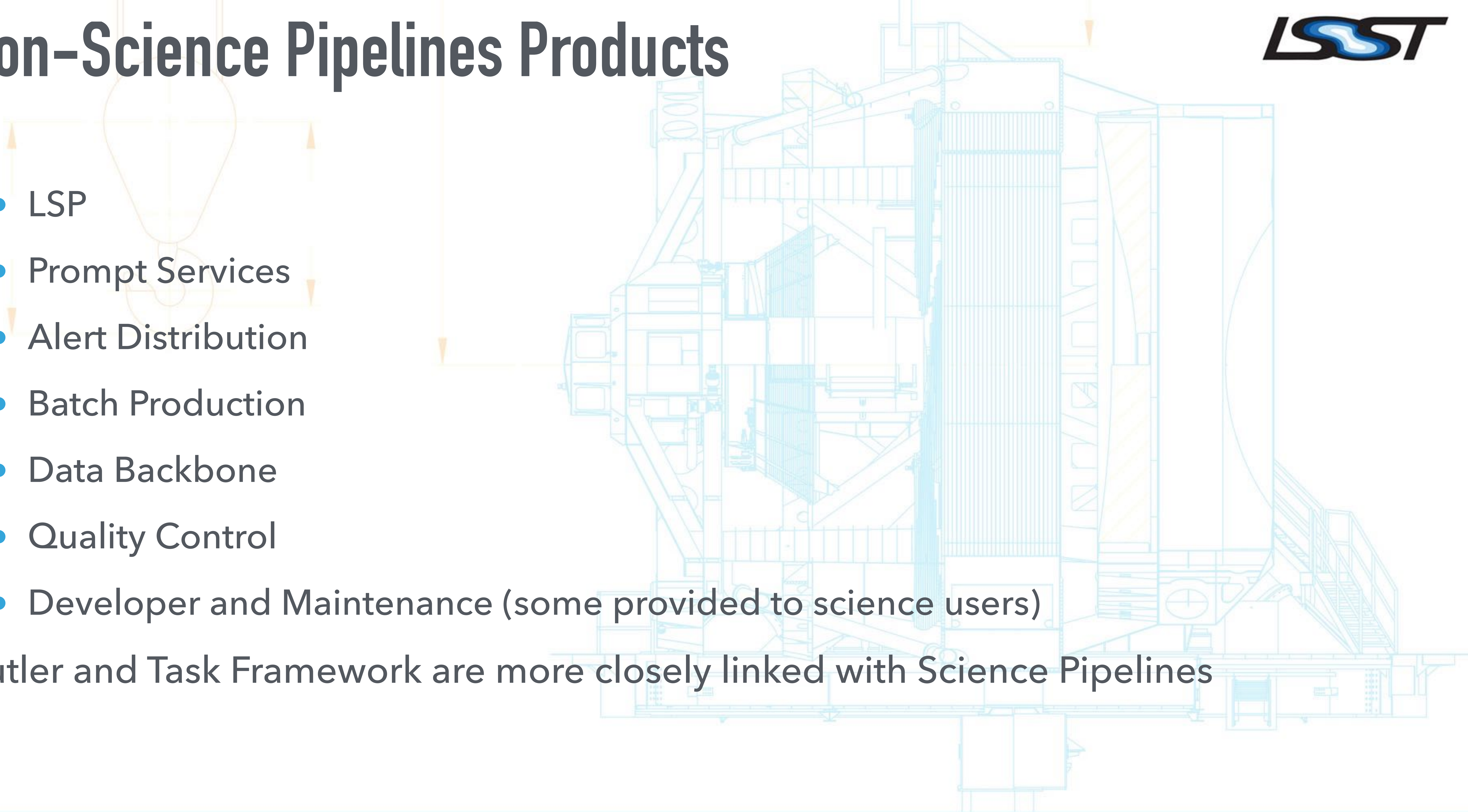
Non-SciPi Processes

Kian-Tat Lim

Large Synoptic Survey Telescope

Non-Science Pipelines Products



- LSP
- Prompt Services
- Alert Distribution
- Batch Production
- Data Backbone
- Quality Control
- Developer and Maintenance (some provided to science users)

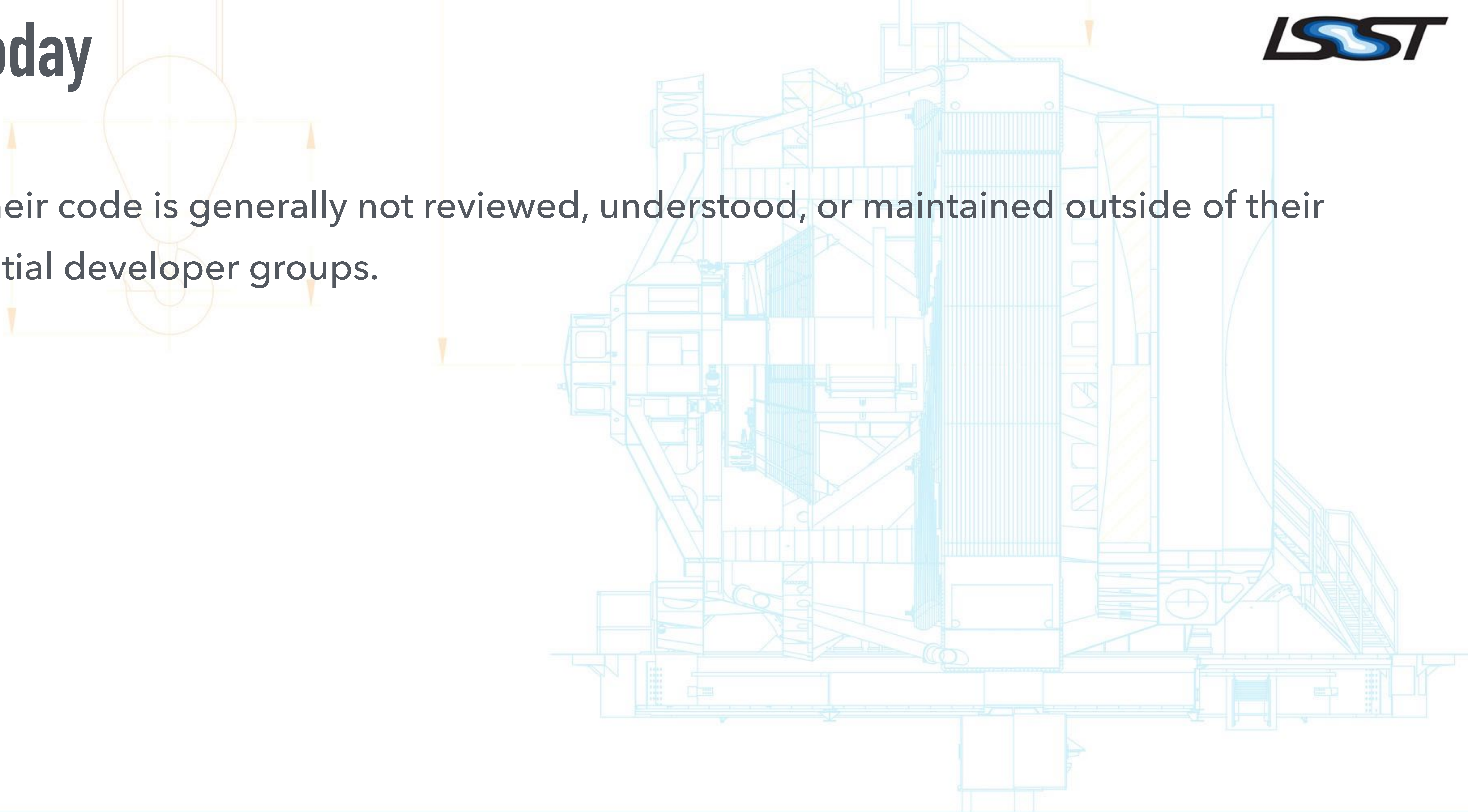
Butler and Task Framework are more closely linked with Science Pipelines

Differences from Science Pipelines

- Languages: Java, Kotlin, JavaScript, SQL, Lua, bash, Ruby, CoffeeScript, Go?
- Dependencies
  - Third-party packages
  - Tooling: code, build, test, deploy
- External drivers
  - Shared with non-LSST projects
- Skillsets
  - Asynchronous, distributed, systems, UX, Web

Today

Their code is generally not reviewed, understood, or maintained outside of their initial developer groups.



Question

What standards and processes are we and should we be applying to these products to ensure that they function well and are maintainable for the lifetime of the project?

Proposed Partial Answer

- LPM-17 (Systems Engineering Management Plan) requires that we maintain certain standards consistent with good (software) engineering practice
- Do not allow operational deployment of code not in LSST GitHub organization
- For long-term operability, must have cross-institution knowledge of the code, with at least one institution being NCSA or Tucson
- Minimize amount of knowledge needed; continue to require RFCs for each language and product
- Common documentation standards
- Common release process principles, even with different tooling and cadences