

Astrobbee Overview



System, Status,
Performance, and specification
November 24th, 2019



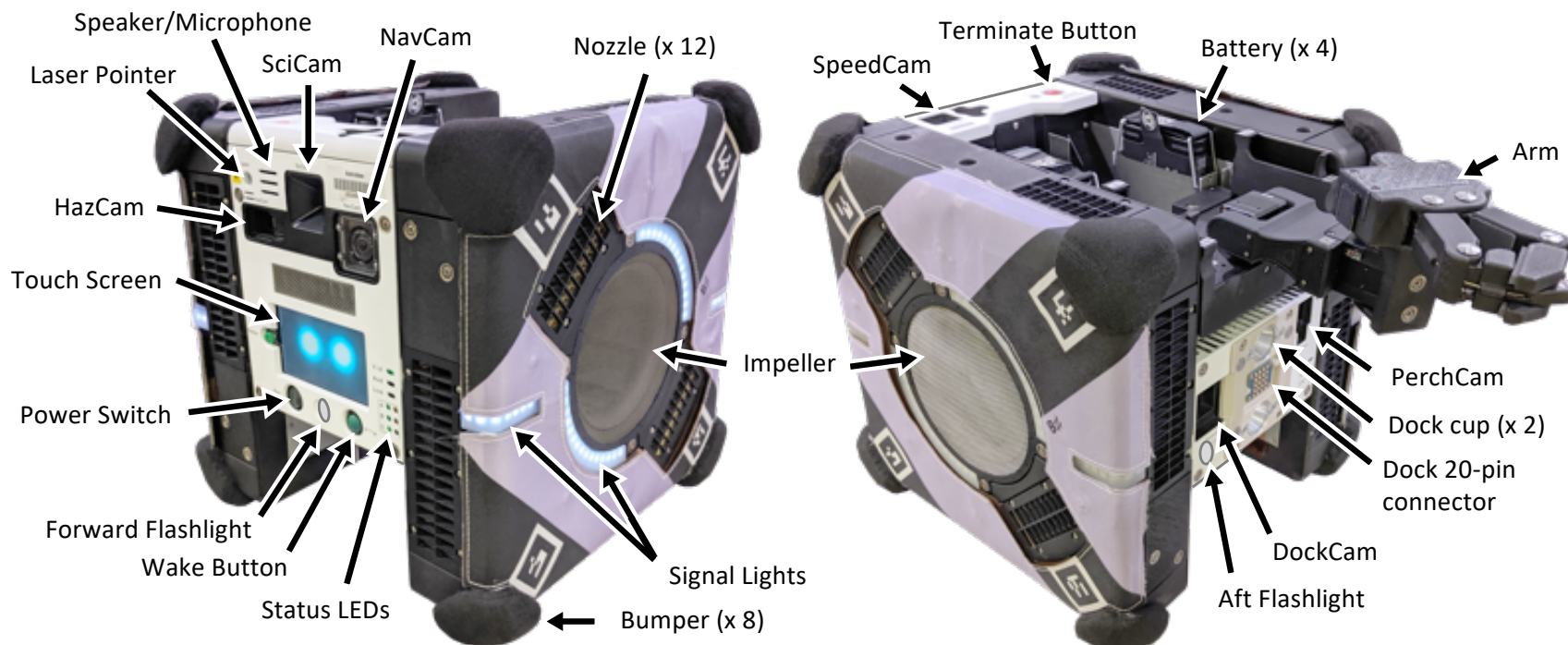
Astrobee Objectives

- Provide a microgravity robotic research facility in the ISS US Orbital Segment (USOS), which will replace the existing SPHERES facility
- Provide remotely operated mobile camera views of the ISS USOS to enhance the situation awareness of mission control
- Perform mobile sensor tasks in the ISS USOS

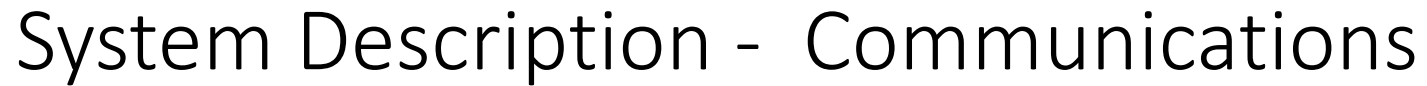




System Description - Hardware



- Free flying robot inside the ISS
- 32 cm wide, ~9.1kg (2 batt., no arm)
- All electric + fan-based propulsion
- Robot arm for “perching”, ~1kg
- Three smartphone computers
- Three payload bays for expansion
- Microphone not currently enabled



-
- The diagram illustrates the ISS Payload LAN architecture. At the top, a small inset shows the ISS. The main diagram shows the ISS components: Dock, Free Flyers, Crew Control Station, and ISS NAS Storage. These are connected to the ISS Payload LAN. The LAN is connected to the White Sands ground station via Ku-Band. The White Sands ground station is connected to the Astrobee Ground Relay (MSFC). The Astrobee Ground Relay is connected to the Multi-Mission Ops Center (ARC), the ISS Payload Ops Integration Center (MSFC), the ISS Mission Control Center (JSC), and the Guest Scientist Institutions. The Multi-Mission Ops Center (ARC) is connected to the Astrobee Data Archive (ARC).

4



System Description - Navigation

- Vision-based navigation
 - Compares features with on-board a priori map
 - Incorporates inertial measurements
- Fiducials used for autonomous docking
 - Requires approximately 1 cm position accuracy
- Visual odometry
 - Robot can continue to navigate where no map features are recognized



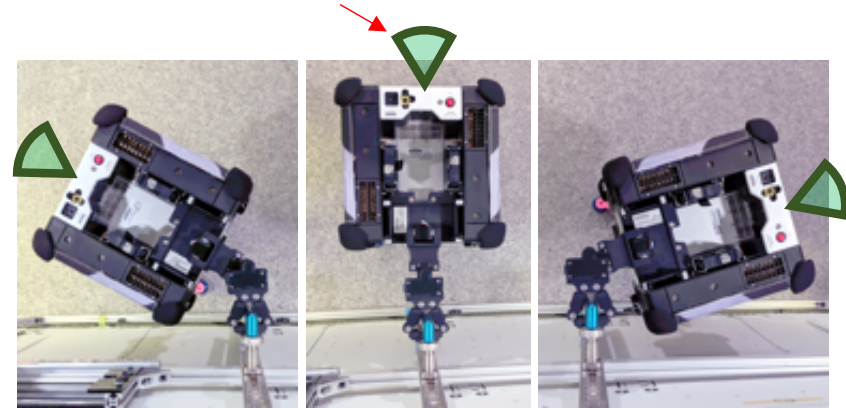
Feature map of the JEM-PM



System Description - Perching Arm

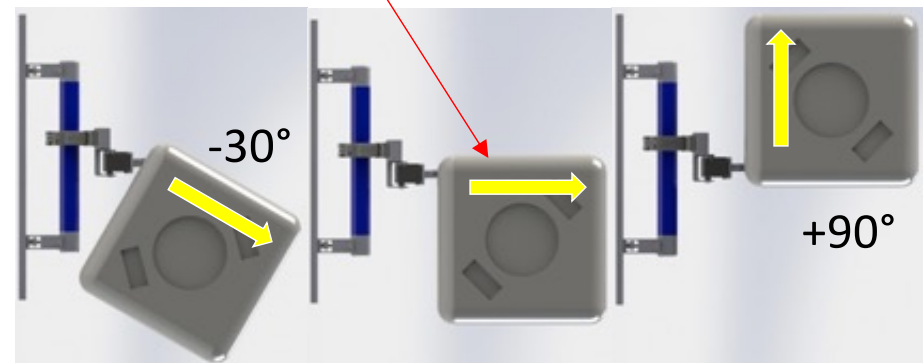
- Designed to grasp handrails
- Stows completely in payload bay
- Acts as a pan-tilt unit while perched
- Flexible and back-drivable
- May be perched manually

Camera View Direction



Astrobee Perching Arm pan motion

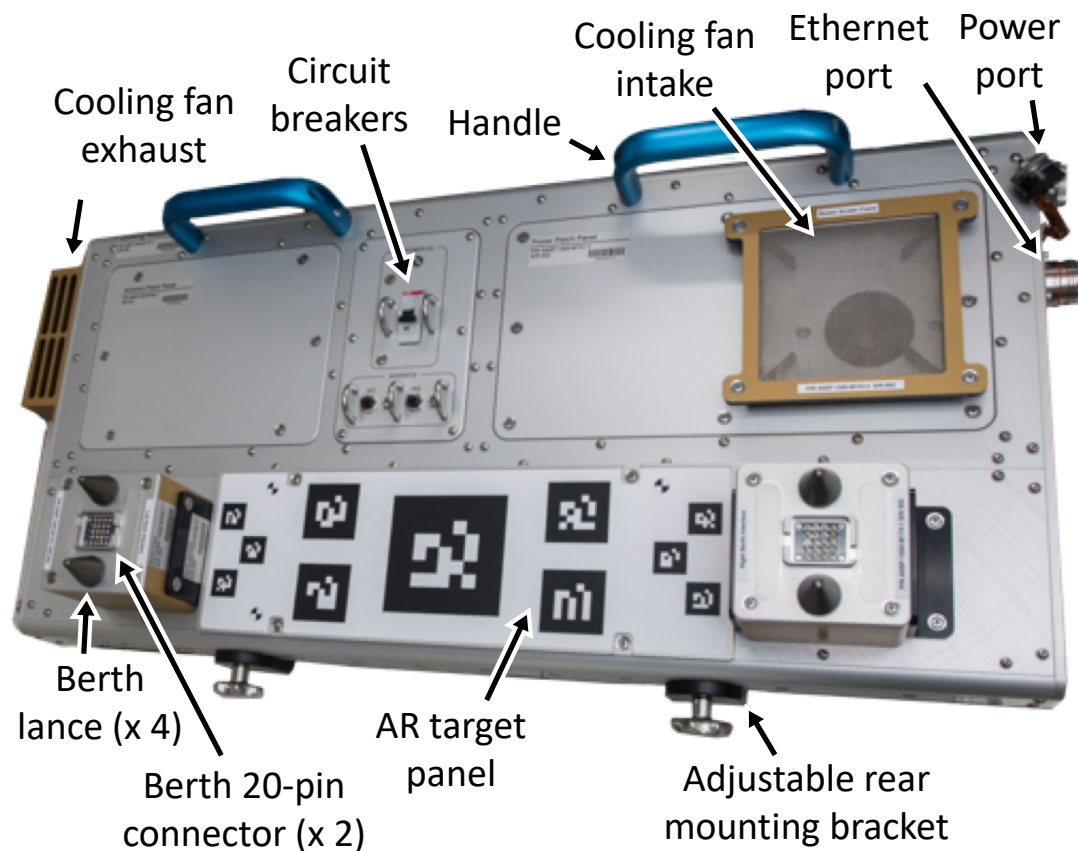
Camera View Direction



Astrobee Perching Arm tilt motion



System Description - Docking Station



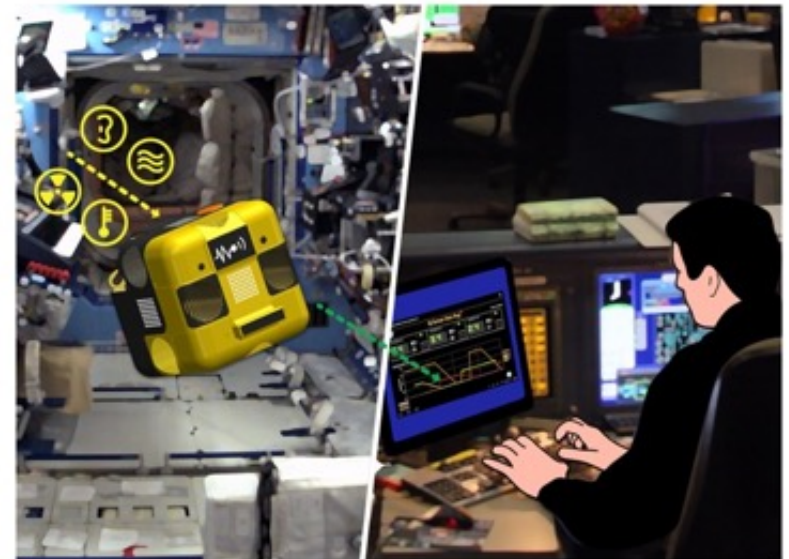
- 85 cm x 38 cm x 28 cm
- Berths for 2 free flyers
- Provides power and Ethernet

- Fiducials used for visual servoing to autonomously dock
- Magnets provide retention force



System Description - Ground Data System

- Astrobee Control Station
 - Sortie planning tool
 - Execution monitoring
 - Live telemetry
 - Image and video streams
 - 3D virtual display
 - Supervisory control (run plans or single commands)
 - Typically used by ground operators
- Crew Control Station runs on an EXPRESS Laptop Computer (ELC)
- Server for archiving and distributing Astrobee data
- Suite of engineering tools to support maintenance and software upgrades





Astrobee Control Station

Crew Control Station

File View Help

Run Plan Teleoperation Guest Science

FreeFlyerA Comm ● Control DW@DW-Windows7-32 Est. Batt 2:39 Docking Station ● GPS 18Mar17 17:38:44

Health and Status

Operating State	Plan Execution
Mobility State	Flying
Operating Limits	Default_Safeguard
Plan	Survey1
Plan Status	Executing

Robot Commanding

File: C:\Users\DW\Desktop\FPlans\Survey1.fplan

Plan Valid

Buttons: Load Run **Pause** Skip Step

Description: Survey European Lab and US Lab

Plan

Total Elapsed Time 00:00:35

Plan Step	Duration	Success
Survey1		
0 Station		Complete
0-1 Segment	00:01:48	Complete
1 Station		Complete
1.0 PowerOnIten		Complete
1.1 Wait	00:00:25	Complete
1.2 PowerOffIten		Complete
1-2 Segment		
2 Station		
2.0 Wait		
2-3 Segment		
3 Station		
3-4 Segment		
4 Station		
4-5 Segment		
5 Station		
5-6 Segment		
6 Station		

Live Telemetry Live Images Live Video

17:38:09 FreeFlyerA: Run Plan Pending ...



Basic Conops

- When an Astrobee is idle, it charges in its dock
- Astrobees can execute complex plans with full autonomy and no astronauts present
 - Including undock, traverse multiple modules, return to dock
- However, Astrobees run with ground operator oversight
 - When an anomaly occurs, an Astrobee generally stops and waits for operator intervention
 - It can continue operating during communication outages until it encounters an anomaly
- The operator can always take over and teleoperate
- Astronauts can also be operators, but this is will likely be a rare occurrence (minimize crew time)



Hardware Status

- Docking Station
 - Launched on NG-10 – 11/17/18
 - Installed on 2/15/19
- 2 Astrobees (Bumble & Honey) and Dock Spares Kit
 - Launched on NG-11 – 4/17/19
 - Checkout of Bumble began 4/30/19
 - Checkout of Honey around 7/25/19
- 1 Astrobee (Queen) and 3 Perching Arms
 - Launching on SpX-18 – 7/21/19
 - All hardware delivered and verifications closed
- Astrobee Spares Kit and Port Tester
 - Launching on NG-12 – 10/19/19
 - On-dock 8/6/19



Astrobee Commissioning Activities

1. Checkout

- Crew inspects the robot
- Functional tests of all hardware

2. Calibration & Mapping

- Crew collects camera and IMU calibration data
- Crew collects mapping data of Kibo

3. Localization & Mobility

- Verify robot localization within Kibo
- Robot performs increasingly complex motions to test mobility system

4. Checkout & Calibration

- Combination of 1 & 2 without mapping



Astrobee Commissioning Activities

6. Ops Demo

- Demonstrate an operational mission scenario

7. Payload Installation

- Crew installs payload (perching arm)
- Functional tests of payload

8. Payload Demo

- Operational demonstration of payload (perching arm)

9. Performance Characterization

- TBD tests to further characterize Astrobee performance

10. SPHERES/Astrobee Hand-off

- Symbolic passing of the torch from SPHERES to Astrobee
- Completion of commissioning



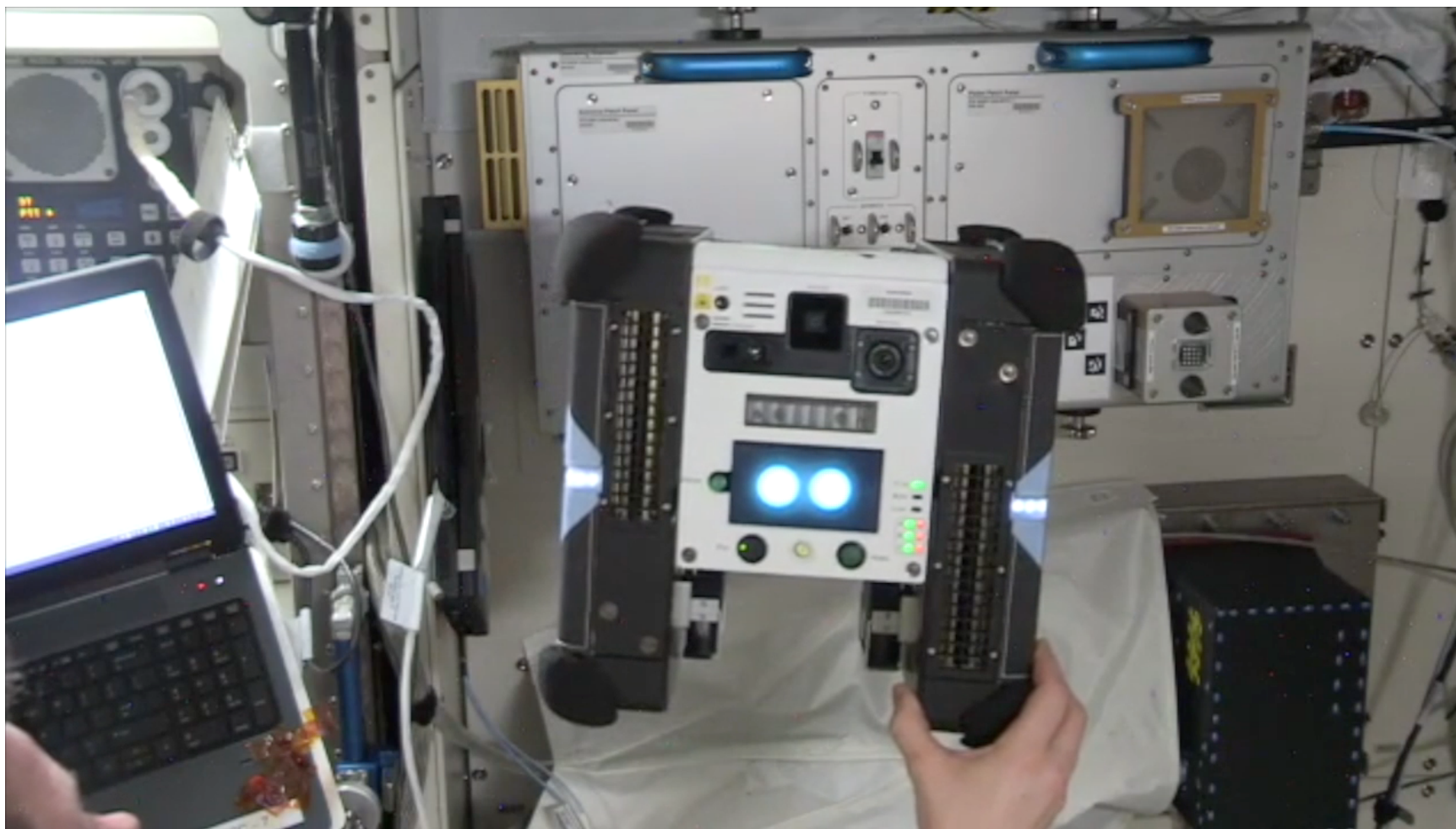
Commissioning Schedule

Completed | **Scheduled** | Desired

Activity	Robot	Date
Checkout	Bumble	4/30/19
Calibration & Mapping	Bumble	5/13/19
Additional Mapping	Bumble	5/23/19
Localization & Mobility	Bumble	6/14/19
Localization & Mobility	Bumble	7/12/19
Checkout & Calibration	Honey	7/25/19
Localization & Mobility	Bumble	7/30/19
Localization & Mobility	Honey	8/13/19
Checkout & Calibration	Queen	8/20/19
Localization & Mobility	Honey	8/27/19



Bumble Checkout





IMU Calibration





1st Flight





Stopping Ability





1st Autonomous Undocking





1st Autonomous Docking





Near-Term Guest Science Payloads

- NASA JSC's RFID-Recon (REALM-2)
 - RFID reader allows inventory and searches for logistics reduction
 - Launch on NG-12
- CASIS's SoundSee (Astrobotic/Bosch)
 - Microphone array that creates a sound map used to monitor health of on-board systems
 - Launch on NG-12
- Gecko Gripper (Stanford)
 - Gecko-inspired end-effector for gripping of smooth surfaces
 - Launch on SpX-18
- Zero Robotics (MIT)
 - Middle school and high school STEM robotics competition
- JAXA Astrobee-IntBall Joint activity, Kibo-RPC
 - Asia-region STEM robotics competition
- Astrobatics (Naval Post-graduate School)
 - Software-only payload investigating propellant-less propulsion using a manipulator



Fun with Robots

Questions?



Astrobees System Overview



Ames Research Center, NASA

Moffett Field, CA, USA

November 25th, 2019



Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



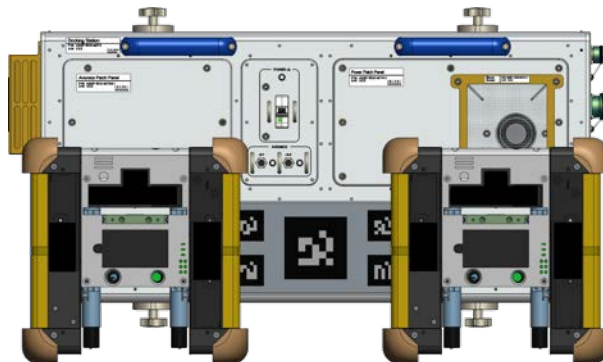
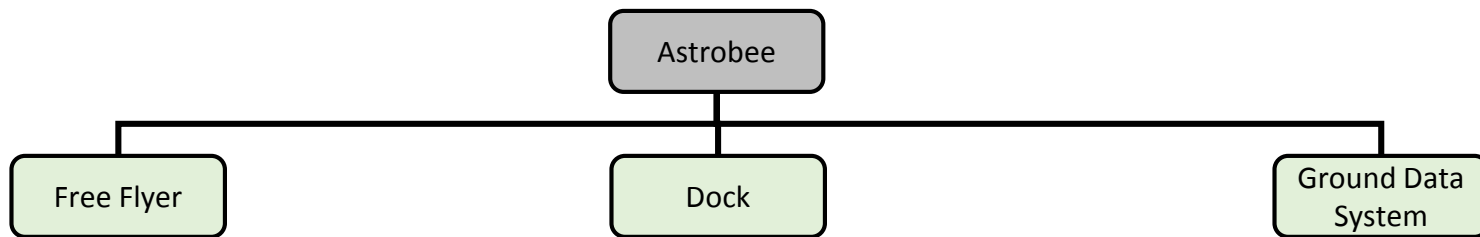
Contents

- [Astrobee System](#)
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



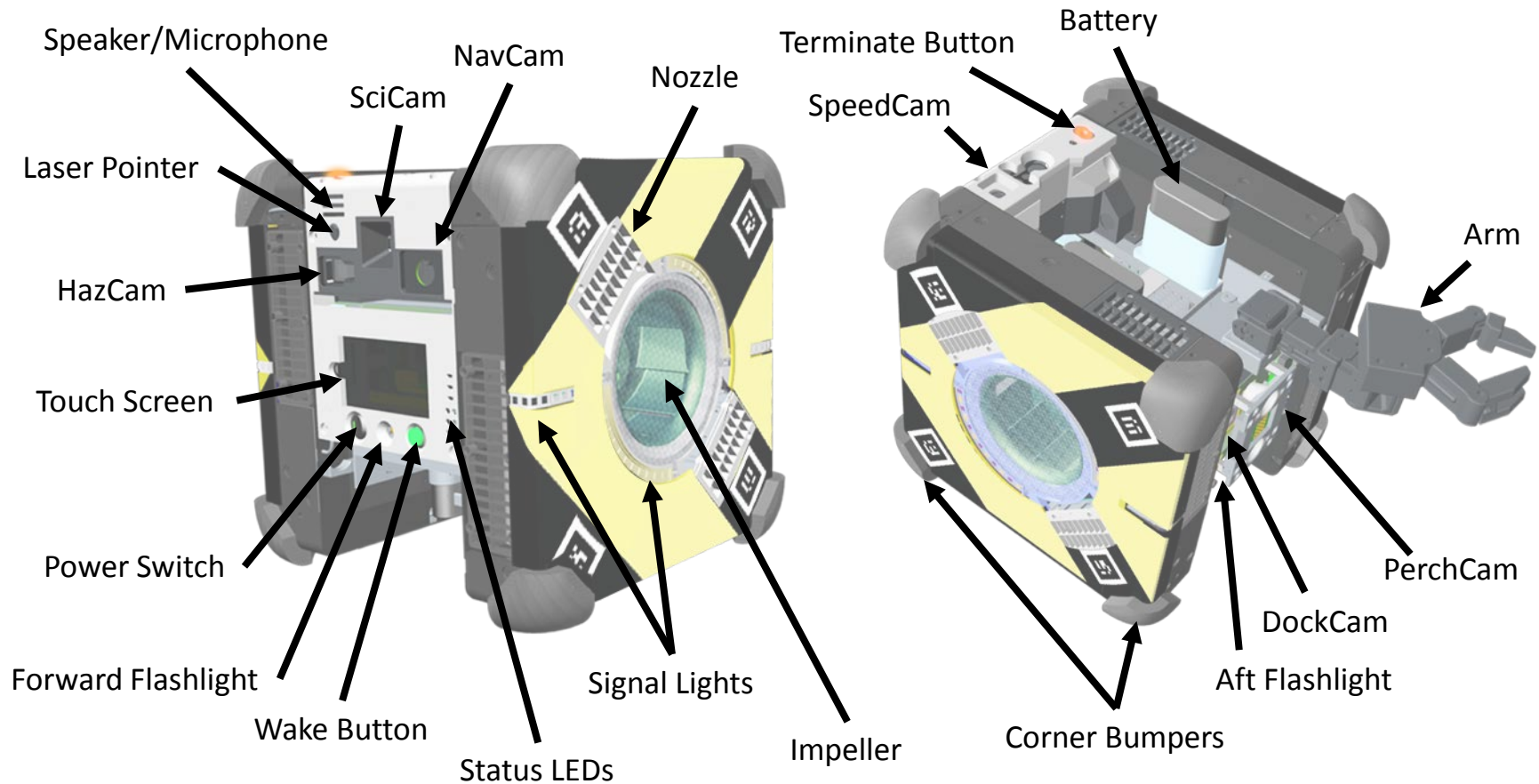
Astrobee System

- Main components





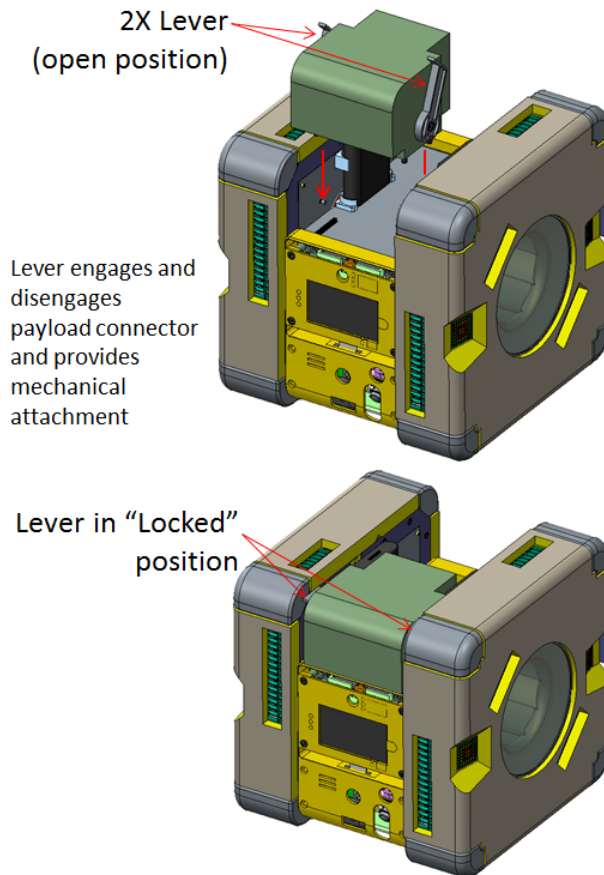
Astrobee System: Free-flyer



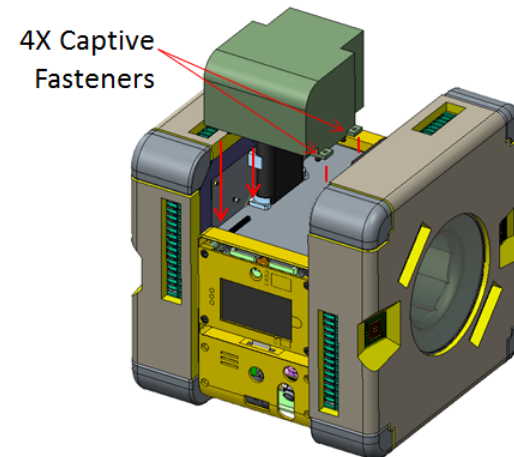


Astrobee System: Free-flyer's Hardware Payloads

Quick "No Tool" Payload Attachment

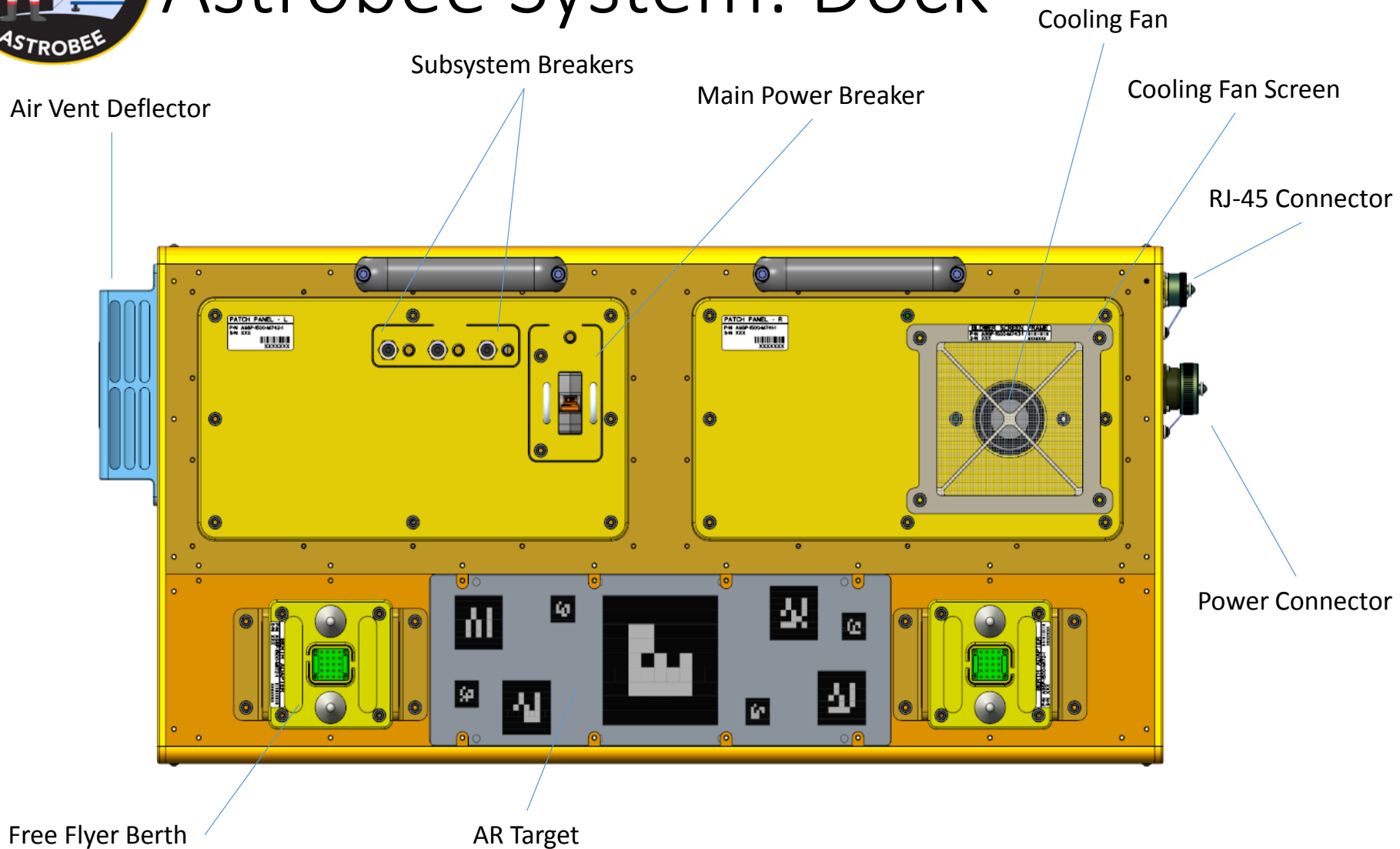


4X Fastener Payload Attachment





Astrobee System: Dock



November, 2019

Asia-Pacific Regional Space Agency Forum
(APRSAF-26), Nagoya, Japan



Astrobee System: Ground Data System

- Provides control over the Astrobee free-flyer
 - by the astronaut
 - by ground controllers
 - by Guest Scientists (GS)
- Enables data exchange between Astrobee and ground
- GDS interfaces Middle Level Processors and Executive
- GDS communicates through Executive with High Level Processor (HLP) where GS Android Apps are deployed





Ethernet:
Internal IP

Ethernet: Internal and Payload LAN

WiFi:
Payload LAN

USB

Other/LAN



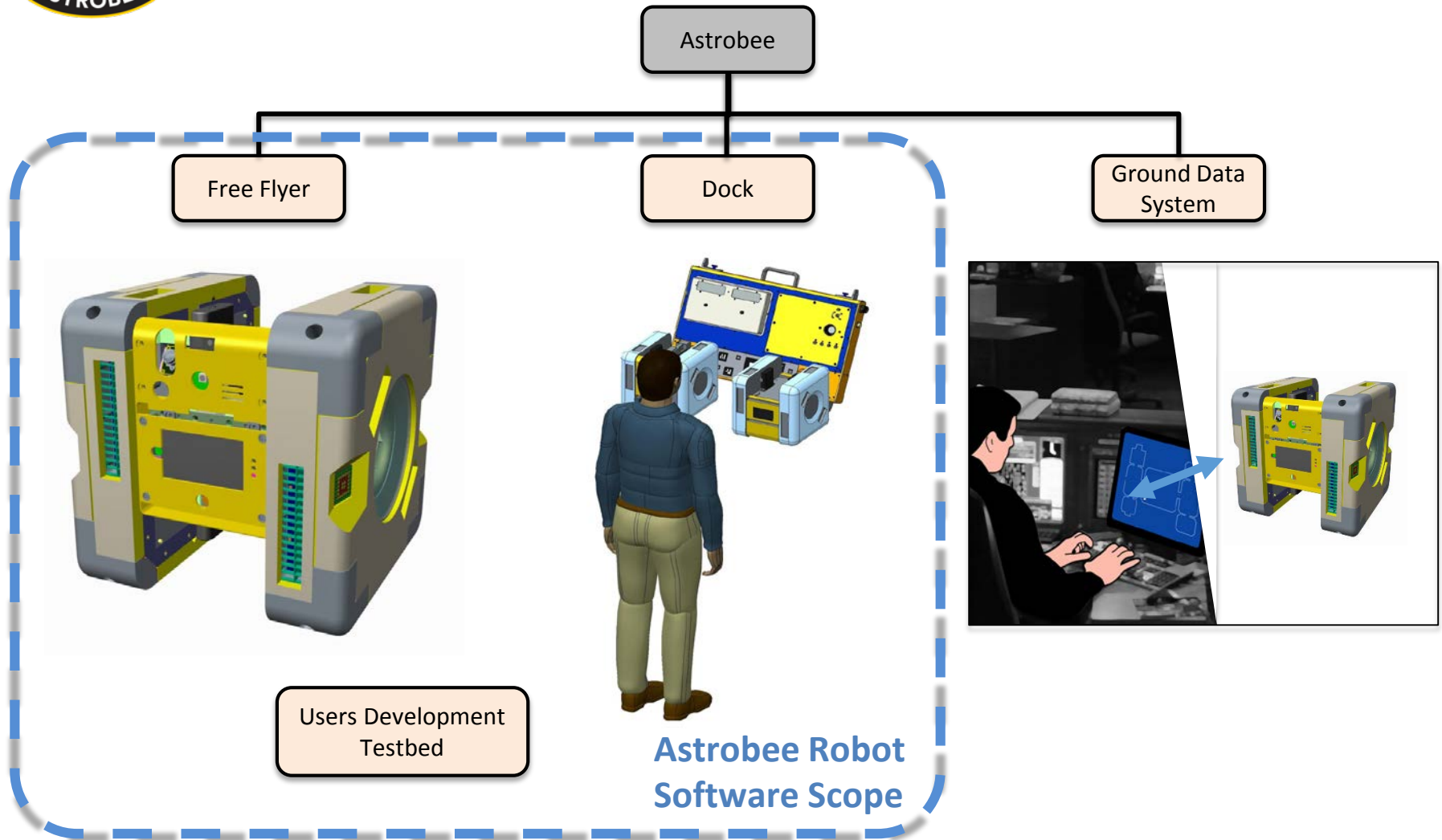


Contents

- Astrobee System
- [Astrobee Robot Software](#)
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



Astrobee Robot Software





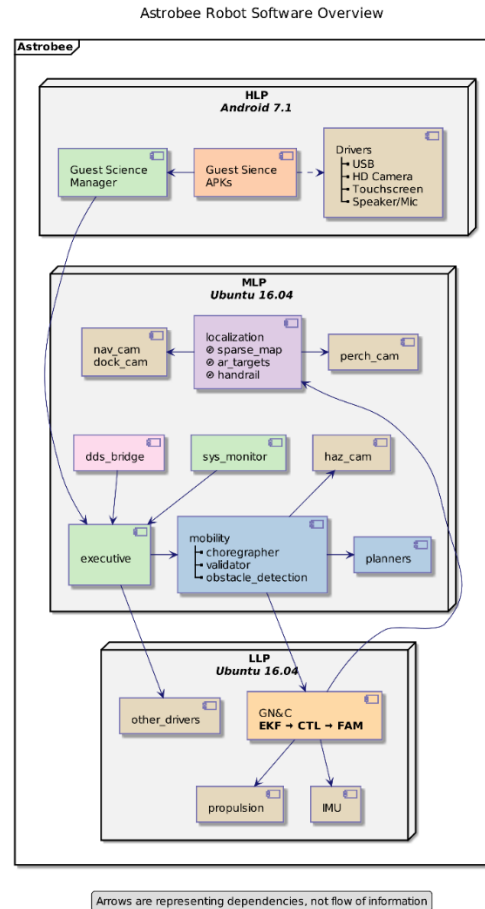
Astrobee Robot Software: Overview

- Astrobee Robot Software (ARS) is deployed on 4 cell phone type processors (Astrobee + Dock) running Linux and Android
- Astrobee contains 7 distinct microprocessors with custom firmware + several microprocessors with Components Off-The-Shelf (COTS) firmware
- Software deliverables includes:
 - Custom firmware(s)
 - Custom tailored Linux kernels
 - Linux and Android Operating Systems
 - ARS dependencies (third-party libraries)
 - **ARS Source Code** ← **Open Source**



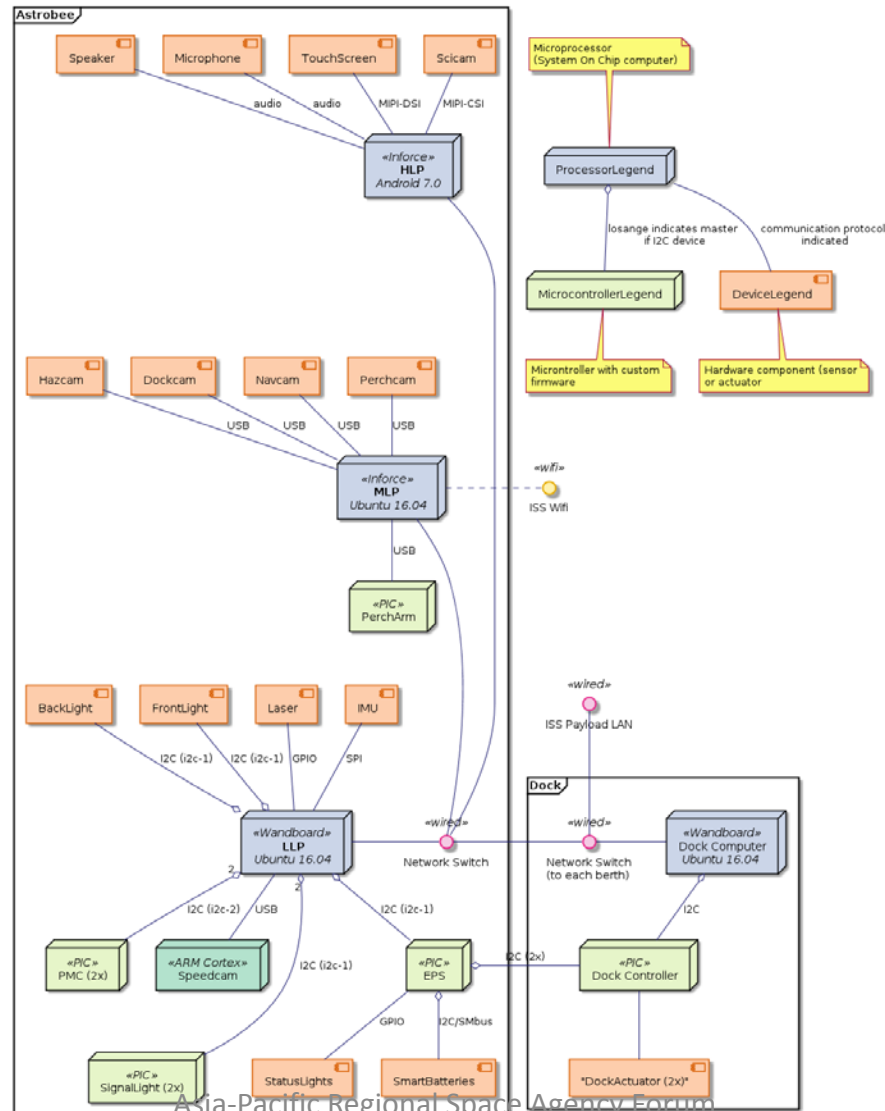


Astrobee Robot Software: Overview





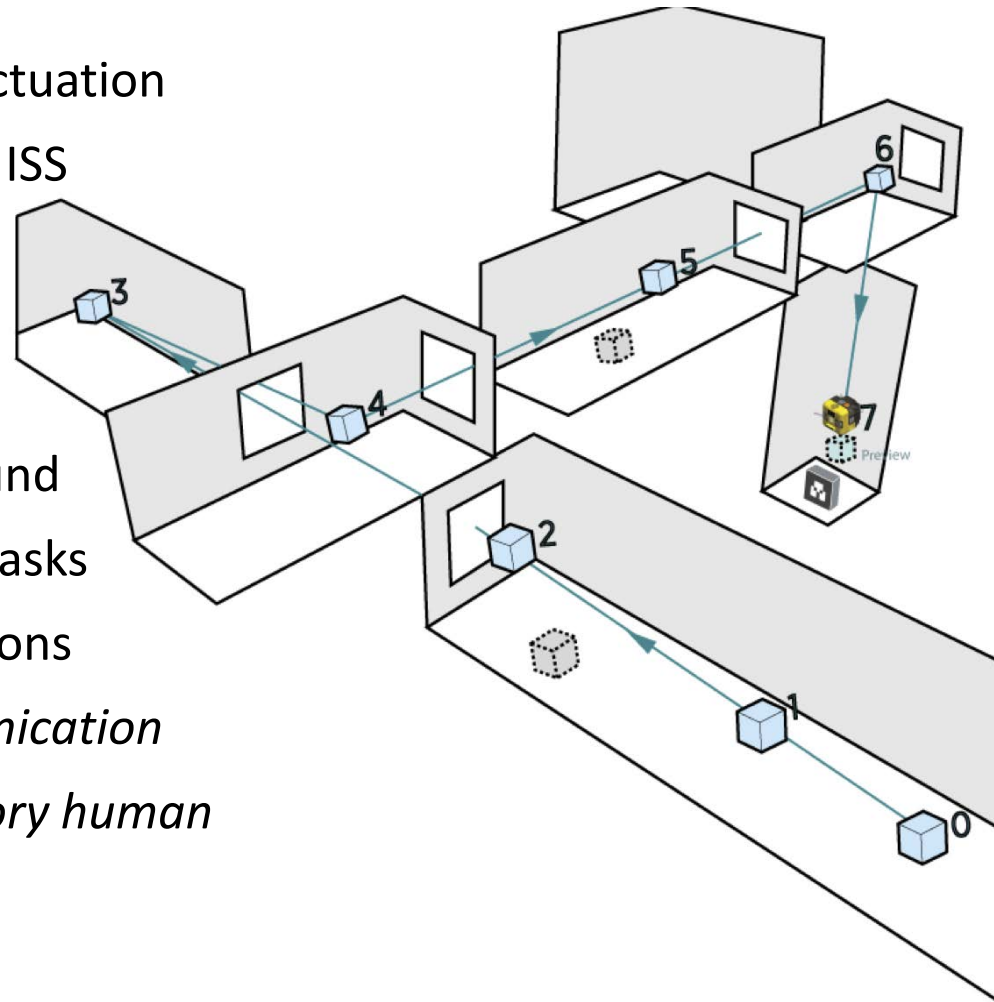
Astrobee Robot Software: Overview





Astrobee Robot Software: Features

- Manage Astrobee sensing and actuation
- Localize and Navigate within the ISS
- Perform autonomous docking (+ return to dock)
- Perform autonomous perching
- Support teleoperation from ground
- Support plan based automated tasks
- Support “Guest Science” operations
- *Support multi Astrobees communication*
- *Support hardware for multisensory human interaction*





Astrobee Robot Software: Components

- Communication Framework (ROS + DDS based)
- Localization
 - Marker less Flying
 - Docking
 - Perching
- Offline mapping for localization
- Pose Estimation + Propulsion Control (GNC)
- Executive
 - Mode Management
 - Sequencer (Plan Execution)
- Mobility
 - Coordinates required subsystem
 - Generates and validates trajectories
 - Performs collision detection
- Fault Management
- Guest Science
- User Interfaces Support
- Simulator
- Platform Management and development tools

Number of ROS nodes for a simulation: ~36

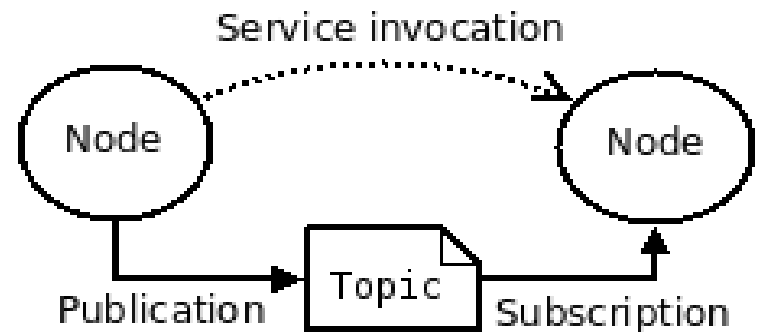
Number of ROS nodes on Astrobee: ~48
(not counting HLP)

Language	files	comment	code
C++	263	11078	45023
Lua and config	76	2179	14380
C/C++ Header	158	4232	8209
Python	58	1798	6706
CMake	150	3631	5624
IDL MSG			
SRV ACTION	148	0	5139
XML	163	309	4271
Java	70	1764	3822
Bourne (Again) Shell	83	853	2522
YAML	4	29	294
make	9	75	238
SUM:	1182	25948	96228



Astrobee Robot Software: ROS

- ARS makes extensive use of the open-source Robot Operating System (ROS):
 - Communication framework linking all “nodes” running on the target platform
 - Try to maximize the re-use of existing ROS messages benefit from existing ROS packages
 - Use ROS introspections tools to rapid debugging
 - Use ROS facilities to record/replay/analyze data
 - Use some ROS/Gazebo components for the simulator





Contents

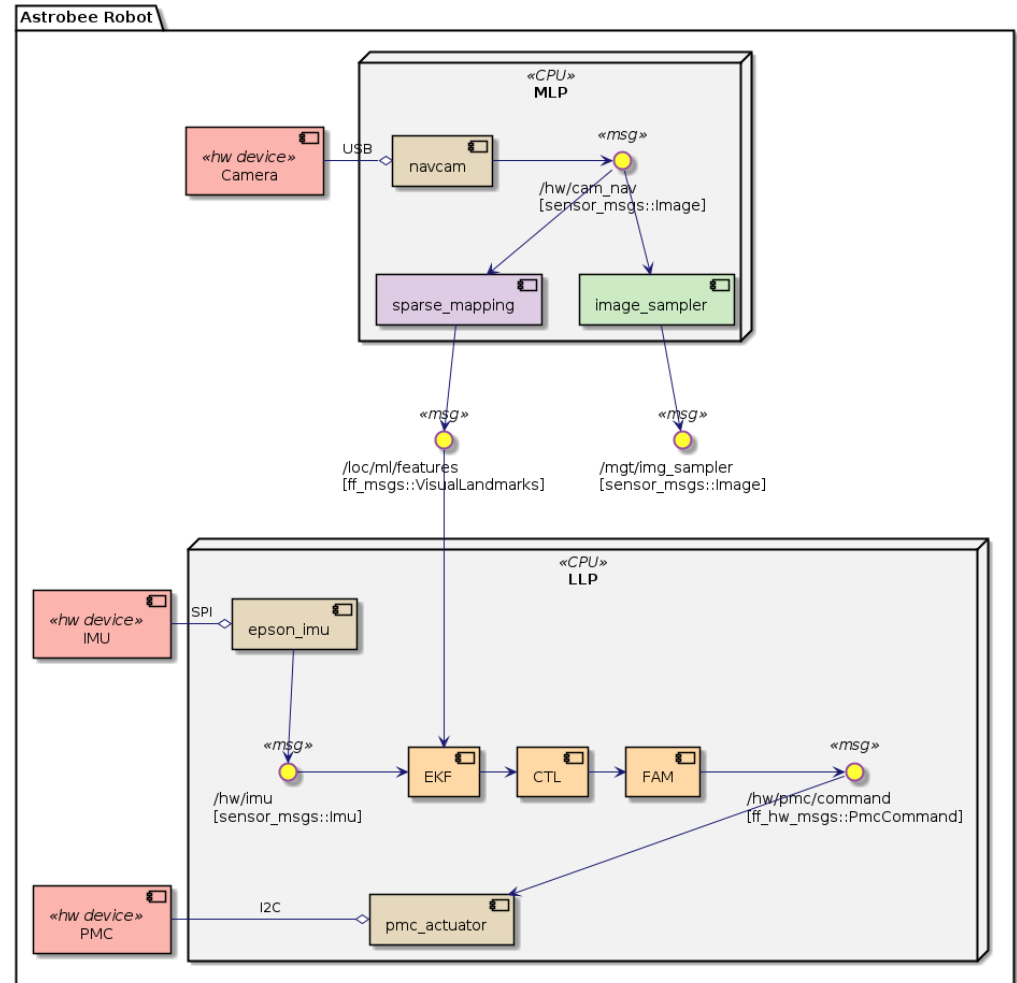
- Astrobee System
- Astrobee Robot Software
- [Simulator](#)
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



Simulator

- ARS can run on:
 - actual hardware on Astrobee (camera, IMU, PMC)
 - simulated robot hardware on a PC
 - both simultaneously
- Simulation is possible by enabling/disabling ARS elements
- Simulator may be thought as a “software-in-the-loop” simulation of the free-flyer’s hardware drivers.
- Hardware drivers are simulated via Gazebo Plug-ins

Astrobee Robot Software deployed on Physical Robot

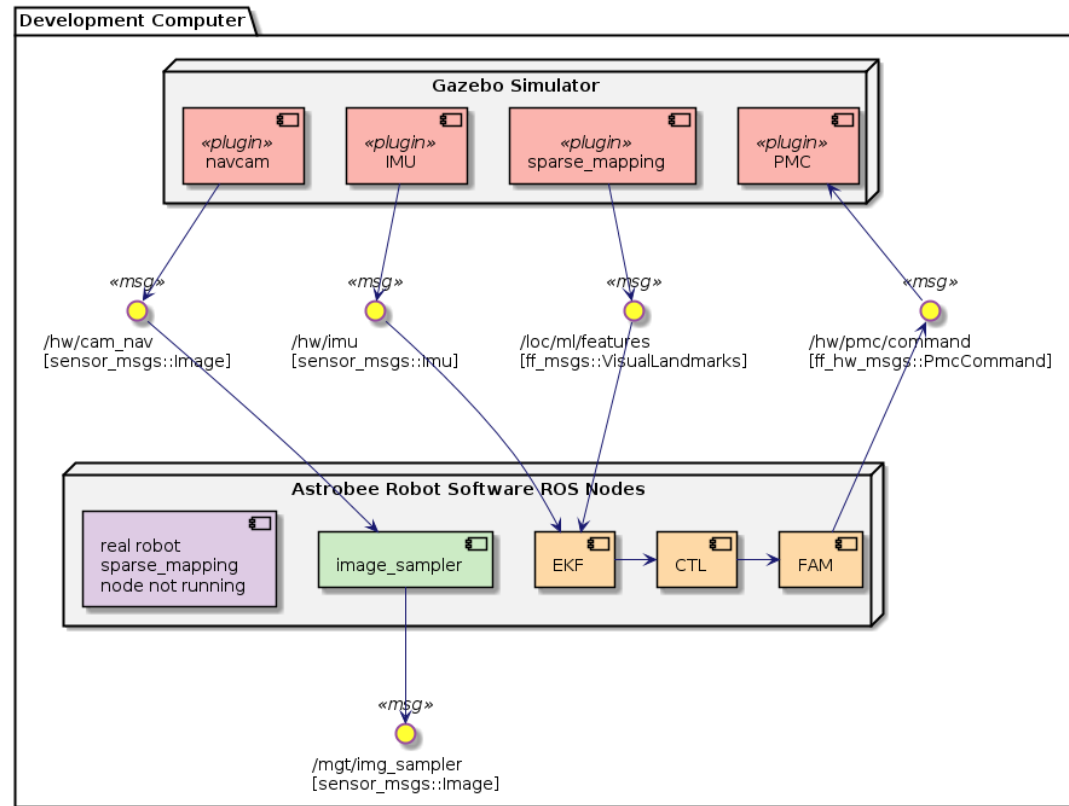




Simulator

Astrobee Robot Software testing with Simulator

- ARS can run on:
 - actual hardware on Astrobee (camera, IMU, PMC)
 - simulated robot hardware on a PC
 - both simultaneously
- Simulation is possible by enabling/disabling ARS elements
- Simulator may be thought as a “software-in-the-loop” simulation of the free-flyer’s hardware drivers.
- Hardware drivers are simulated via Gazebo Plug-ins

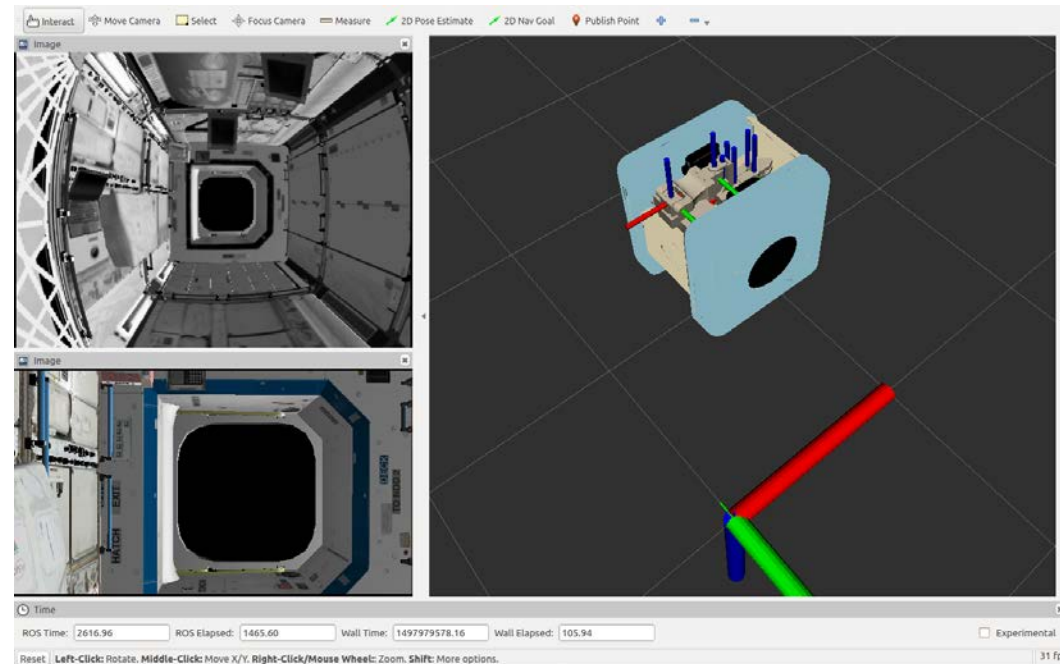


PMC: Propulsion Module Controller
 CTL: GN&C Control
 FAM: GN&C Force Allocation Module



Simulator

- Contains custom propulsion system and some localization sensors
- Gazebo-based dynamics model, imagers, arm, lights and ISS model
- Can run all nodes on desktop or some nodes on target development board



Gazebo Rviz (ROS visualization) used to debug ARS



Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- [Visualizers](#)
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



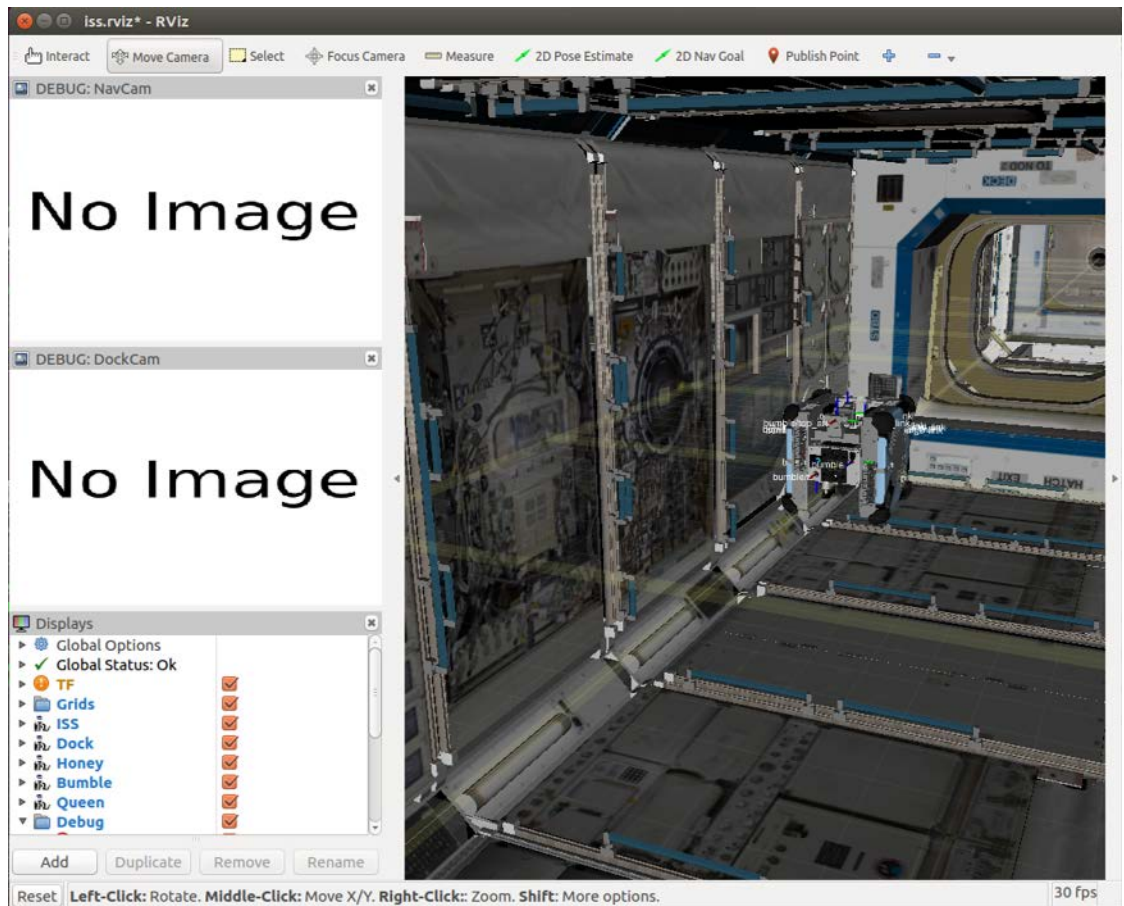
Astrobee Robot Software: Visualizers

- ARS supports 3 main visualizers: Rviz, Gviz, and Sviz
- Simulation can run headless (without any visualizer) or with any combination of them
- Rviz serves as an internal debugging tool from the robot's perspective
- Gviz describes several data streams (e.g. EKF, PMC, etc.) to determine attitude/localization
- Sviz shows a 3D view of the ISS, the Astrobee free-flyer, its arm, and obstacles



Astrobee Robot Software: Visualizers

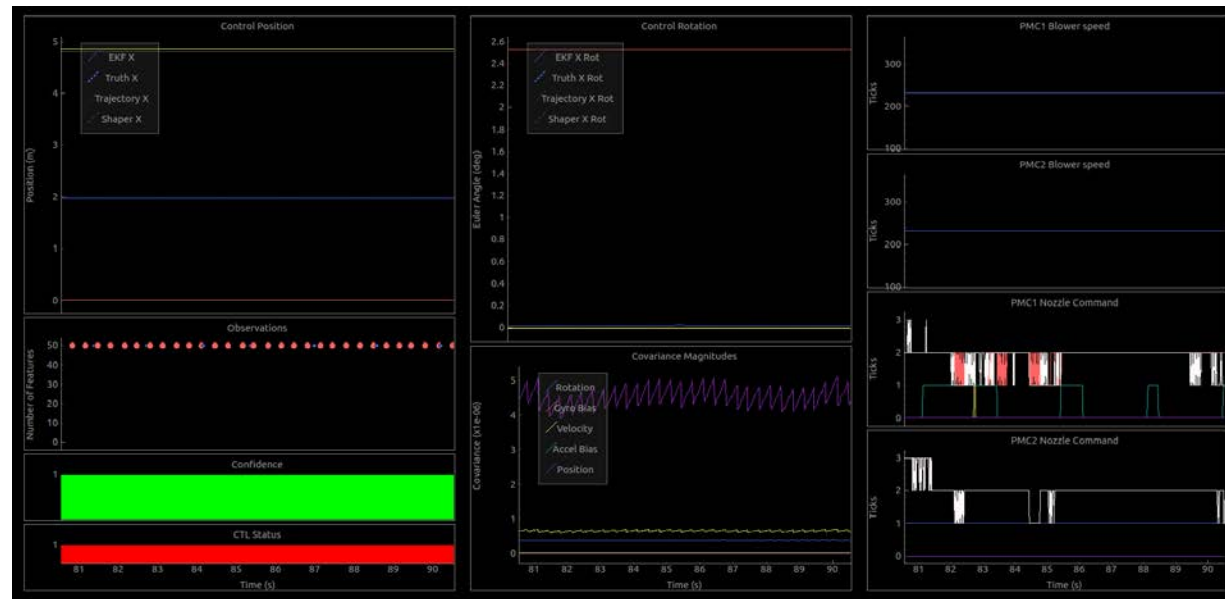
- Rviz





Astrobee Robot Software: Visualizers

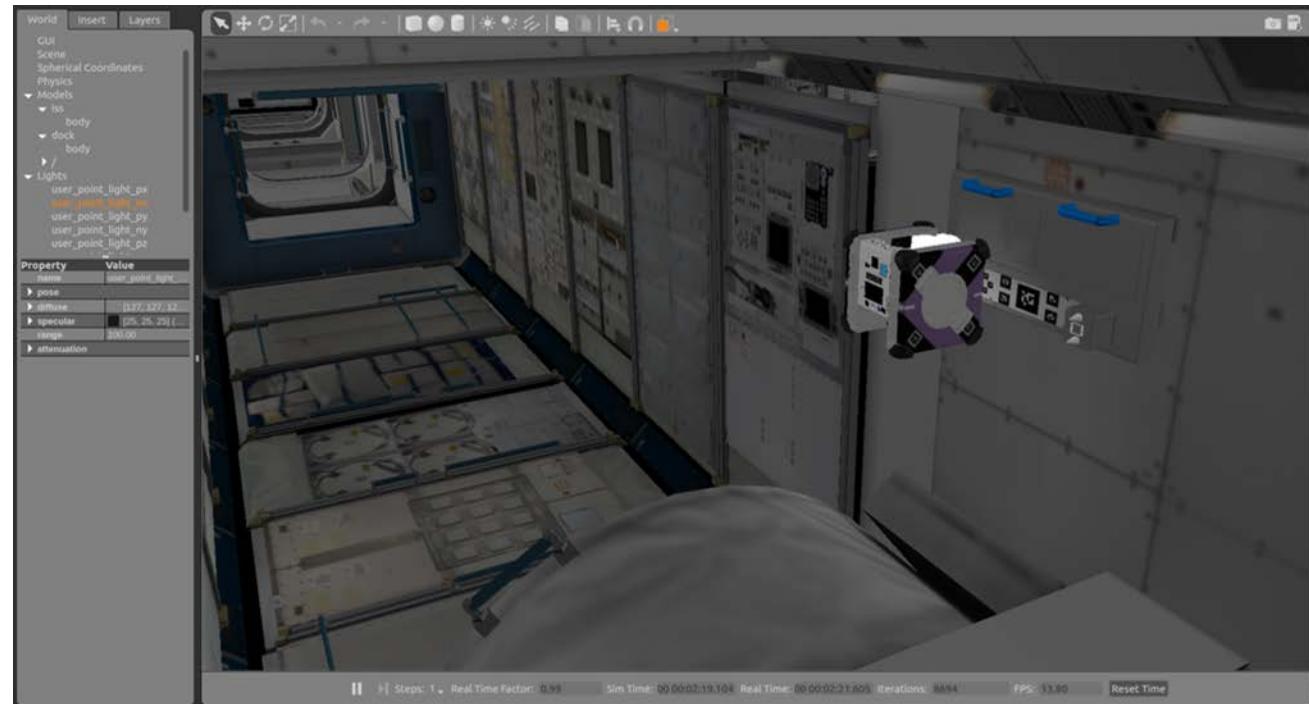
- Gviz





Astrobee Robot Software: Visualizers

- Sviz





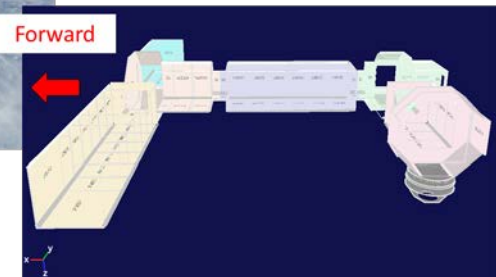
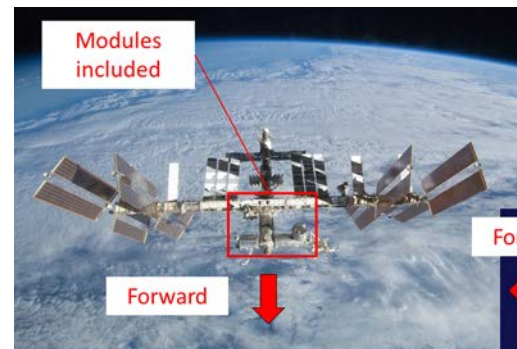
Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- [Ground Data System](#)
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



Ground Data System (GDS)

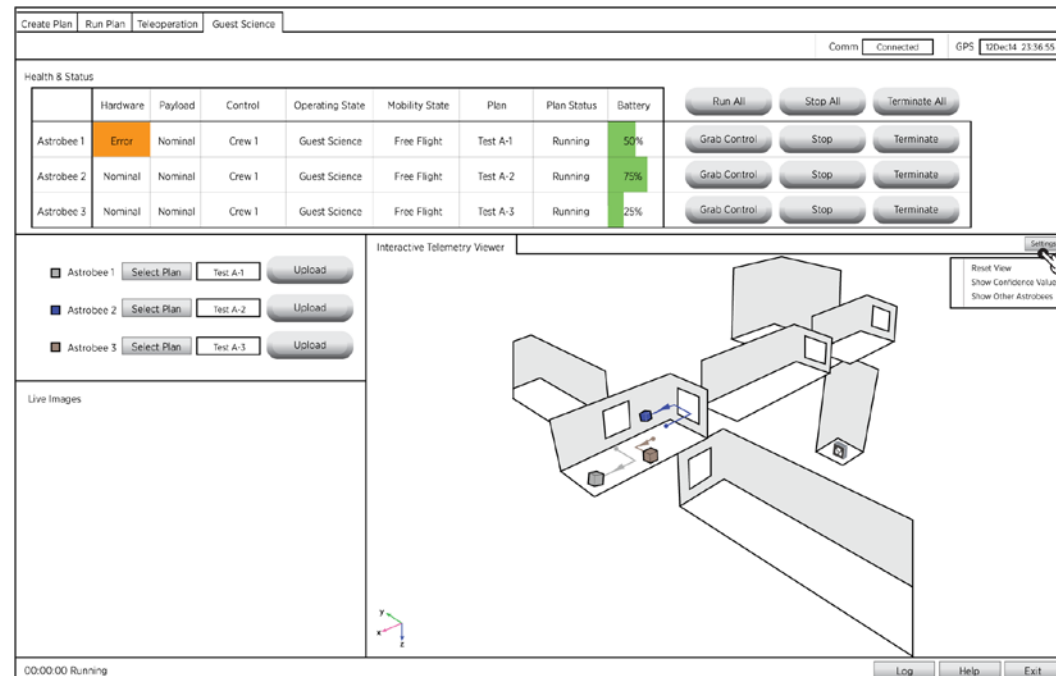
- Ground communication uses DDS middleware to command Astrobee and receive telemetry
- Data Distribution Service (DDS): standard enabling secure real-time information exchange, modular application development and rapid integration (www.rti.com/products/dds-standard)





Ground Data System (GDS)

- Commands:
 - Commands are defined using XP-JSON schema
 - ARS defined a “ROS Command” mirroring the DDS command structure
 - Onboard Astrobee Guest Science or Ground Applications **share the same command dictionary** with either DDS or ROS transport
- Telemetry:
 - Internal uses ROS Messages (using ROS messages when possible)
 - External uses DDS Messages (subset only)



Simplified image for illustrative purposes



Contents

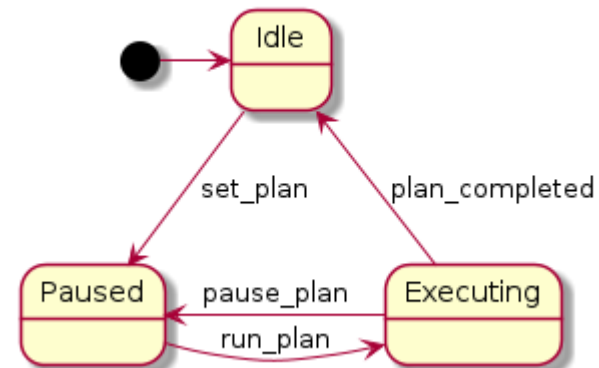
- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- [Executive](#)
- Command Dictionary
- Libraries
- Guest Science Applications



Executive

- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)

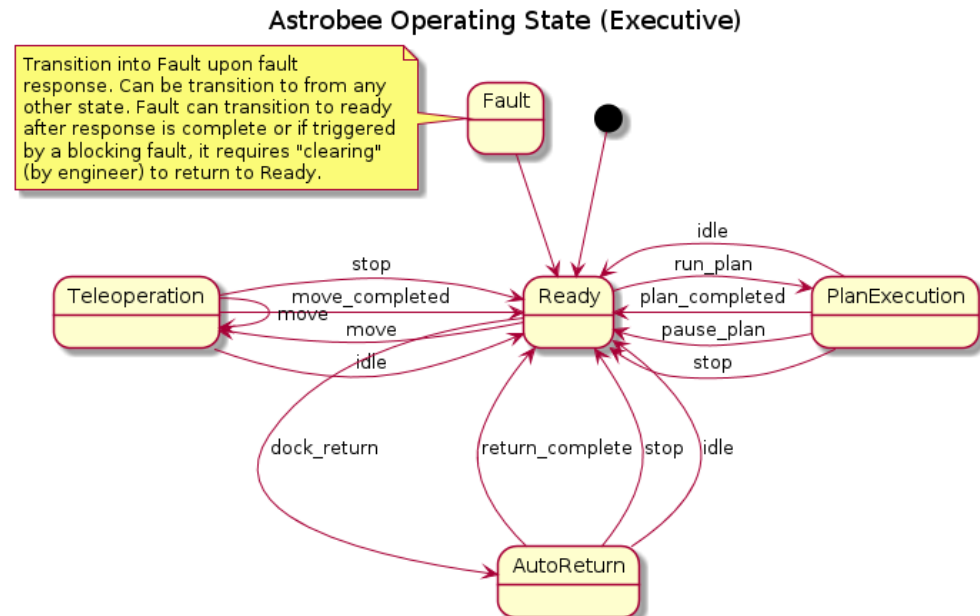
Astrobee PlanExecution State (Executive)





Executive

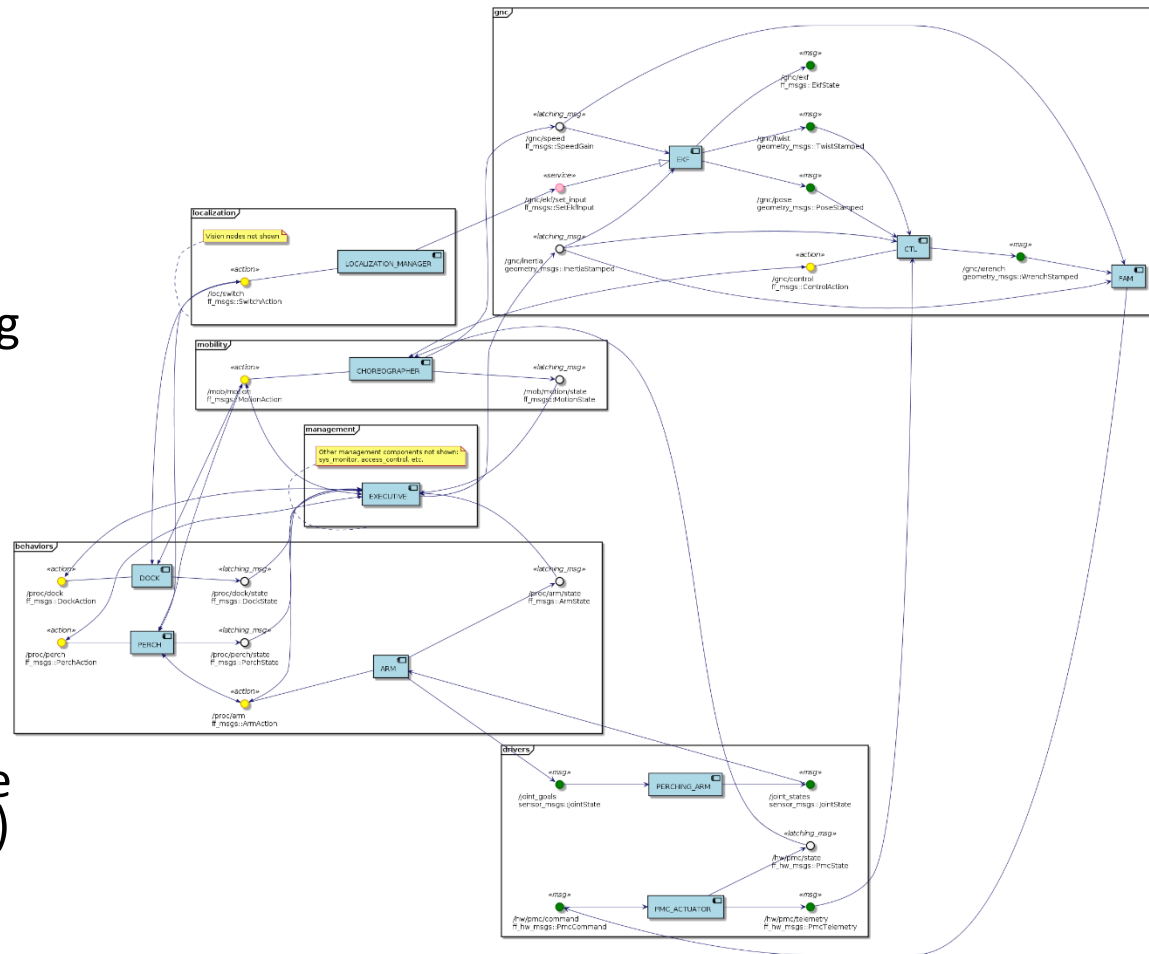
- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)





Executive

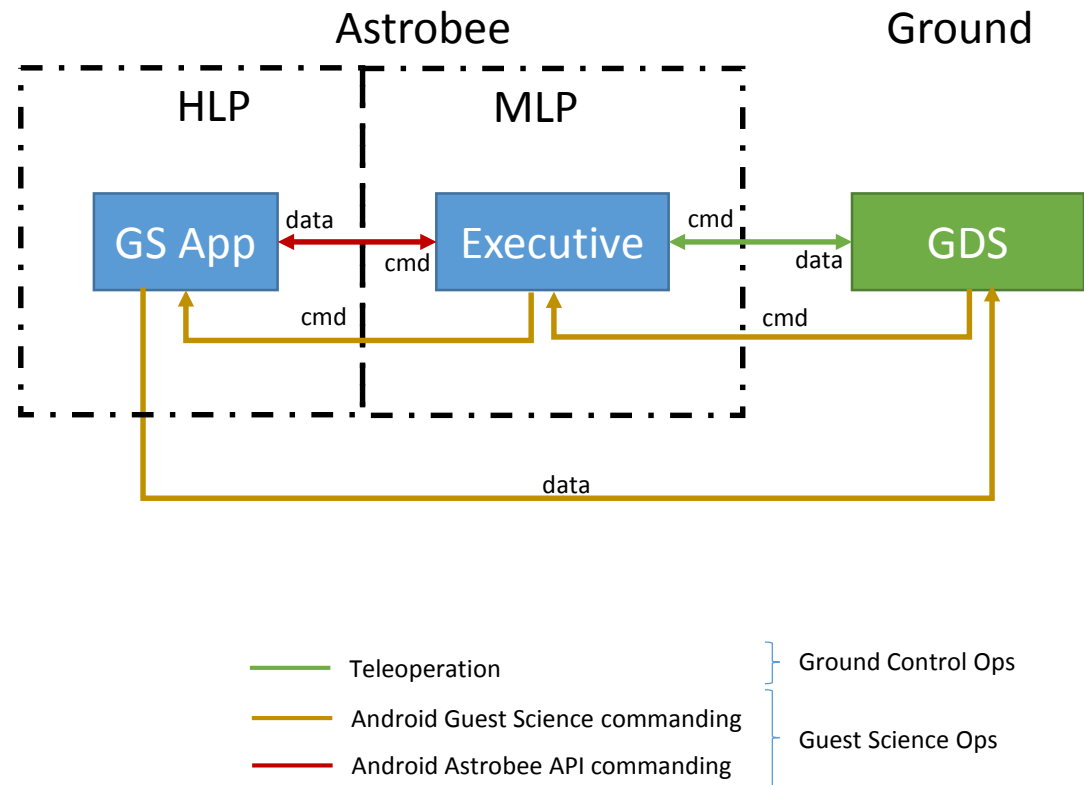
- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)





Executive

- The executive is responsible for keeping track of the operating and mobility states of the robot.
- Responsible for forwarding accepted commands to the correct nodes and informing the ground and system on the status of the command.
- Communicates with ground via DDS-ROS bridge
- Interfaces ground with the High Level Processor (HLP)





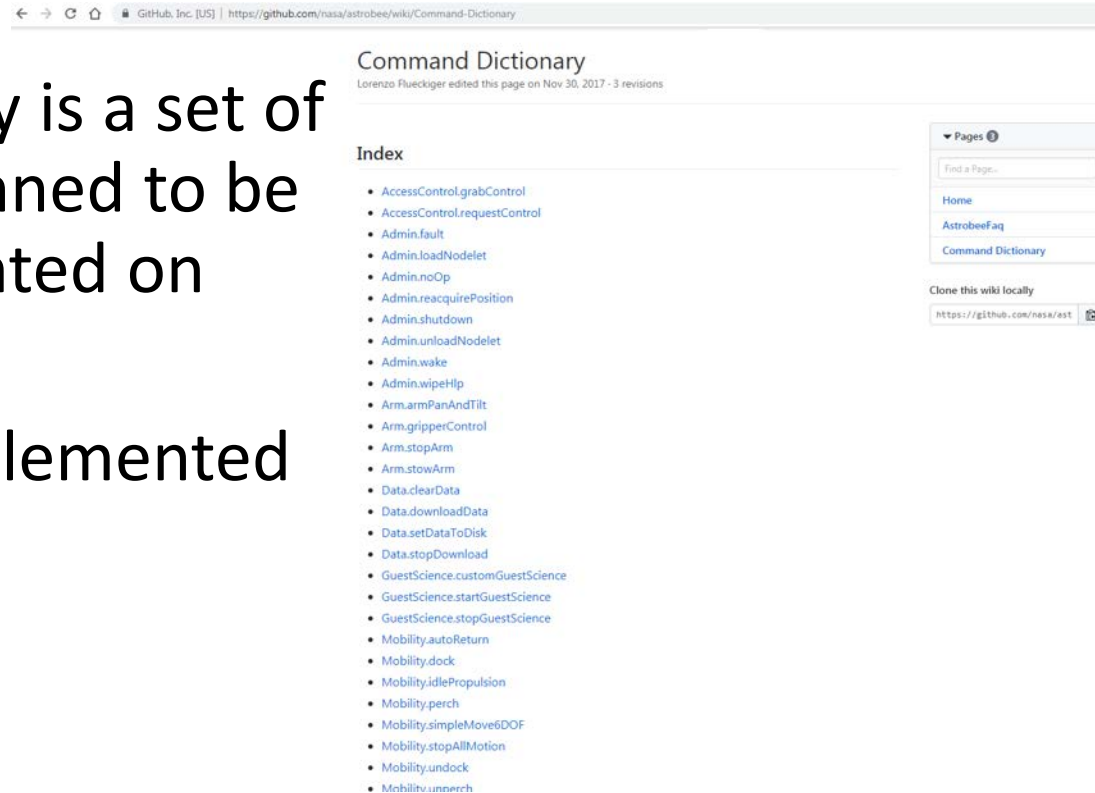
Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- [Command Dictionary](#)
- Libraries
- Guest Science Applications



Command Dictionary

- Command Dictionary is a set of functions either planned to be or already implemented on Astrobee.
- Its functions are implemented in the Executive





Command Dictionary

- These functions
 - help control the robot via commands
 - allow guest scientist to start/stop their Android apps from the ground

← → ↺ 🏠 GitHub, Inc. [US] | https://github.com/nasa/astrobee/wiki/Command-Dictionary#command_GuestScience.startGuestScience

GuestScience.startGuestScience

Start guest science APK

Parameter	Type	Default	Notes
Apk Name	string	required	Specify which guest science APK to start

GuestScience.stopGuestScience

Terminate guest science APK

Parameter	Type	Default	Notes
Apk Name	string	required	Specify which guest science APK to terminate

Mobility.autoReturn

Mobility.dock

Dock Astrobee. Must meet dock approach preconditions (positioned at dock approach point, etc).

Parameter	Type	Default	Notes
Berth Number	long	required	Berth number can only be 1 or 2.

Mobility.idlePropulsion

Stop propulsion impeller motors

Mobility.perch

Mobility.simpleMove6DOF

Astrobee teleop move command

Parameter	Type	Default	Notes
Reference Frame	string	"ISS"	which reference frame to use
Xyz	Point	required	target point
Xyz Tolerance	array[3].double	[0, 0, 0]	Not used! Tolerance is dictated by the flight mode.
Rot	quaternion	required	target attitude

Mobility.stopAllMotion

Stop teleop motion. Stop plan execution and pause plan.

Mobility.undock

Undock Astrobee



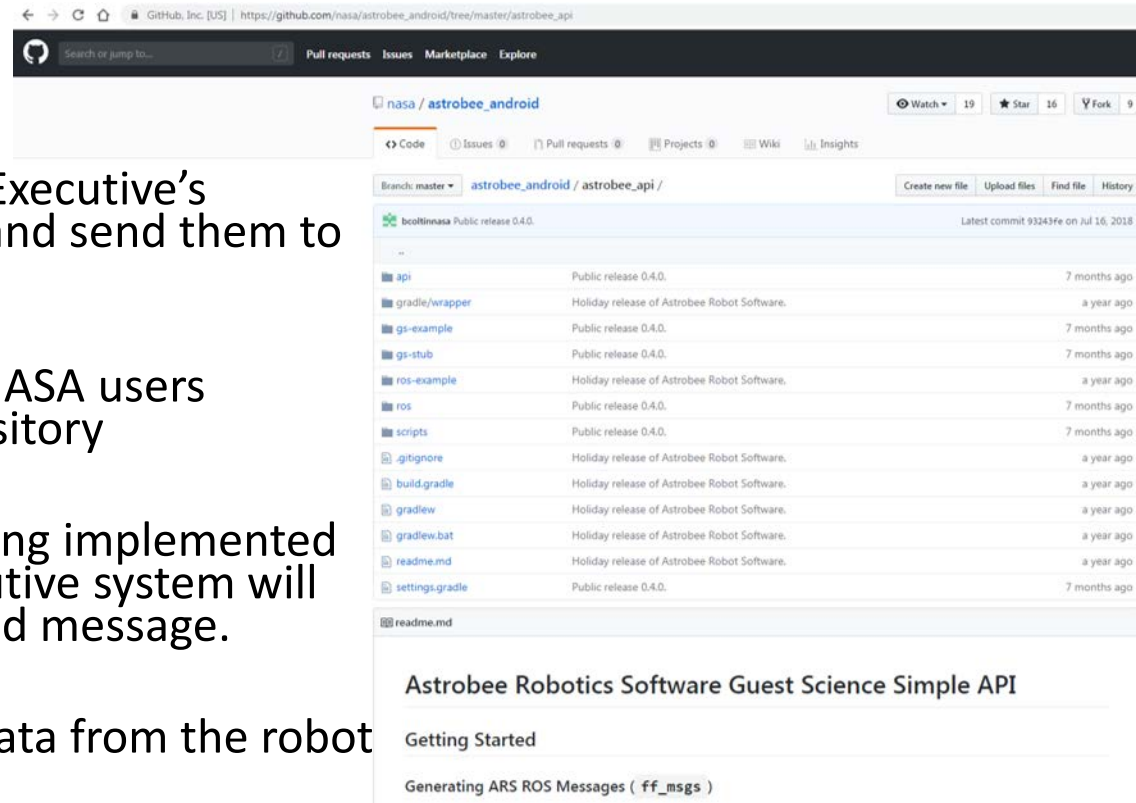
Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



Libraries

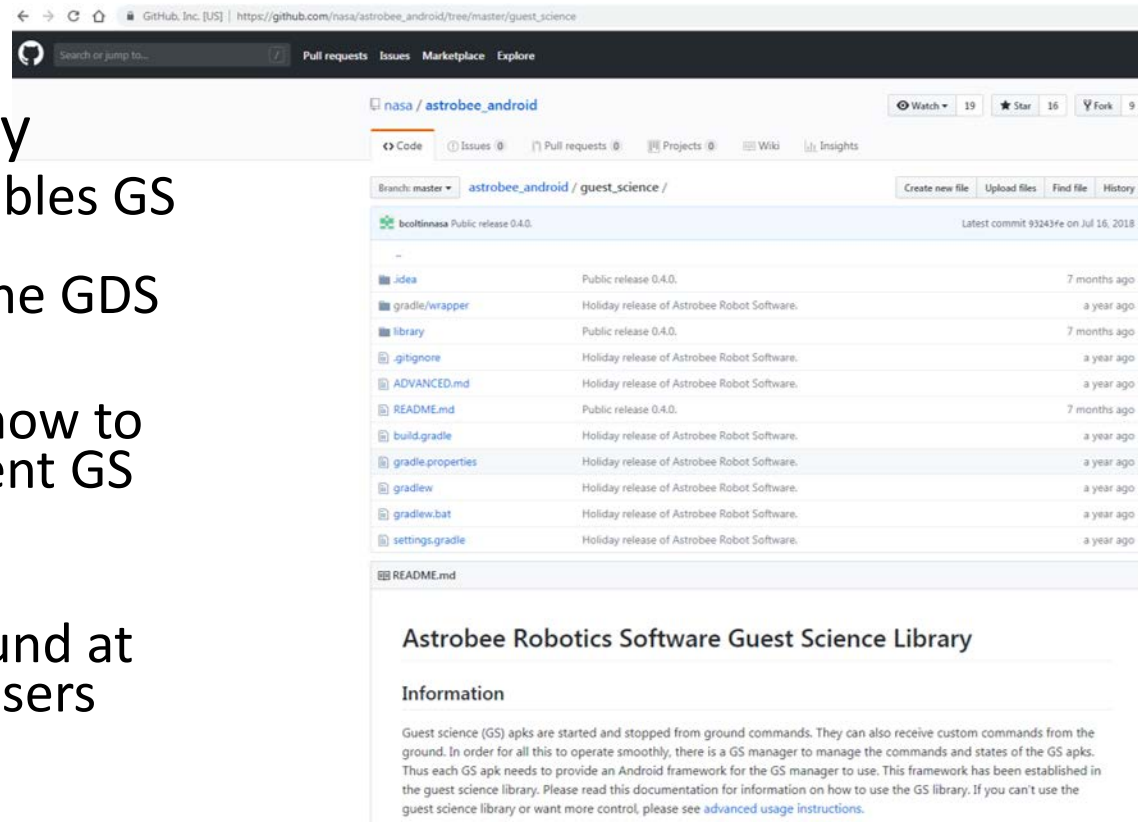
- Astrobee API Library
 - Java API/library wraps Executive's Dictionary commands and send them to the robot.
- Found at GitHub Non-NASA users Astrobee Android repository
- If a command is not being implemented yet, the Astrobee Executive system will return a failed command message.
- Provide means to get data from the robot
- Utilized by the Guest Science Code (Android App) at the HLP





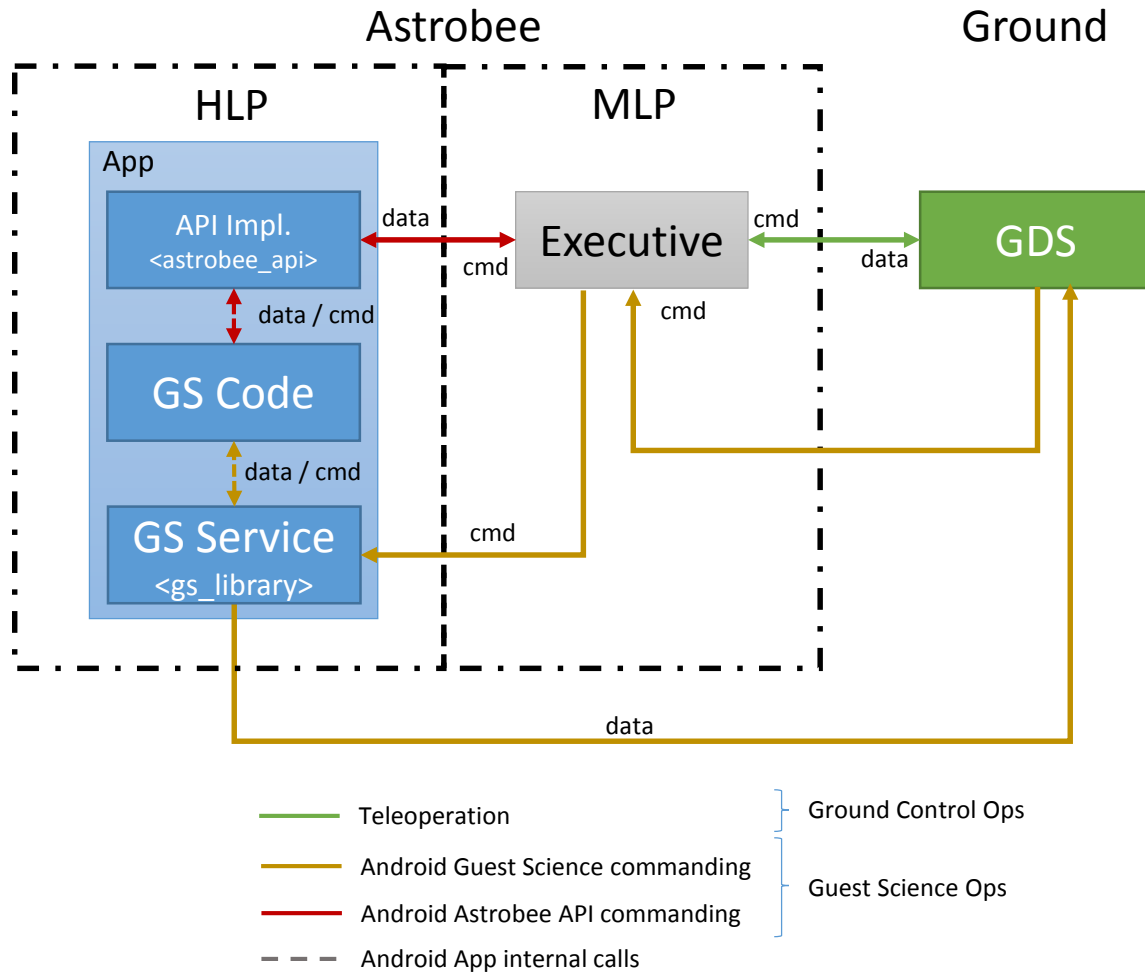
Libraries

- Guest Science Library
 - Java library that enables GS applications to be commanded from the GDS
 - Sets a standard on how to design and implement GS apps
 - Examples can be found at GitHub Non-NASA users repository
 - Utilized by the Guest Science Service (Android App) at the HLP





Libraries





Contents

- Astrobee System
- Astrobee Robot Software
- Simulator
- Visualizers
- Ground Data System
- Executive
- Command Dictionary
- Libraries
- Guest Science Applications



Guest Science Applications

- Guest Science Applications
 - Two types:
 - Primary (commands robot, gets data, internal computation, payload communication)
 - Secondary (gets data, internal computation, payload communication)
 - Standard components:
 - GS API implementation (`astrobee_api`)
 - GS Service implementation (`gs_science_lib`)
 - Remote capabilities from GDS:
 - Starting/Stopping
 - Custom commanding and data feedback

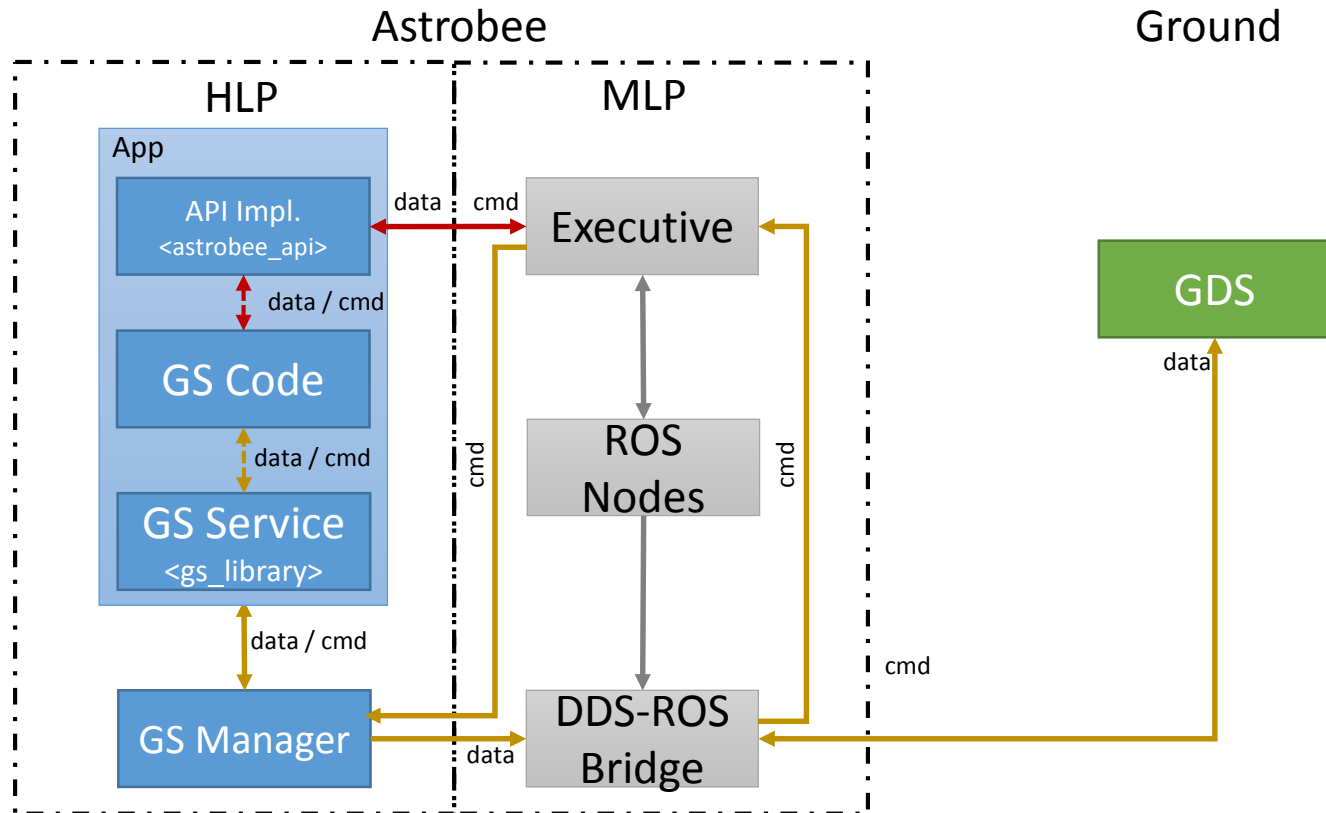


Guest Science Applications

- Guest Science Manager
 - GS Manager is an Android App of its own (Android Service)
 - Manage the commands and states of the GS Apps
 - Each GS App needs to provide an Android framework for the GS Manager to use
 - In charge of interfacing GS Android Apps and the GDS through the Executive system and the DDS-ROS Bridge



Guest Science Applications



*Teleoperation is not represented in this data/cmd flow diagram

- Android Guest Science commanding
 - Android Astrobee API commanding
 - - - Android App internal calls
- Guest Science Ops



Guest Science Applications

Astrobee Guest Science Architecture

