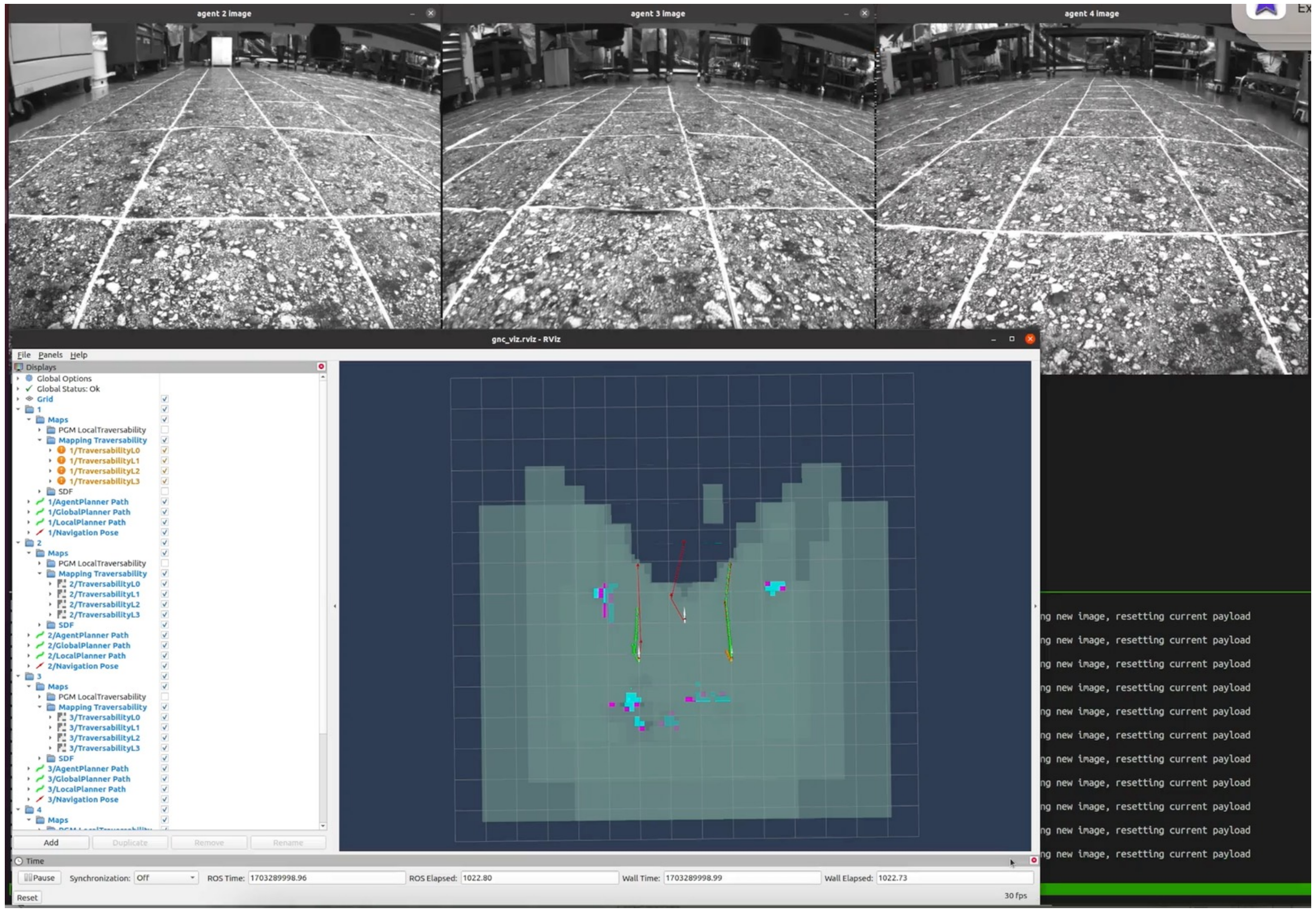
Flight models

Recent Results

- FMs cooperatively and autonomously completed a successful formation drive (as they would for a distributed measurement)
  - Base station executed leader functionality by planning and scheduling for the team

- When driving out-of-formation, rovers reported to leader and the system replanned and continued drive automatically
  - Three (3) full (autonomous) planning cycles were performed
  - A second experiment verified that the team stops driving when any rover violates its SoC constraint (i.e., battery too low)
  - A third experiment included a previously unmapped obstacle around which the team replanned.

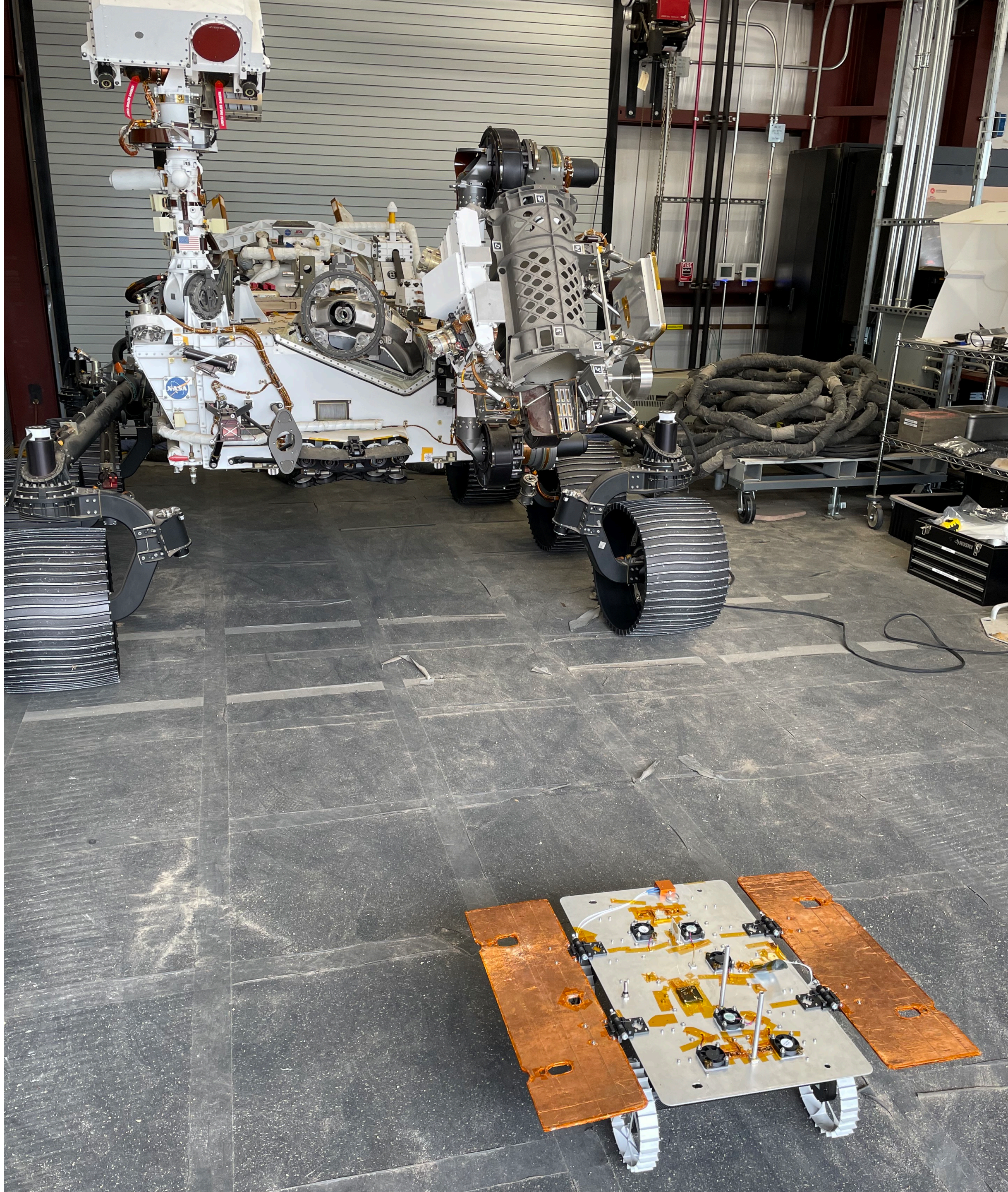
- Operators started autonomy, but did not intervene at any point.



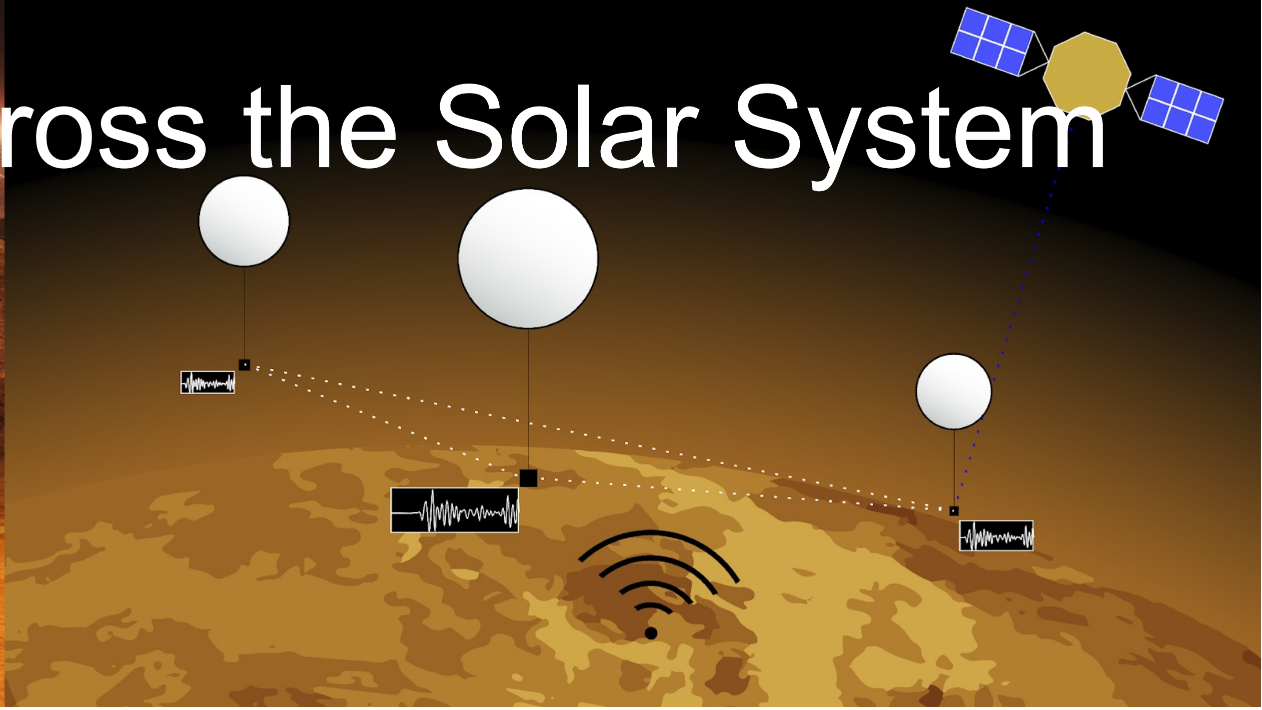
Summary

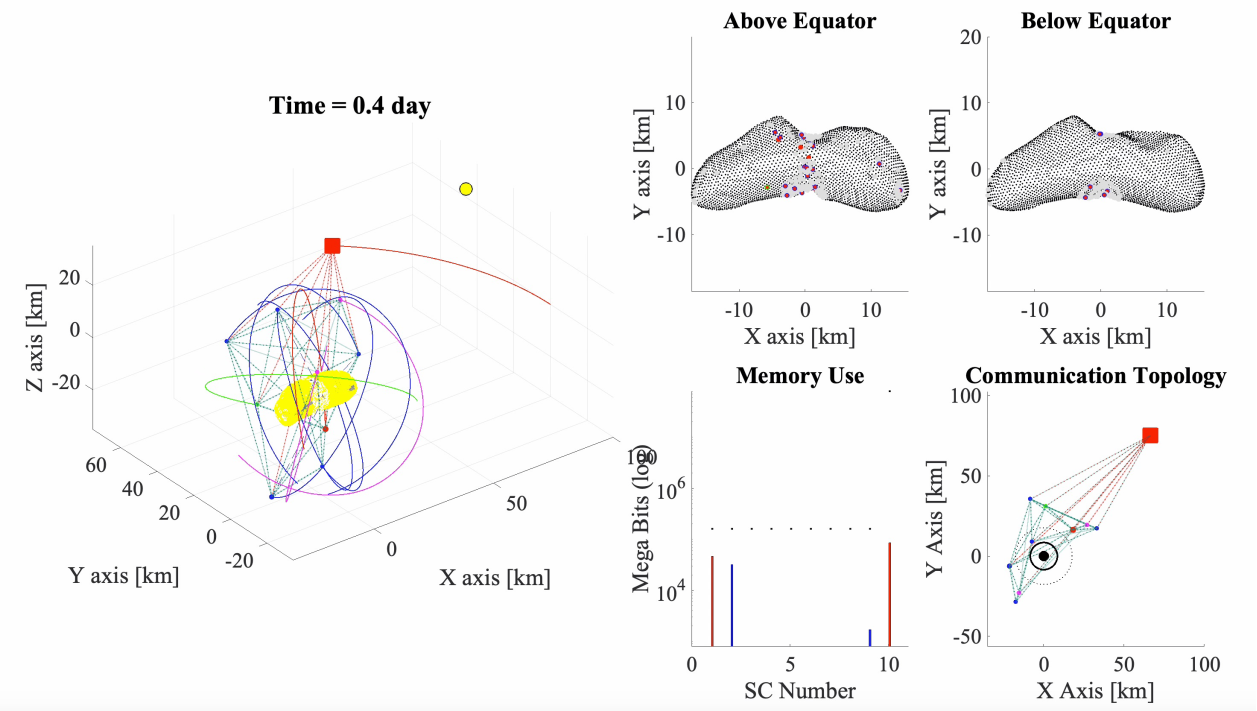
CADRE VALIDATES A NEW GENERATION OF ROBOTIC TECHNOLOGIES:

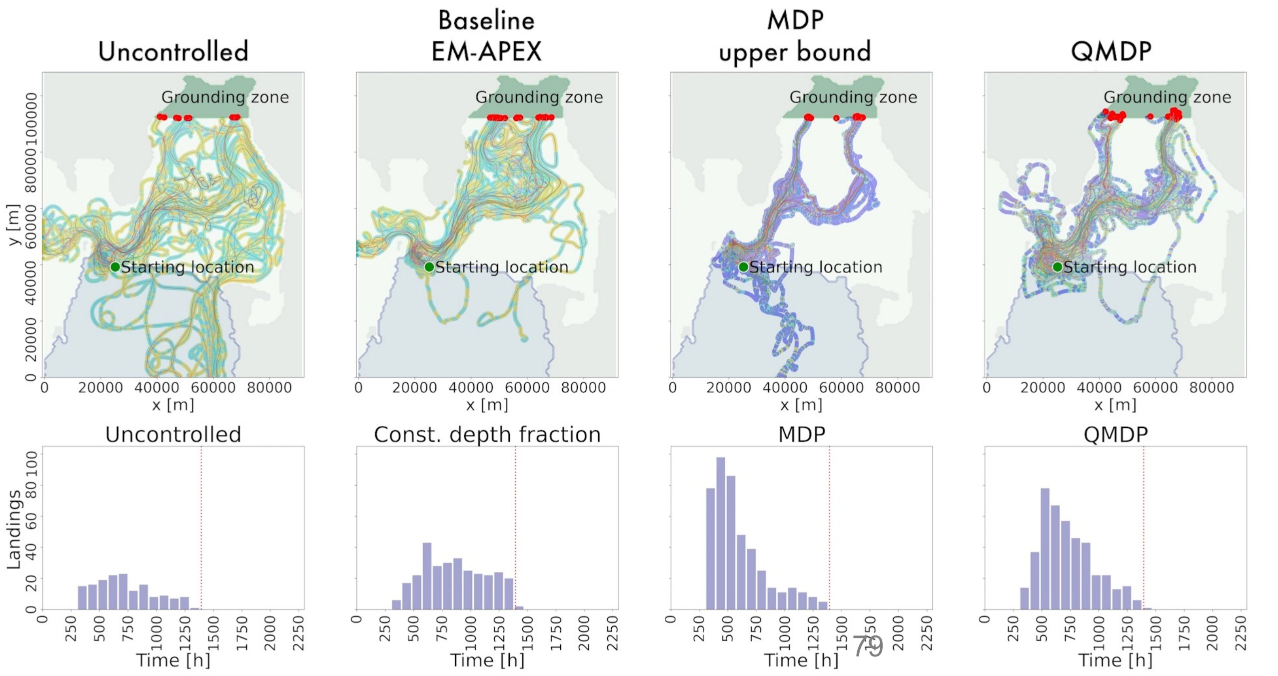
- Small-scale, autonomy capable rovers
- Multi-agent autonomy that enables robots to work cooperatively as a team (requires localization!)
- Ability to perform distributed science measurements



Multi-robot systems across the Solar System

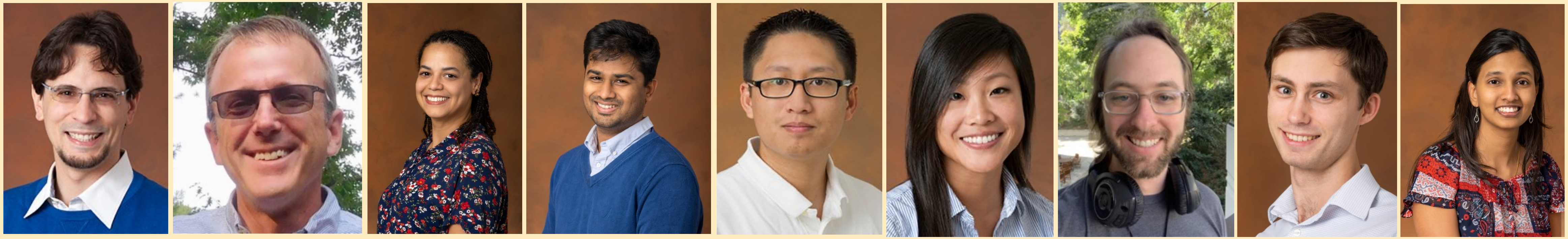






CADRE's Autonomy Team

Multi-Agent Planning, Scheduling, and Execution



Federico Rossi, Gregg Rabideau, Maira Saboia, Yashwanth Nakka, Viet Nguyen, Grace Lim, Joe Russino, Keenan Albee, Ramya Bhamidipati

"Motion Planning", Navigation, and Control

Roland Brockers, Libby Boroson, Abhishek Cauligi, Ben Morrell, Rob Hewitt, Dustin Aguilar, Pedro Proecna, Dima Kogan

And many more working on avionics, FSW, mobility, system engineering, etc. to make CADRE happen!