E4S: The Extreme-scale Scientific Software Stack for Collaborative Open Source Software

Michael A. Heroux, Sandia National Laboratories
Director of Software Technology, US Exascale Computing Project

The 2nd E4S Forum
September 24, 2020

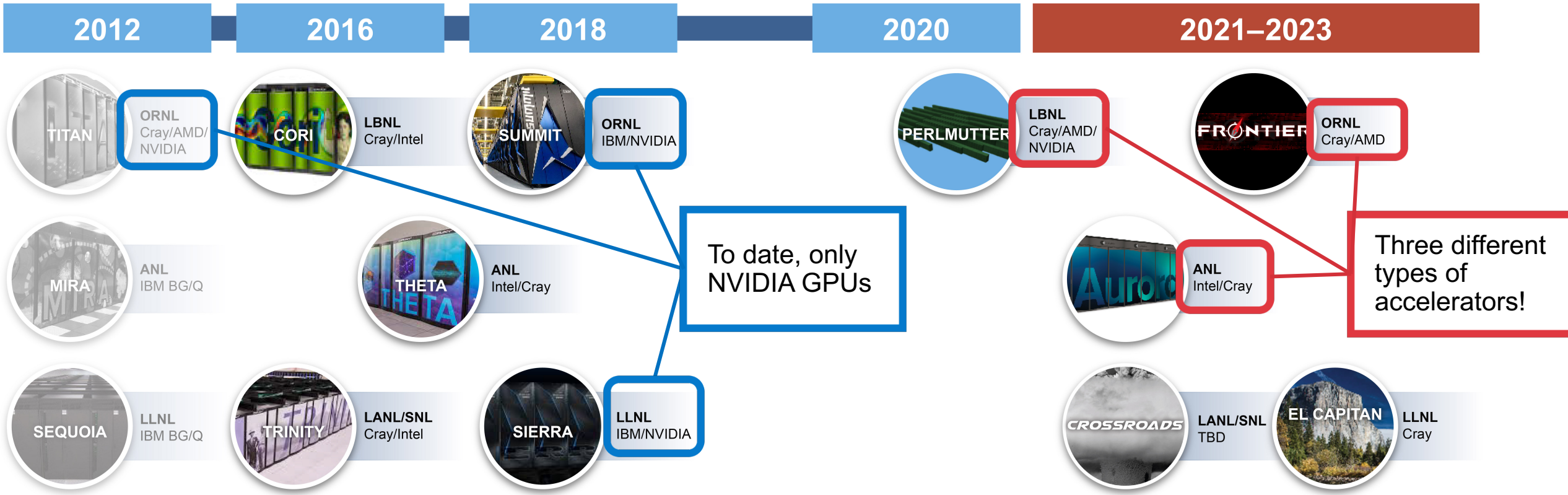
NNSA National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY | Office of Science

Department of Energy (DOE) Roadmap to Exascale Systems

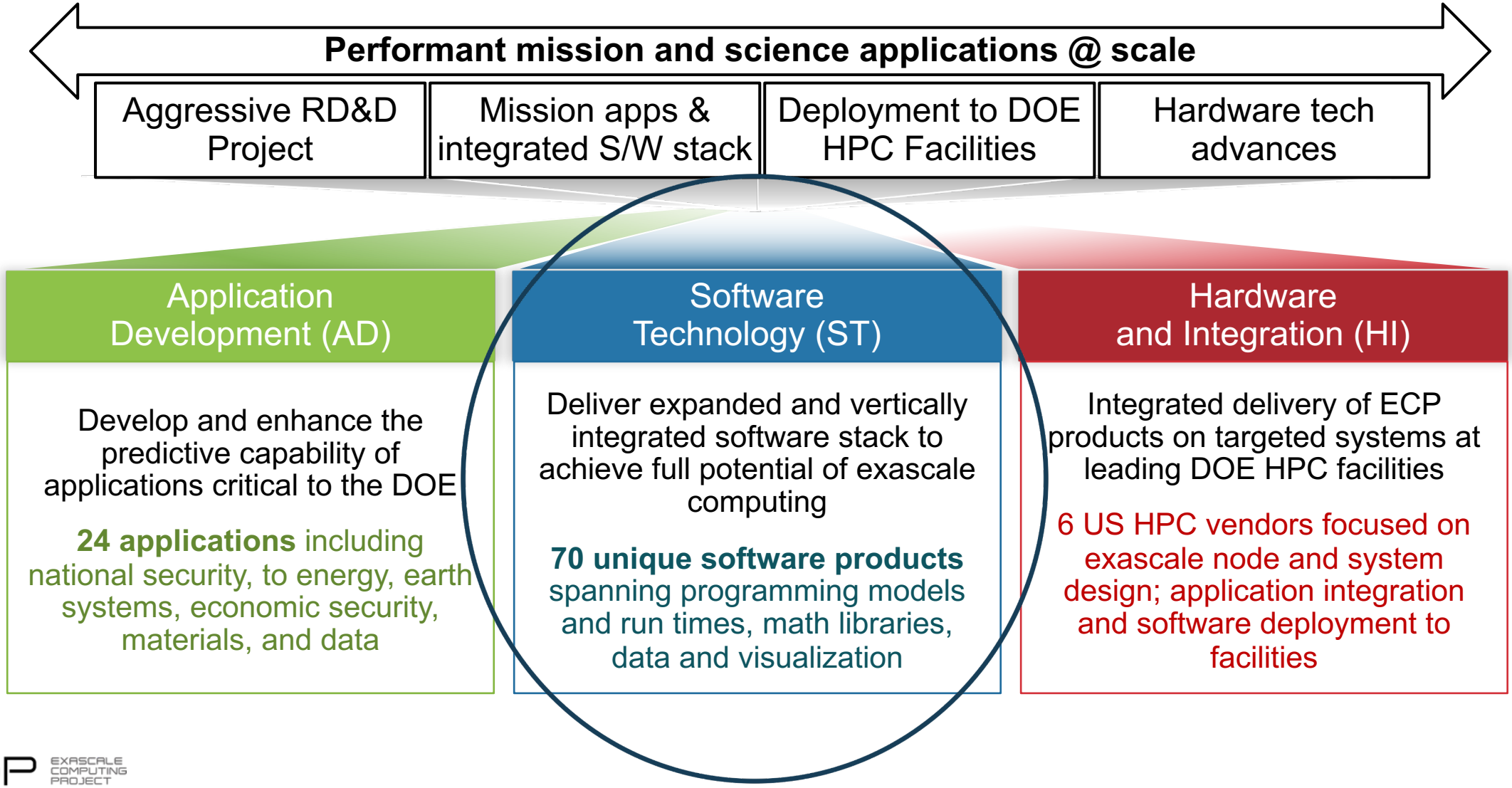
An impressive, productive lineup of *accelerated node* systems supporting DOE's mission

Pre-Exascale Systems

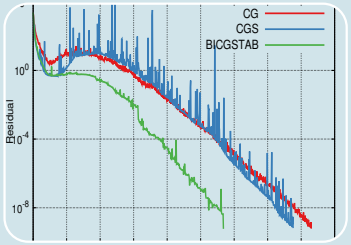
Future Exascale Systems

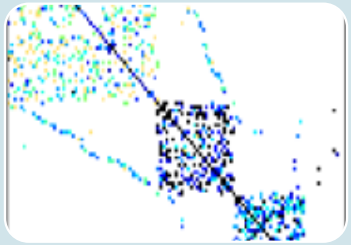


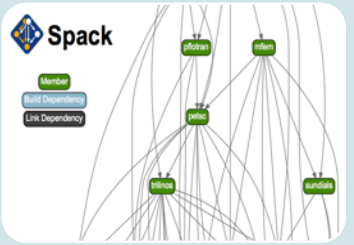
ECP Software Technology (ST) is one of three focus areas



ECP ST has six technical areas







Programming Models & Runtimes

- Enhance and get ready for exascale the widely used MPI and OpenMP programming models (hybrid programming models, deep memory copies)
- Development of performance portability tools (e.g. Kokkos and Raja)
- Support alternate models for potential benefits and risk mitigation: PGAS (UPC++/GASNet) ,task-based models (Legion, PaRSEC)
- Libraries for deep memory hierarchy and power management

Development Tools

- Continued, multifaceted capabilities in portable, open-source LLVM compiler ecosystem to support expected ECP architectures, including support for F18
- Performance analysis tools that accommodate new architectures, programming models, e.g., PAPI, Tau

Math Libraries

- Linear algebra, iterative linear solvers, direct linear solvers, integrators and nonlinear solvers, optimization, FFTs, etc
- Performance on new node architectures; extreme strong scalability
- Advanced algorithms for multi-physics, multiscale simulation and outer-loop analysis
- Increasing quality, interoperability, complementarity of math libraries

Data and Visualization

- I/O via the HDF5 API
- Insightful, memory-efficient in-situ visualization and analysis – Data reduction via scientific data compression
- Checkpoint restart

Software Ecosystem

- Develop features in Spack necessary to support all ST products in E4S, and the AD projects that adopt it
- Development of Spack stacks for reproducible turnkey deployment of large collections of software
- Optimization and interoperability of containers on HPC systems
- Regular E4S releases of the ST software stack and SDKs with regular integration of new ST products

NNSA ST

- Open source NNSA Software projects
- Projects that have both mission role and open science role
- Major technical areas: New programming abstractions, math libraries, data and viz libraries
- Cover most ST technology areas
- Subject to the same planning, reporting and review processes

We work on products and applications needed now and into the future

Key themes:
- Exploration/development of new algorithms/software for emerging HPC capabilities:
- High-concurrency node architectures and advanced memory & storage technologies.
- Enabling access and use via standard APIs.

Software categories:
- The next generation of well-known and widely used HPC products (e.g., MPICH, OpenMPI, PETSc)
- Some lesser used but known products that address key new requirements (e.g., Kokkos, RAJA, Spack)
- New products that enable exploration of emerging HPC requirements (e.g., SICM, zfp, UnifyCR)

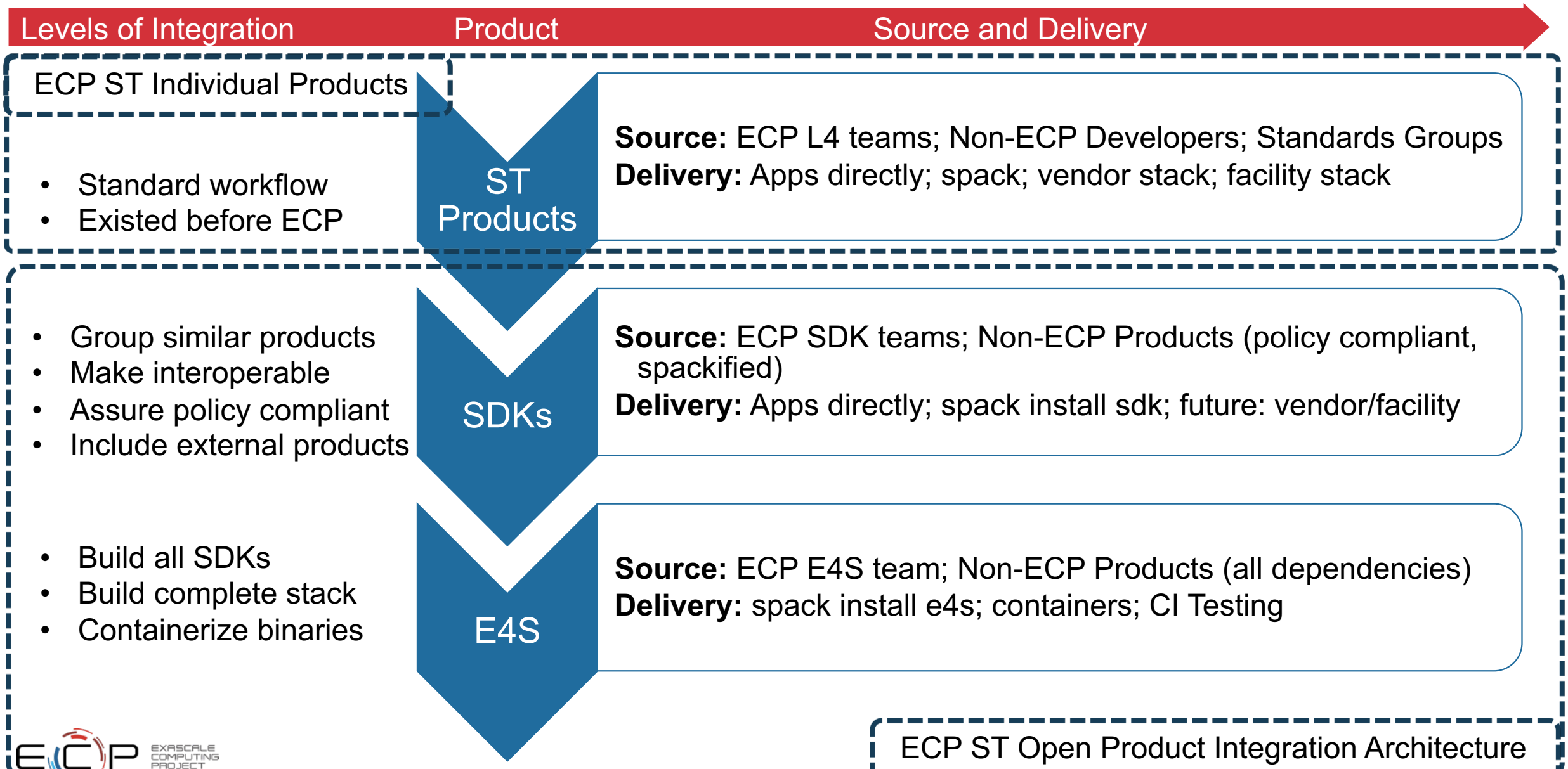
Example Products	Engagement
MPI – Backbone of HPC apps	Explore/develop MPICH and OpenMPI new features & standards.
OpenMP/OpenACC –On-node parallelism	Explore/develop new features and standards.
Performance Portability Libs: Kokkos, RAJA	Lightweight APIs for compile-time polymorphisms.
LLVM/Vendor compilers	Injecting HPC features, testing/feedback to vendors.
Perf Tools - PAPI, TAU, HPCToolkit	Explore/develop new features.
Math Libraries: BLAS, sparse solvers, etc.	Scalable algorithms and software, critical enabling technologies.
IO: HDF5, MPI-IO, ADIOS	Standard and next-gen IO, leveraging non-volatile storage.
Viz/Data Analysis	ParaView-related product development, node concurrency.

The Extreme-Scale Scientific Software Stack (E4S):

A collaborative HPC Linux Ecosystem

Delivering an open, hierarchical software ecosystem

Delivering an open, hierarchical software ecosystem



E4S Components

- E4S is a curated release of ECP ST products based on Spack [http://spack.io].
- E4S Spack cache to support bare-metal installs at facilities and custom container builds:
  - x86_64, ppc64le, and aarch64
- Container images on DockerHub and E4S website of pre-built binaries of ECP ST products.
- Base images and full featured containers (GPU support).
- GitHub recipes for creating custom images from base images.
- e4s-cl for container launch and for replacing MPI in application with system MPI libraries.
- Validation test suite on GitHub provides automated build and run tests.
- Automates build process via GitLab Continuous Integration to ensure packages can be built.
- E4S Doc Portal aggregates and summarizes documentation and metadata by raking product repos.
- E4S VirtualBox image with support for Docker, Shifter, Singularity, and Charliecloud runtimes.
- AWS image to deploy E4S on EC2.

https://e4s.io

Extreme-scale Scientific Software Stack (E4S)

- E4S: A Spack-based distribution of ECP ST and related and dependent software tested for interoperability and portability to multiple architectures
- Provides distinction between SDK usability / general quality / community and deployment / testing goals
- Will leverage and enhance SDK interoperability thrust

- Oct 2018: E4S 0.1 - 24 full, 24 partial release products
- Jan 2019: E4S 0.2 - 37 full, 10 partial release products
- Nov 2019: E4S 1.0 - 50 full, 5 partial release products
- Feb 2020: E4S 1.1 - 50 full,10 partial release products

e4s.io

Lead: Sameer Shende (U Oregon)

E4S 1.1 Full Release: 50 ECP Packages and all dependencies

- Adios
- AML
- Argobots
- Bolt
- Caliper
- Darshan
- Dyninst
- Faodel
- Flecsi
- Gasnet
- GEOPM
- Gotcha
- HDF5
- HPCToolkit
- Hypre
- Kokkos
- Legion
- Libnrm
- Libquo
- Magma
- Mercury
- MFEM
- MPICH
- MPIFileUtils
- Ninja
- OpenMPI
- PAPI
- Papyrus
- Parallel netCDF
- PDT
- PETSc
- Qthreads
- Raja
- Rempi
- SCR
- Spack
- Strumpack
- Sundials
- SuperLU
- SZ
- Tasmanian
- TAU
- Trilinos
- Turbine

Packages installed using Spack

- Umpire
- UnifyFS
- UPC++ Veloc
- Zfp

All ST products will be released through E4S

Partial (in-progress) Release: ASCENT, Catalyst, Flang, libEnsemble, LLVM, Visit, others

E4S Software Development Kits (SDKs):

Collaborative community development of complementary software capabilities

Software Development Kits (SDKs): Key delivery vehicle for ECP

A collection of related software products (packages) where coordination across package teams improves usability and practices, and foster community growth among teams that develop similar and complementary capabilities

- **Domain scope**
  Collection makes functional sense
- **Interaction model**
  How packages interact; compatible, complementary, interoperable
- **Community policies**
  Value statements; serve as criteria for membership
- **Meta-infrastructure**
  Invokes build of all packages (Spack), shared test suites
- **Coordinated plans**
  Inter-package planning. Augments autonomous package planning
- **Community outreach**
  Coordinated, combined tutorials, documentation, best practices

ECP ST SDKs: Grouping similar products for collaboration & usability

Programming Models & Runtimes Core

Tools & Technologies

Compilers & Support

Math Libraries (xSDK)

Viz Analysis and Reduction

Data mgmt., I/O Services & Checkpoint/ Restart

"Unity in essentials, otherwise diversity"

xSDK version 0.5: November 2019
(21 math libs, 2 domain-specific packages)

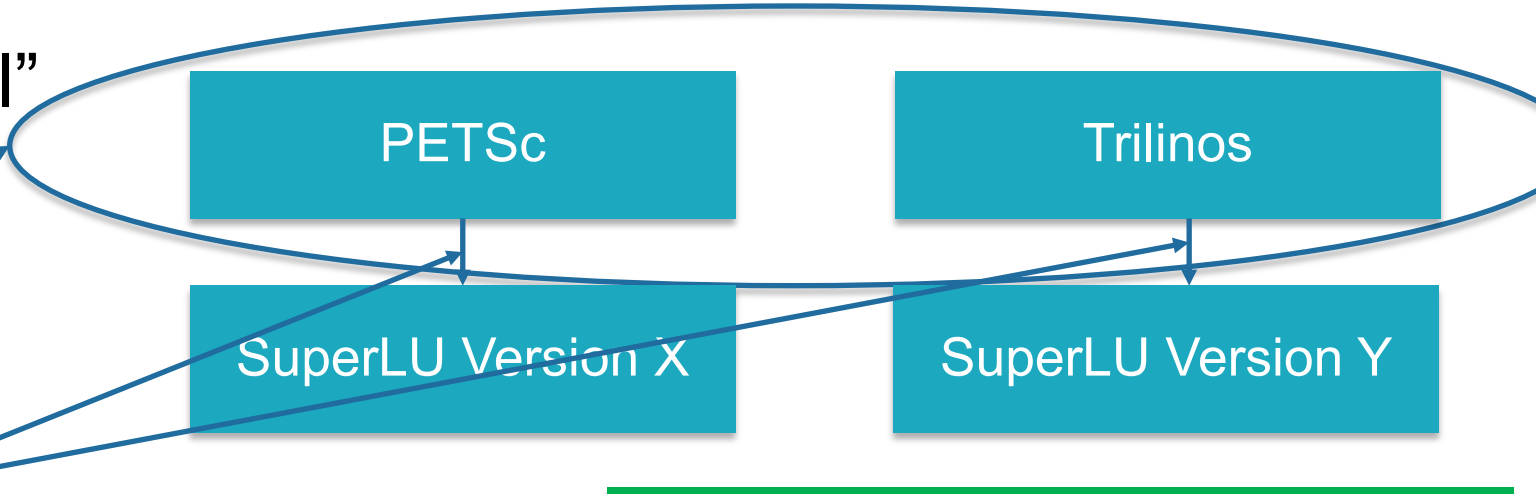
- AMReX
- ButterflyPACK
- DTK
- deal.ii
- Ginkgo
- hypre
- libEnsemble
- MAGMA
- MFEM
- Omega_h
- PETSc/TAO
- PHIST
- PLASMA
- preCICE
- PUMI
- SLEPc
- STRUMPACK
- SUNDIALS
- SuperLU
- Tasmanian
- Trilinos
- Pflotran
- Alquimia

Notes:

- Growth:
  - 5 in release 0.1.
  - 7 in 0.2
  - 9 in 0.3
  - 19 in 0.4
  - 23 in 0.5

- You do not need to build all packages.
- We build and test all packages.
- Any subset is guaranteed to build if using the same build parameters, platforms.
- Similar builds should work or require less effort for success.

SDK “Horizontal” Grouping: Adressing “Dependency Hell”



Horizonal (vs Vertical) Coupling

- Common substrate
- Similar function and purpose
  - e.g., compiler frameworks, math libraries
- Potential benefit from common Community Policies
  - Best practices in software design and development and customer support
- Used together, but not in the long vertical dependency chain sense
- Support for (and design of) common interfaces
  - Commonly an aspiration, not yet reality

Horizontal grouping:
- Assures X=Y.
- Protects against regressions.
- Transforms code coupling from heroic effort to turnkey.

xSDK community policies

https://xsdk.info/policies

xSDK compatible package: Must satisfy mandatory xSDK policies:

M1. Support xSDK community GNU Autoconf or CMake options.
M2. Provide a comprehensive test suite.
M3. Employ user-provided MPI communicator.
M4. Give best effort at portability to key architectures.
M5. Provide a documented, reliable way to contact the development team.
M6. Respect system resources and settings made by other previously called packages.
M7. Come with an open source license.
M8. Provide a runtime API to return the current version number of the software.
M9. Use a limited and well-defined symbol, macro, library, and include file name space.
M10. Provide an accessible repository (not necessarily publicly available).
M11. Have no hardwired print or IO statements that cannot be turned off.
M12. For external dependencies, allow installing, building, and linking against an outside copy of external software.
M13. Install headers and libraries under <prefix>/include/ and <prefix>/lib/.
M14. Be buildable using 64 bit pointers. 32 bit is optional.
M15. All xSDK compatibility changes should be sustainable.
M16. The package must support production-quality installation compatible with the xSDK install tool and xSDK metapackage.

Also **recommended policies**, which currently are encouraged but not required:

R1. Have a public repository.
R2. Possible to run test suite under valgrind in order to test for memory corruption issues.
R3. Adopt and document consistent system for error conditions/exceptions.
R4. Free all system resources it has acquired as soon as they are no longer needed.
R5. Provide a mechanism to export ordered list of library dependencies.
R6. Document versions of packages that it works with or depends on, preferably in machine-readable form
R7. Have README, SUPPORT, LICENSE, and CHANGELOG files in top directory.

xSDK member package: Must be an xSDK-compatible package, *and* it uses or can be used by another package in the xSDK, and the connecting interface is regularly tested for regressions.

We welcome feedback. What policies make sense for your software?

better scientific software

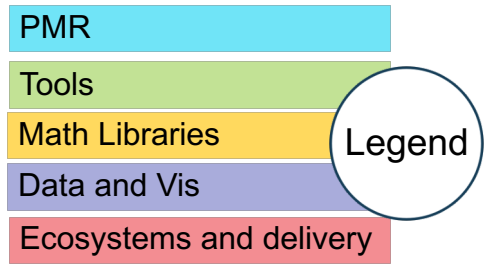
BSSw blog article:
P. Luszczek and U. Yang, Aug 2019,
https://bssw.io/blog_posts/building-community-through-software-policies

ECP ST SDKs will span all technology areas

Motivation: Properly chosen cross-team interactions will build relationships that support interoperability, usability, sustainability, quality, and productivity within ECP ST.

Action Plan: Identify product groupings where coordination across development teams will improve usability and practices, and foster community growth among teams that develop similar and complementary capabilities.

PMR Core (17)	Compilers and Support (7)	Tools and Technology (11)	xSDK (16)	Visualization Analysis and Reduction (9)	Data mgmt, I/O Services, Checkpoint restart (12)	Ecosystem/E4S at-large (12)
QUO	openarc	TAU	hypre	ParaView	SCR	mpiFileUtils
Papyrus	Kitsune	HPCToolkit	FleSCI	Catalyst	FAODEL	TriBITS
SICM	LLVM	Dyninst Binary Tools	MFEM	VTK-m	ROMIO	MarFS
Legion	CHiLL autotuning comp	Gotcha	Kokkoskernels	SZ	Mercury (Mochi suite)	GUFI
Kokkos (support)	LLVM openMP comp	Caliper	Trilinos	zfp	HDF5	Intel GEOPM
RAJA	OpenMP V & V	PAPI	SUNDIALS	VisIt	Parallel netCDF	BEE
CHAI	Flang/LLVM Fortran comp	Program Database Toolkit	PETSc/TAO	ASCENT	ADIOS	FSEFI
PaRSEC*		Search (random forests)	libEnsemble	Cinema	Darshan	Kitten Lightweight Kernel
DARMA		Siboka	STRUMPACK	ROVER	UnifyCR	COOLR
GASNet-EX		C2C	SuperLU		VeloC	NRM
Qthreads		Sonar	ForTrilinos		IOSS	ArgoContainers
BOLT			SLATE		HXHIM	Spack
UPC++			MAGMA			
MPICH			DTK			
Open MPI			Tasmanian			
Umpire			TuckerMPI			
AML						



E4S Community Candidate Policies V 1.0 Beta

- **Spack-based Build and Installation**

  Each E4S member package supports a scriptable Spack build and production-quality installation in a way that is compatible with other E4S member packages in the same environment. When E4S build, test, or installation issues arise, there is an expectation that teams will collaboratively resolve those issues.

- **Minimal Validation Testing**

  Each E4S member package has at least one test that is executable through the E4S validation test suite (https://github.com/E4S-Project/testsuite). This will be a post-installation test that validates the usability of the package. The E4S validation test suite provides basic confidence that a user can compile, install and run every E4S member package. The E4S team can actively participate in the addition of new packages to the suite upon request.

- **Sustainability**

  All E4S compatibility changes will be sustainable in that the changes go into the regular development and release versions of the package and should not be in a private release/branch that is provided only for E4S releases.

- **Product Metadata**

  Each E4S member package team will provide key product information via metadata that is organized in the E4S DocPortal format. Depending on the filenames where the metadata is located, this may require minimal setup.

- **Public Repository**

  Each E4S member package will have a public repository, for example at GitHub or Bitbucket, where the development version of the package is available and pull requests can be submitted.

- **Imported Software**

  If an E4S member package imports software that is externally developed and maintained, then it must allow installing, building, and linking against a functionally equivalent outside copy of that software. Acceptable ways to accomplish this include (1) forsaking the internal copied version and using an externally-provided implementation or (2) changing the file names and namespaces of all global symbols to allow the internal copy and the external copy to coexist in the same downstream libraries and programs.

- **Error Handling**

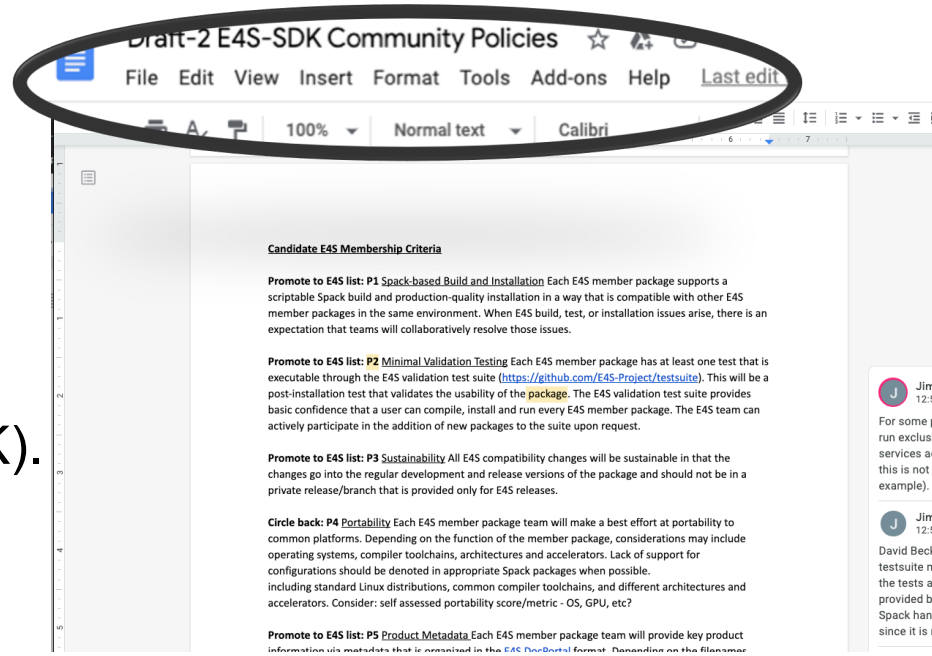
  Each E4S member package will adopt and document a consistent system for signifying error conditions as appropriate for the language and application. For e.g., returning an error condition or throwing an exception. In the case of a command line tool, it should return a sensible exit status on success/failure, so the package can be safely run from within a script.

- **Test Suite**

  Each E4S member package will provide a test suite that does not require special system privileges or the purchase of commercial software. This test suite should grow in its comprehensiveness over time. That is, new and modified features should be included in the suite.

E4S/SDK Policy Initiative Status

- Community policies are important for several reasons:
  - Commitment to quality
  - Membership criteria for the future
  - Community discussion
- Each SDK community developing policies like Math Libs (xSDK).
- Policies common to all SDKs will be promoted to E4S level
- Policies will determine:
  - Quality label
  - Membership in E4S and the SDKs
- Version 1.0 of policies due by end of 2020



E4S DocPortal

A Single Portal with Redirect to Product Documentation

Product Documentation Challenges: User Perspective

Finding info for specific product

What it does
License
Support
Contact info
More …

Finding new products

What can solve my problem

Trusting accuracy of information

Up to date
Complete

Hierarchical

Summary to deep dive

Product Documentation Challenges: Developer Perspective

Efficient and Effective generation and maintenance

Getting noticed by new users

Conveying summary information *and* details

E4S DocPortal Requirements

Provide a single online location for *accurate* product descriptions for ECP software products.

Derived requirements

Sustainable: Must be integrated into software team workflows.

Incremental: Must build on community approaches to providing this information.

Extensible: Must be usable by any open source software team.

E4S Documentation Portal Strategy

All content resides in product repositories

Use open source community approach of specially-name files in software repositories.

Adopt commonly used file names when available.

ID new information items not already being requested.

Documentation portal provides single point of access

Web-based raking tool capture information from product repositories and present in summary form.

Aggregates and summarizes documentation and metadata for E4S products

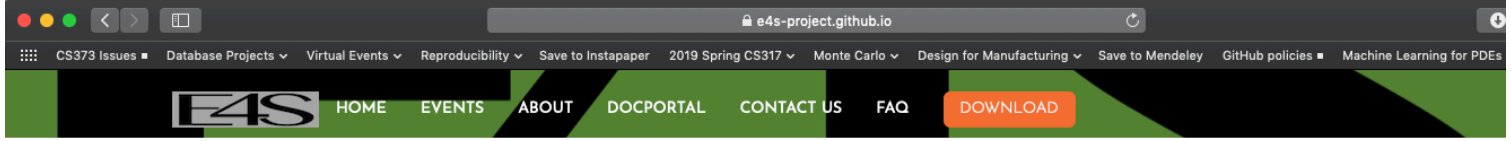
Regularly updates information directly from product repositories

Location: https://e4s-project.github.io/DocPortal.html

E4S DocPortal Status

- Completed DocPortal Prototype and Design Document
- Reviewed prototype/design with Facilities and ST developers
- Version 1.0 available now:
  - https://e4s-project.github.io/DocPortal.html
  - Available from e4s.io
- Next steps:
  - Work with ECP ST teams to improve amount and quality of raked content
  - Develop policy and acceptance criteria for E4S DocPortal membership

E4S DocPortal



- Summary Info
  - Name
  - Functional Area
  - Description
  - License
- Searchable
- Sortable

E4S Products

*: Member Product

Search:

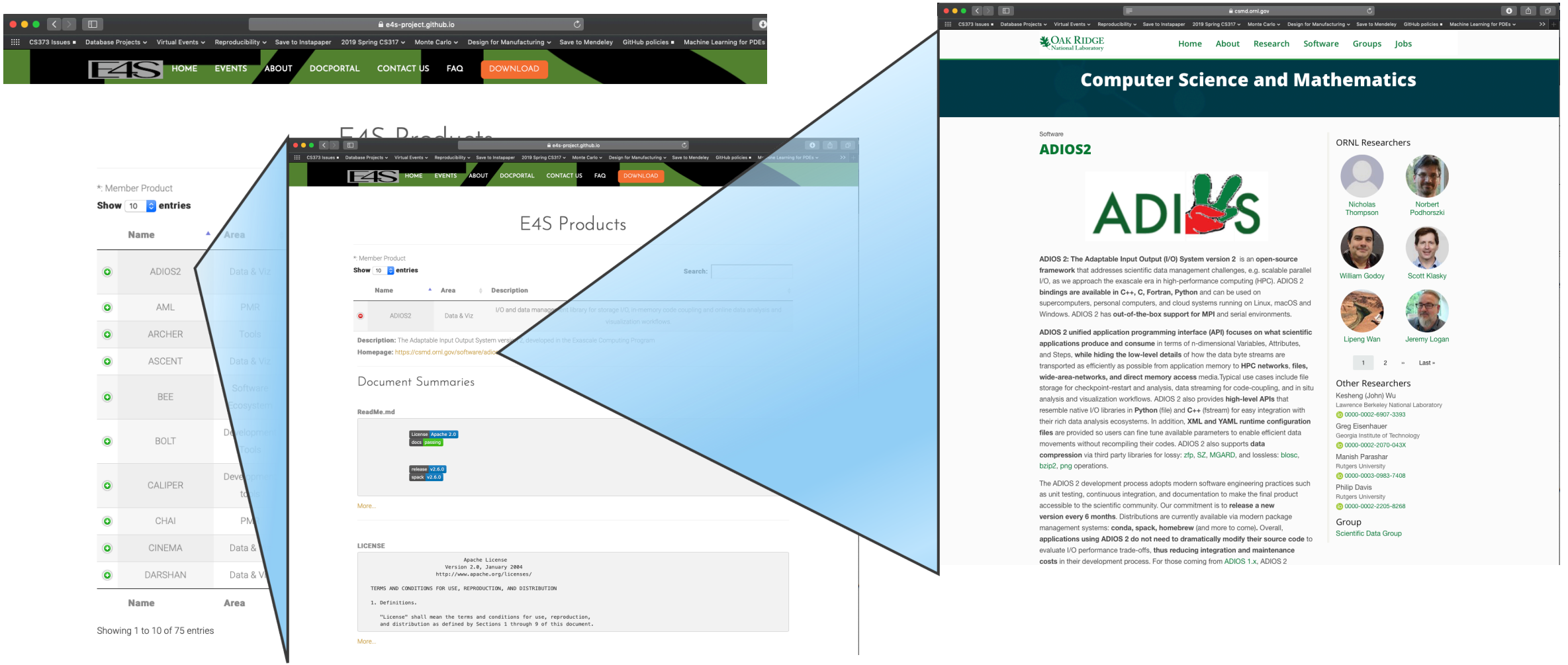
Name	Area	Description
ADIOS2	Data & Viz	I/O and data management library for storage I/O, in-memory code coupling and online data analysis and visualization workflows.
AML	PMR	Hierarchical memory management library from Argo.
ARCHER	Tools	Data race detection tool for OpenMP applications
ASCENT	Data & Viz	Flyweight in situ visualization and analysis runtime for multi-physics HPC simulations
BEE	Software Ecosystem	Container-based solution for portable build and execution across HPC systems and cloud resources
BOLT	Development Tools	OpenMP over lightweight threads.
CALIPER	Development tools	Performance analysis library.
CHAI	PMR	A library that handles automatic data migration to different memory spaces behind an array-style interface.
CINEMA	Data & Viz	Data analysis and visualization tool suite.
DARSHAN	Data & Viz	I/O characterization tool.
Name	Area	Description

Showing 1 to 10 of 75 entries

Previous 1 2 3 4 5 … 8 Next

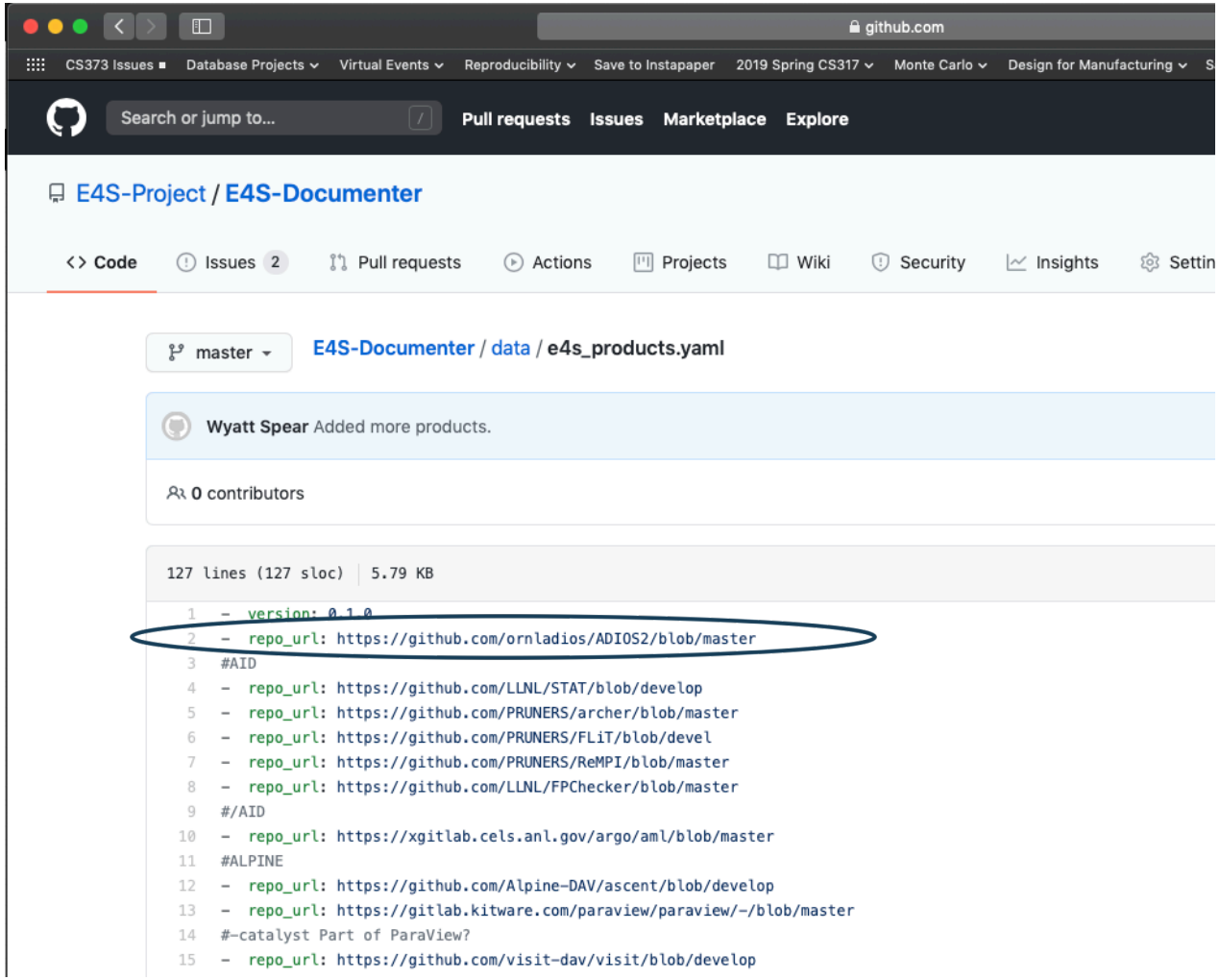
https://e4s-project.github.io/DocPortal.html

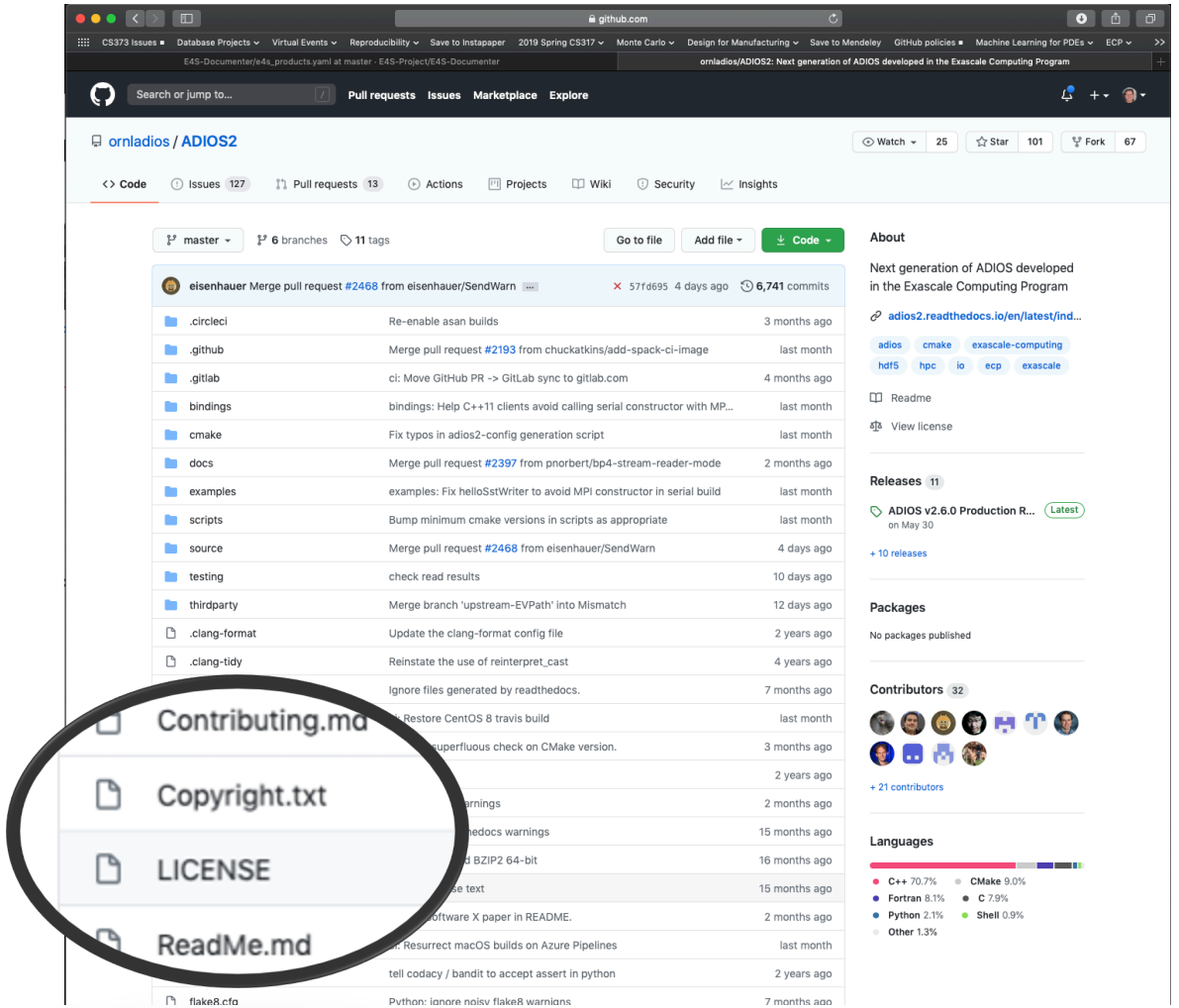
Goal: All E4S Product Documentation Accessible from single portal on E4S.io (Working Mock Webpage below)



https://e4s-project.github.io/DocPortal.html

Q: What do we need for adding a product to the DocPortal?
A: A repo URL + up-to-date meta-data files





E4S DocPortal Initiative Status: Fall 2020

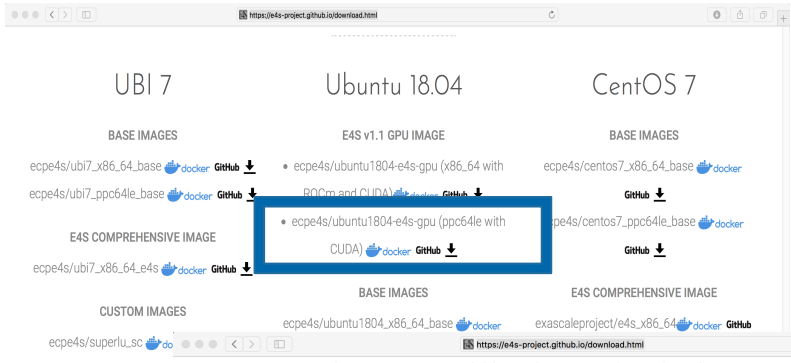
- Completed DocPortal Prototype and Design Document
- Reviewed prototype/design with Facilities and ST developers
- Version 1.0 available now: https://e4s-project.github.io/DocPortal.html
- Next steps:
  - Work with ECP ST teams to improve amount and quality of raked content
  - Develop policy and acceptance criteria for E4S DocPortal membership

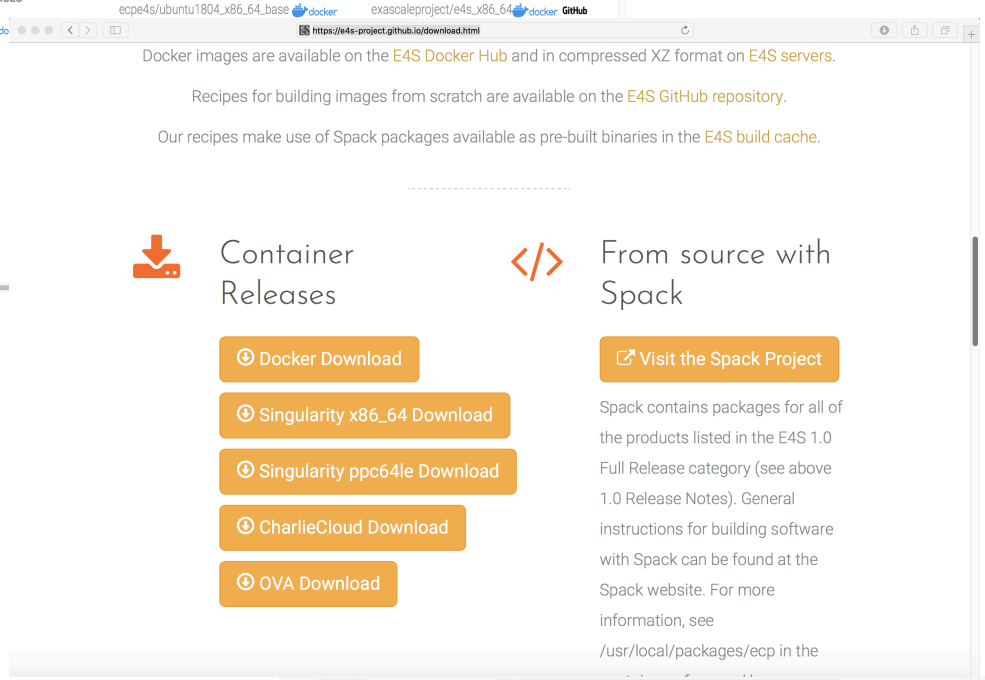
How to Access E4S Software

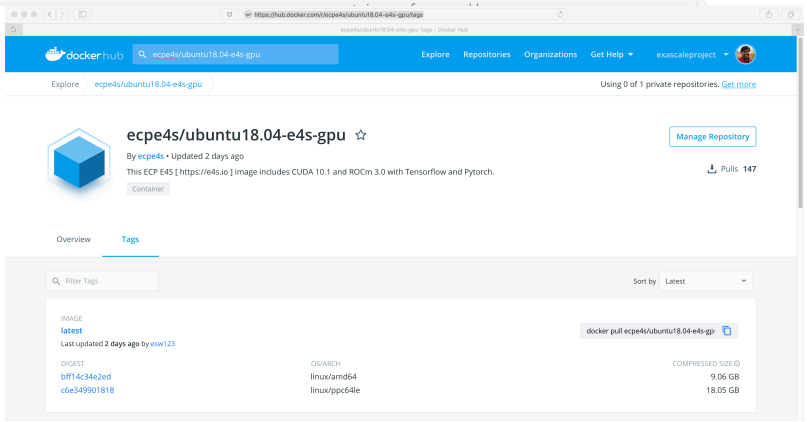
From Source and Many Other Ways

E4S v1.1 Access

- From e4s-project.github.io
  - Source via Spack
  - E4S v1.1 GPU image
  - Docker, Singularity (ppc64le, x86_64) …
- E4S v1.1 Release Available at DockerHub
  - 40+ ECP ST Products
  - Support for GPUs
    - NVIDIA (CUDA 10.1.243)
    - ppc64le and x86_64
- Visit https://e4s.io for more details





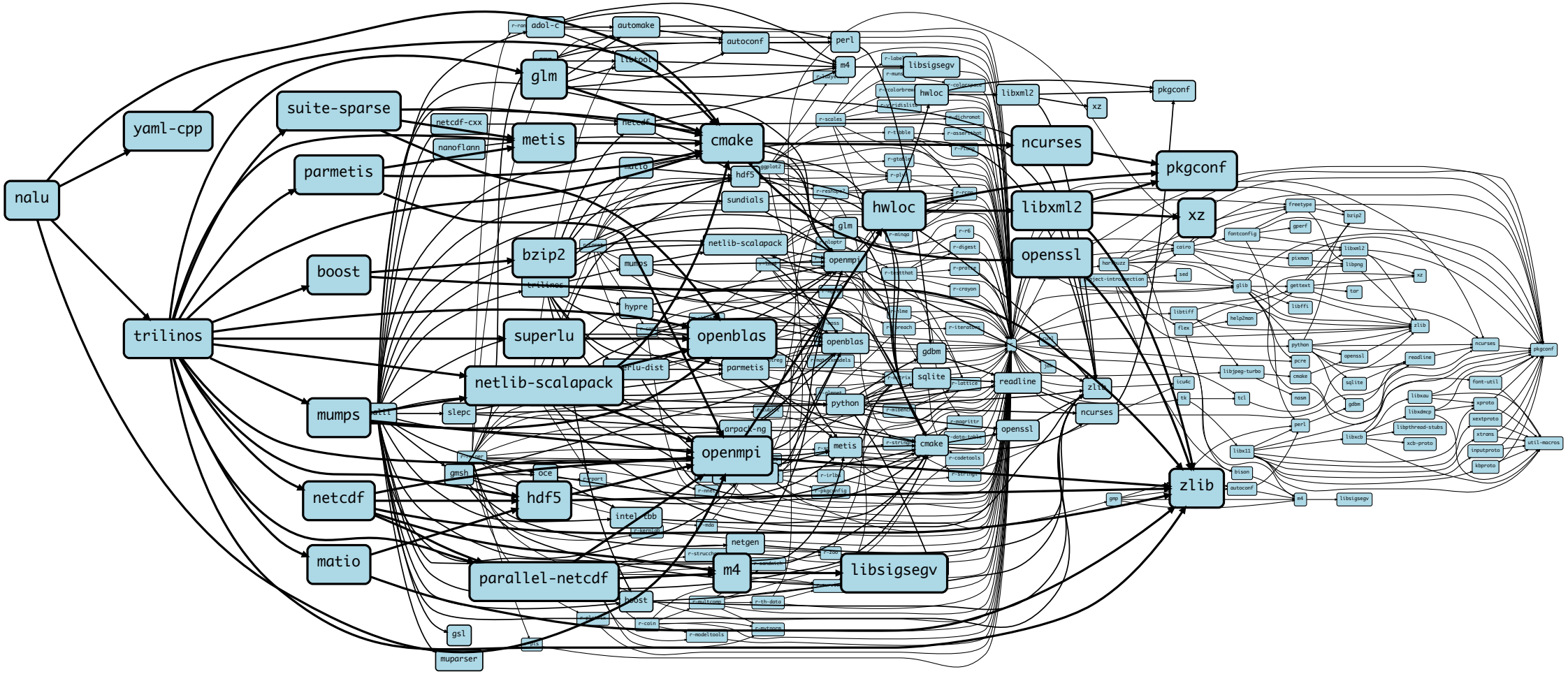


Spack

Spack

- E4S uses the Spack package manager for software delivery
- Spack provides the ability to specify versions of software packages that are and are not interoperable.
- Spack is a build layer for not only E4S software, but also a large collection of software tools and libraries outside of ECP ST.
- Spack supports achieving and maintaining interoperability between ST software packages.

Scientific software is becoming extremely complex



Nalu: Generalized Unstructured Massively Parallel Low Mach Flow

deal.ii: C++ Finite Element Library

RMiner: R Data Mining Library

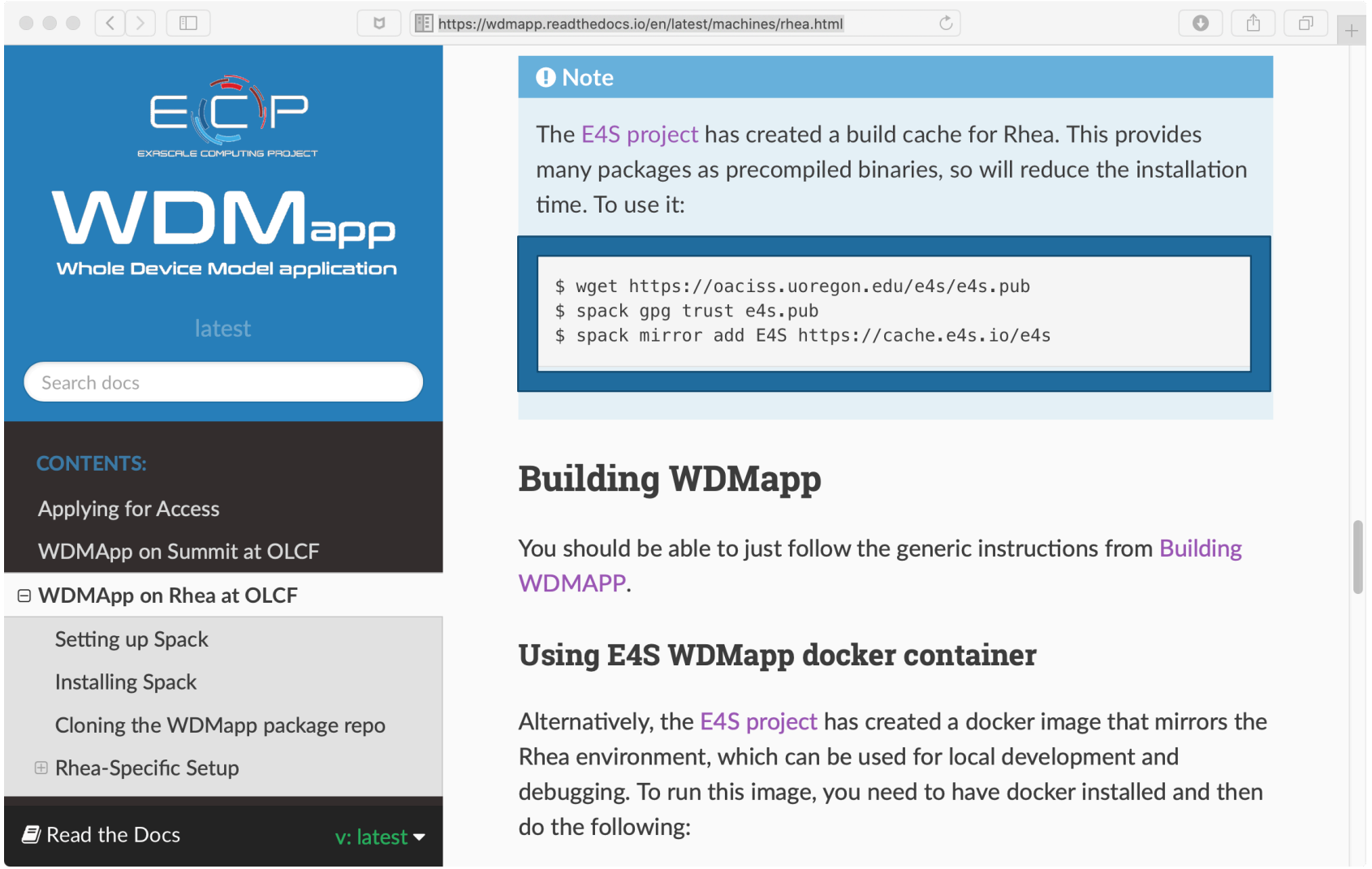
E4S Spack Build Cache and Container Build Pipeline

E4S: Spack Build Cache at U. Oregon

- 10,000+ binaries
- S3 mirror
- No need to build from source code!

- https://oaciss.uoregon.edu/e4s/inventory.html

WDMApp: Speeding up bare-metal installs using E4S build cache



- E4S Spack build cache
- Adding E4S mirror
- WDMApp install speeds up!

- https://wdmapp.readthedocs.io/en/latest/machines/rhea.html

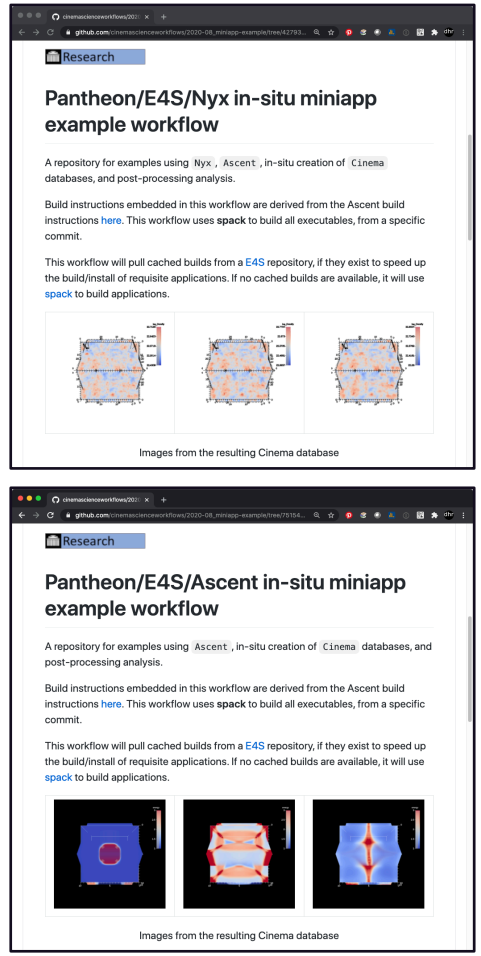
Pantheon and E4S support end-to-end ECP examples

Overview: The Exascale Computing Project (ECP) is a complex undertaking, involving a myriad of technologies working together. An outstanding need is a way to capture, curate, communicate and validate workflows that cross all of these boundaries.

The **Pantheon** and **E4S** projects are collaborating to advance the integration and testing of capabilities, and to promote understanding of the complex workflows required by the ECP project. Utilizing a host of ECP technologies (spack, Ascent, Cinema, among others), this collaboration brings curated workflows to the fingertips of ECP researchers.

Contributions

- Curated end-to-end application/in-situ analysis examples can be run quickly by anyone on Summit. (https://github.com/pantheonscience/ECP-E4S-Examples)
- Pantheon/E4S integration speeds up build/setup times over source builds due to cached binaries (approx.10x speed up).



Instructions page for (top) Nyx, Ascent and Cinema workflow repository, and (bottom) Cloverleaf3d, Ascent, Cinema workflow. These curated workflows use Pantheon, E4S and spack to provide curated workflows for ECP.

E4S: Building on top of previous efforts

- E4S did not emerge from nothing
- Leveraging the work of many others
- HPC Linux: Work done at U. of Oregon, and at ParaTools, Inc
- IDEAS-Classic: xSDK – the original SDK continuing under ECP
- Spack – Pre-dates E4S
- All the useful open source software created by the scientific community

E4S Summary

What E4S is not	What E4S is
• A closed system taking contributions only from DOE software development teams.	• Extensible, open architecture software ecosystem accepting contributions from US and international teams. • Framework for collaborative open-source product integration.
• A monolithic, take-it-or-leave-it software behemoth.	• A full collection if compatible software capabilities **and** • A manifest of a la carte selectable software capabilities.
• A commercial product.	• Vehicle for delivering high-quality reusable software products in collaboration with others.
• A simple packaging of existing software.	• The conduit for future leading edge HPC software targeting scalable next-generation computing platforms. • A hierarchical software framework to enhance (via SDKs) software interoperability and quality expectations.

Some E4S Collaboration Models

Approach	Comments/Potential
Use E4S containers for tutorials and webinars	Many ST technologies offer tutorial/webex forums to learn more; range from introductory to advanced
Evaluate new capabilities using E4S software products	**Accelerator-enabled software stack** (compilers, programming environments, tools, math libraries, in situ), **next-generation IO** (HDF5, ADIOS, PNetCDF)
Install E4S	A goal for us: Want to explore how to make this possible. Collaboration can help us improve our product development and delivery.
Contribute to E4S and SDKs	• E4S and SDKs are open architectures enabling light-weight integration of similar products. • Contributions already exist from NSF and European funded projects • Talk with us about contributing your product to the DocPortal • Assess your product against our community policies
Overall	Full collaborative interactions benefit everyone.

E4S/SDK Summary

- E4S/SDK Software: Curated release of complete production-quality HPC Linux software stack:
  - **Latest ECP-developed features** for 50+ products.
  - **Ported and validated** regularly on all common and emerging HPC platforms.
  - **Single DocPortal access** to all product documentation.
  - **Collaborative development communities** around SDKs to build culture of quality.
  - **Policies** for SW and user experience quality.
  - **Containers, build caches** for (dramatic) reduction in build time and complexity.
- E4S: A new member of the HPC ecosystem:
  - **A managed portfolio** of HPC software teams and products.
  - **Enabling first-of-a-kind collaboration:** vendors, facilities, US agencies, industry and internationally.
  - Extensible to new domains: **AI/ML.**
  - **A new way of delivering reusable HPC software** with ever-improving quality and functionality.

https://e4s.io

ECP Software Technology Capability Assessment Report (CAR) Version 2.0

- Comprehensive document about ECP ST structure, progress and planning
- Version 2.0:
  - E4S/SDK details
  - 2-page writeups for each product
  - Released February 1, 2020
- **Version 2.5:**
  - **Refresh of 2-page writeups**
  - **Foreshadowing of next ECP features in E4S**
  - **Scheduled for November 15, 2020**

https://exascaleproject.org/wp-content/uploads/2020/02/ECP-ST-CAR-V20-1.pdf
All ECP Reports: https://www.exascaleproject.org/reports/