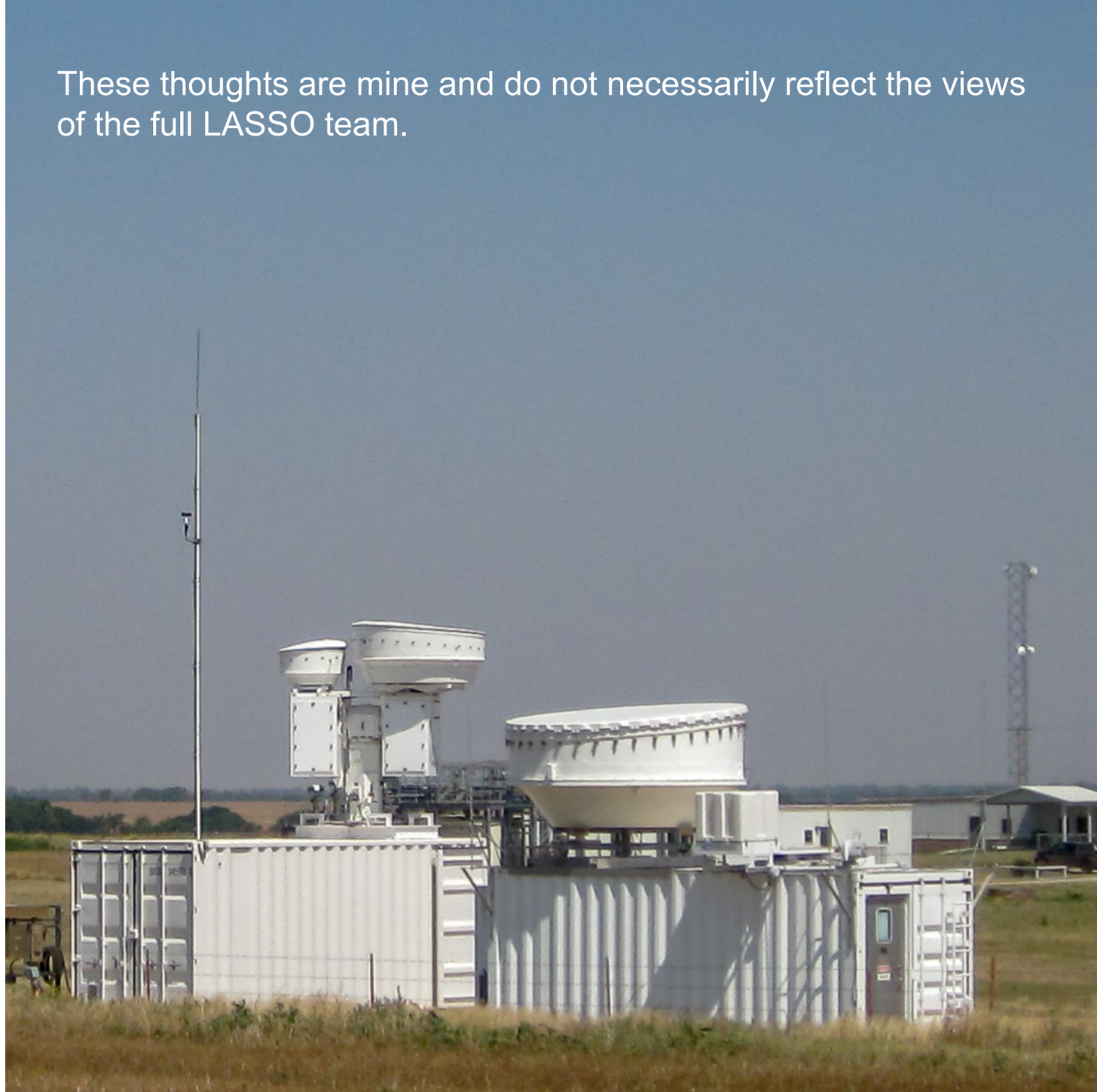
Pacific Northwest
NATIONAL LABORATORY

# Thinking About LASSO and Open Sourcing

William I. Gustafson Jr.

These thoughts are mine and do not necessarily reflect the views of the full LASSO team.

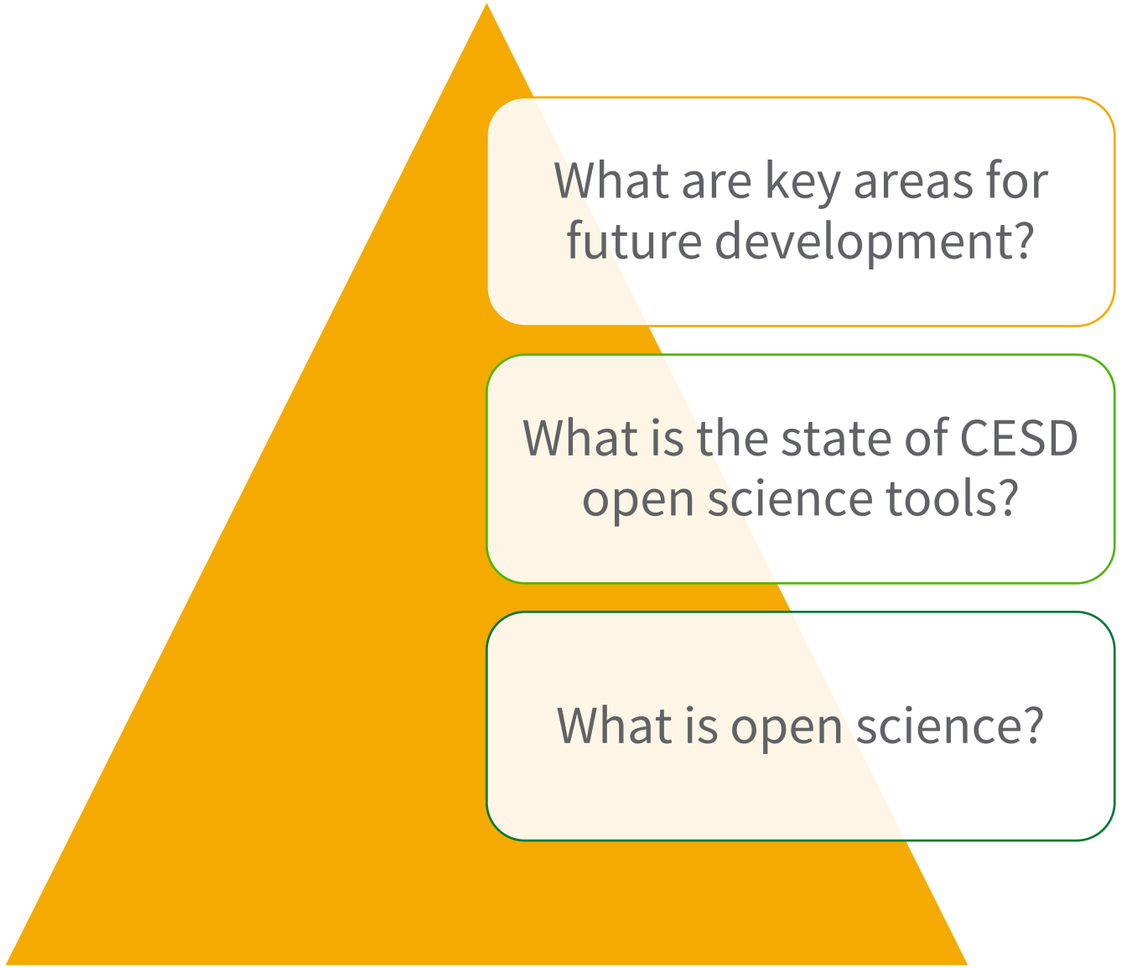


Open Science for Advancing Knowledge Transfer across CESD Breakout Session, 2019 ARM/ASR PI Meeting, 13-Jun-2019

U.S. DEPARTMENT OF ENERGY BATTELLE

PNNL is operated by Battelle for the U.S. Department of Energy

# Organized my thoughts around the session's guiding questions




**Developing LASSO has forced hard choices in the area of code development and sharing**

- Initially envisioned making all code publicly available and able to easily run by anybody
- Resources and practicality trump vision

Pacific Northwest NATIONAL LABORATORY

# How much of ARM's software should be publicly available?

## What motivates making the code open?

1. Ethical & legal responsibilities (what you have to do)
   - Reproducibility
   - Journal requirements

2. Programmatic outcomes (how you can benefit from open sourcing)
   - Increase ARM data usage
   - Save money by getting others to contribute features

3. Altruistic desires (how you can help others by open sourcing)
   - Vision of simplifying research
   - Making code available for educational purposes

Pacific Northwest NATIONAL LABORATORY

# The state of the released code depends heavily on the motivation

## Motivated by ethical and legal reasons

- Releasing the code for documentation & reproducibility reasons does not necessarily mean external users will be able to run the code
- How much of the code needs to be released to meet journal data sharing requirements?
- Does the code have to work on non-ARM computers?

## Motivated by programmatic outcomes and/or altruistic desires

- Do users care if the code gets released?
- How will releasing the code lead to improved outcomes/statistics?
- If users are meant to run the code, do we need to release everything? Or, only the parts that would be meaningfully used?

Pacific Northwest NATIONAL LABORATORY

# What responsibilities come with releasing code?

- If the code is solely released for documentation purposes, then there is no expectation of user support
- Providing tools for others to use implies a need for ongoing support
- Should a lack of a plan for, and/or ability to provide, support lead to a decision not to release the code?

Pacific Northwest NATIONAL LABORATORY

# What is the right balance of making code general vs. efficient for ARM's needs?

- Funding and available time determine much of what gets formally released
- Grand vision for making LASSO fully turnkey and publicly runnable has been reshaped by practicality
- Making LASSO software more efficient to run within ARM essentially has meant making it harder for others to use

Pacific Northwest NATIONAL LABORATORY

# Current LASSO thinking: release code where ARM added value and where it would be scientifically useful to external users

- WRF model would be released, but is 99% already openly available elsewhere
  - We essentially have a LASSO patch that applies on top of WRF, so this is a fork from the main WRF repository
- Code to run the model is somewhat specific to ARM's computing environment and would not be very useful to others, so it would not be released
- Code to compute model statistics vs. observations
  - Will be released, but users may need to pull it apart to be useable for their needs
  - This is where the most effort has gone for automation
  - Chose to use ARM Data Integrator (ADI) software library (almost required for code within ARM), which hampers external usage

Areas for future devel.

Pacific Northwest NATIONAL LABORATORY

# All ARM code should be publicly available

- Assumptions made in the codes can often impact results and, without the code, many of these assumptions are only known by the developer
- Not all code should have the expectation of external runnability. However, this should be as widely sought as practically possible
- Ability to release useful, runnable code for ARM products relies heavily on the ability to integrate with ADI externally
  - What changes are needed for ADI to assist with open sourcing code?
  - How can code be modularized to work around ADI?
  - Can ADI be modularized and reduced to simpler common features necessary for external use?
    - ✓ Workflow tracking, data retrieval, file subsetting, or unit conversion?
- Releasing code ups the need for (and cost of) quality documentation