Reference Architecture for Deep Learning Environments (RADeLE)

Abdul Qayum, Mengqi Zhang, Jim Buckley, Ashish Sai, Simon Colreavy, Muslim Chochlov, Dayi Lin

1 MOTIVATION:

- Deep learning Environments (DLEs) increasingly important/prevalent, with substantial architecture requirements regarding NFRs like:
  - Security
  - Performance
  - Maintainability
  - Operability
- But little explicit literature re DLE architecture exists

GOALS:

- Architecture recovery approaches for DLEs
- Resultant architectural perspectives for individual DLEs
  - TensorFlow
  - Pytorch
  - Mindspore
- A reference architecture across DLEs

3 PROTOTYPE 1: SOCIAL ARCHITECTURE

- Based on TF users frequently visited code locations:

Packages frequently mentioned in developer mailing lists

2 REVIEWS:

Software Architecture Recovery (SAR) Taxonomy findings:

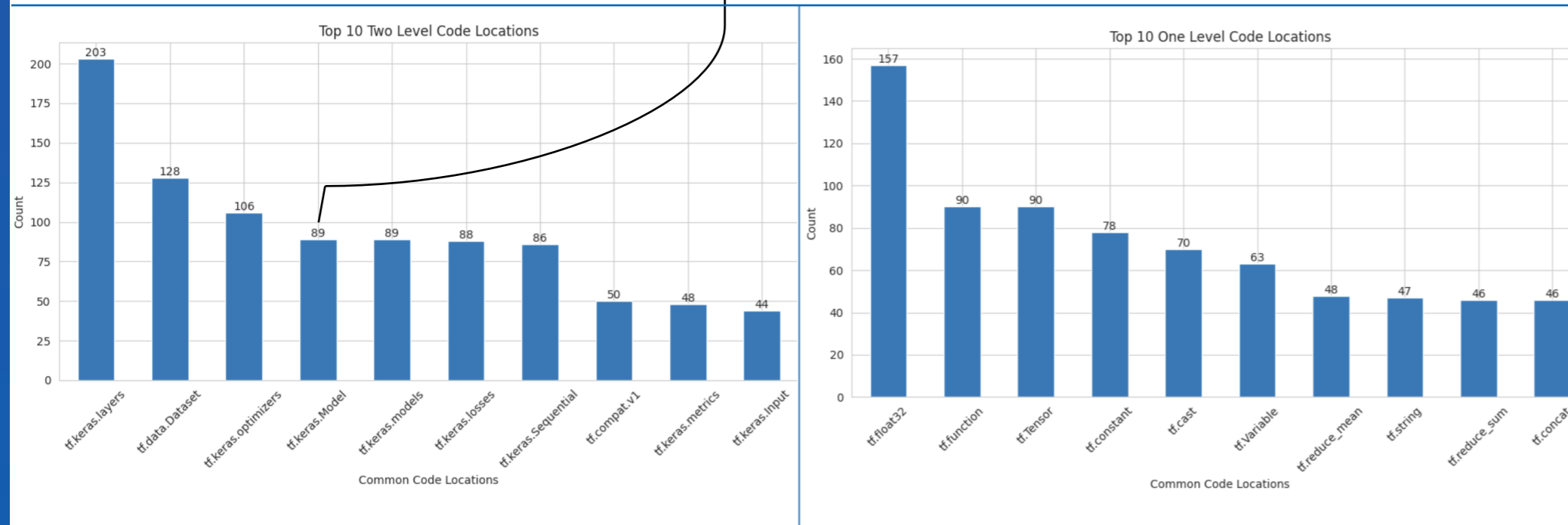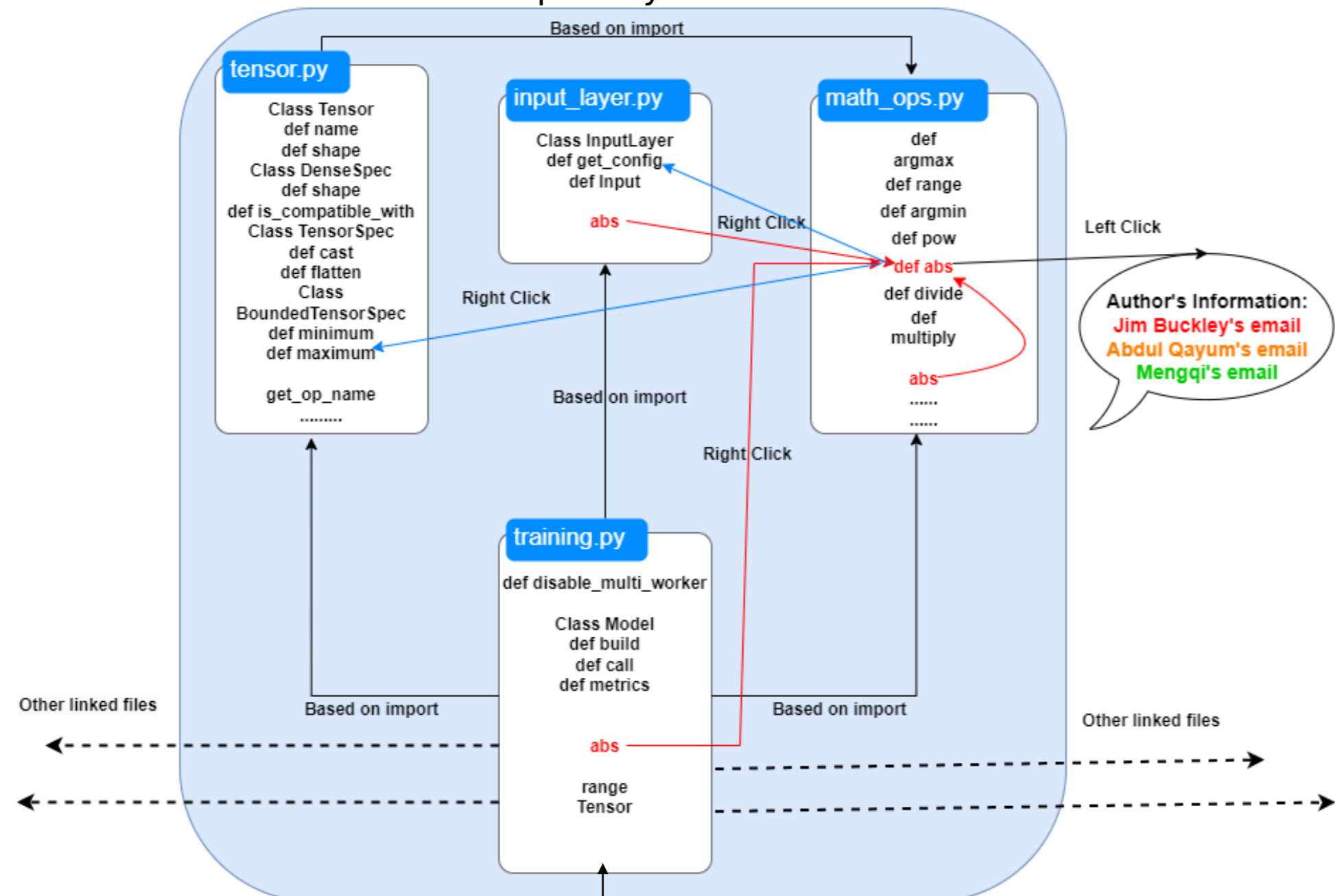Dali [1]   Softwarenaut [2]   ARCADE [3]   Arch [4]

1. Lack of SAR-approach tooling (i.e., most identified SARs aren't open source).
2. Most identified SARs are bottom-up; Top-down and hybrid are seldom seen.
3. 12% of SAR approaches have not been evaluated.
4. Most identified SARs are evaluated using case studies and field studies, which means less generalizable empirical evaluation.
5. The number of systems SARs are evaluated against is small.
6. Expert opinion is seldom used in empirical studies.

Architecture insights to API systems:

1. API systems have emphasised certain Non-Functional-Requirements(NFRs), such as operability, portability, over other NFRs such as security.

2. API systems have some common architecture patterns: Layered, Vertical, Sectioned. However, these patterns are not well exposed in the existing literature.

3. Those existing patterns seem to have different foci regarding NFRs, as judged by association.
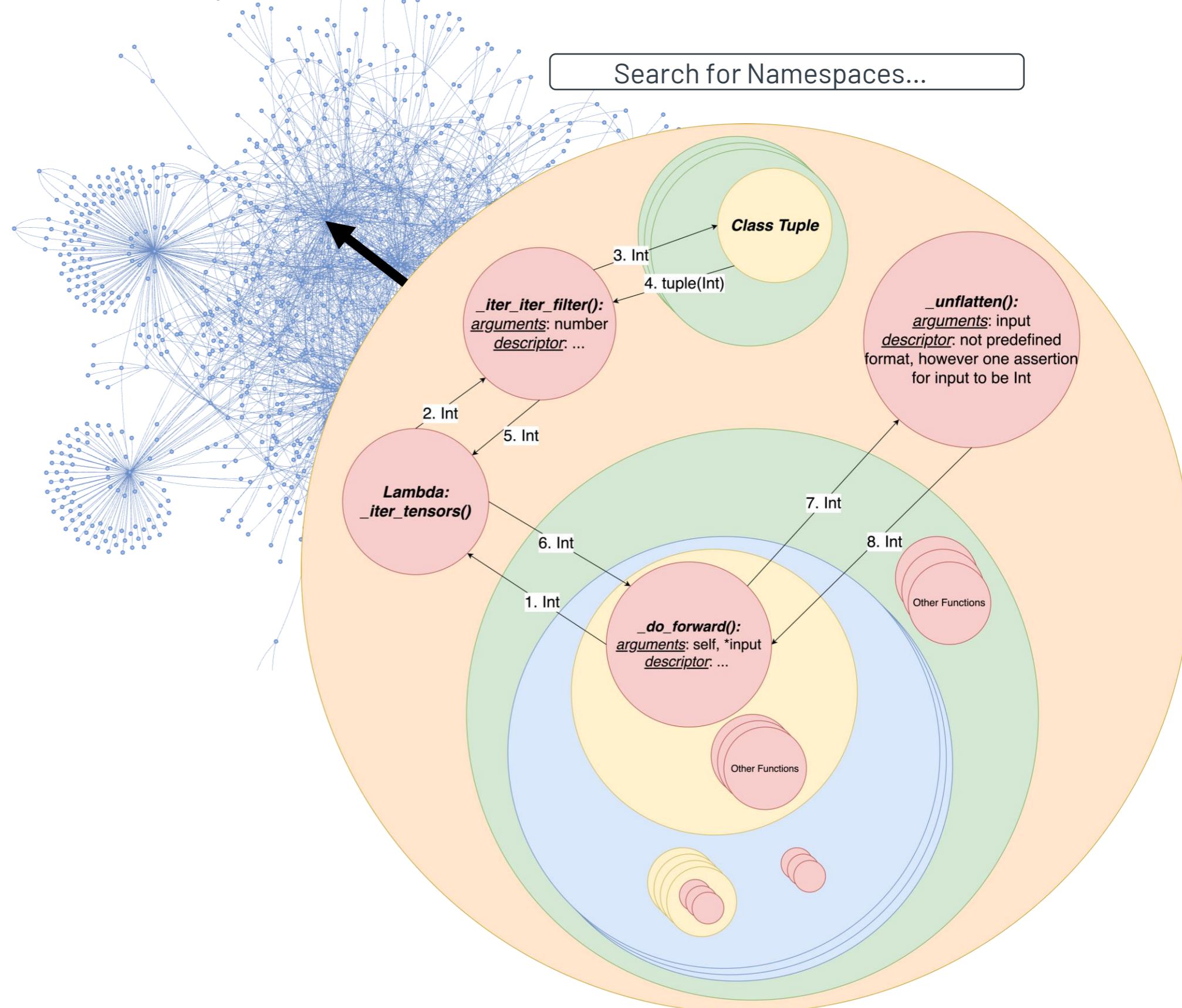
4 PROTOTYPE 2: DATAFLOW ARCHITECTURE

- Based on PyTorch community members frequently discussed data-type errors in GitHub