The Molecular Sciences Software Institute

Cecilia Clementi, T. Daniel Crawford, Robert Harrison,
Teresa Head-Gordon, Shantenu Jha*, Anna Krylov,
Vijay Pande, and Theresa Windus

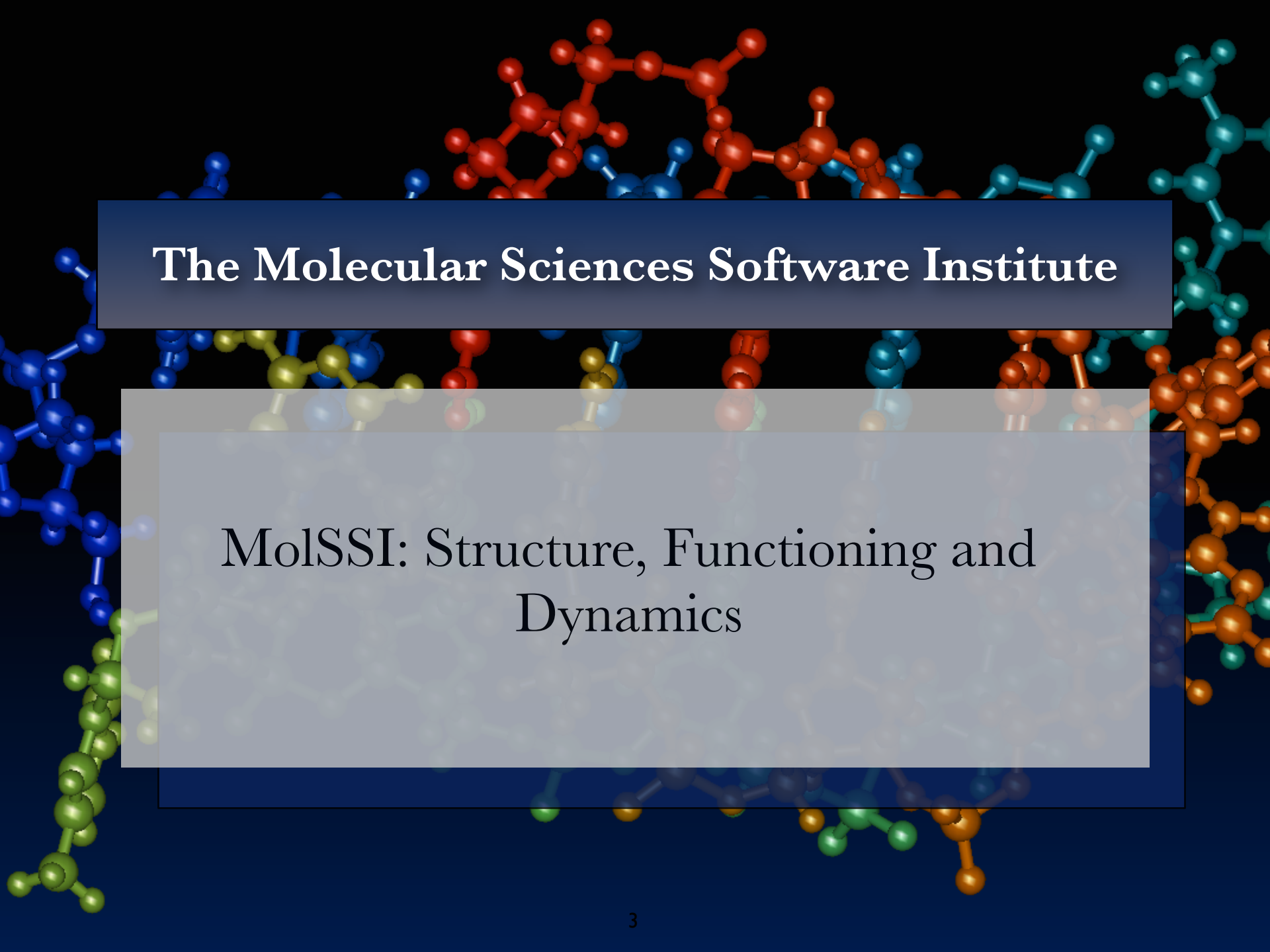
http://molssi.org

D3R

23 August, 2019

The Molecular Sciences Software Institute (MolSSI)

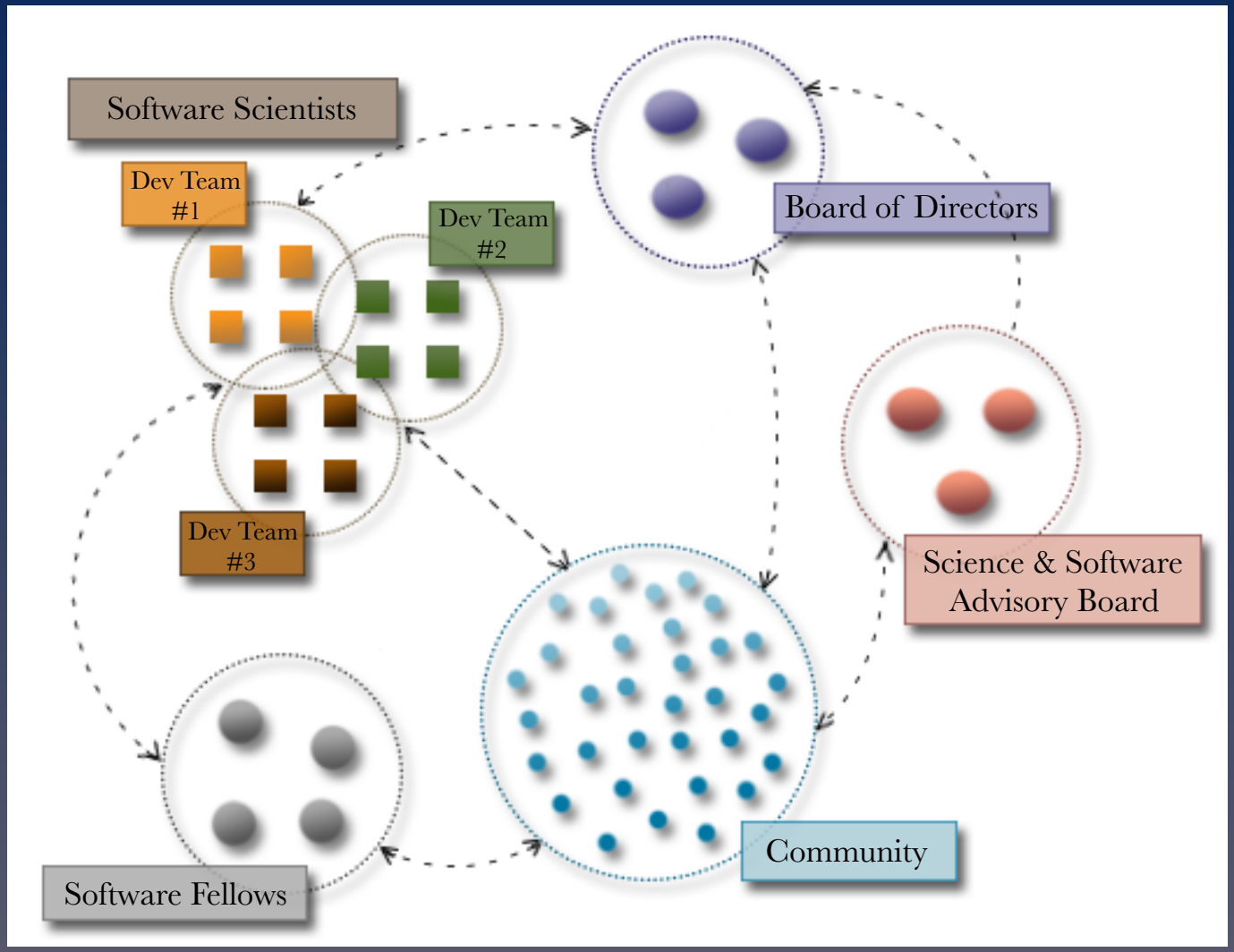
- Project (start date of August 1st, 2016) funded by the National Science Foundation.
- Collaborative effort by Virginia Tech, Rice U., Stony Brook U., U.C. Berkeley, Stanford U., Rutgers U., U. Southern California, and Iowa State U.
- Total budget of $19.42M for five years, potentially renewable to ten years.
- Joint support from numerous NSF divisions: Advanced Cyberinfrastructure (ACI), Chemistry (CHE), Division of Materials Research (DMR), Office of Multidisciplinary Activities (OMA)
- Designed to **serve** and **enhance** the software development efforts of the broad field of computational molecular science.



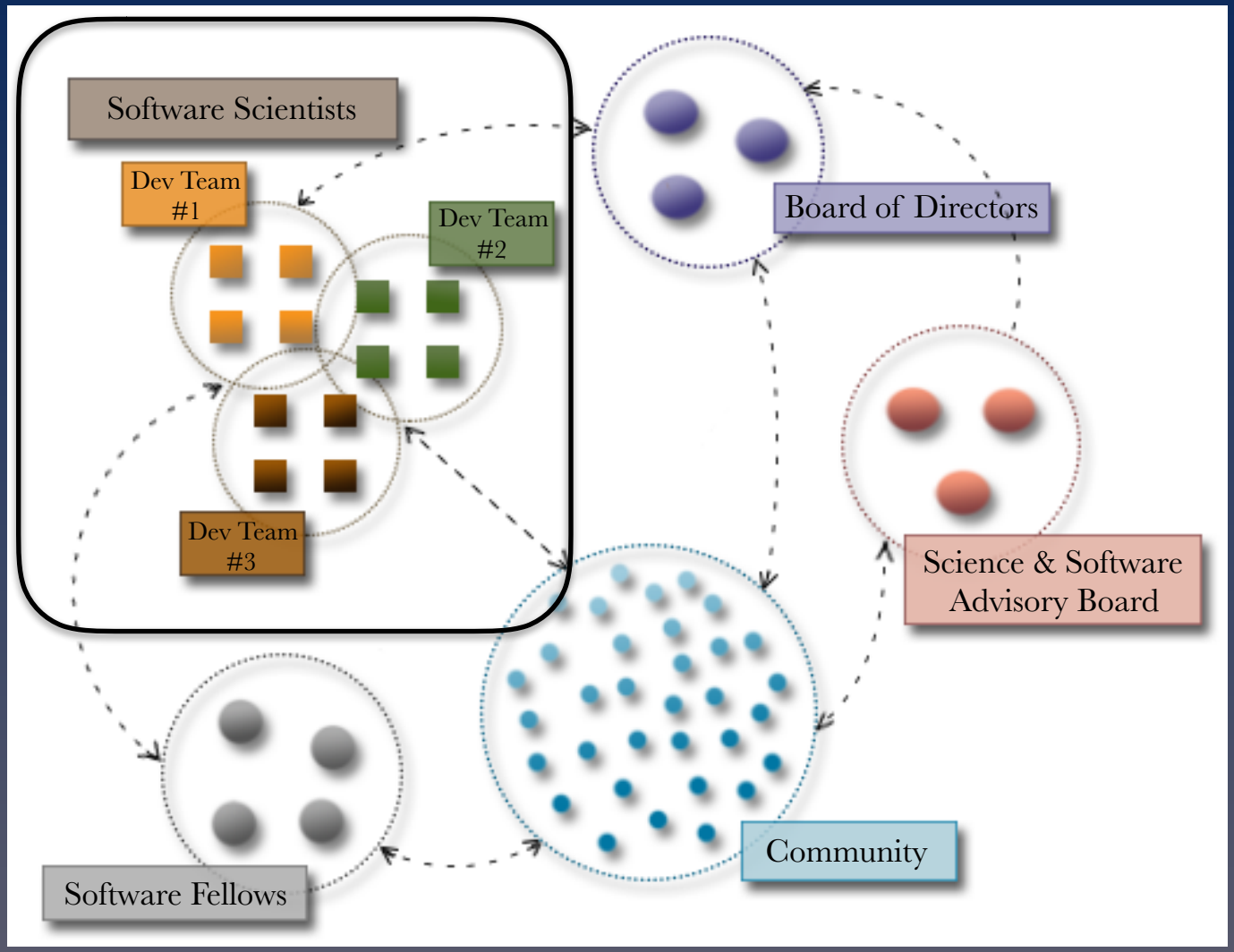
The Molecular Sciences Software Institute

MolSSI: Structure, Functioning and Dynamics

The Molecular Sciences Software Institute (MolSSI)



The Molecular Sciences Software Institute (MolSSI)

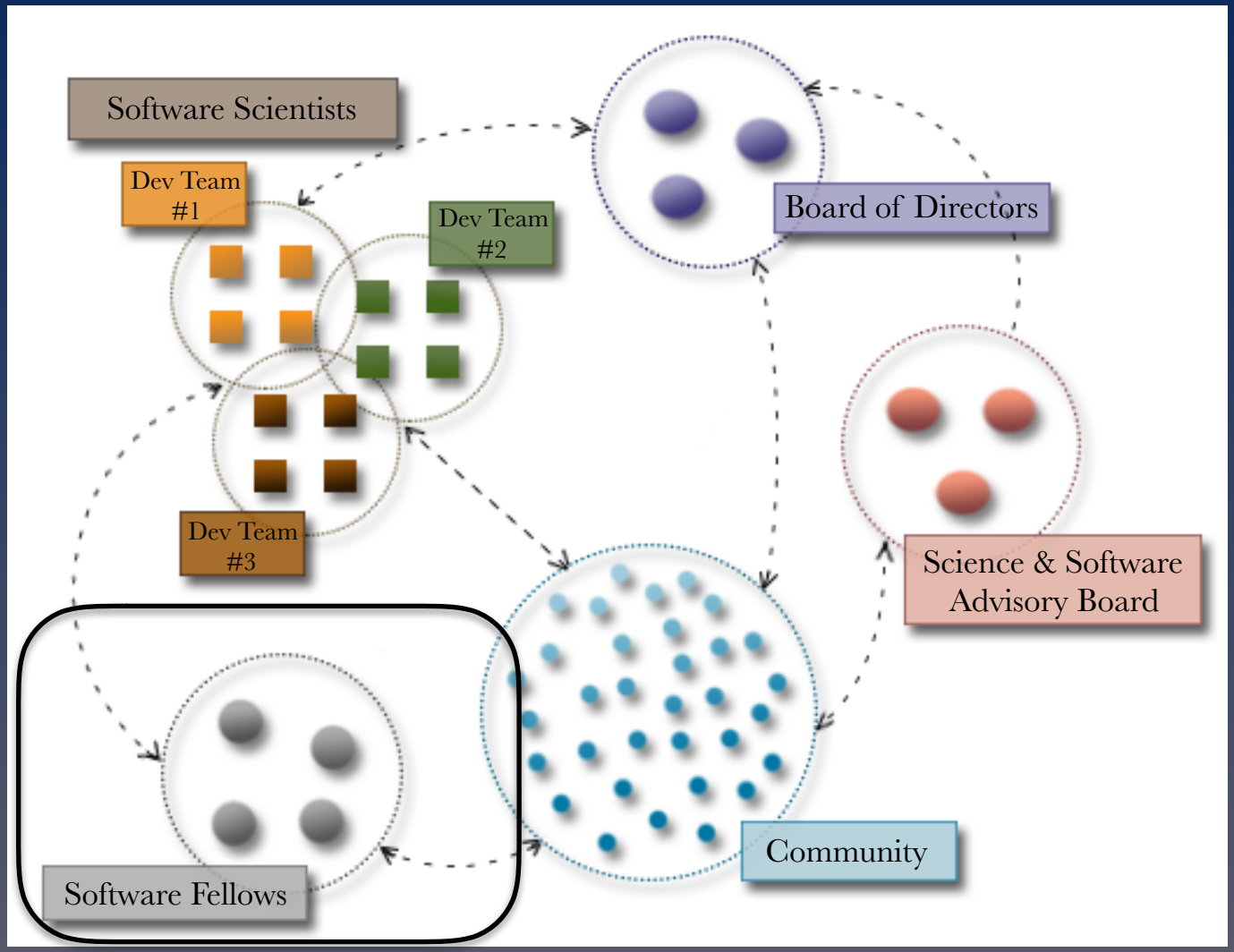


The MolSSI Software Scientists (MSSs)

- A team of ~12 software engineering experts, drawn both from newly minted Ph.D.s and established researchers in molecular sciences, computer science, and applied mathematics.
- Dedicated to multiple responsibilities:
  - Developing software infrastructure and frameworks;
  - Interacting with CMS research groups and community code developers;
  - Training and Education Mission (summer schools, bootcamps..)
  - Serving as mentors to MolSSI Software Fellows;
  - Working with industrial, national laboratory, and international partners;

Approximately 50% of the Institute's budget will directly support the MolSSI Software Scientists.

The Molecular Sciences Software Institute (MolSSI)

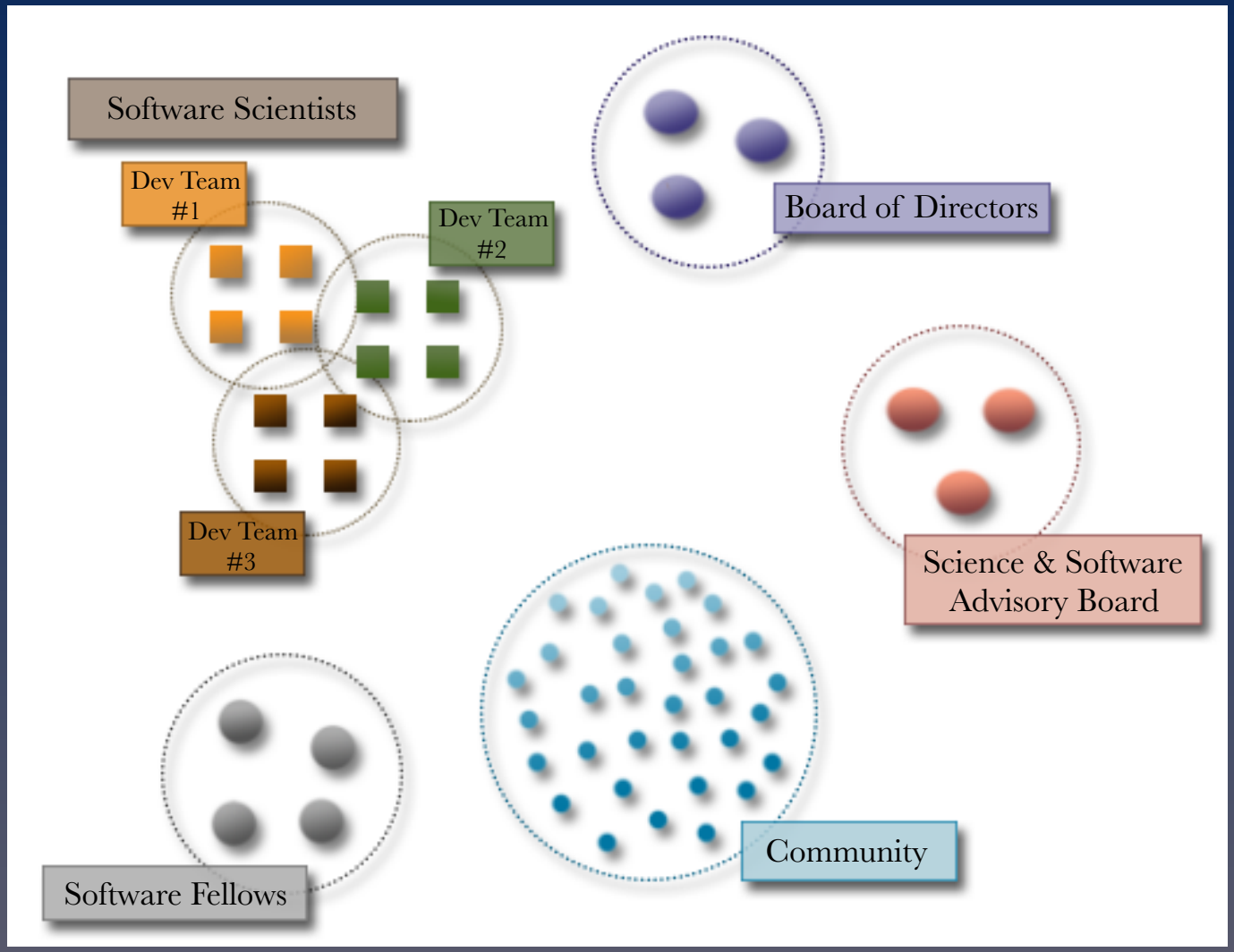


The MolSSI Software Fellows (MSFs)

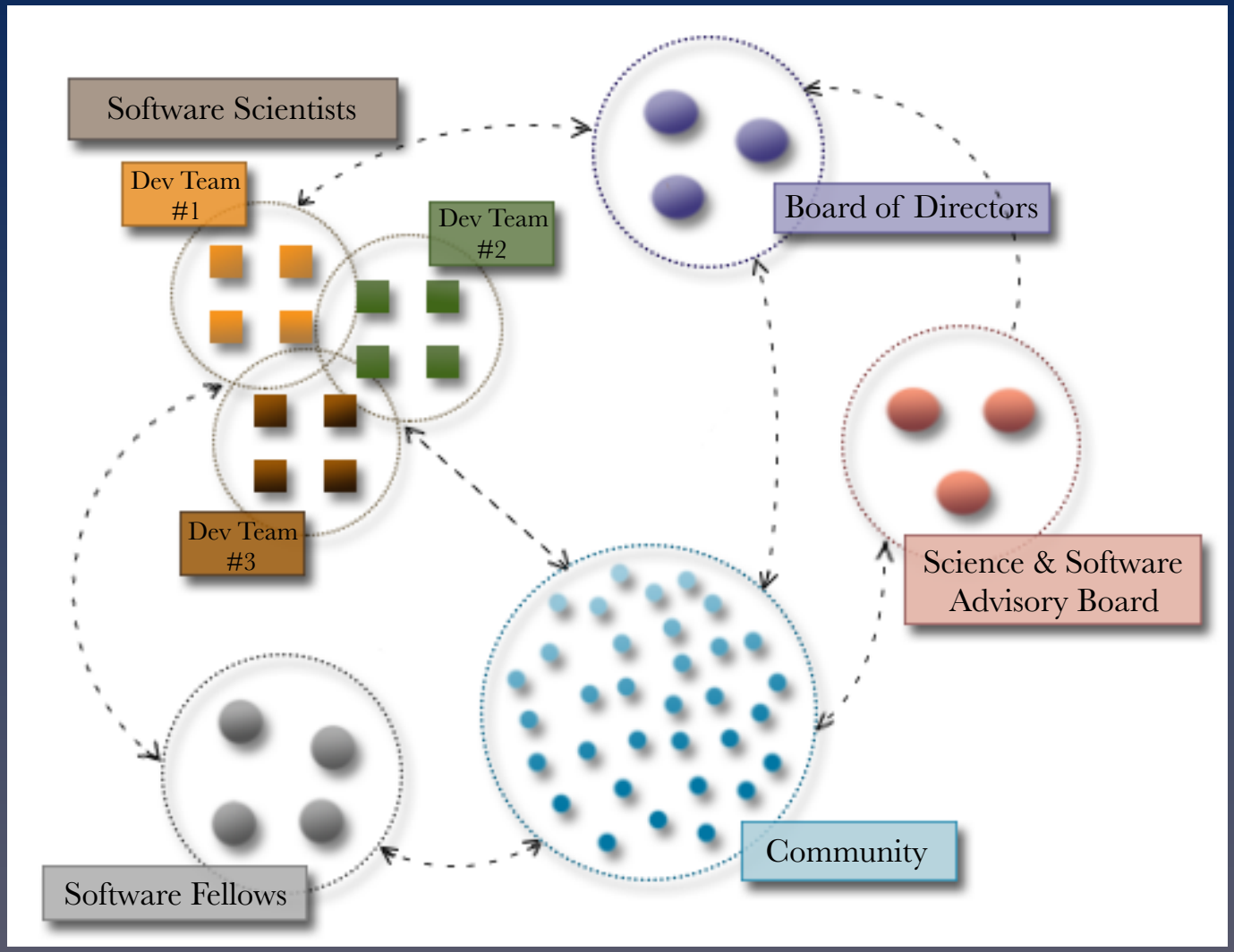
- A cohort of ~16 Fellows supported simultaneously – graduate students and postdocs selected by the Science and Software Advisory Board from research groups across the U.S.
- Fellows will work directly with both the Software Scientists and the MolSSI Directors, thus providing a conduit between the Institute and the CMS community itself.
- Fellows will work on their own projects, as well as contribute to the MolSSI development efforts, and they will engage in outreach and education activities under the Institute guidance.
- Funding for MolSSI Software Fellows will follow a flexible, two-phase structure, providing up to two years of support.

Approximately 25% of the Institute's budget will directly support the MolSSI Software Fellows.

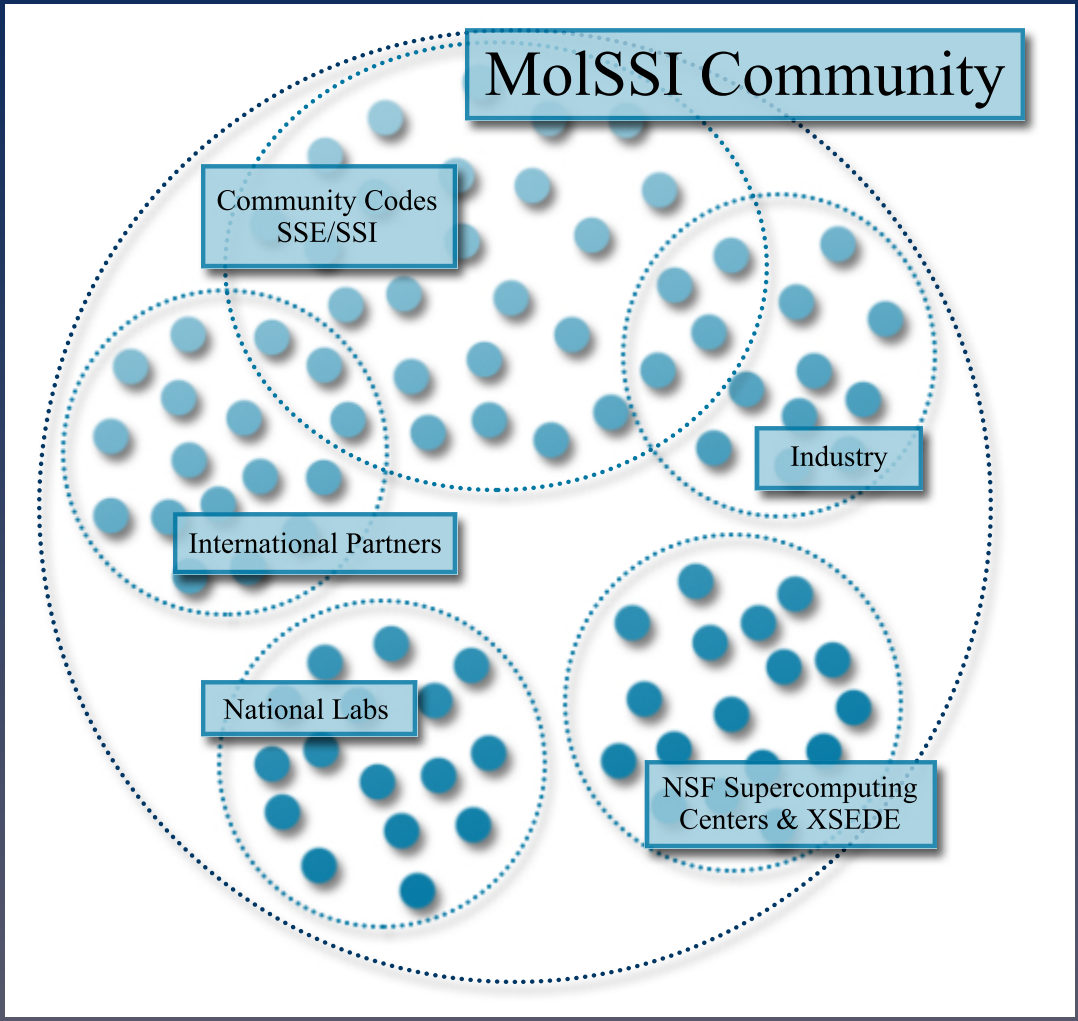
The Molecular Sciences Software Institute (MolSSI)



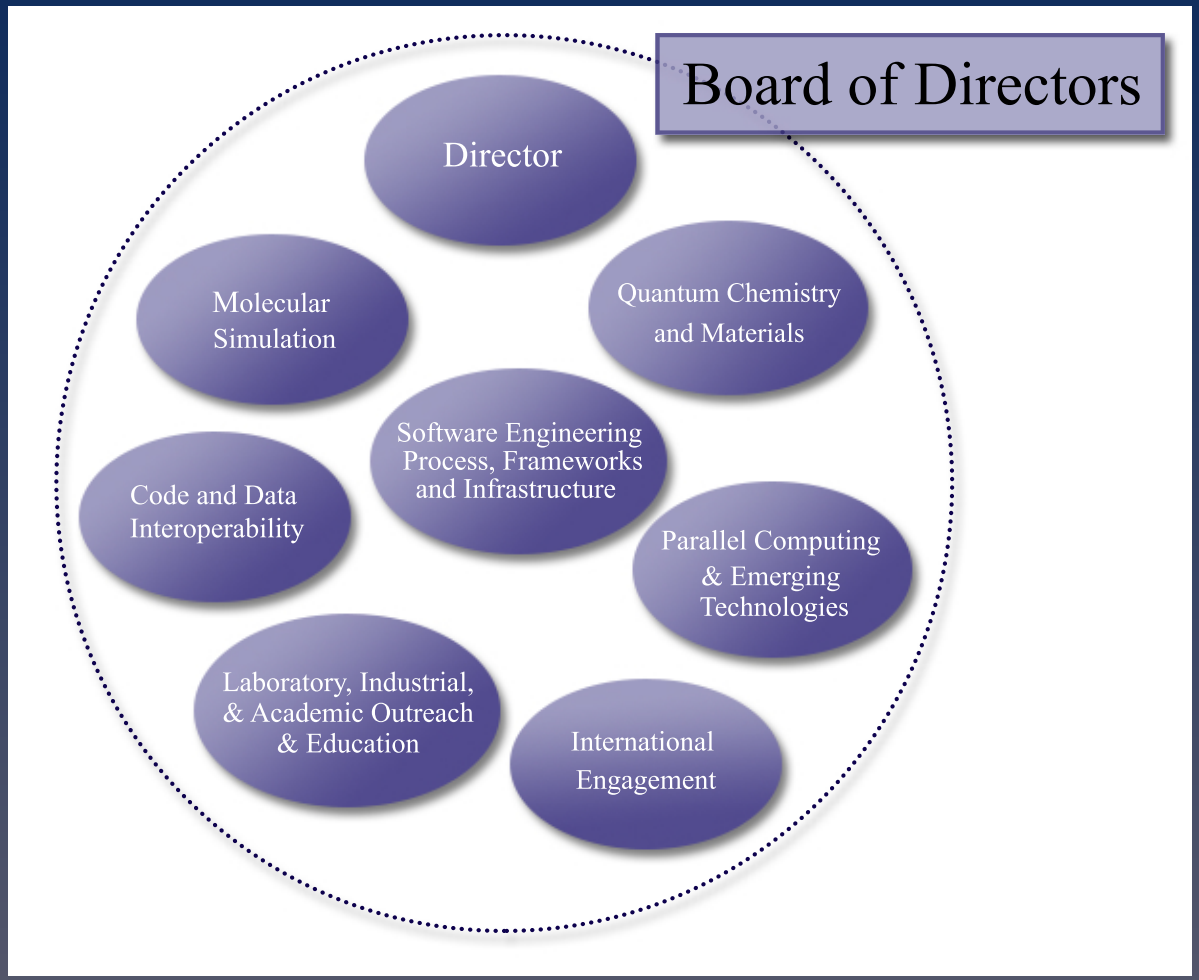
The Molecular Sciences Software Institute (MolSSI)



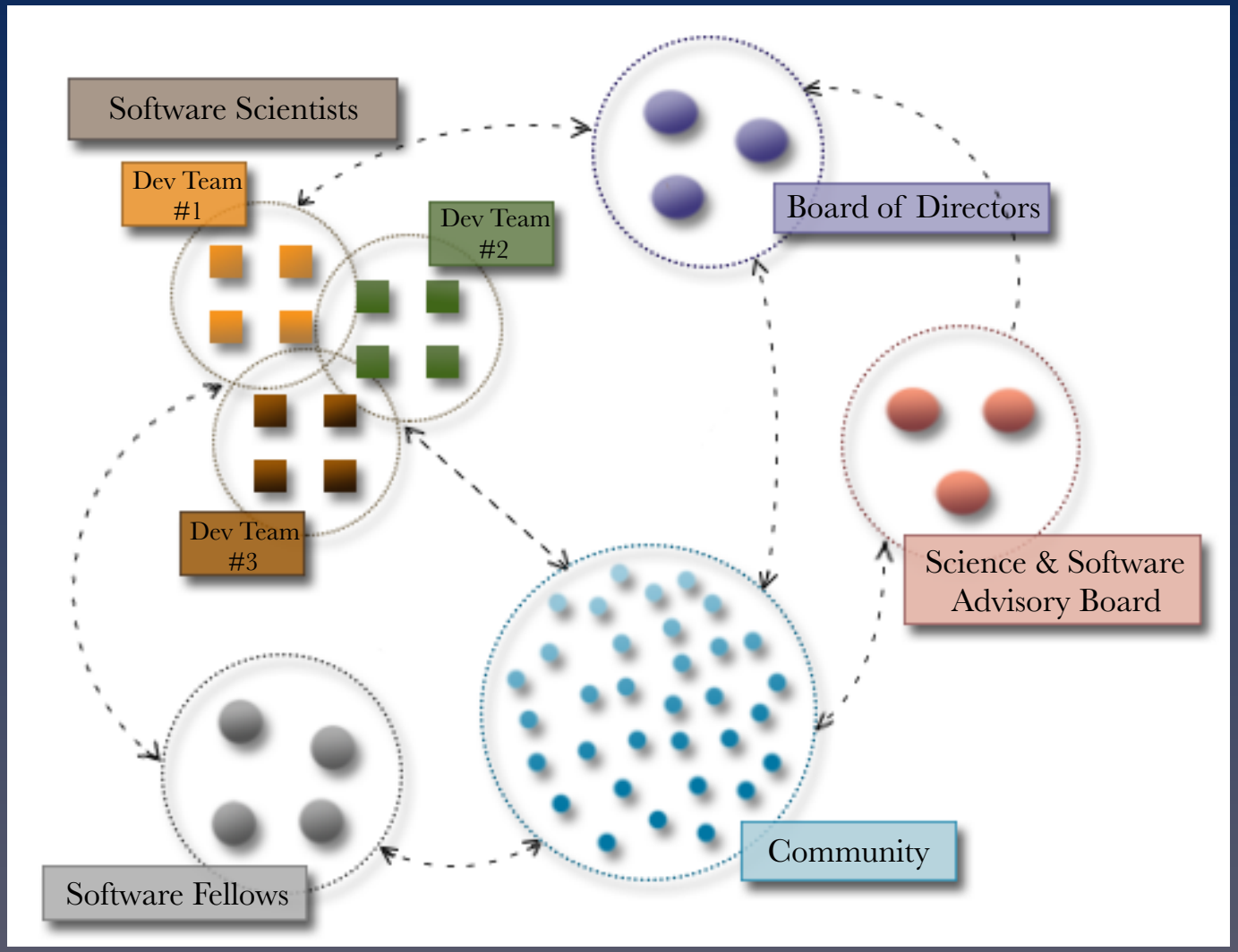
The MolSSI Community



The MolSSI Board of Directors



The Molecular Sciences Software Institute (MolSSI)



Middleware Building Blocks for Workflow Systems

Shantenu Jha
Rutgers University and Brookhaven National Laboratory

http://radical.rutgers.edu

Outline

- **Middleware Building Blocks for Workflow Systems**
- **Ensemble Computational Model**
  - System & performance sensitive components versus user-facing component
- **Building Blocks for Workflow Systems for Adaptive Ensembles**
  - RADICAL-Cybertools: RADICAL-Pilot and Ensemble ToolKit (EnTK)
  - Case Studies

Some Statements About Workflows (2019) ….

- "*All computational problems require workflows*"
- *"Everyone has a different workflow"*
- *"The optimization of the end-to-end performance of a workflow is important (and different..)"*
- *"**Nothing tends so much to the advancement of knowledge as the application of a new instrument.** The native intellectual powers of people in different times are not so much the causes of the different success of their labors, as the peculiar nature of the means and artificial resources in their possession" -- Humphrey Davy*

A Brief History of Workflows & Systems

- **1970s:** “Business Workflows”, early 1990s: Workflow Management Coalition
- **Late 1990’s** (Early 2000s): Increased uptake of Scientific WMS
  - **Grid/Distributed workflows -- driven by LHC**
  - HPC Workflows (ASCI Program)
- **2001:** MyGrid / MyExperiment emphasized provenance and reproducibility,
  - Advances in workflow sharing, e.g., Taverna (cross-disciplinary WMS)
  - Implementations rely upon changing technologies. Sustainability?
- **2014:** DOE ASCR Workflow Modelling Program (Rich Carlson)
- **2019:** Approximately 240 computational & data analysis Workflow Systems
  - https://s.apache.org/ existing- workflow- systems
    - Caveat: Diverse systems; complete - partial; extensible - standalone, …
  - “Unwittingly developed” !
  - Most workflow users don’t use a “formal” WMS, but “roll their own”

Perspective on Workflows & Systems

- Initially workflow **management** systems provided **end-to-end** capabilities:
  - “Big Science”; software infrastructure was fragile, missing services
  - Run many times, for many users: amortisation of development overhead
- Workflows aren’t what they used to be
  - HTC important but other design points: automation, sophistication, …
  - **The workflow** is a manifestation of algorithmic & methodological innovation
- The infrastructure is not what it used to be either!
  - Python ecosystem, e.g., task distribution and coordination systems
  - Apache (big) data stack of analysis tools; container technologies ..

Status Quo: Workflows & Workflow Systems

- Need **sustainable** ecosystem of both existing and new software components from which **tailored workflow systems** can be composed
  - Lower barrier to integration of components
  - **Supply** (workflow system development) and **Demand** (workflows) side!
- Separate performance sensitive from application facing components
  - Engineer for design points: Usability vs Functionality vs Scalability
- A **systems approach** which addresses both **technical** & **social factors**
  - Incentivize sharing and collective community capability
  - Enable expert contributions, lower expertise to contribute and **use***

* “Which workflow system should I **use**?” was the most frequently asked question at BW 2017

Middleware Building Blocks for Workflow Systems

- Building Block Approach:
  - Principled approach to the **architectural** design of middleware systems;
  - Applies traditional notion of modularity at the **software systems level**
  - Enable composability among **independent** software systems

- The four design principles:
  - **Self-sufficient:** Implements functionalities; not dependent on other blocks
  - **Composable:** Caller can compose functionalities from independent BB
  - **Interoperable:** Usable in diverse system without semantic modification
  - **Extensible:** Building block functionality and entities can be extended

Middleware Building Blocks for Workflow Systems https://arxiv.org/abs/1903.10057

Building Blocks (BB): Properties

- A BB is a semantically well-defined independent software system, agnostic to coordination, and communication **patterns**, and exposed via an API.
  - **Stronger** (stringent) property than modularity
- BB have well defined state, event, and error models
  - Reduce challenge of composability of independent components
- BB work stand-alone, or integrated with other BB, or with 3rd party software
- Architecturally building blocks require:
  - Stable interfaces & distinction between computation and composition
  - Conversion layers -- multiple representation of the same entity

Outline

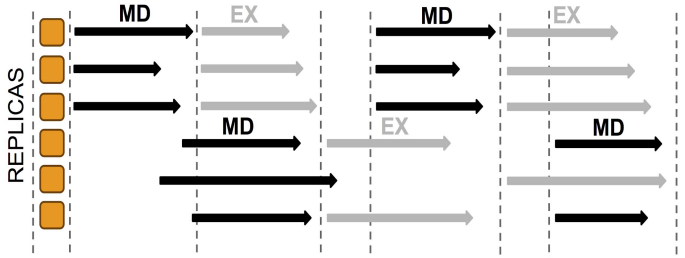
- **Middleware Building Blocks for Workflow Systems**

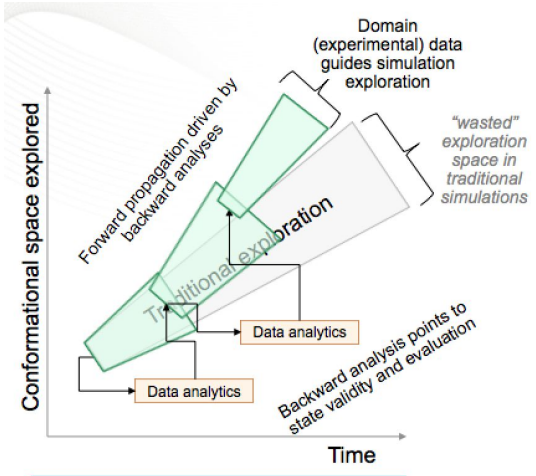
- **Ensemble Computational Model**
  - System & performance sensitive components versus user-facing component

- **Building Blocks for Workflow Systems for Ensemble Computing**
  - RADICAL-Cybertools: RADICAL-Pilot and Ensemble ToolKit (EnTK)
  - Case Studies: Performance, Functionality and Extensibility

Adaptive Ensemble Algorithms: Variation on a theme

- Ensemble-based methods necessary, but not sufficient !
- **Adaptive** Ensemble-based Algorithms: Intermediate data, determines next stages
- Adaptivity: How and What
  - Internal data used: Simulation generated data used to determine “optimal” adaptation

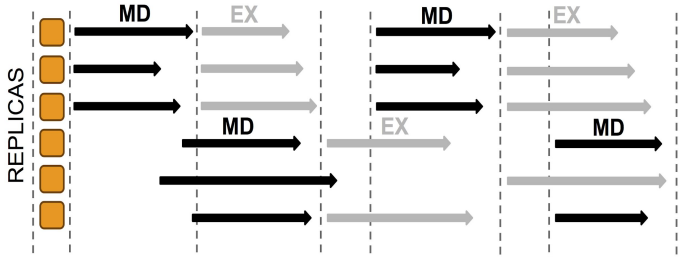


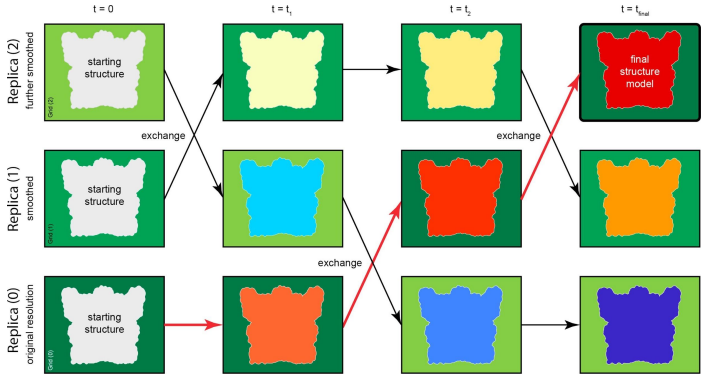


*Chodera, J.D., Noe, F., Curr. Opin. Struct. Biol. (2014)

Adaptive Ensemble Algorithms: Variation on a theme

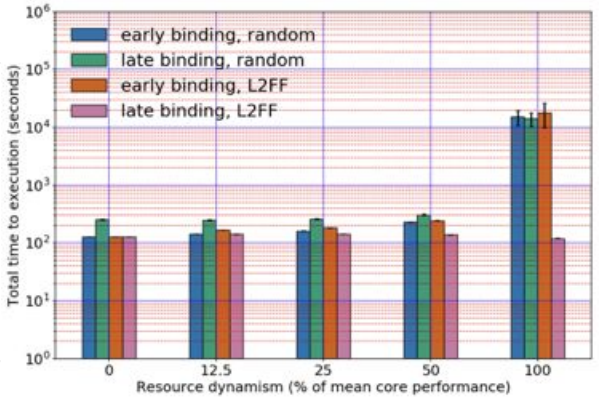
- Ensemble-based methods necessary, but not sufficient !
- **Adaptive** Ensemble-based Algorithms: Intermediate data, determines next stages
- Adaptivity: How and What
  - Internal data used: Simulation generated data used to determine “optimal” adaptation
  - External data used, e.g., experimental or separate computational process.
  - What: Task parameter(s), order, count, ….





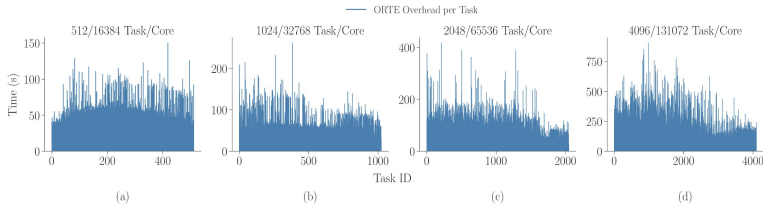
Ensemble Simulations at Scale: Challenges

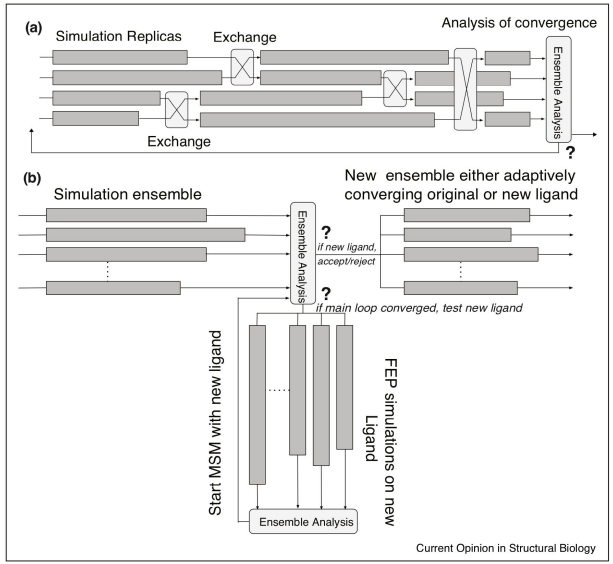
- **Resource Management** for $O(10^{5-6})$ tasks -- each is independent executing program!
  - Exascale ~$O(10^{6-9})$
- **Application requirements and resource performance must be dynamic**
  - Abstraction of static perf. is inadequate!
  - Implications on perf. portability & scaling
- **Execution Model** of heterogeneous tasks on heterogeneous and dynamic resources.
  - Early-binding: A->B->C->D
  - Late-binding: C->B->A->D



Ensemble Simulations at Scale: Challenges

- **Resource Management** for O(10^{5-6}) tasks -- each is independent executing program!
  - Exascale ~O(10^{6-9})
- **Application requirements and resource performance must be dynamic**
  - Abstraction of static perf. is inadequate!
  - Implications on perf. portability & scaling
- **Execution Model** of heterogeneous tasks on heterogeneous and dynamic resources.
- **Adaptive Ensemble Algorithms:** Encoding algorithms that express adaptivity, even statistically ("approximately")?
  - Managing interactions (coupling) between tasks
  - **…..**

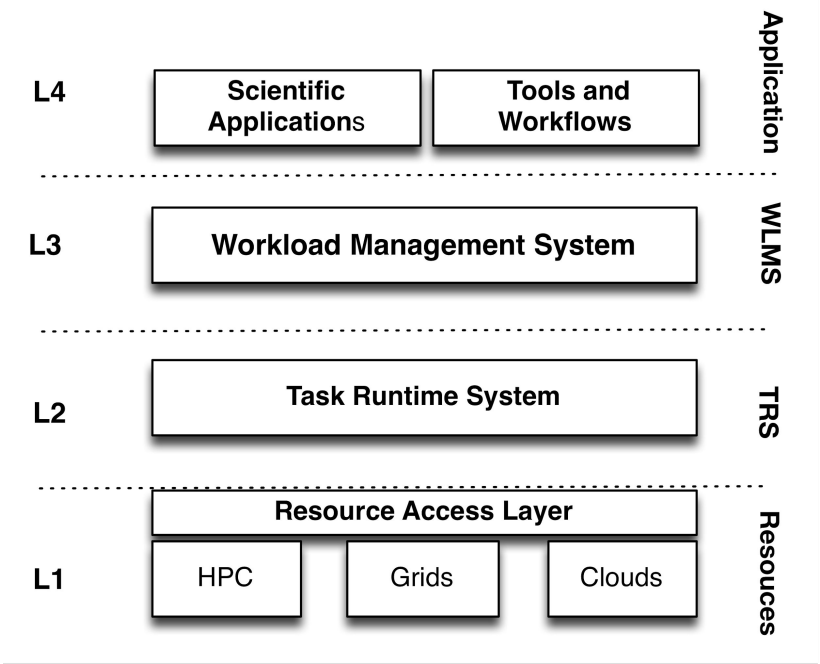




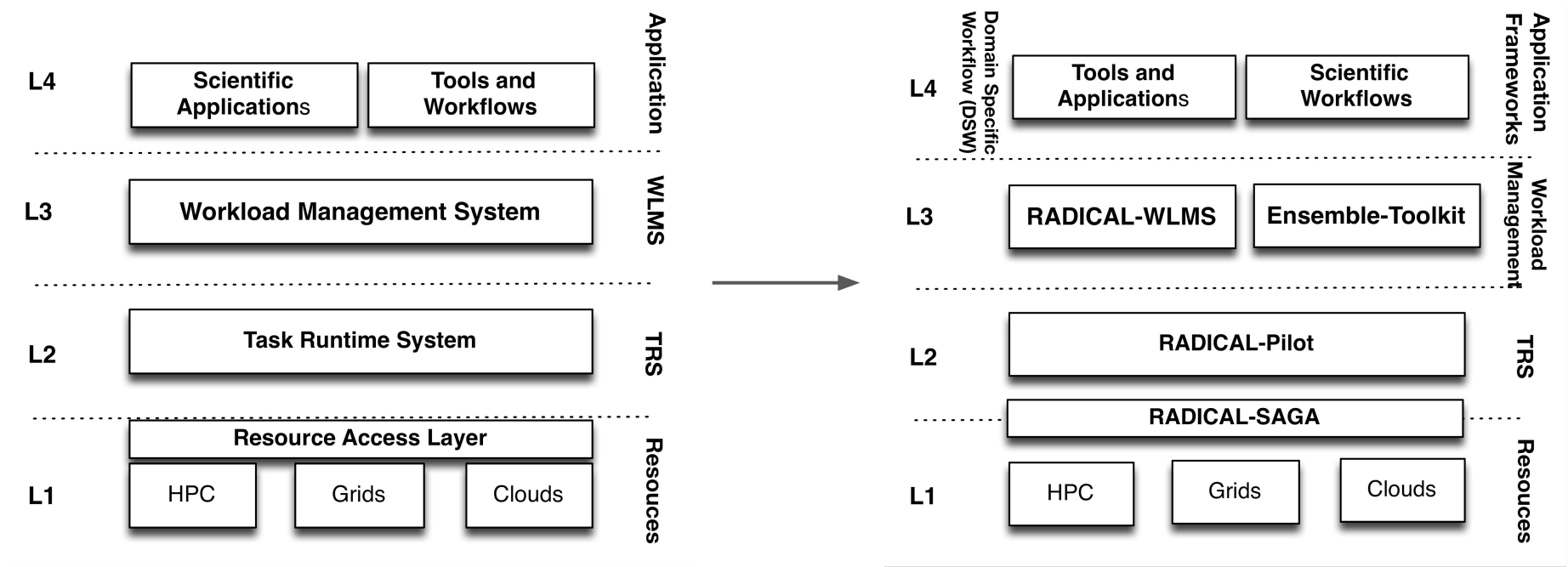
Outline

- Middleware Building Blocks for Workflow Systems
- Ensemble Computational Model
  - System & performance sensitive components versus user-facing component
- **Building Blocks for Workflow Systems for Adaptive Ensembles**
  - RADICAL-Cybertools: RADICAL-Pilot and Ensemble ToolKit (EnTK)
  - Case Studies: Adaptive Sampling and Adaptive Ensemble

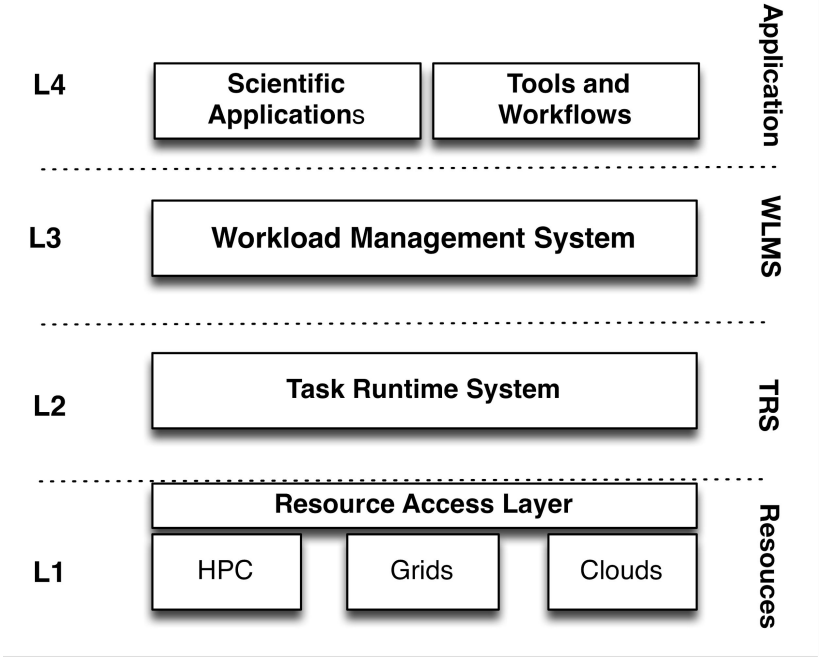
Developing Workflow Tools Using Building Blocks

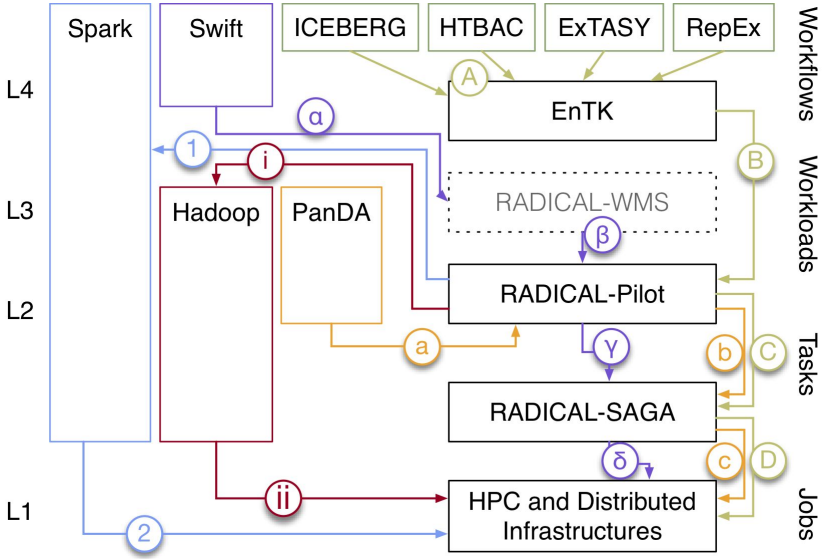


RADICAL-Cybertools: Building Blocks for Workflows

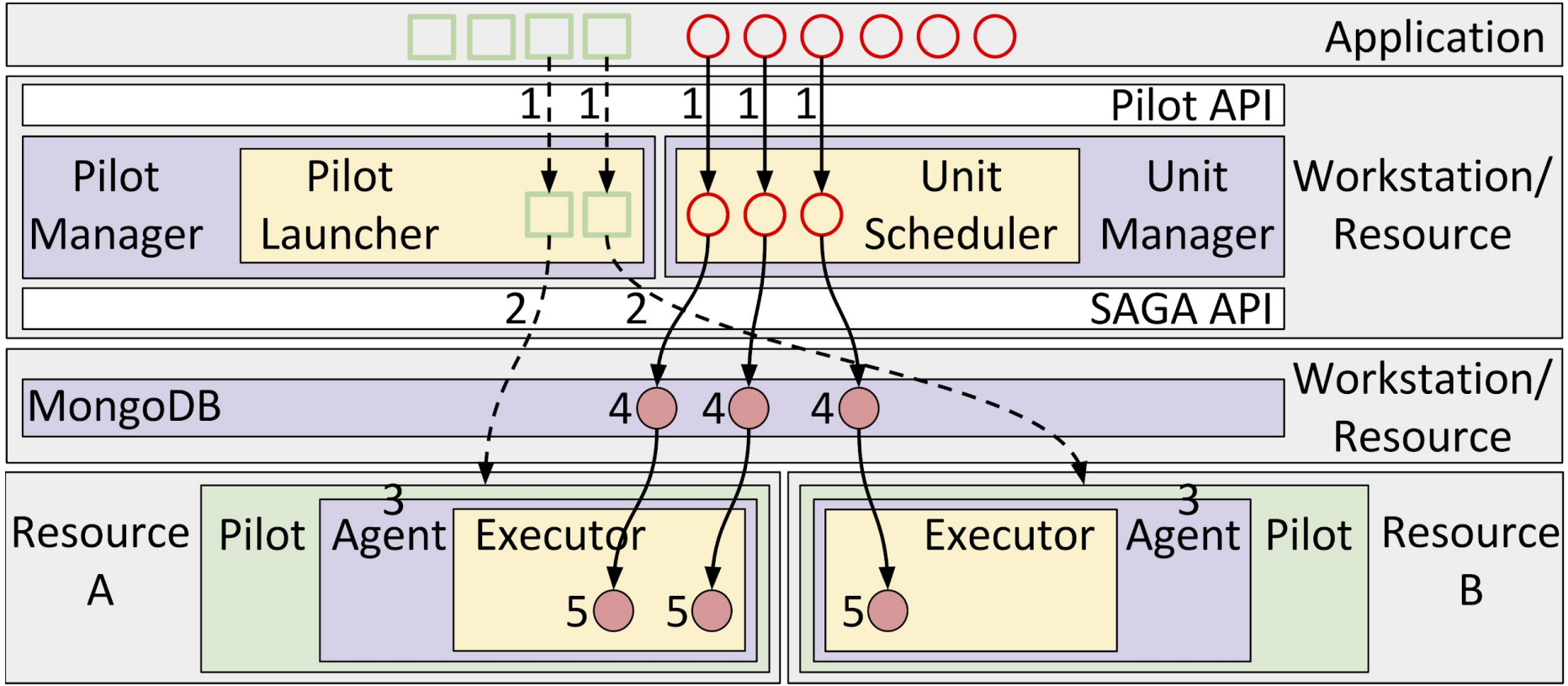


Developing Workflow Tools Using Building Blocks





RADICAL-Pilot: Execution Model



RADICAL-Pilot: Resource Utilization Performance (Titan)

RADICAL-Pilot on Leadership Class Machine

- Can we get performance agnostic of batch queue systems and MPI flavour?
  - LSF, PBS, SLURM, … ?
  - MVAPICH, … MPI flavours?
- **PMI-X: P**rocess **M**anagement **I**nterface for E**X**ascale https://github.com/pmix/pmix/wiki
  - **PRRTE: P**MI-X **R**eference **R**un**T**ime **E**nvironment https://github.com/pmix/prrte
- PMI used by MPI implementations, batch system
- Private DVM, concurrent tasks
- Pros: heterogeneous tasks (as with JSRUN), (potentially) fast, **portable**
- Cons: Emerging official support

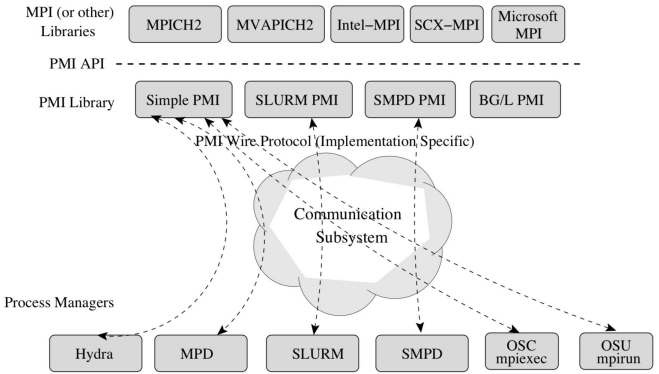
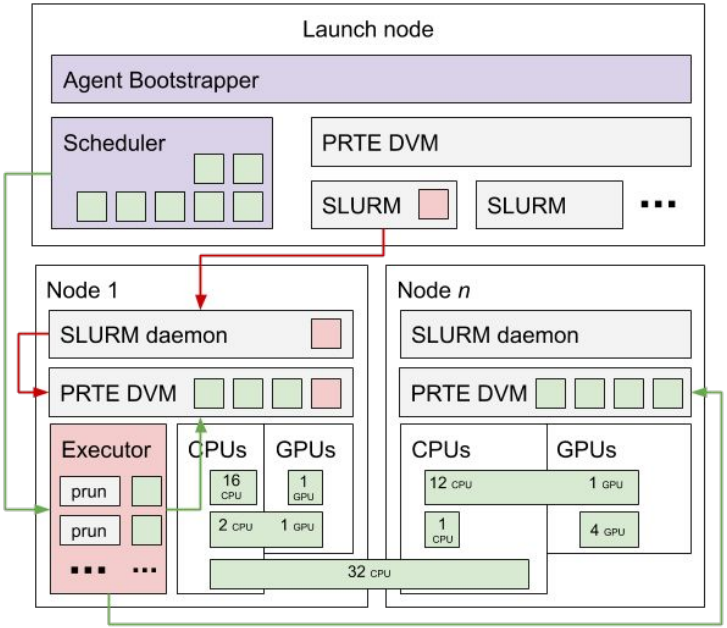


Fig. 1. Interaction of MPI and the process manager through PMI

PMI: A Scalable Parallel Process-Management Interface for Extreme-Scale Systems, Balaji et al

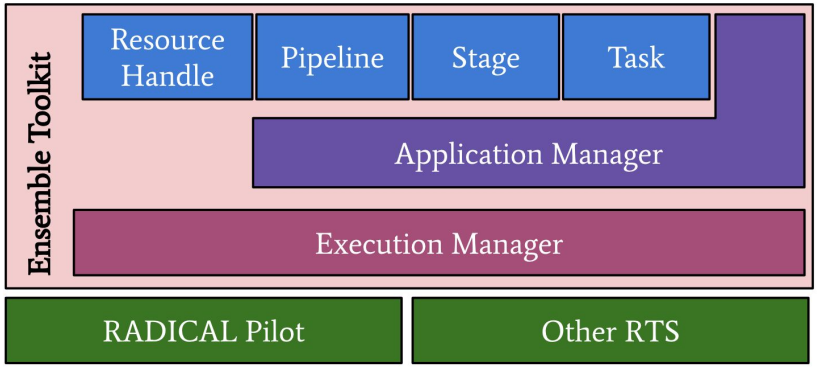
RADICAL-Pilot on Leadership Class Machine

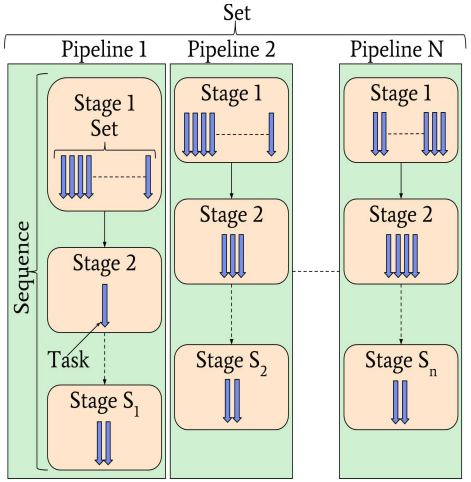
- Can we get performance agnostic of batch queue systems and MPI flavour?
  - LSF, PBS, SLURM, … ?
  - MVAPICH, … MPI flavours?
- **PMI-X: P**rocess **M**anagement **I**nterface for E**X**ascale https://github.com/pmix/pmix/wiki
  - **PRRTE: P**MI-X **R**eference **R**un**T**ime **E**nvironment https://github.com/pmix/prrte
- PMI used by MPI implementations, batch system
- Private DVM, concurrent tasks
- Pros: heterogeneous tasks (as with JSRUN), (potentially) fast, **portable**
- Cons: Emerging official support



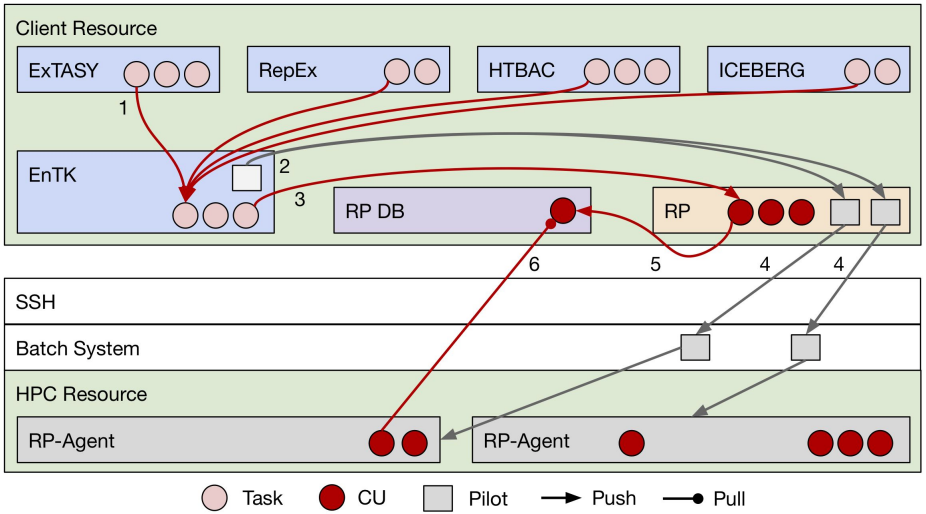
EnTK: Building Block for Ensemble based Applications

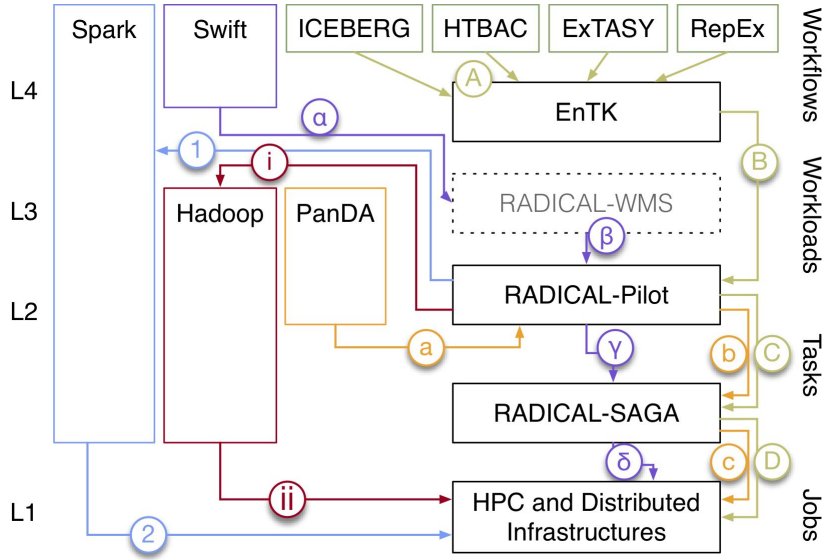
- **Ensemble-Toolkit (EnTK):** Promote ensembles as a first-class programming and execution entity.
  - (i) Facilitate expression of ensemble based applications, (ii) manage complexity of resource acquisition, and (iii) task execution.
- **Architecture:**
  - User facing components (blue); Workflow management components (purple); Workload management components (red) via runtime system (green)
- **PST Programming Model:**
  - **Task:** an abstraction of a computational process and associated execution information
  - **Stage:** a set of tasks without dependencies, which can be executed concurrently
  - **Pipelines:** a list of stages, where stage “i” can be executed after stage “i−1”





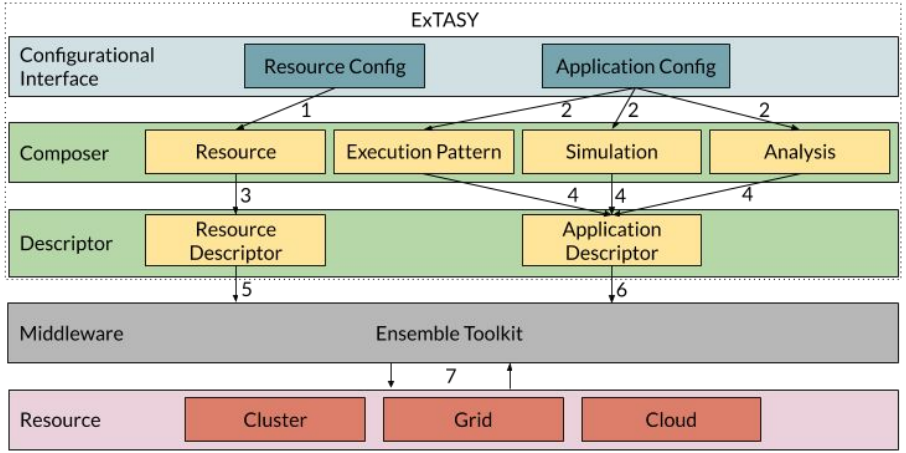
Software Systems Challenge: Extensibility & Performance

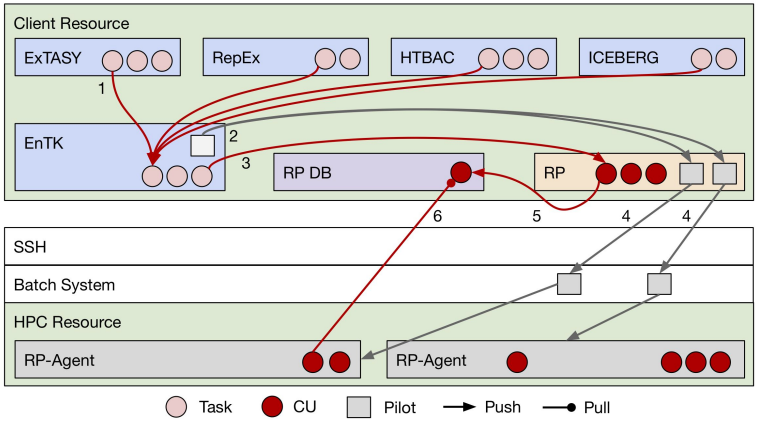


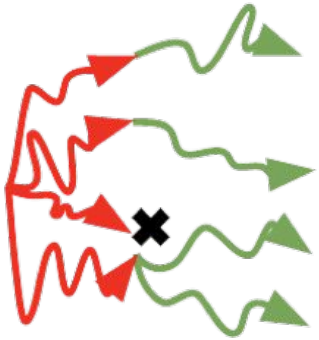


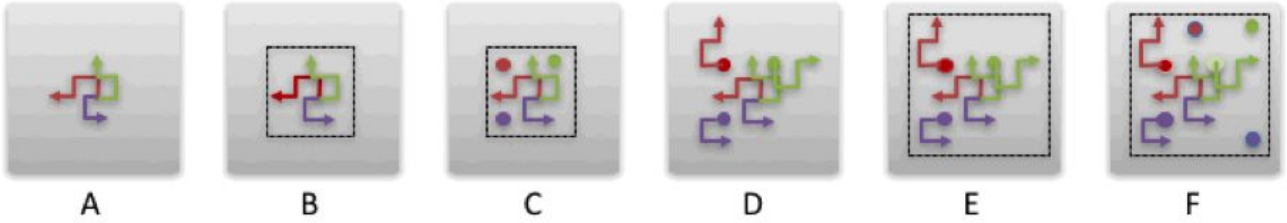
Middleware Building Blocks for Workflow Systems https://arxiv.org/abs/1903.10057

Case Study: Advanced Sampling

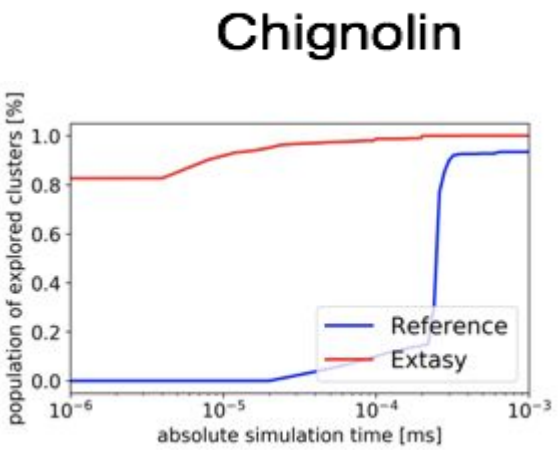


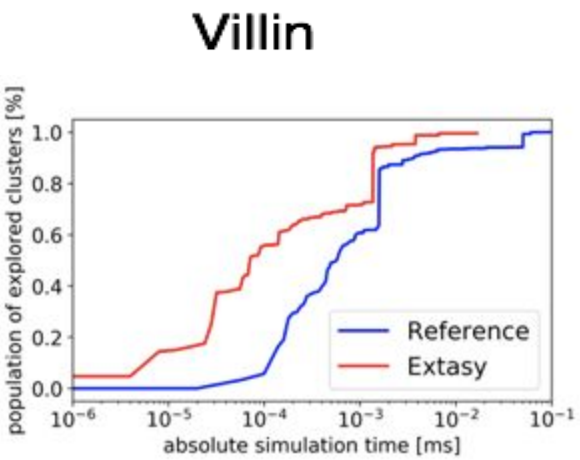


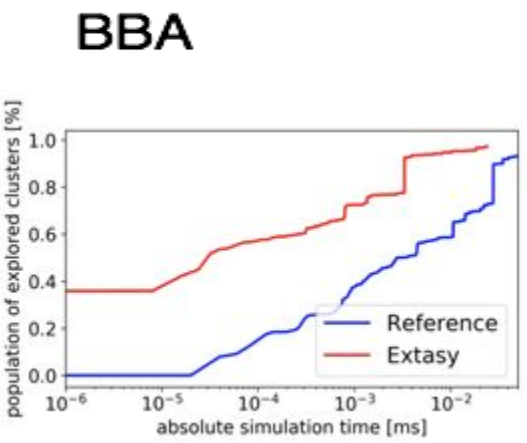


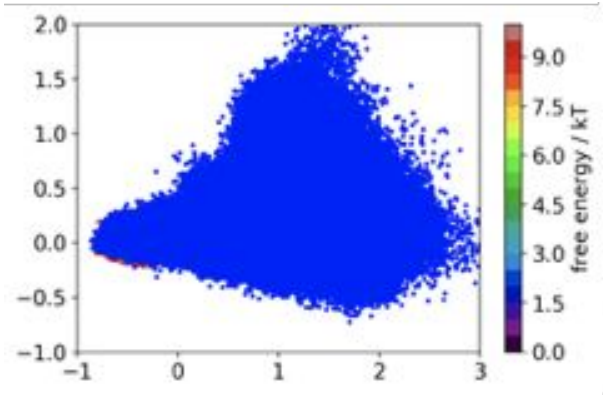


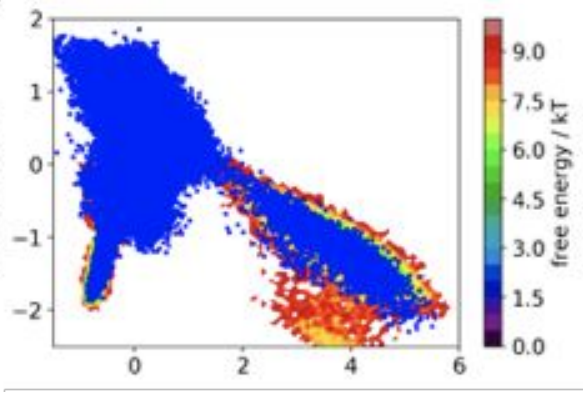
ExTASY (MSM) vs Conventional MD

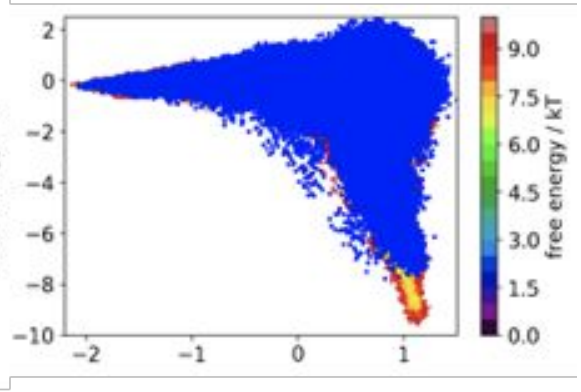




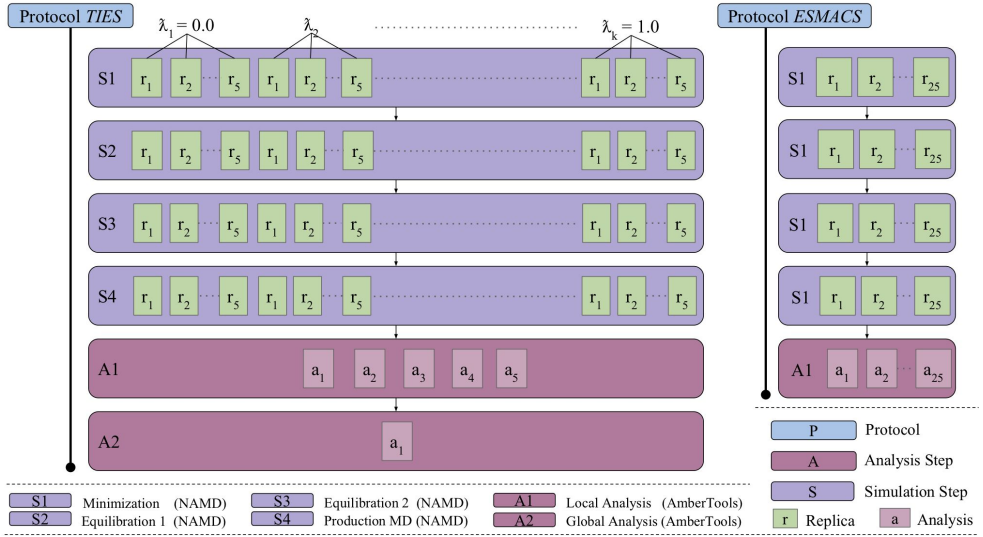


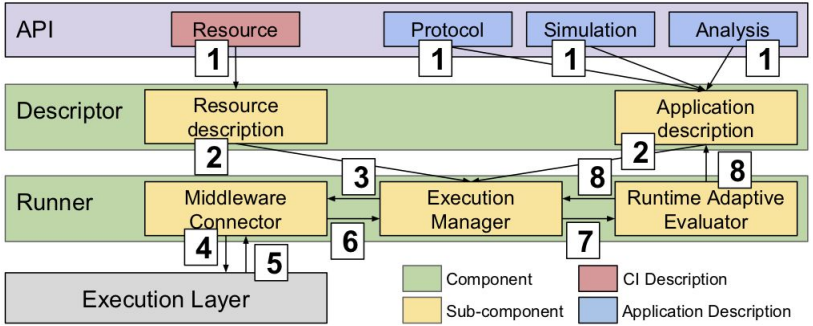


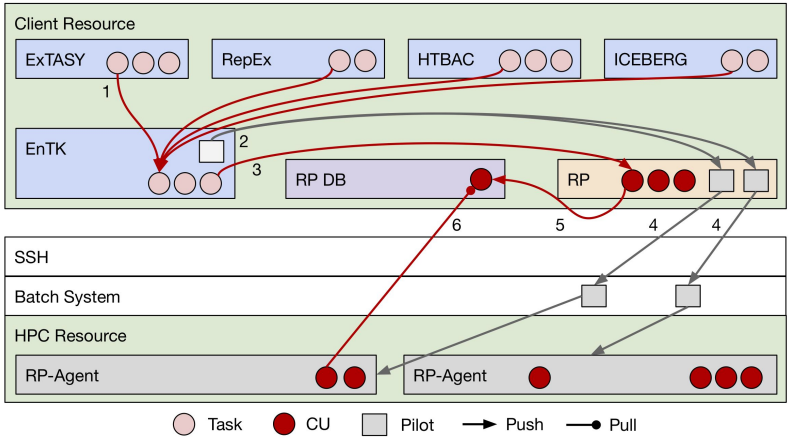




High-Throughput Binding Affinity Calculator

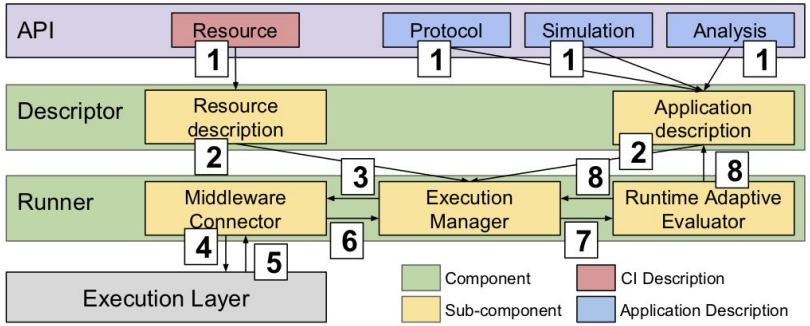






Advantage of Adaptive Ensemble Algorithms

- Original study GSK and UCL
  - 16 drug candidates, BRD4 inhibitor
- Non-adaptive implementation of ESMACS and TIES protocols at scale on average:
  - ESMACS 10K core-hours
  - TIES 25K core-hours
- Millions of candidates to “hits” to leads
- Typical lead optimization involves 10,000 small molecule (drug) candidates
  - 250 Million Core Hours!!
- Without **specialized tools using building blocks,** otherwise sequential & separate



[1] S. Wan, A. P. Bhati, S. J. Zasada, I. Wall, D. Green, P. Bamborough, and P. V. Coveney. Rapid and reliable binding affinity prediction of bro- modomain inhibitors: a computational study. *J. Chem. Theory Comput.*, 13(2):784–795, 2017.

[2] J. Dakka et al., "Enabling Trade-offs Between Accuracy and Computational Cost: Adaptive Algorithms to Reduce Time to Clinical Insight," 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Washington, DC, USA, 2018, pp. 572-577. doi:10.1109/CCGRID.2018.00005

Advantage of Adaptive Ensemble Algorithms

- Original study GSK and UCL
  - 16 drug candidates, BRD4 inhibitor
- Non-adaptive implementation of ESMACS and TIES protocols at scale on average:
  - ESMACS 10K core-hours
  - TIES 25K core-hours
- Millions of candidates to “hits” to leads
- Typical Lead Optimization involves 10,000 small molecule (drug) candidates
  - 250 Million Core Hours!!
- Without **specialized tools using building blocks,** otherwise sequential & separate


18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2018)

Washington, DC
May 1 - 4, 2018

SCALE Challenge Award 2018

Winner

Jumana Dakka, Kristof Farkas-Pall, Vivek Balasubramanian, Matteo Turilli, Shunzhou Wan, David W. Wright, Stefan Zasada, Peter V. Coveney and Shantenu Jha

with

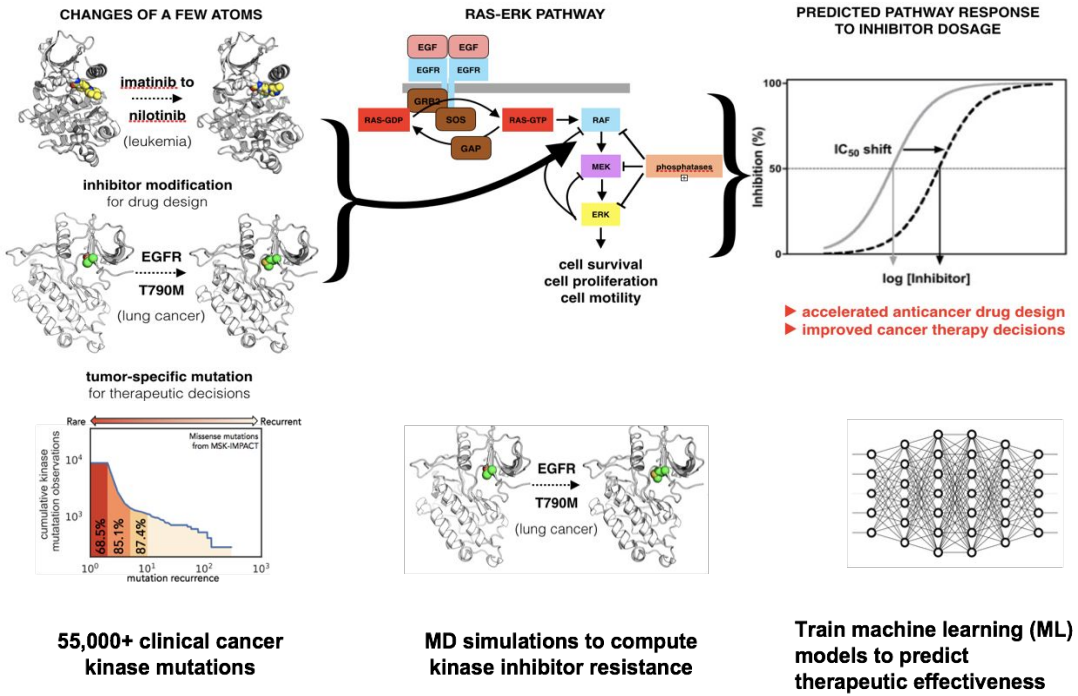
"Enabling Trade-off Between Accuracy and Computational Cost: Adaptive Algorithms to Reduce Time to Clinical Insight"

Sandra Gesing
Chair of IEEE SCALE 2018
May 2018

Amy Apon
Chair of IEEE SCALE 2018
May 2018

INSPIRE: Integrated (ML-MD) **S**calable **P**rediction of **RE**sistance

- Chemical space of drug design in response to mutations very large. 10K -100K mutations; too large for HPC simulations alone!
- Develop methods that use: (i) simulations to train machine learning (ML) models to predict therapeutic effectiveness; (ii) use ML models to determine which drug candidates to simulate.

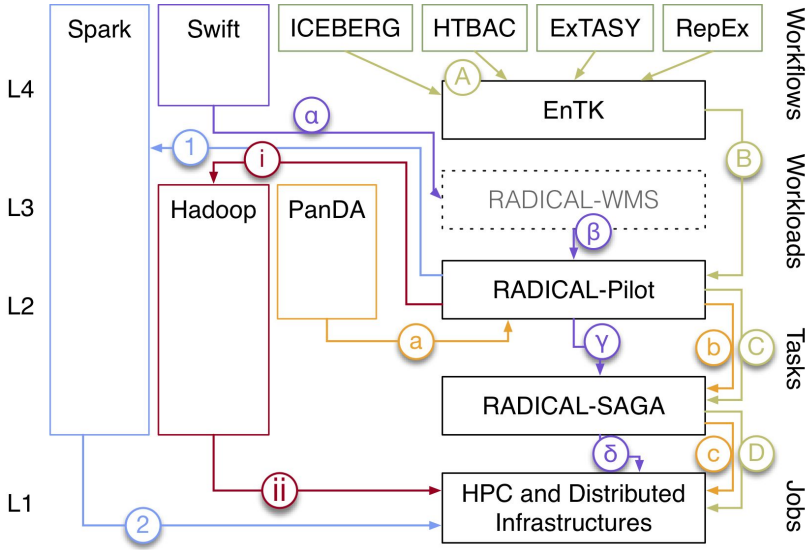


Early Science Project on NSF Frontera. DD Award on Summit.

A collaboration between BNL/Rutgers (Jha), Chicago (Stevens), Memorial Sloan Kettering (Chodera), UCL (Coveney)

Building Blocks as Software Systems

- **Different levels / types of Integration**
  - Domain Specific Workflows (DSW)
  - End-to-end Workflow Systems
  - Workload management systems
  - General purpose computing systems
- Can the community (developers, facilities, users..) provide Building Blocks for Workflow Systems as components of “Open Workflow Systems” ?
  - Not every scenario, but the most common and HPC relevant ..



Summary

- **Many advances in workflows, but many challenges in workflow systems**
  - Landscape is changing in many ways; also need focus on systems developers
- **RADICAL-Cybertools: BB for Ensemble Computational Model**
  - Supports a range of functionality, use cases and platforms
  - RCT designed for separation of performance and functional extensibility
- **BB Approach: Promise but many open questions**
  - *Qualitative:* Need more formally rigorous definitions building block? Differentiability?
  - *Quantitative:* Develop a hypothesis & validation of how/when/if BB are more scalable and sustainable than monolithic approaches?
  - *Best Practice:* A formal understanding of granularity, type and how domain specific?
- **BB for Workflow Systems as components of community "Open Workflow" ?**
  - A plausible outcome from a systems approach ?
  - Social and technical issues, also functional composition address "which" to use?

Thank you!