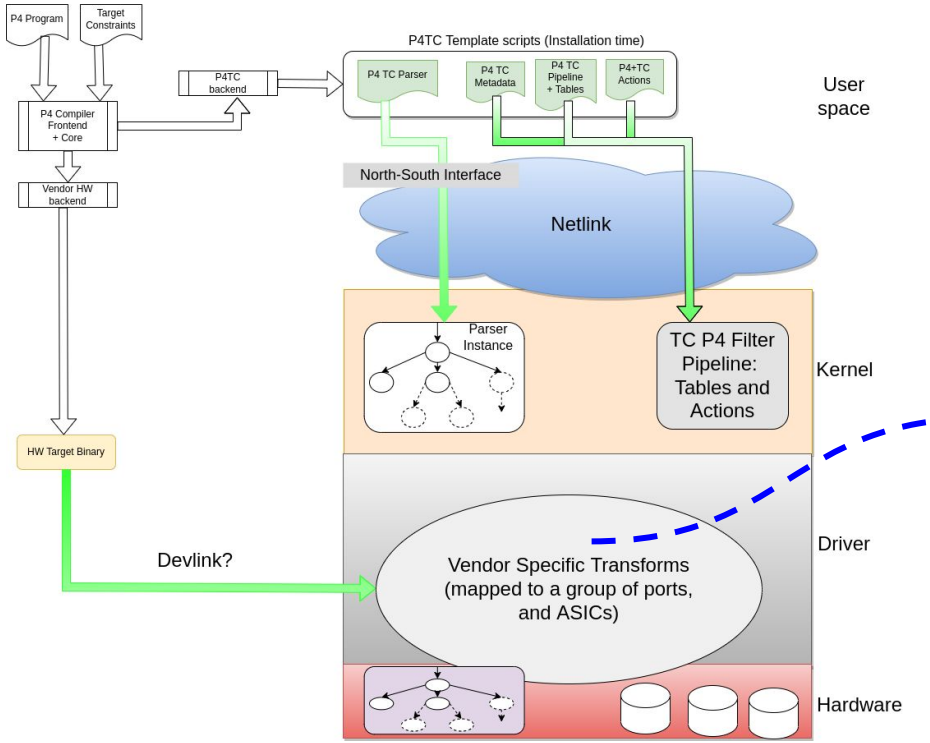
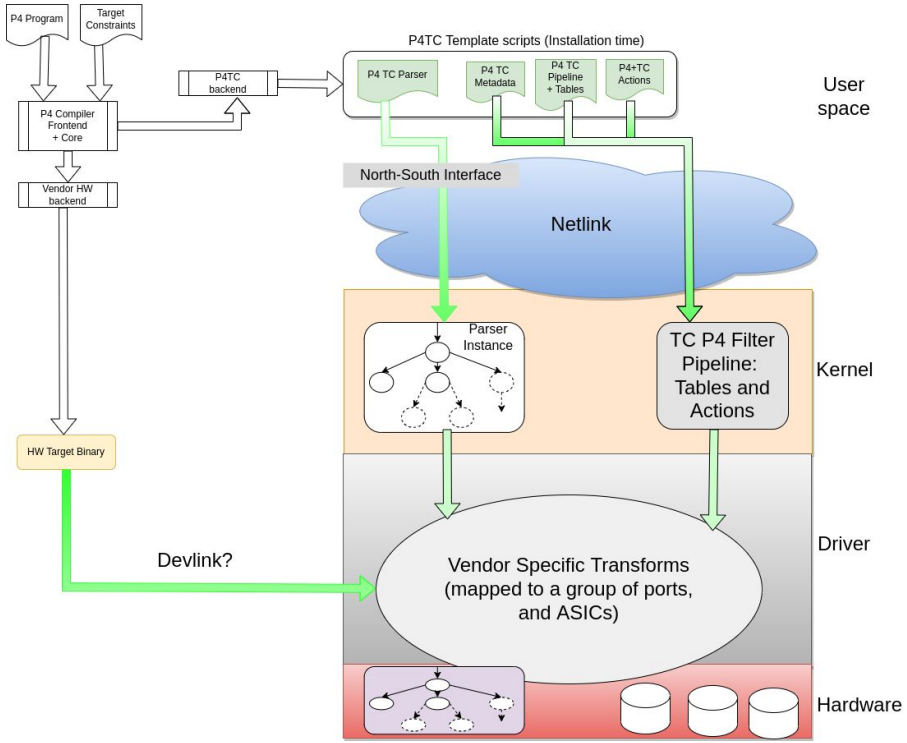
P4TC Potential Program Installs

Model 1: Separate Loading



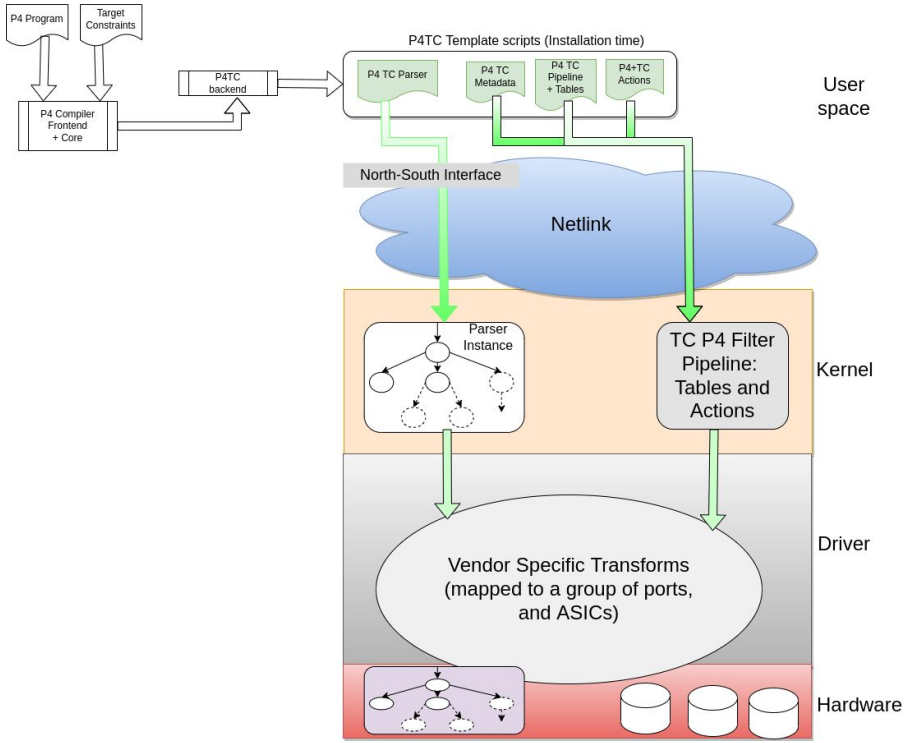
- HW-equivalent loaded via sideband
  - Package includes one or more of
    - Binary
    - Extra configuration for vendor
    - Anjali: We could standardize how the configuration looks like
      - Json file?
- SW-equivalent install time verifies that the two programs are the same
- Vendor driver is responsible for "*transforming*" P4 abstractions
  - Table reordering, merge, sort, etc
  - LPM, TCAM, SRAM etc
  - P4 pipeline to hardware pipeline mapping
    - RMT vs DRMT
    - Mesh processing/cross-bar

Model 2: Hybrid Loading



- Parts of HW-equivalent loaded via direct system calls in conjunction with SW-equivalent
- Parts loaded on sideband
- Driver handles P4 transforms

Model 3: Joint Loading



- Both HW-equivalent and SW-equivalent loaded via direct system calls
  - Assumes all objects are <u>provisionable</u> in hw
    - Consensus so far says: This is IMpossible
- Driver transforms P4 abstraction

Issues/Challenges

- Driver Transformation complexity
  - Mapping one to one on P4 table pipeline vs merging/splitting tables in HW
    - We allow both models - up to vendor
- Verification of h/w vs s/w program version
- For hybrid model - we need use cases. Matty and Co?
- Should we standardize the “target constraints” definition that is fed into the compiler?
- Should we allow parser-value-sets?
  - P4 Language SPEC 1.2.1, "12.11. Parser Value Sets".
    - https://p4.org/p4-spec/docs/P4-16-v1.2.0.html#sec-value-set