Lunar Mobility as a Service in the Next Five Years: A Software Perspective. Taylor JL Whitaker[1], Chris Rampolla[2], Cedric Corpa De La Fuente[3], Mike Provenzano[4], Astrobotic Technology, 1016 N. Lincoln Ave. Pittsburgh, PA, {[1]Taylor.Whitaker, [2]Chris.Rampolla, [3]Cedric.Corpadelafuente, [4]Mike.Provenzano}@astrobotic.com

Introduction: The next five years of Lunar surface exploration will play a major role in the preparation for and the eventual establishment of a persistent presence on the Moon. Though, it's not humans that will be doing the initial exploration and experimentation, but rather robotic systems laying the foundation for permanent human presence. An increased understanding of the Lunar environment and infrastructure in terms of mobility, power, and surface structures are required to support a sustained presence, and robotic systems are an invaluable resource for this endeavor. A core requirement of any surface asset contributing to this effort is mobility.

Mobility as a Service (MaaS) is a familiar concept, here, on Earth as the proliferation of car-sharing and ride-sharing industries has virtually removed the need for individuals to rely on personal methods for transportation. However, simply attaching a payload to a rover destined for the Lunar surface, has up to this point not been possible. MaaS is at the core of Astrobotic's CubeRover, a small rover platform with colossal benefits for surface exploration as its affordable, modular, and scalable. From academic sensors to robotic arms, the CubeRover platform can provide surface mobility for science applications, laying power cables, or even major structural construction.

Though CubeRover will be among the first to establish the MaaS model on the Lunar surface, it will not be the last and thus the eventual diverse ecosystem of surface assets was a vital consideration in the development of the platform. The CubeRover platform has given significant insights into this eventuality, especially regarding the software perspective.

A MaaS Software Ecosystem: Developing a MaaS software ecosystem in the context of establishing Lunar infrastructure extends beyond just the flight and ground software components of a single rover platform or even a single MaaS provider. Flight software (FSW) must facilitate the specific mission of a platform but should also maintain a high degree of adaptability to easily integrate into an environment of diverse Lunar assets. Ground software (GSW) must support the functionalities of individual platforms, but also enable higher-level management and coordination of fleets of assets.

Flight Software Architecture. FSW for a spacecraft is inherently tied to the craft's mission and the available hardware resources, processors, sensors, actuators, etc., and can lead to large development overheads for recurring missions. FSW frameworks such as NASA's core Flight System (cFS) [1] aid in this respect as NASA and community contributions enable a vast toolbox of functionalities to be leveraged and the modular framework reduces overheads in core software development to support the spacecraft and allows reuse across missions. Considering CubeRover is a mobility platform, FSW supporting CubeRover's base functionality also requires that payloads be supported. Additionally, considering the range of complexity in possible payloads and use-cases, payload support can range from basic power supply and heat management to provisioning processing power for tighter integration of payload applications. The former case could easily be supported with a cFS framework and specific power and temperature management application plugins. However, the latter case shouldn't impose the software framework used for the base functionalities of the platform on the payload developers, but rather allow software ecosystems such as ROS [2] to be supported alongside.

Astrobotic has approached the design of FSW such that core functionalities are isolated away from sandboxed payload functionalities with both hardware and software APIs providing an interface with the core system. This prevents CubeRover from requiring payload developers to concern themselves with the same rigorous qualifications needed to classify the platform as flight-safe and enables a larger range of supported payloads and missions. This segmentation of FSW drastically improves the modularity of the platform even beyond payload support, as high-level software systems can run safely with core applications.

Ground Software Architecture. The need for a front-end UI to control and monitor planetary assets is clear. But current open-source command and control solutions do not account for MaaS models that must handle significant increases in demand for data throughput of surface assets. Astrobotic aims to take an aggressive position in the nascent Lunar MaaS market by leveraging proven big-data technology that can handle high message throughput with built-in redundancy and structuring its ground services around it. This will make ground capabilities more attractive for payload customers who will require ever more demanding computational and data resources. Further, it guarantees the ability to scale gracefully without the need to re-architect the entirety of the ground segment to integrate new surface assets and missions.

References:

[1] D. McComas (2012) *NASA/GSFC's Flight Software Core Flight System*. [2] M. Quigley et al. (2009) *ROS: an open-source Robotic Operating System.*