Sandia National Laboratories

Exceptional service in the national interest

WE MADE 40 TRILINOS SPACK PACKAGES?

Joe Frye

Sandia National Labs

TRILINOS

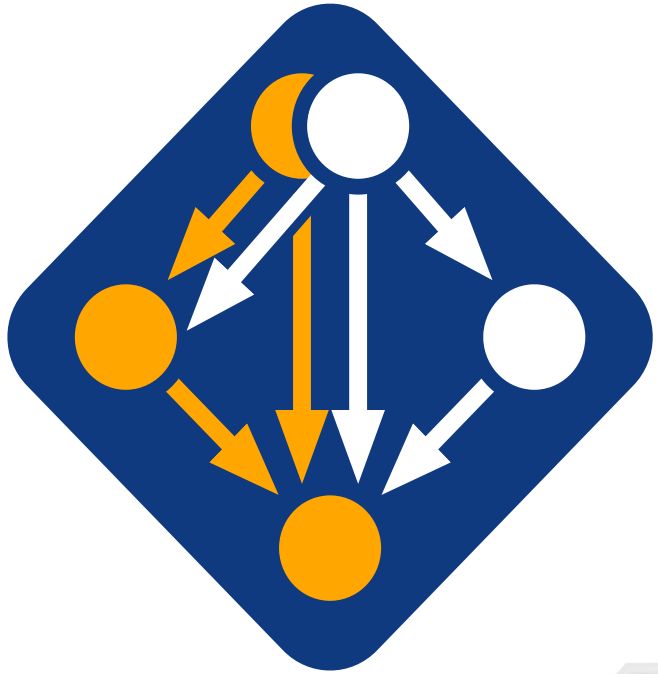
Spack

HPSF TUG March 19, 2026

SAND2026-18295C

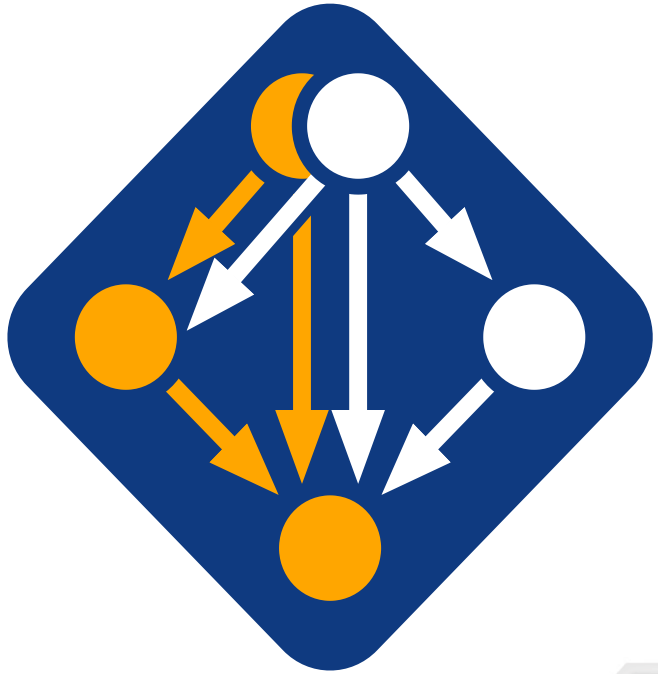
WHY INVEST IN SPACK?

- Spack can handle the dependencies between Trilinos packages
- Spack can build all the Trilinos TPL dependencies (Trilinos AT2 containers, SEMS, and AUE are already using Spack for Trilinos' TPL dependencies)
- Internal ASC codes are using Spack for their processes and want better integration with Trilinos
- Projects are maintaining their own Trilinos Spack packages
- Spack is the most common package manager for scientific software/HPC
- New/interested users likely have familiarity with Spack



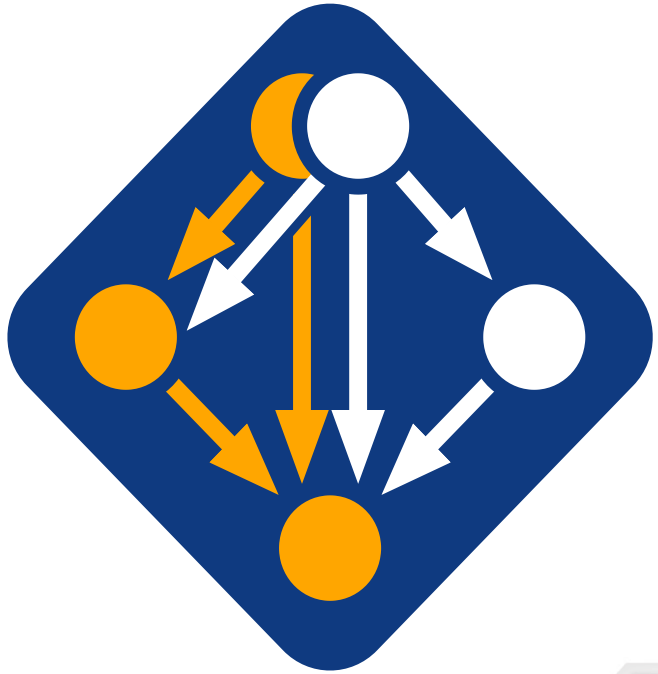
COMMON CONCERNS

- Will this break the current trilinos spack package?
  - No I do not have plans to replace the current trilinos spack package. This will be available alongside that package. Hopefully this will be easier, and projects will want to switch
- Where is this work and when can I use it?
  - The repository is with the trilinos group on github
  - https://github.com/trilinos/trilinos-spack-packages.git
  - Some of the packages are not concretizing correctly, but most are



HOW TO TRY IT OUT

- Will this break the current trilinos spack package?
  - No I do not have plans to replace the current trilinos spack package. This will be available alongside that package. Hopefully this will be easier, and projects will want to switch
- Where is this work and when can I use it?
  - The repository is with the trilinos group on github
  - https://github.com/trilinos/trilinos-spack-packages.git
  - Some of the packages are not concretizing correctly, but most are
  - See README.md on the repo for setup instructions

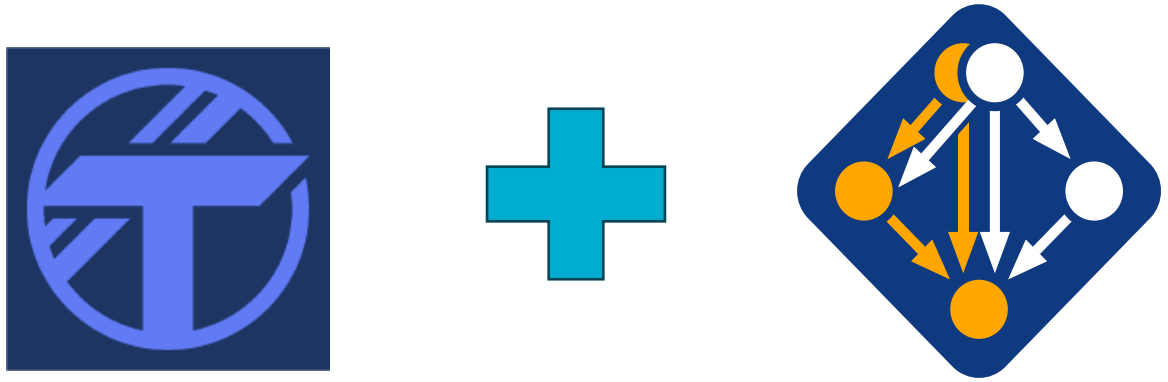


BUILDS TPLS

```
$ spack spec trilinos-teuchos

-   trilinos-teuchos@jfrye-spack-changes~cuda~ipo+kokkos+kokkoscomm+kokkoscompat+mpi~rocm~tests
build_system=cmake build_type=Release commit=70176b5e3337cff8bc7bb8d75dec384ca0da5ecb

[+]      ^cmake@3.31.9~doc+ncurses+ownlibs~qtgui build_system=generic build_type=Release …

         ...

[+]      ^gmake@4.4.1~guile build_system=generic platform=linux os=rhel8 target=skylake_avx512 %c=gcc@8.5.0

         ...

[+]      ^kokkos@4.6.02~aggressive_vectorization~cmake_lang~compiler_warnings+complex_ ...

         ...

[+]      ^openmpi@5.0.8+atomics~cuda~debug+fortran~gpfs~internal-hwloc~internal-libevent  ...
```

PARAMETERS

- Each Trilinos package is it's own package
  - Use the Tribits option "-DTPL_ENABLE_<PackageName>" and <PackageName>Config.cmake to allow packages to be built independently
- Minimized rework for Spack
- We want the packages to maintain the Trilinos "branding" / "association"
- Let package options live with the packages and Trilinos wide options live in a base class
- Easy for new users to build the Trilinos package(s) they are interested
- Minimize duplication of Tribits logic implemented in Spack (generate as much as we can)

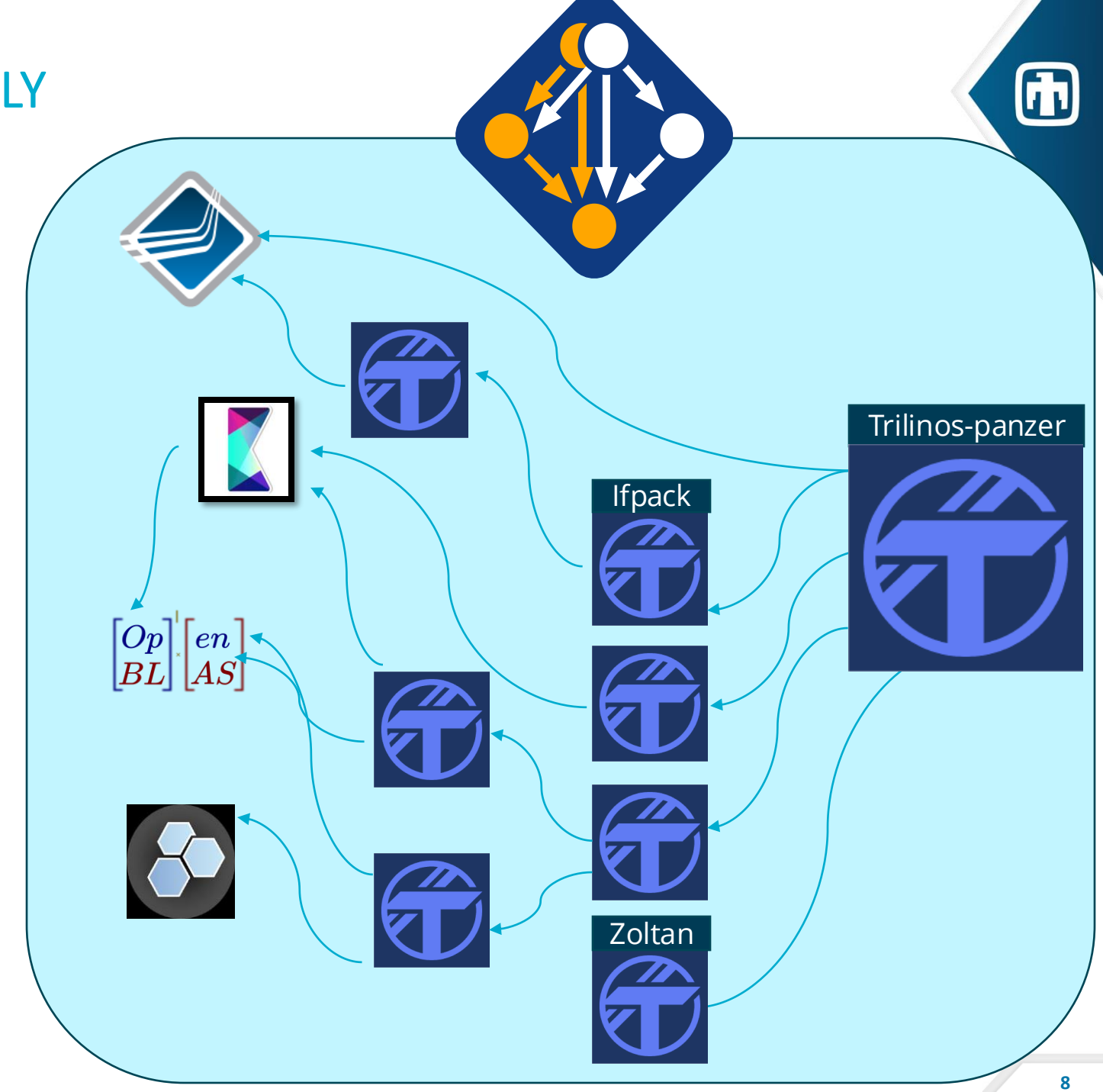


COMMON FORMAT FOR PROJECTS USING TRILINOS

- Projects that depend on Trilinos build in different ways
- Some are maintaining their own Spack package
- Spack can help to standardize how Trilinos configurations are specified between projects
- Testing the Spack packages that downstream projects are using

BUILD EACH PACKAGE INDIVIDUALLY

- Each package is its own Trilinos build with only one package enabled and all dependencies are expected as TPLs
- Allows Spack to handle dependencies
- Spack will build only one copy of each Trilinos package and one version of TPLs by default
- Not faster than building everything in one Tribits build
- Allows rebuilds of individual packages (may be faster for certain workflows)
- Requires some duplication of Tribits logic in the Spack packages



KEEPING TRIBITS AND SPACK IN SYNC

- Minimize duplication of Tribits logic implemented in Spack (generate as much as we can)
  - We (I) don't want to be in the business of monitoring dependency information in cmake files and then implementing that in Spack package files
- Inter-package dependency information that is handled by Tribits needs to be reproduce in Spack
- This has a risk of becoming out of sync and is hard to manage without automation
- Will also require implementing workflows to keep the Spack package current
- We have testing of our Spack package files, but need to automate

KEEPING TRIBITS AND SPACK IN SYNC

- What information does Spack need?
  - Spack is good at managing complex dependency graphs
  - Spack has the idea of 'variants' which are ways to modify the build (map to some set of configure options)
- What information can be easily generated for Spack to use?
  - Tribits will makes an xml file with all the package and TPL dependencies for each package
  - Easy to express as dependencies or variants in Spack
- Can we assume reasonable defaults in our Spack packages to reduce the amount of duplicated logic?
  - Remember this? "Easy for new users to build the Trilinos package(s) they are interested"
  - Set defaults to build as much as possible so that it "just works" (at the cost of build time and size)



<Package name="MueLu" dir="packages/ifpack2" type="PT">	**spack_repo/trilinos/packages/trilinos_mue_lu/package.py**
<LIB_REQUIRED_DEP_PACKAGES value="Teuchos, ... ">	cmake_args.append('-DTRILINOS_ENABLE_Teuchos=ON')
<LIB_OPTIONAL_DEP_PACKAGES value="Thyra, ... ">	variant('thyra', default=True, description='Enable thyra') Depends_on_trilinos_package('trilinos-thyra') cmake_args.append(self.define_from_variant('TPL_ENABLE_Thyra'), 'thyra')
<LIB_REQUIRED_DEP_TPLS value="..., BLAS, ... ">	depends_on(blas)
<LIB_OPTIONAL_DEP_TPLS value="..., Boost, ... ">	variant('boost', default=True, description='Enable TPL boost') depends_on('boost', when='+boost') cmake_args.append(self.define_from_variant('TRILINOS_ENABLE_Boost'), 'boost')

TRILINOS BRANDING/ASSOCIATION

- All packages are prefixed with "trilinos-" to maintain an obvious connection to Trilinos for users
- "spack install trilinos-teuchos"
- Package description is inherited from a base class which is the same for all the Trilinos packages
- The package.py files (except the base class) are generated using the python script so they have a consistent look/feel/style

```
24 Jul 24 20:19 trilinos_adelus
24 Jul 24 20:19 trilinos_amesos2
24 Jul 24 20:19 trilinos_anasazi
40 Jul 24 20:19 trilinos_base_class
24 Jul 24 20:19 trilinos_belos
24 Jul 24 20:19 trilinos_compadre
24 Jul 24 20:19 trilinos_galeri
24 Jul 24 20:19 trilinos_gtest
24 Jul 25 19:47 trilinos_ifpack2
24 Jul 24 20:19 trilinos_intrepid2
43 Jul 24 20:19 trilinos_krino
24 Jul 24 20:19 trilinos_mini_tensor
24 Jul 24 20:19 trilinos_muelu
24 Jul 24 20:19 trilinos_nox
24 Jul 24 20:19 trilinos_pamgen
24 Jul 24 20:19 trilinos_panzer
43 Jul 24 20:19 trilinos_percept
24 Jul 24 20:19 trilinos_phalanx
43 Jul 24 20:19 trilinos_piro
24 Jul 24 20:19 trilinos_pytrilinos2
24 Jul 24 20:19 trilinos_rol
24 Jul 24 20:19 trilinos_rtop
24 Jul 24 20:19 trilinos_sacado
24 Jul 24 20:19 trilinos_shards
24 Jul 24 20:19 trilinos_shylu
24 Jul 24 20:19 trilinos_stk
24 Jul 24 20:19 trilinos_stokhos
24 Jul 24 20:19 trilinos_stratimikos
```

PACKAGE VARIANTS LIVE WITH THEIR PACKAGES

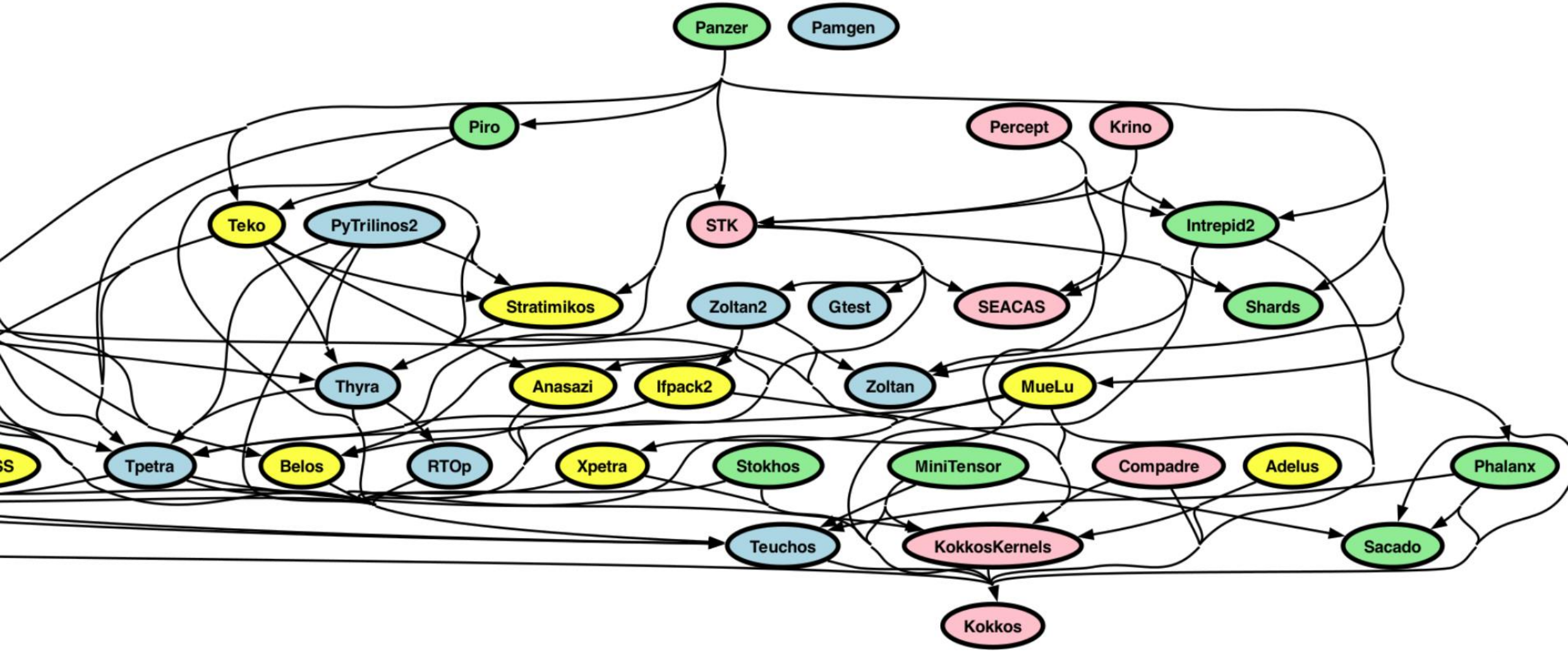
- Each package has variants that correspond to:
  - Optional TPLs
  - Optional subpackages
  - Optional Trilinos package dependencies
- Each package file also contains the cmake defaults for:
  - Required TPLs
  - Required subpackages
  - Required Trilinos package dependencies
- Each package only has variants relevant tot that package and a few from the base class

```
Variants:
    cuda [false]                false, true
        Build with CUDA

    cuda_arch [none]            none, 10, 100, 100a, 100f,
101, 101a, 101f, 103, 103a, 103f, 11, 12, 120, 120a, 120f,
121, 121a, 121f, 13, 20, 21, 30, 32, 35, 37, 50, 52,
                                53, 60, 61, 62, 70, 72, 75,
80, 86, 87, 89, 90, 90a
      when +cuda
        CUDA architecture

    cxxstd [17]                 17, 20
        C++ standard to use when building
    generator [make]            none
      when  build_system=cmake
        the build system generator to use
    ipo [false]                 false, true
      when  build_system=cmake
        CMake interprocedural optimization
    kokkos [true]               false, true
        Enable TPL Kokkos
      kokkoscomm [true]          false, true
        Enable TeuchosKokkosComm
    kokkoscompat [true]         false, true
        Enable TeuchosKokkosCompat
    mpi [true]                  false, true

        Enable TPL MPI
```

SOME VARIANTS NEED TO BE THE SAME BETWEEN PACKAGES

- Base class that defines and enforces synchronization on configuration that needs to be consistent for all packages
  - Kokkos version
  - ETI settings
  - CXX standard
- Base Class inherits from cuda, rocm, and cmake packages to add some support for those
- Individual packages inherit from the base class and define package specific configuration
  - <PackageName>_ENABLE_SomeOption

```
Variants:
    cuda [false]                false, true
        Build with CUDA

    cuda_arch [none]            none, 10, 100, 100a, 100f,
101, 101a, 101f, 103, 103a, 103f, 11, 12, 120, 120a, 120f,
121, 121a, 121f, 13, 20, 21, 30, 32, 35, 37, 50, 52,
                                53, 60, 61, 62, 70, 72, 75,
80, 86, 87, 89, 90, 90a
      when +cuda
        CUDA architecture

    cxxstd [17]                 17, 20
        C++ standard to use when building
    generator [make]            none
      when  build_system=cmake
        the build system generator to use
    ipo [false]                 false, true
      when  build_system=cmake
        CMake interprocedural optimization
    kokkos [true]               false, true
        Enable TPL Kokkos
      kokkoscomm [true]          false, true
        Enable TeuchosKokkosComm
     kokkoscompat [true]        false, true
        Enable TeuchosKokkosCompat
     mpi [true]                 false, true

        Enable TPL MPI
```


Panzer
Pamgen
Piro
Percept
Krino
Teko
PyTrilinos2
STK
Intrepid2
Stratimikos
Zoltan2
Gtest
SEACAS
Shards
Thyra
Anasazi
Ifpack2
Zoltan
MueLu
Tpetra
Belos
RTOp
Xpetra
Stokhos
MiniTensor
Compadre
Adelus
Phalanx
Teuchos
KokkosKernels
Sacado
Kokkos

```
[+]  trilinos-tpetra@jfrye-spack-changes~all-optional-packages~cuda+explicit-instantiation~fortran
ca0da5ecb cxxstd=17 generator=make arch=linux-rhel8-cascadelake %c,cxx=gcc@11.2.0
[+]      ^cmake@3.31.6~doc+ncurses+ownlibs~qtgui build_system=generic build_type=Release arch=linu
[+]          ^curl@8.11.1~gssapi~ldap~libidn2~librtmp~libssh~libssh2+nghttp2 build_system=autotool
[+]              ^nghttp2@1.65.0 build_system=autotools arch=linux-rhel8-cascadelake %c,cxx=gcc@11
[+]                  ^diffutils@3.10 build_system=autotools arch=linux-rhel8-cascadelake %c=gcc@11
[+]              ^openssl@3.4.1~docs+shared build_system=generic certs=mozilla arch=linux-rhel8-ca
[+]                  ^ca-certificates-mozilla@2025-02-25 build_system=generic arch=linux-rhel8-cas
[+]          ^ncurses@6.5~symlinks+termlib abi=none build_system=autotools patches:=7a351bc arch=l
[+]          ^zlib-ng@2.2.4+compat+new_strategies+opt+pic+shared build_system=autotools arch=linux
[+]      ^compiler-wrapper@1.0 build_system=generic arch=linux-rhel8-cascadelake
[e]      ^gcc@11.2.0~binutils+bootstrap~graphite~nvptx~piclibs~profiled~strip build_system=autotoo
[+]      ^gcc-runtime@11.2.0 build_system=generic arch=linux-rhel8-cascadelake
[e]      ^glibc@2.28 build_system=autotools arch=linux-rhel8-cascadelake
[+]      ^gmake@4.4.1~guile build_system=generic arch=linux-rhel8-cascadelake %c=gcc@11.2.0
[+]      ^kokkos@4.6.02~aggressive_vectorization~cmake_lang~compiler_warnings+complex_align~cuda~c
x_async_dispatch~hwloc~ipo~memkind~numactl~openmp~openmptarget~pic~rocm+serial+shared~sycl~tests~t
inux-rhel8-cascadelake %c,cxx=gcc@11.2.0
[+]      ^kokkos-kernels@4.6.02~blas~cblas~cublas~cuda~cusolver~cusparse~execspace_cuda~execspace_
p~rocblas~rocsolver~rocsparse~serial+shared~superlu~threads build_system=cmake build_type=Release
=gcc@11.2.0
[+]      ^openblas@0.3.29~bignuma~consistent_fpcsr+dynamic_dispatch+fortran~ilp64+locking+pic+shar
[+]      ^openmpi@5.0.7+atomics~cuda~debug~gpfs~internal-hwloc~internal-libevent~internal-pmix~ipv
 fabrics:=none patches:=38529b5 romio-filesystem:=none schedulers:=none arch=linux-rhel8-cascadela
```