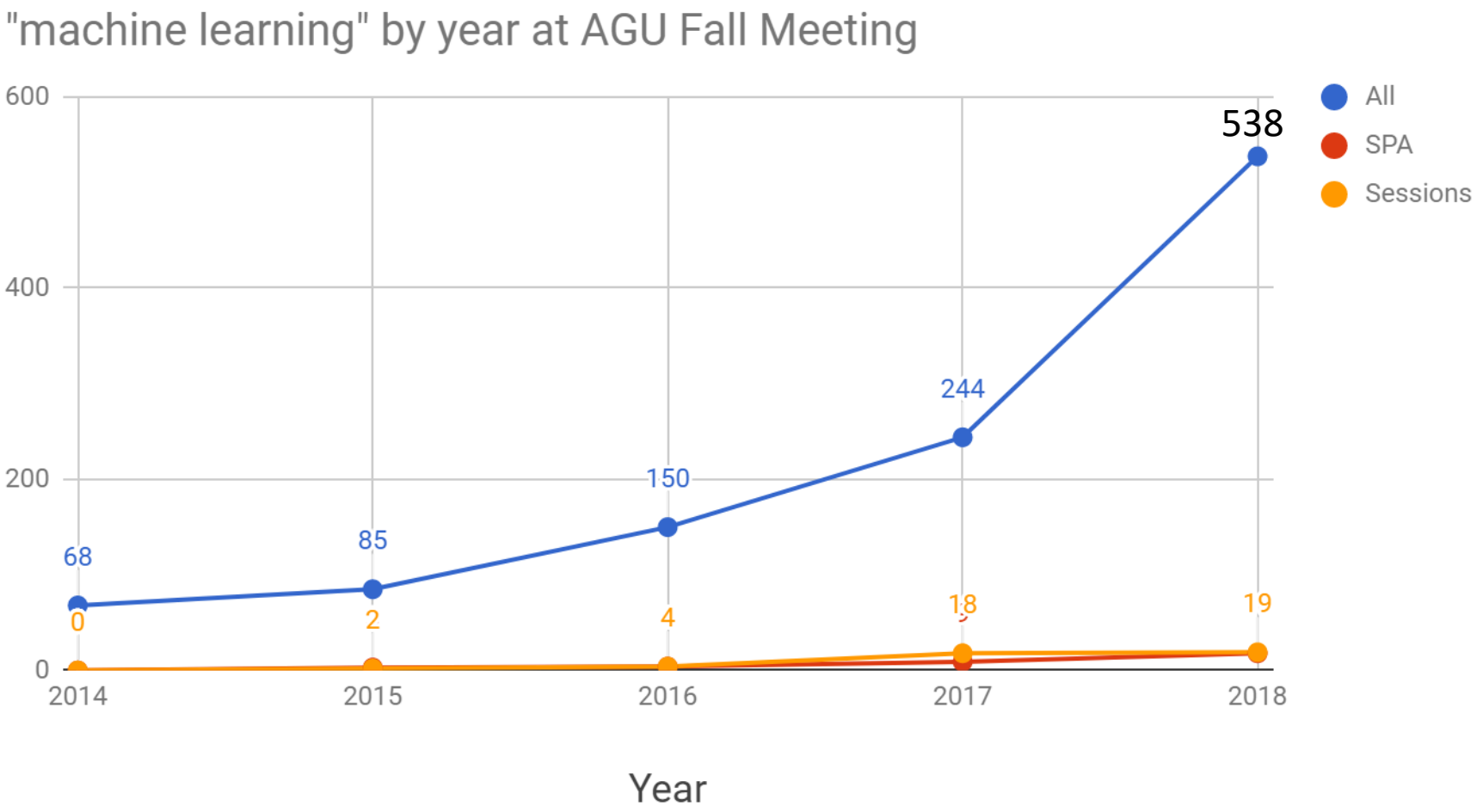
Software Engineering for Heliophysics

CEDAR 2019 Workshop

Monday 17 June 2019

Zia / Eldorado

Heliophysics data science by the numbers



CEDAR Workshop

“machine learning”:

- 2016: 1
- 2017: 1
- 2018: 4
- 2019: 11

Science / Engineering Frameworks

- accelerate time to completion (paper, project)
- essential to heliophysics reproducibility & archiving (agency directions)
- infrequently discussed in formal sessions or literature in our field

- Why does software / data science work inevitably link to open-source?
- Why is everyone talking about Python / Julia / R ?

Jan 24, 2019: https://github.blog

Top Machine Learning Languages on GitHub

1	Python
2	C++
3	JavaScript
4	Java
5	C#
6	Julia
7	Shell
8	R
9	TypeScript
10	Scala

Packages Imported by Machine Learning Projects on GitHub

1	numpy	74%
2	scipy	47%
3	pandas	41%
4	matplotlib	40%
5	scikit-learn	38%
6	six	31%
7	tensorflow	24%
8	requests	23%
9	python-dateutil	22%
10	pytz	21%

True cost of proprietary software

- Proprietary software can accelerate initial conceptual work
- Transition to working, deployable, scalable, replicable algorithm is more problematic

Specific examples:

- Matlab: GNU Octave provides an alternative, but plotting can be buggy and no easy path to parallelization
- IDL: likewise, GDL is compatible with a lot of IDL code (that uses proper syntax) but plotting is onerous

Both GNU Octave and GDL have active communities, but they are a fraction of the size of Python, Julia and R communities

Agency pathways include

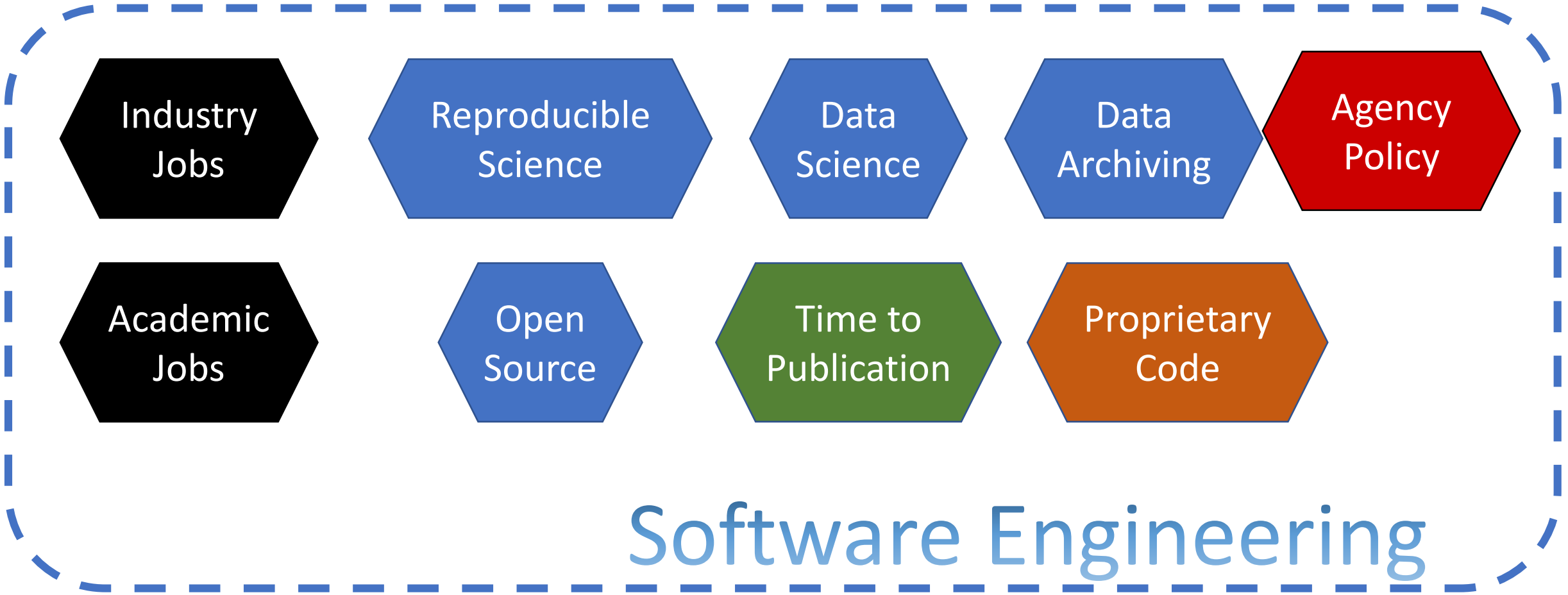
"…software needs to be as open as possible; as closed as necessary."

Open Source Software Policy Options for NASA Earth and Space Sciences, National Academies Press (2018) doi: 10.17226/25217

moving toward FAIR (Findable-Accessible-Interoperable-Reusable) science principles

Open Science by Design: Realizing a Vision for 21st Century Research, National Academies Press (2018) doi: 10.17226/25116

Nexus of Interests



New work: suggest latest version with fallbacks for older versions

Status of Python branches

Branch	Schedule	Status	First release	End-of-life
master	PEP 596	features	*TBD*	*TBD*
3.8	PEP 569	prerelease	*2019-10-21*	*2024-10*
3.7	PEP 537	bugfix	2018-06-27	*2023-06-27*
2.7	PEP 373	bugfix	2010-07-03	*2020-01-01*
3.6	PEP 494	security	2016-12-23	*2021-12-23*
3.5	PEP 478	security	2015-09-13	*2020-09-13*

https://devguide.python.org/#status-of-python-branches

Outline

1. Software Version Control – GitHub Tutorial (13:35-13:45)
2. Continuous Integration: Automated software testing (13:45-14:00)
3. Accessible heliophysics data: lightweight standard HAPI (14:00-14:15)
4. OpenMPI physics model: Fortran 2018 design patterns & Python integration (14:15-14:30)
5. Mobile / Web app dev for crowd-sourced science (14:30-14:45)
6. Science software workflow (14:45-15:00)
7. Intro to concurrent / parallel programming in Python (15:00-15:10)
8. Room discussion (15:10-15:30)