Open source, open standards, and collective invention in the space industry. J. Woods and C. McMahon, Open Lunar Foundation, 399 Webster Street, San Francisco, CA, 94117. (Contact: juno@translunar.io)

Introduction: Collective invention occurs when free exchange of information enables rapid technological advance, and differs from individual invention and commercial invention (e.g. research and development). In an academic context, one example of collective invention is the open science movement. In the for-profit world, nominal competitors may work together on key infrastructure, called pre-competitive collaboration.

Normally, the products of these efforts are eventually privately recaptured, but the free/libre and open source software (FLOSS) movement has created a legal mechanism to prevent that recapture. Moreover, collective invention is often a product of specific engineering cultures or participant ideologies. Silicon Valley engages in pre-competitive collaboration by producing open source infrastructure as a foundation for proprietary, closed source innovations; however, space industry collaborations are much rarer.

We briefly review advantages and disadvantages of FLOSS, open hardware, and open standards. We discuss key barriers in the aerospace industry, as well as potential motivators for renewed participation, and make recommendations based on interviews conducted with anonymous space industry executives and several years of experience running open source projects.

Background: While the commercial space industry relies extensively on collective invention (e.g. academic and NASA research and development, as well as substantial tech industry infrastructure), it contributes back much more rarely than the tech industry. While there exist a large number of successful FLOSS projects in the space industry, most are primarily government or academic in origin and support. The culture of the open source movement has existed independently of the legal mechanism, having inherited much from its origins in the academic open science movement.

Pre-competitive collaboration: Explicit in the term ‘pre-competitive’ is that such collaborations revolve around non-differentiators (those technologies that help businesses compete against others in the economic niche are known as differentiators). An example of pre-competitive collaboration is the open consortium model, as demonstrated in the GENIVI Alliance, which produced a Linux-based platform for in-vehicle entertainment, and publishes a variety of open standards. GENIVI was founded in 2009 between auto industry competitors including OEMs such as BMW, Honda, and Hyundai, as well as other supply chain participants like Clarion, Bosch, LG, Garmin and Nvidia. The space industry would greatly benefit from similar open consortia.

Economics of open source: To be adopted broadly, FLOSS projects must nearly always be coded and well documented with reuse in mind. In studies on closed source software, building in reusability entails an up-front cost (~2–5x), but produces a positive return-on-investment within a few years; moreover, the cost of integrating components written with reuse in mind is a fraction of the cost of writing new components.

Incentives for collaboration: Commercial entities considering open source business models ought to consider not only the existing market but how to facilitate the existence of a future market. Organizations working outside of low-Earth orbit would benefit more from actions that increase the market size than from finding customers among the currently extremely limited market.

We stand at an inflection point in the growth of the commercial space industry. The cultural norms at space industry companies today will shape the norms and laws of the societies we build in space.

References: [1] G. Andersson and T. Guild, “Common Vehicle Interface Initiative: A standards-based approach to vehicle data & services” (2021) [2] J. Margono and T. E. Rhoads, Proceedings of the Software Reuse and Reengineering Conference, (1991). [3] A. Lynex and P. J. Layzell, Proceedings of the International Workshop on Software Technology and Engineering Practice, STEP, pp. 339–349, (1997).