Fusion Data Framework structure:
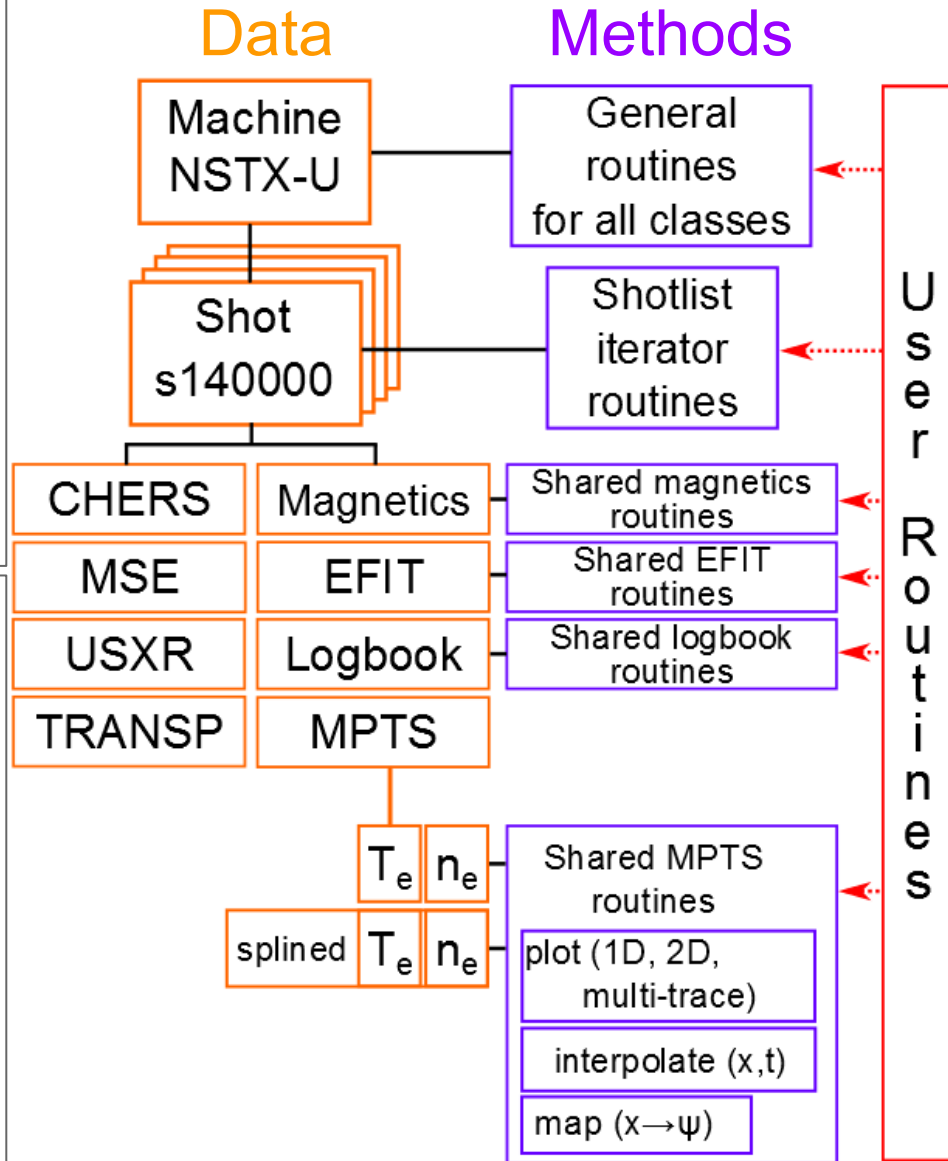
(A new approach to working with data)

J. Schmitt, D. Smith, K. Tritz, H. Yuh

FDF integrates data with routines via signal classes

A unified MDS+ to code data hierarchy. FDF structure is initialized, but data retrieved just in time. Signals in FDF referenceable prior to retrieval

Groups of signals grouped into different python classes, allowing namespace differentiation (e. g. mpts.plot can be different than efit.plot)

Data

Methods

Machine NSTX-U

Shot s140000

CHERS | Magnetics
MSE | EFIT
USXR | Logbook
TRANSP | MPTS

T_e | n_e
splined T_e | n_e

General routines for all classes

Shotlist iterator routines

Shared magnetics routines
Shared EFIT routines
Shared logbook routines

Shared MPTS routines
plot (1D, 2D, multi-trace)
interpolate (x,t)
map (x→ψ)

User Routines

Methods can work at machine, shot, signal group, or signal level
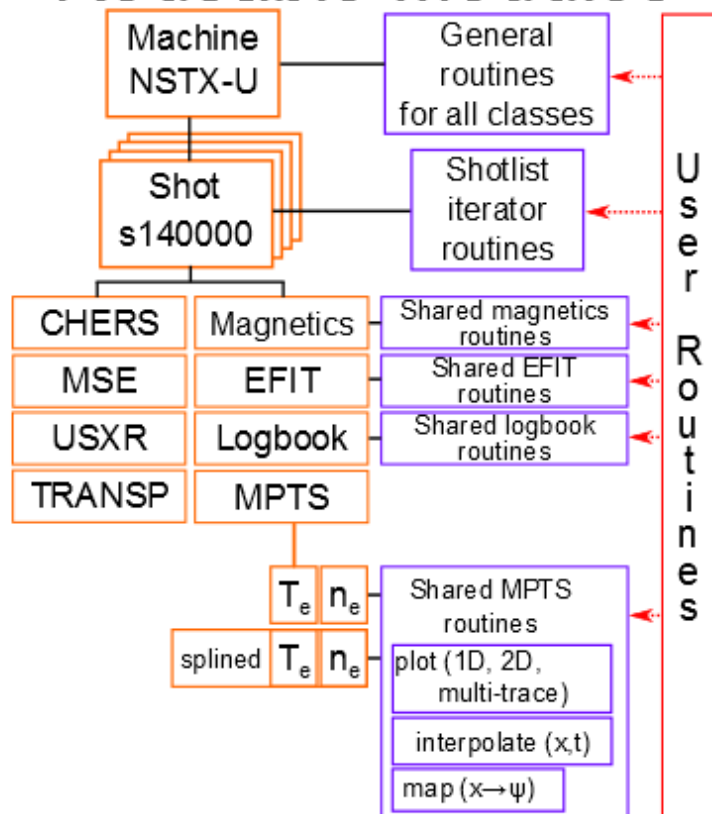
Users can modify by modifying and adding methods in their own FDF setup. This can be done for single session or made persistent

Useful modifications and additions have an established merge path into default distro

Data+method objects create reusable modules

1. I/O, analysis, plotting/visualization, databasing, archiving become reusable modular components

2. Analysis codes input FDF signals AND return FDF signals, methods stay attached to signals

3. As the library grows, physics users should increasingly be coding ONLY the analysis part

4. App development becomes a much less monolithic task, and more like combining modules