DYFLOW: A flexible framework for orchestrating scientific workflows on supercomputers

Swati Singhal (University of Maryland)
Alan Sussman (University of Maryland)

Matthew Wolf (Oak Ridge National Lab)
Kshitij Mehta (Oak Ridge National Lab)
Jong Choi (Oak Ridge National Lab)

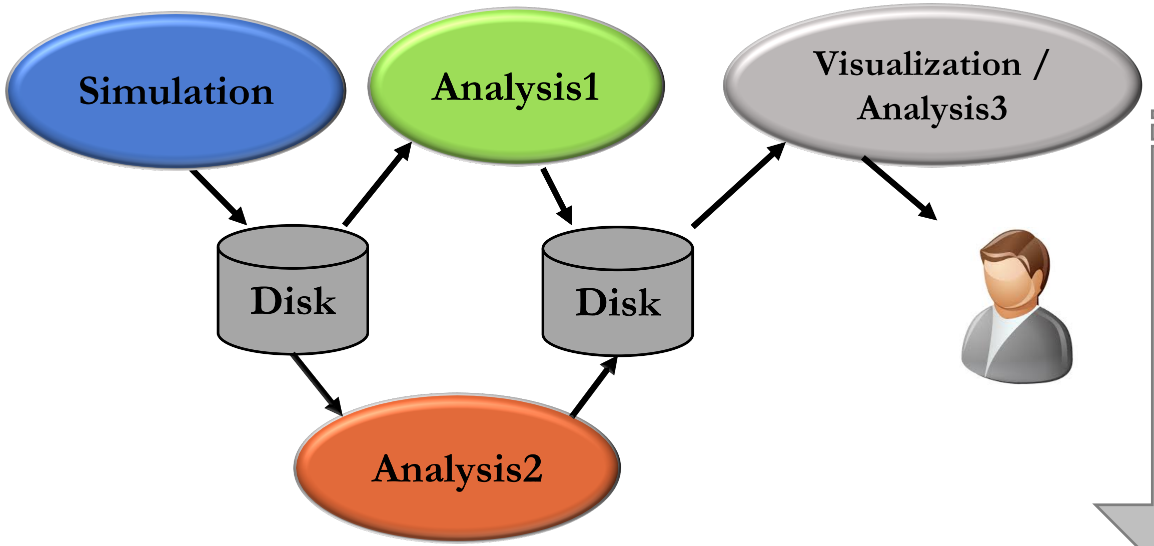
Outline

- **Motivation**
- DYFLOW Approach
- DYFLOW capabilities : demonstration with example
- Future Work

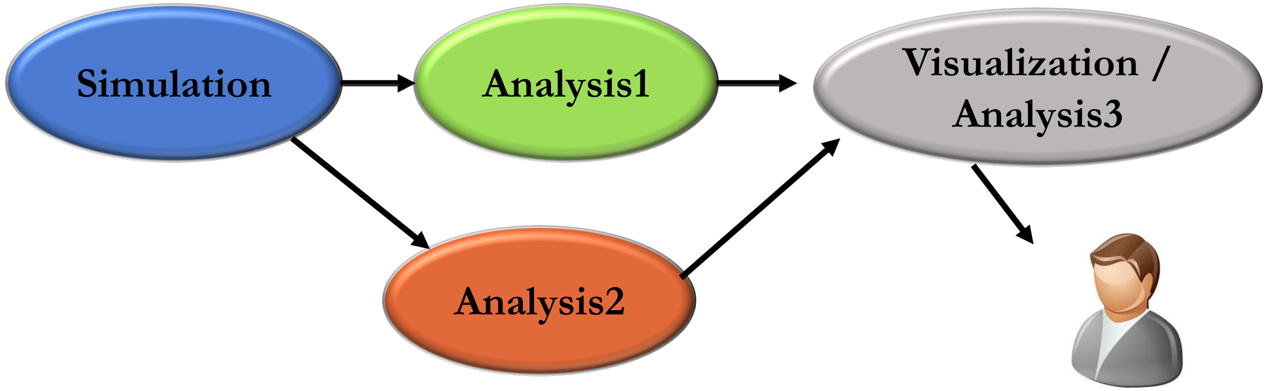
Modern scientific workflows are complex :

Support for adaptive workflows is important now more than ever

Traditional workflows – loosely coupled



In situ workflows – tightly coupled



- Involves complex tasks(e.g., graph and machine learning based approaches)
- Large-scale coupling involving petabytes of data
  ✘ I/O becomes the performance bottleneck

Paradigm shift towards *in situ* workflows

✔ Memory to memory transfers replace disk writes

- ▶ On-node memory buffers, e.g. **ADIOS**
- ▶ Off-node dedicated servers, e.g. **Dataspaces**

Dynamic challenges

- Dealing with interference since tasks share compute nodes
- Dealing with changes in data-flow rate
- Automating handling of data-driven events
- Handling failures

…

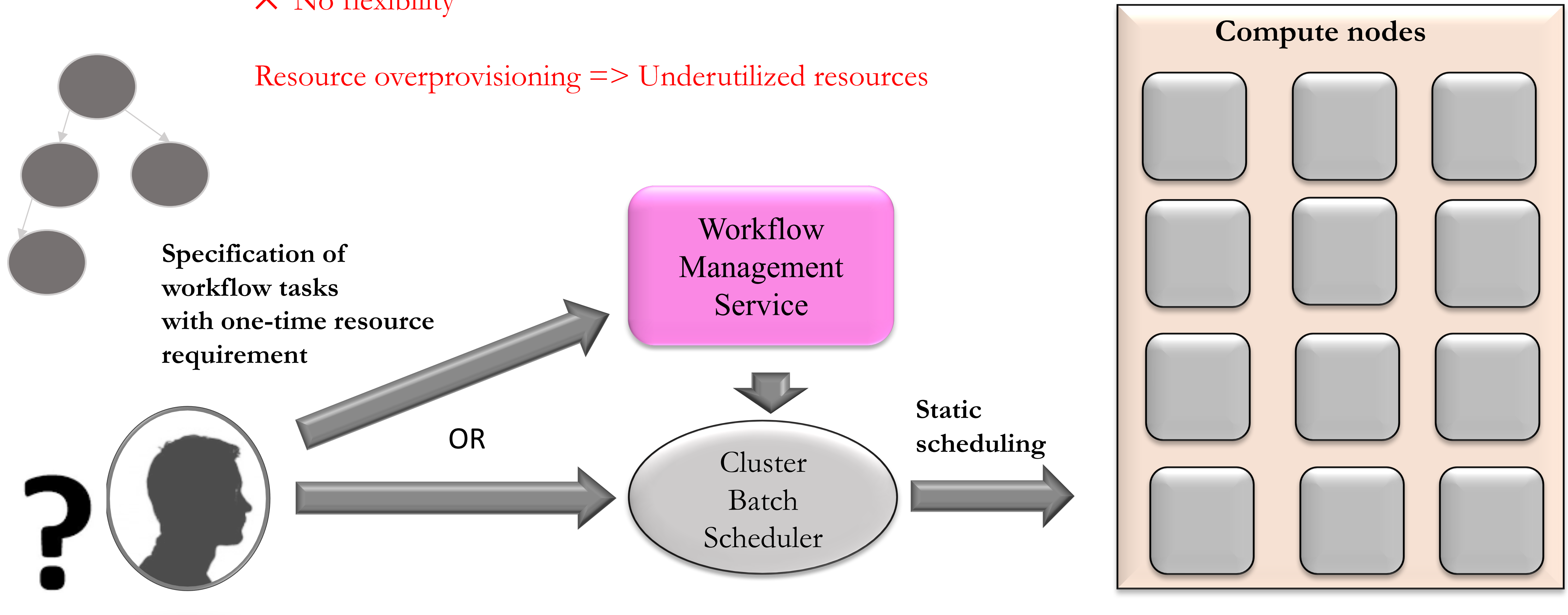
Modern scientific workflows are complex :

Static nature of resource assignment on supercomputers

- ✕ No dynamic resource assignment
- ✕ No or limited support for handling data-driven events or failures
- ✕ No flexibility

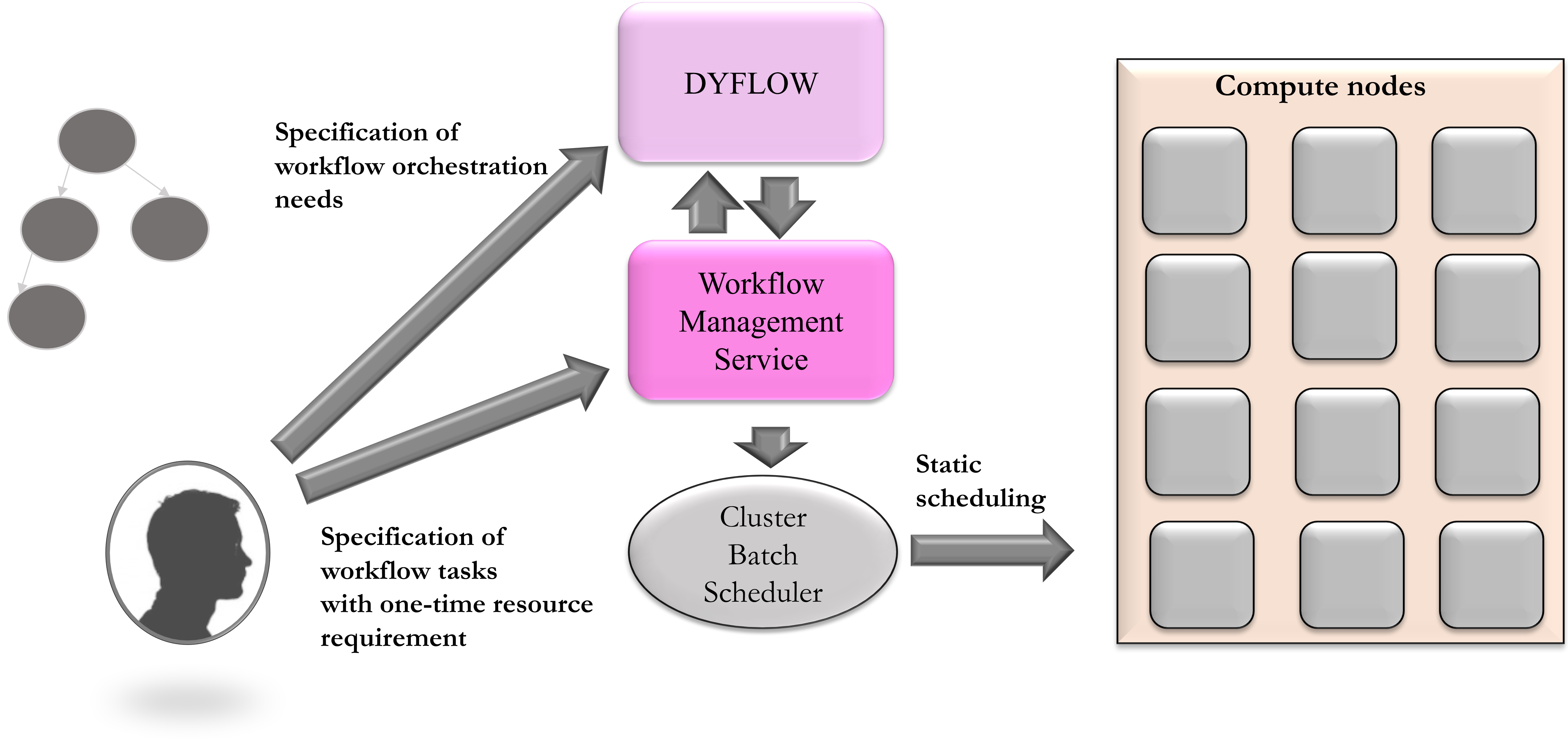
Resource overprovisioning => Underutilized resources

No orchestration to adapt resources based on changing requirements of workflow tasks at runtime



DYFLOW

A workflow orchestration service that reuses existing support from widely used workflow management services



Outline

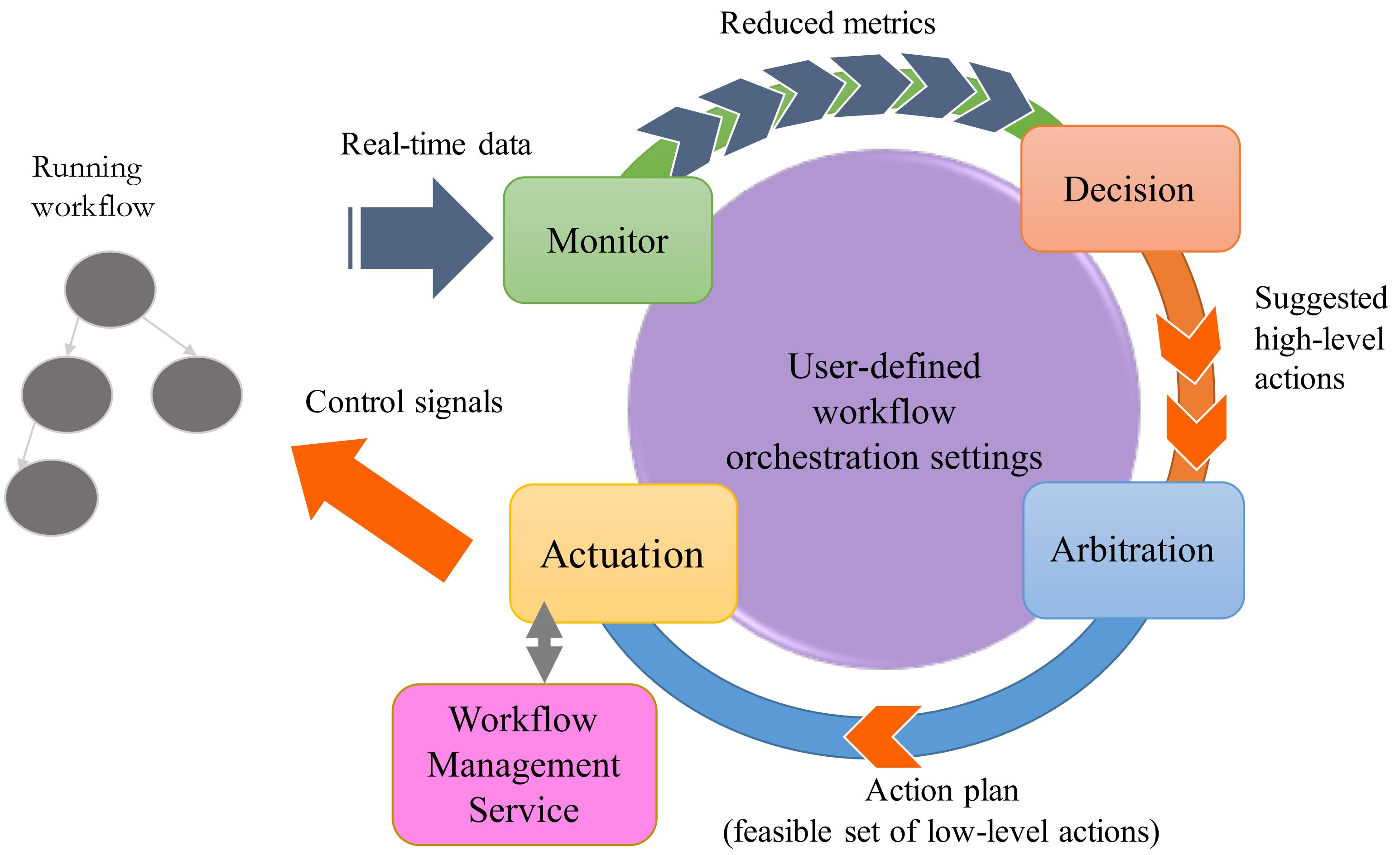
- Motivation
- **DYFLOW Approach**
- DYFLOW capabilities : demonstration with example
- Future Work

DYFLOW: Dynamic model

Features:

Flexibility:
Provides comprehensive constructs at different stages that enable end-users to configure and automate workflow orchestration as a user desires

Ease of expression and **reusability** across workflows tasks and parallel architectures



DYFLOW: Dynamic model (Monitor)

User defined sensors

What data to gather?

Method to use for data gathering?

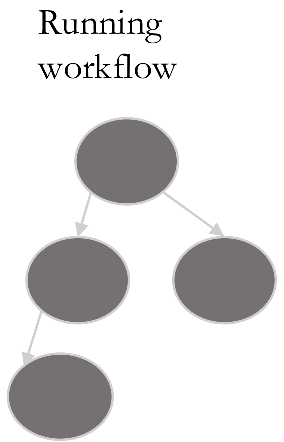
How to **group** and **reduce** data to form a metric?

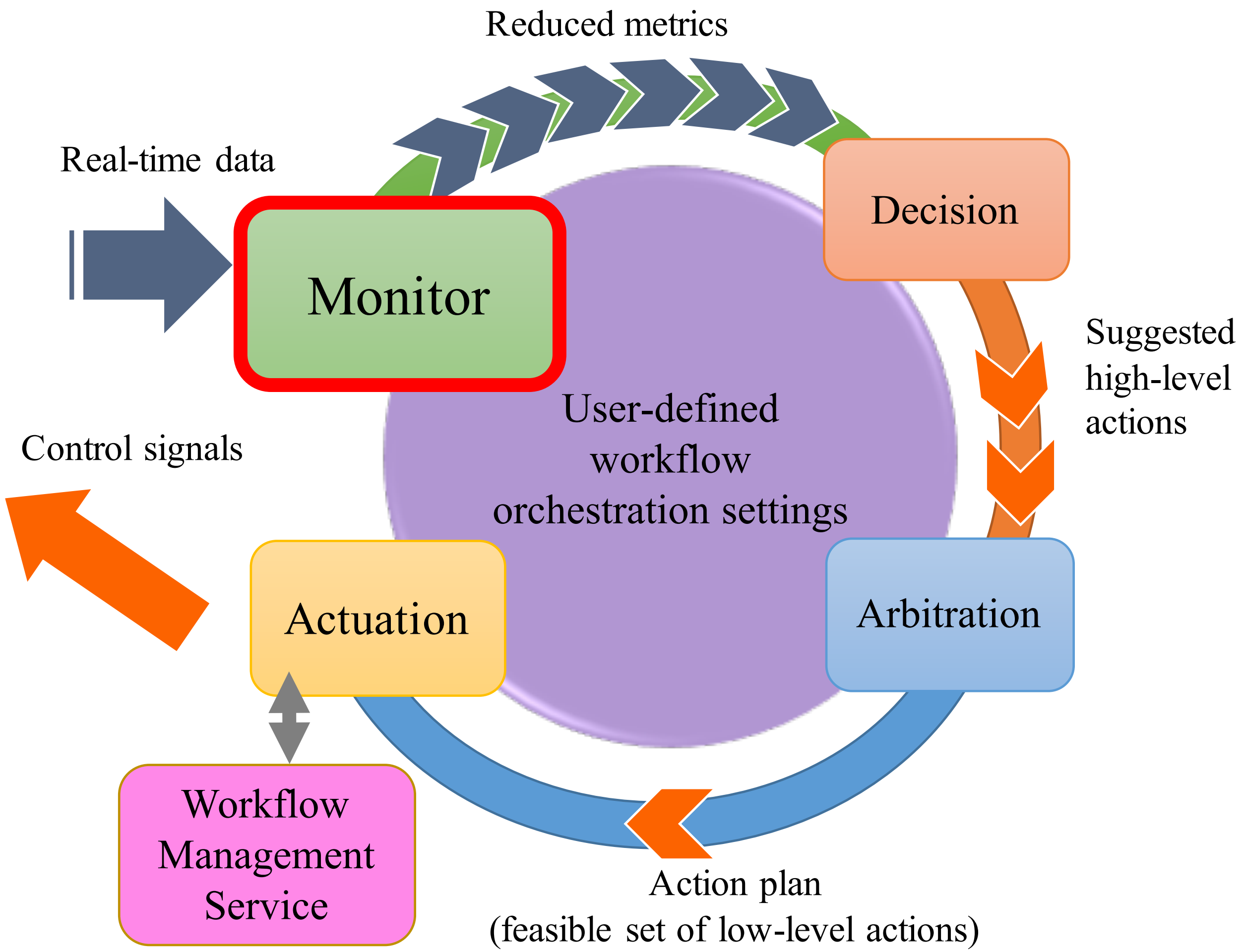
Framework:

Manages sensors -> set and reset sensors as task run and stop

Manage collection and reduction of distributed data at scale

Support different types of data **gathering,** grouping and reduction methods





DYFLOW: Dynamic model (Decision)

User defined policies

What evaluation condition determines an event of interest?

Suggested high-level actions?

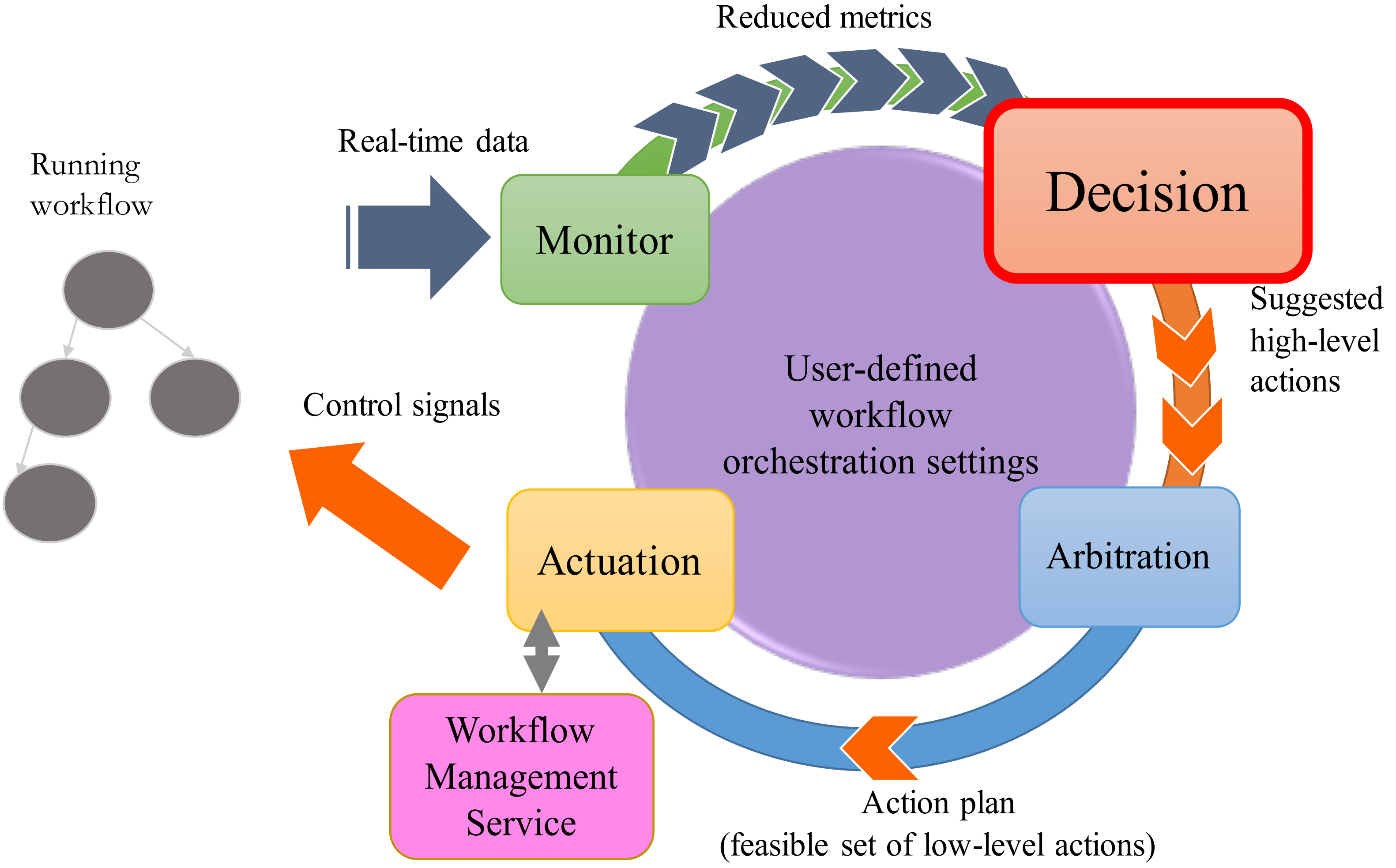
Frequency of evaluation?

Framework:

Manages policies

Trigger evaluation conditions at the desired frequency

Support different types of evaluation conditions and high-level actions



DYFLOW: Dynamic model (Arbitration)

User defined rules

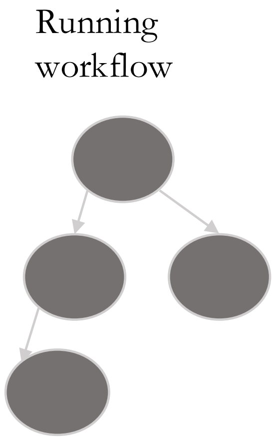
End-users can assist in

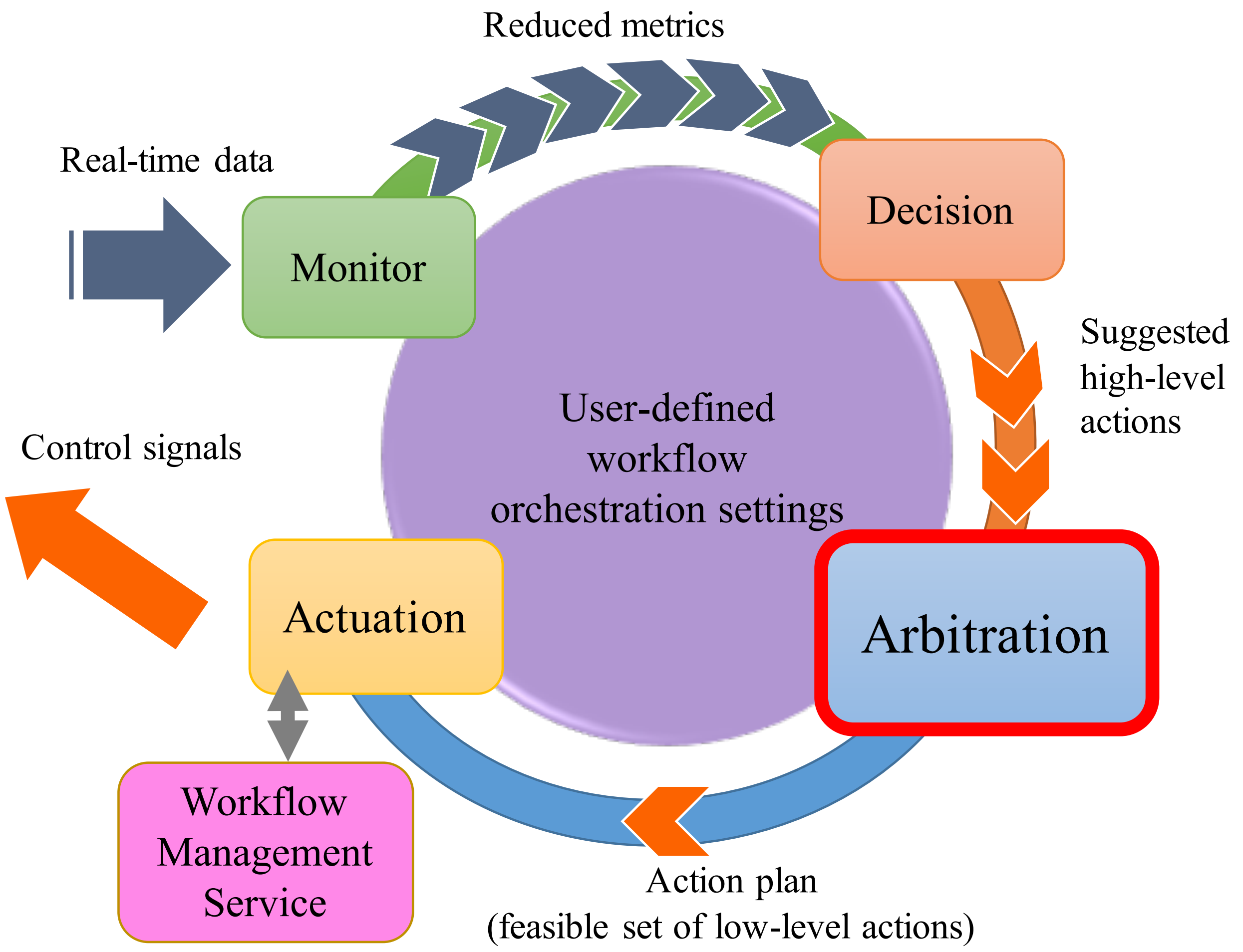
- Conflict resolution by setting **task priorities**
- Identification of dependent actions by declaring **task dependencies**

Framework:

- Resolves conflicting high-level actions
- Identifies dependent actions
- Maps high-level to low-level actions
- Determine resource assignments
- Suspend processing incoming suggestions to avoid repeatedly applying the same actions

………..





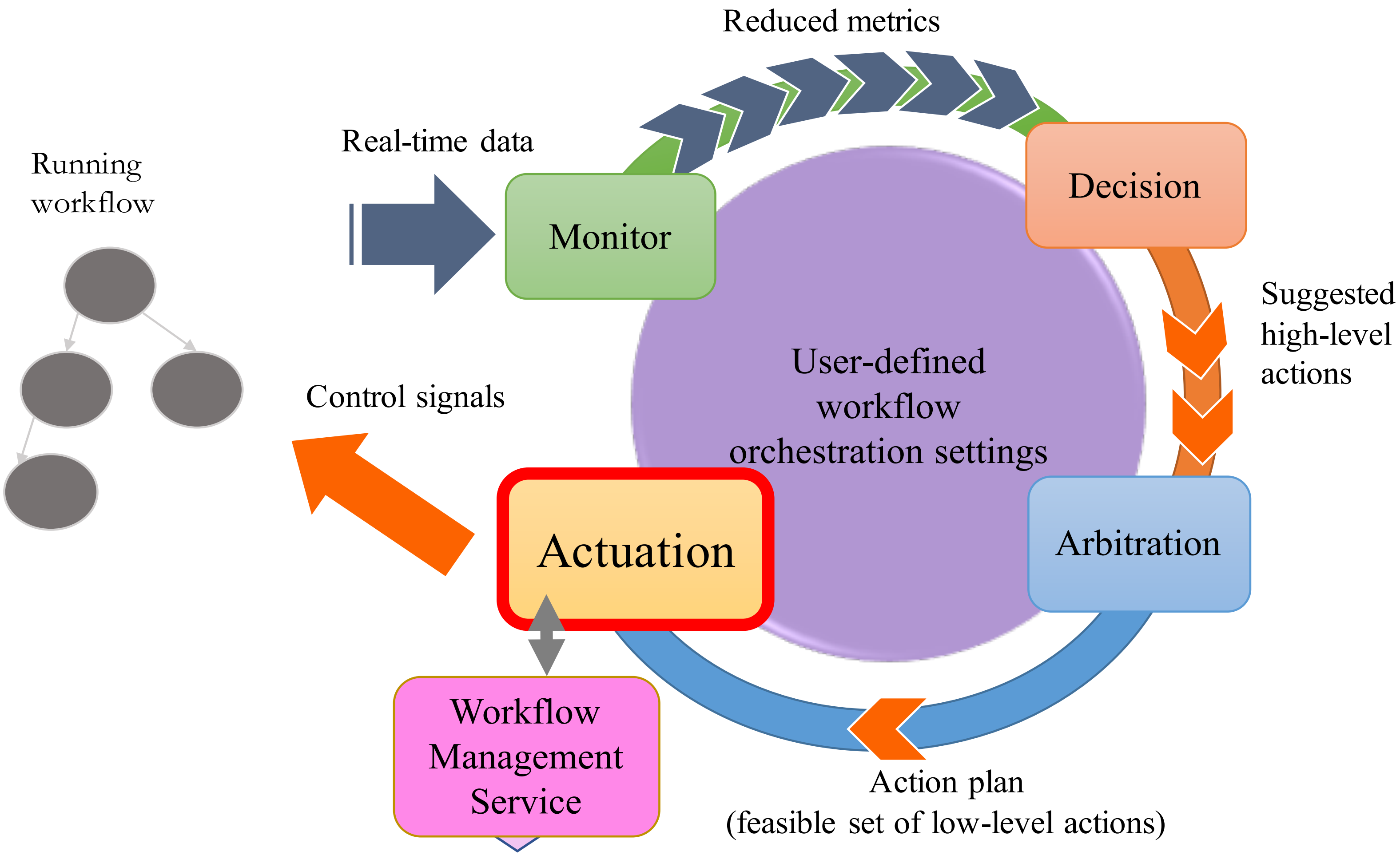
DYFLOW: Dynamic model(Actuation)

Workflow developers:

- Extend DYFLOW via plugin interface to enable Actuation to invoke the workflow management services

Framework:

- Provide wrappers for plugin interfaces with the services (low-level actions) requested by the Actuation. E.g. stopping the tasks



- Stop/Start or relaunch tasks with a new resource assignment
- Inquire about current resources assignment
- Inquire about resource status
- ....

Outline

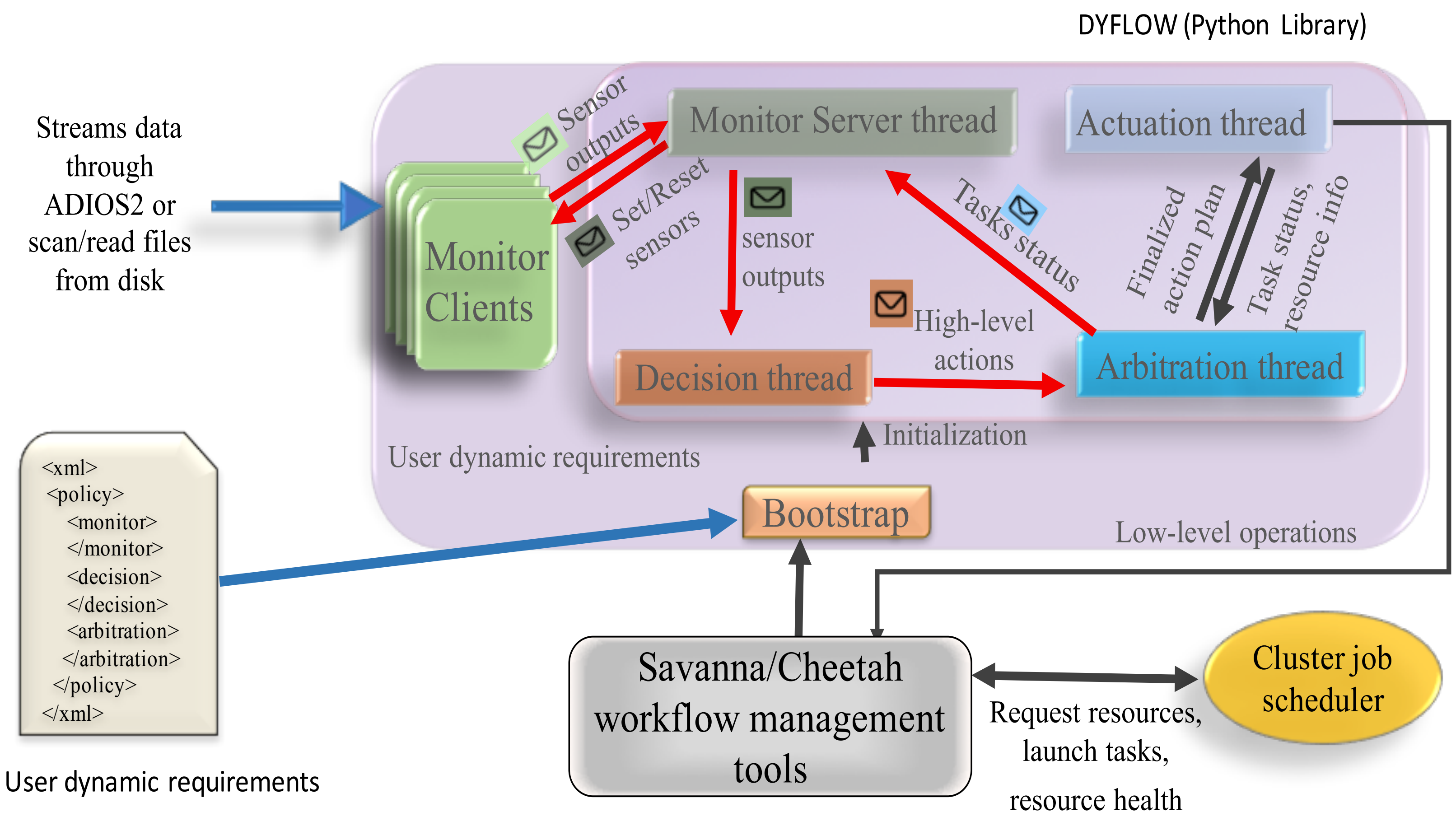
- Motivation
- DYFLOW Approach
- DYFLOW capabilities : demonstration with example
- Future Work

DYFLOW: IMPLEMENTATION

Capabilities

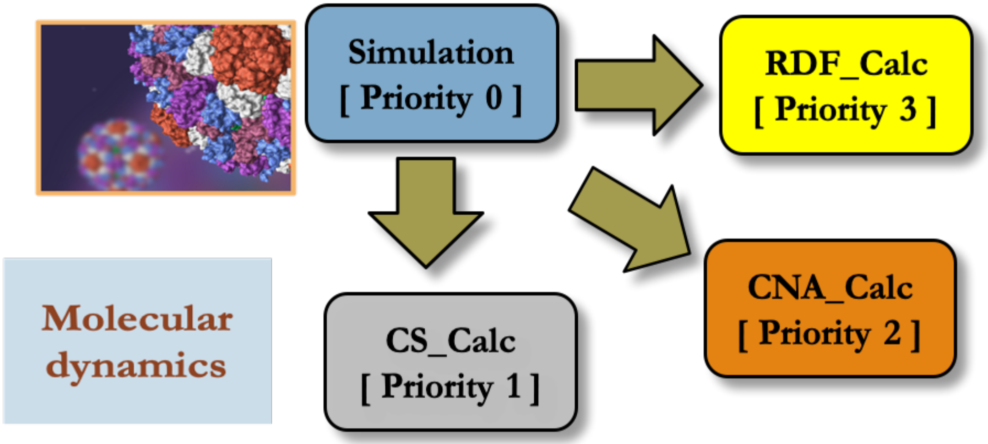
- Improve Throughput
- Improve Performance
- Handle Node Failure
- Handle data-driven Events

…



DYFLOW: Handling Node Failure

LAMMPS



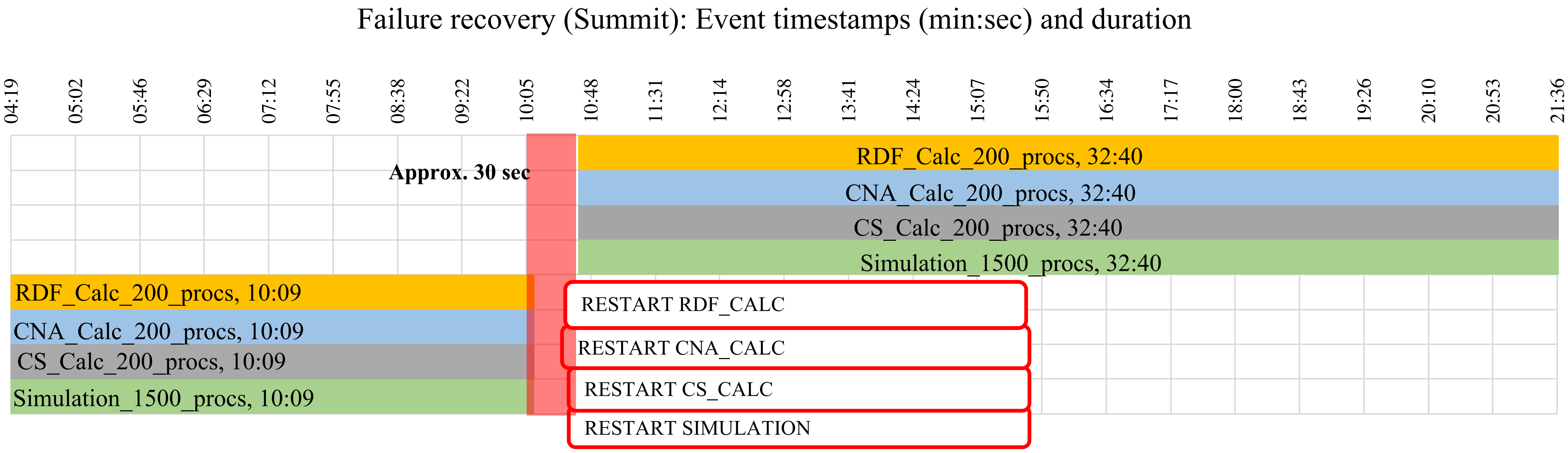
Tightly coupled Workflow

Summit
A high-end supercomputer at Oak Ridge National Lab

```
<sensor id="STATUS" type="ERRORSTATUS">
        <group-by>
                      <group granularity="task" reduction-operation="FIRST"/>
                 </group-by>
 </sensor>
```

```
<policy id="RESTART_ON_FAILURE" >
      <eval operation=”GT" threshold=”128" />
      <sensors-to-use>
             <use-sensor id="STAUS" granularity="task">
             </use-sensor>
      </sensors-to-use>
      <action> RESTART </action>
      <frequency seconds=”30”/>
</policy>
```

DYFLOW: Handling Node Failure



- Workflow restart after node failure using additional node in the allocation from the job scheduler
- Simulation restarts from the last checkpoint
- Time to restart reflects the decision frequency of 30 sec

Outline

- Motivation
- DYFLOW Approach
- DYFLOW capabilities : demonstration with example
- **Future Work**

Future work

- Improving Arbitration stage
  - To manage the heterogeneous resources assignments at runtime
  - Use machine learning to make better arbitration decisions, based on history, to improve conflict resolution, policy priorities, etc.

- Improving Actuation stage
  - Exploring controllable actions that allows users to dynamically alter running workflow, for instance, controlling what and how data flows between tasks.

DYFLOW: A flexible framework for orchestrating scientific workflows on supercomputers

To summarize ...

- DYFLOW is a generic framework that automates the orchestration of scientific workflows on supercomputers based on user-defined criteria
- Provides end-users with comprehensive and easy-to-use constructs to express and customize the orchestration of their workflow at different stages of runtime management
- Supports adaptive workflows with various capabilities to improve throughput and performance, handle failures and data-driven events
- Integrates with existing workflow management services and utilizes their workflow management support

Questions?

Swati Singhal
(University of Maryland)
swati@cs.umd.edu