Performance Architecture within ICENI

Dr Andrew Stephen McGough

Laurie Young, Ali Afzal, Steven Newhouse and John Darlington

London e-Science Centre

Department of Computing, Imperial College London

Outline

- Overview of ICENI
- Performance Framework
- Example
- Conclusion

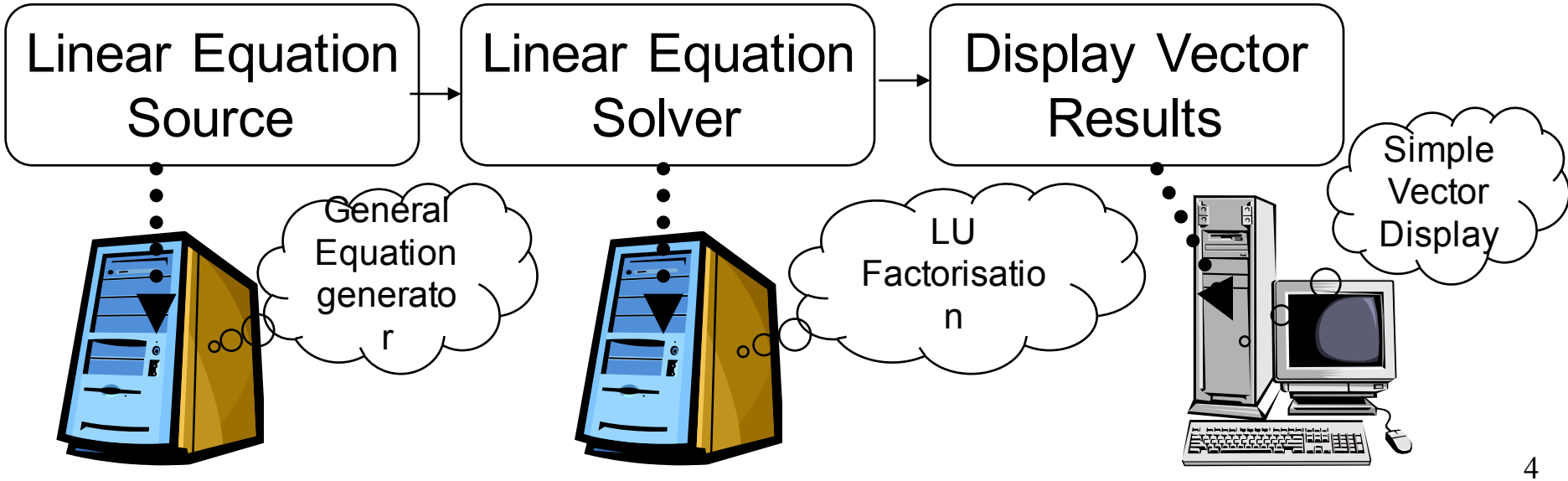
ICENI: Imperial College e-Science Network Infrastructure

- Collect and provide relevant Grid Meta-Data
- Pluggable architecture
- Test Architecture for Grid Research
- Foundation for higher-level Services and Autonomous Composition
- Integrated Grid Middleware Solution
- Interoperability between architectures, APIs
- Added value layer to other middleware
- Usability: Interactive Grid Workflows
- Role and policy driven security
- ICENI Open Source licence (extended SISSL)

The Iceni, under Queen Boudicca, united the tribes of South-East England in a revolt against the occupying Roman forces in AD60.

Scheduling Workflows in ICENI

- Applications consist of a number of components linked together in a dataflow manner
- The abstract workflow needs to be mapped down to a set of component implementations which will run on resources



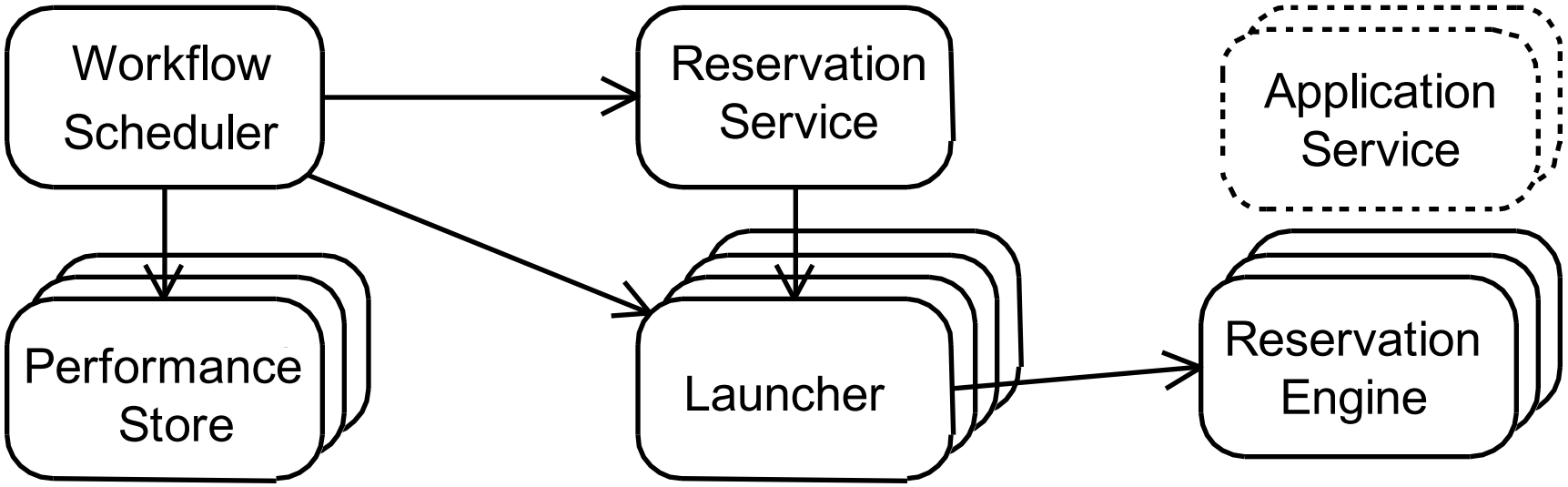
The Problem

- We have:
  - Multiple resources where components can run
  - Multiple implementations of components
  - The choice of one resource component mapping can affect the others
- User wants predictable performance
- How to choose the “best” mapping of workflow over resources to give user predictability?

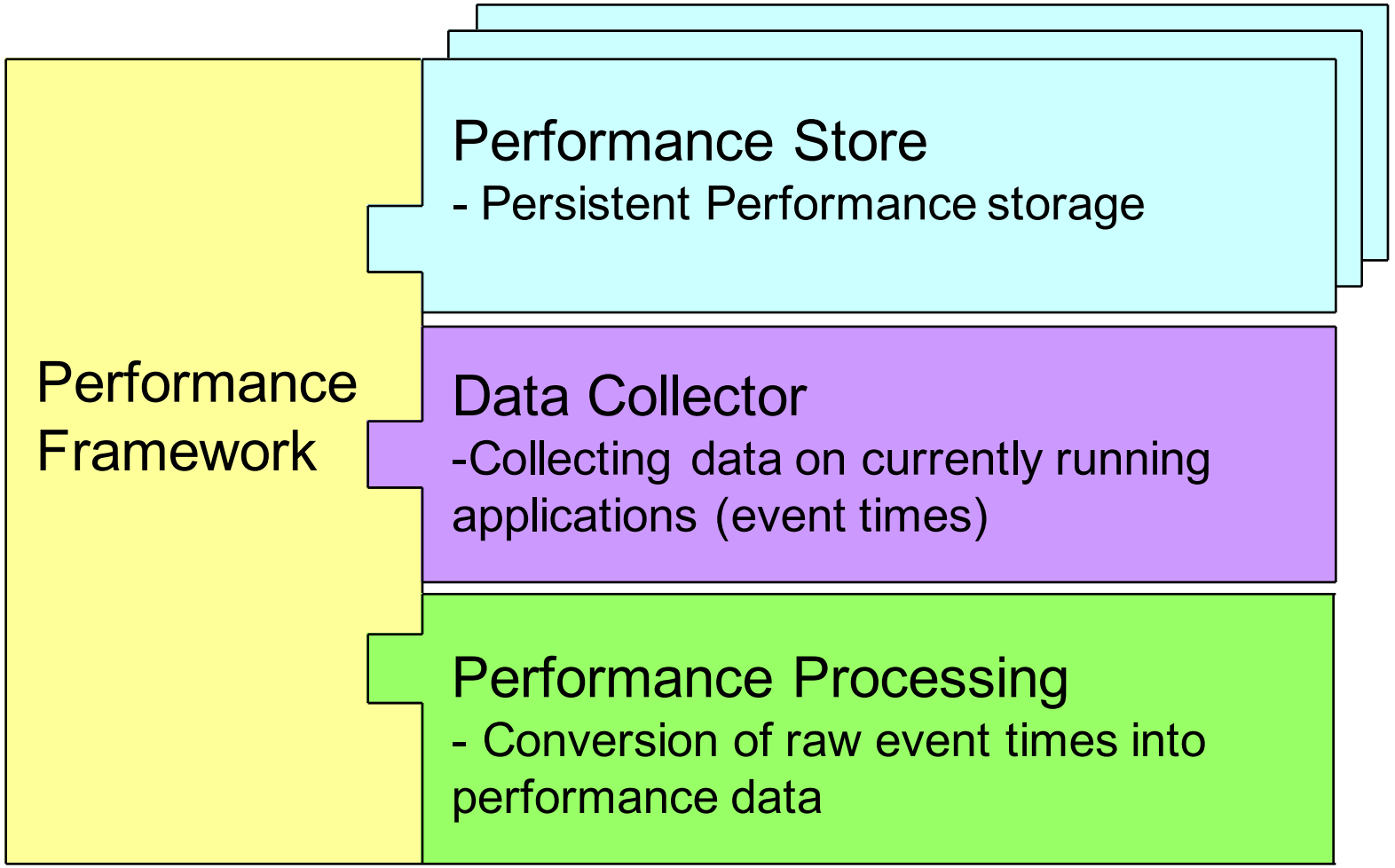
The Solution

- We need to take into account:
  - Execution Times of components on Resources
    - Performance Data
  - Inter-component effects of workflows
    - Workflow aware Schedulers
  - Workload on resources, making sure they are free when we need them
    - Reservation systems

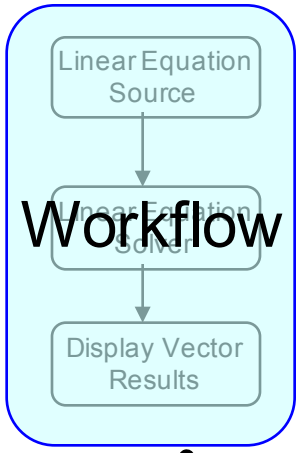
The Architecture



The Performance Repository Framework:



Collection of Performance Results



Event:
Start Linear Equation Source

Data Collector

Time	Event
12:00	Linear Equation Source Start
12:04	Send out Equations
12:03	Linear Equation Solver Start
12:05	Receive Equations
12:12	……..

Performance Processing

Performance Store

Storing Performance Data

- Multiple stores can be used in one framework

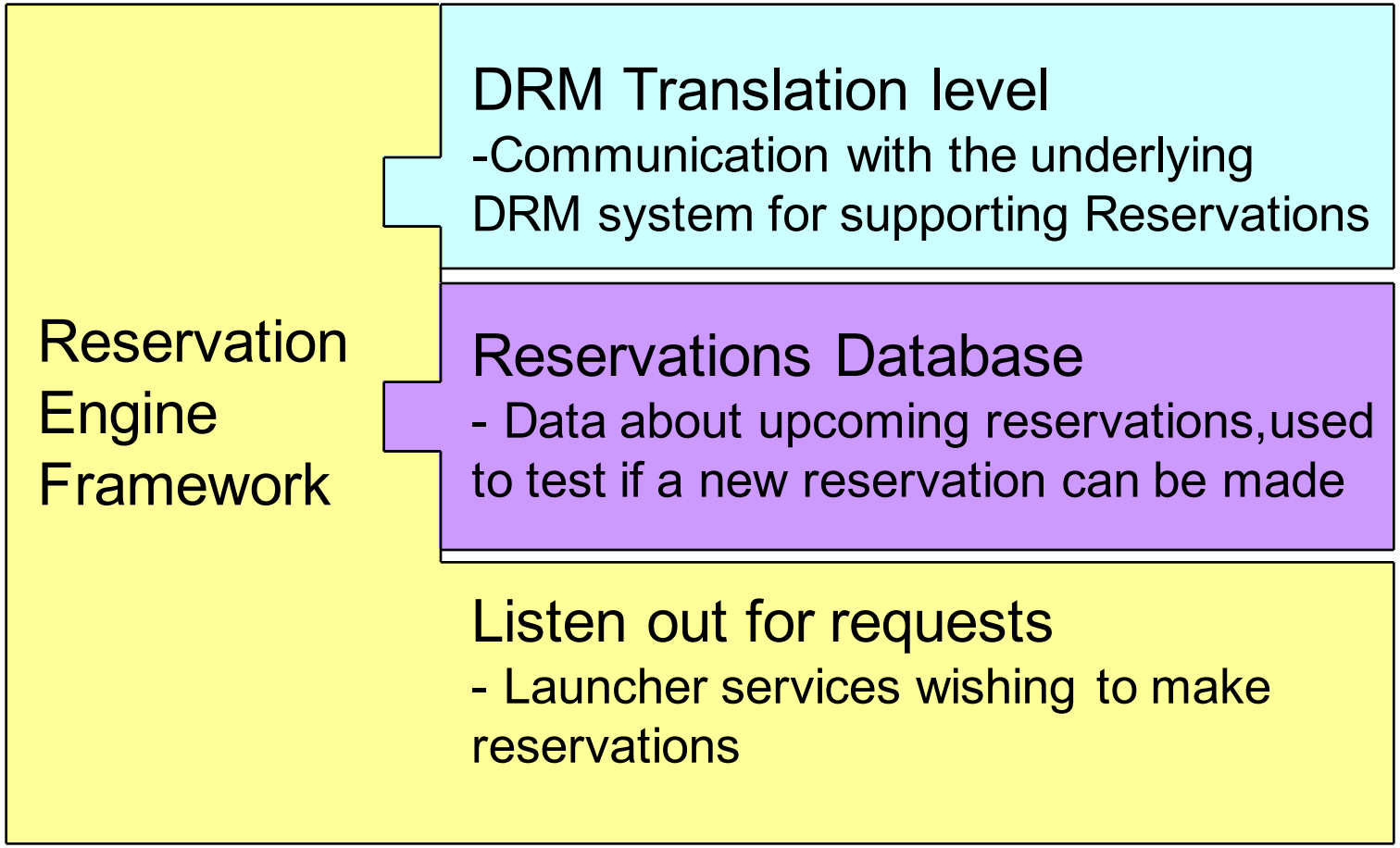
Performance Store

- The stores may be data stores or analytical models
- All assumed to be persistent
- Allows requests for predictions to be made
- New Data can be added to the stores
- Store data is aggregated together
  - based upon reliability of store data
    - Provided by the store

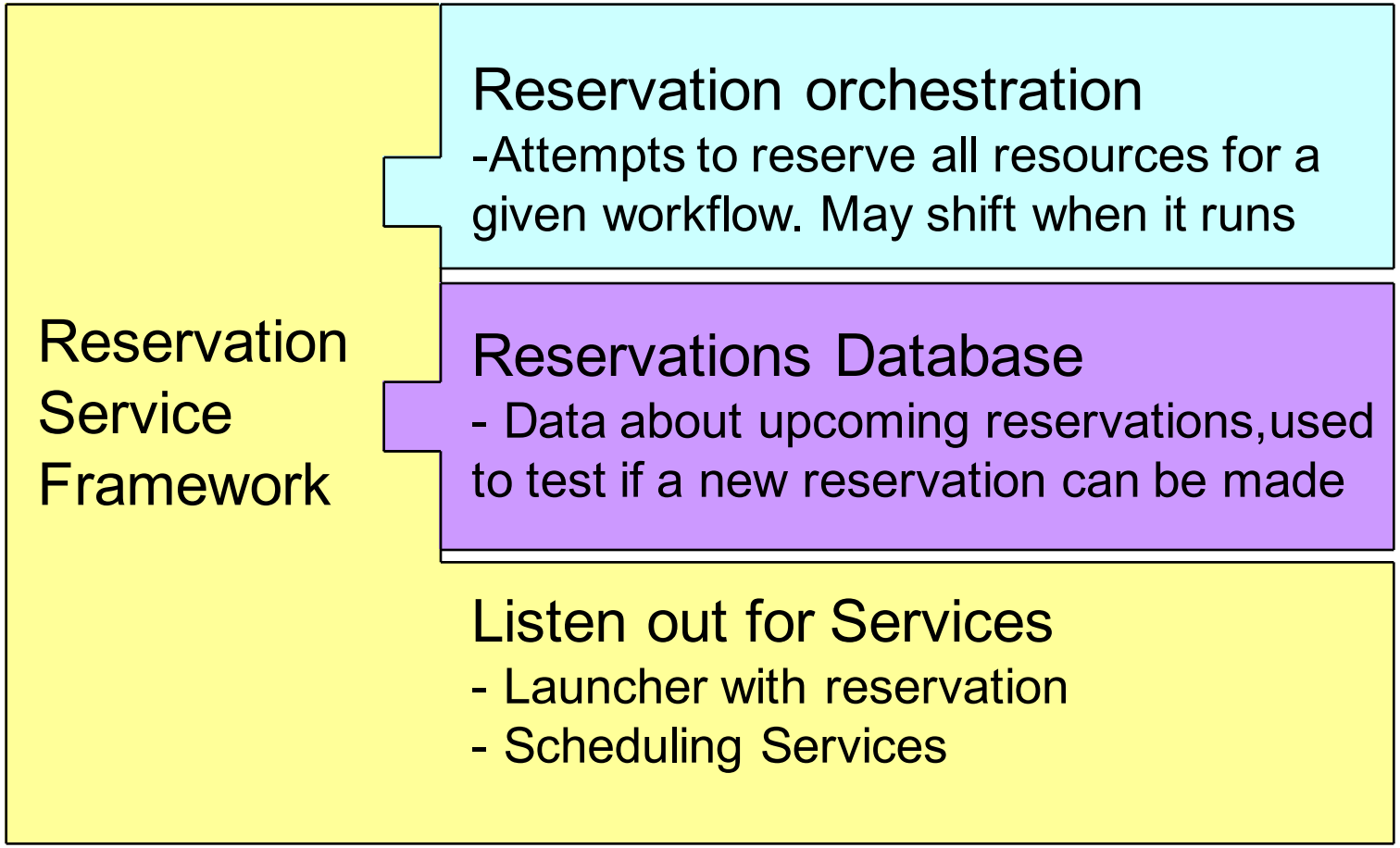
Using Performance Data

- Scheduler Builds up workflow graph with timings requested from the Performance Repository
- Timings are based on component implementation, resource, co-allocation count and other properties defined by the component implementer
- As the store will not contain all possible combinations of these properties regression is used to provide estimates for the missing values
  - This is an area of ongoing research

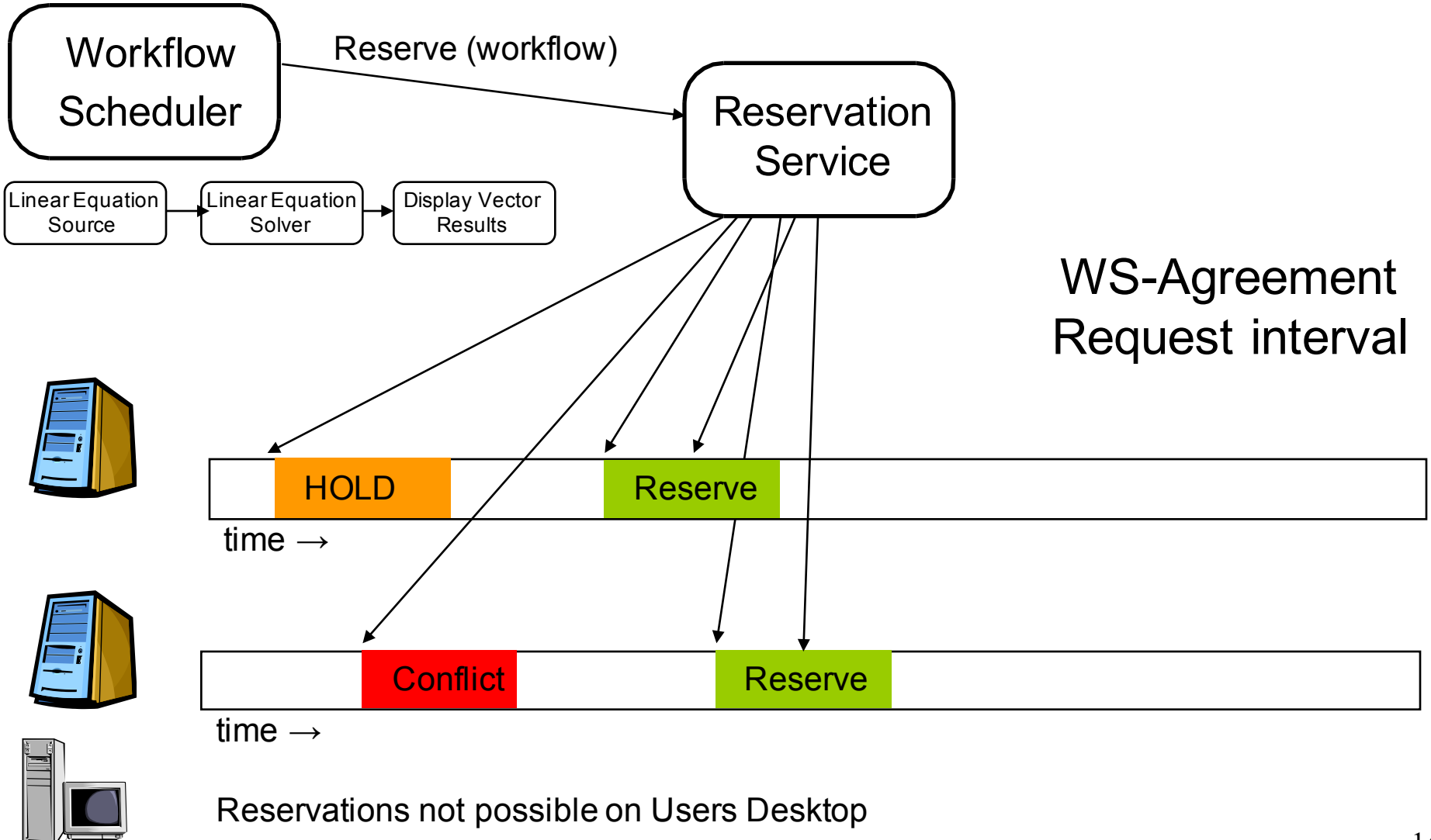
Reservation Engine



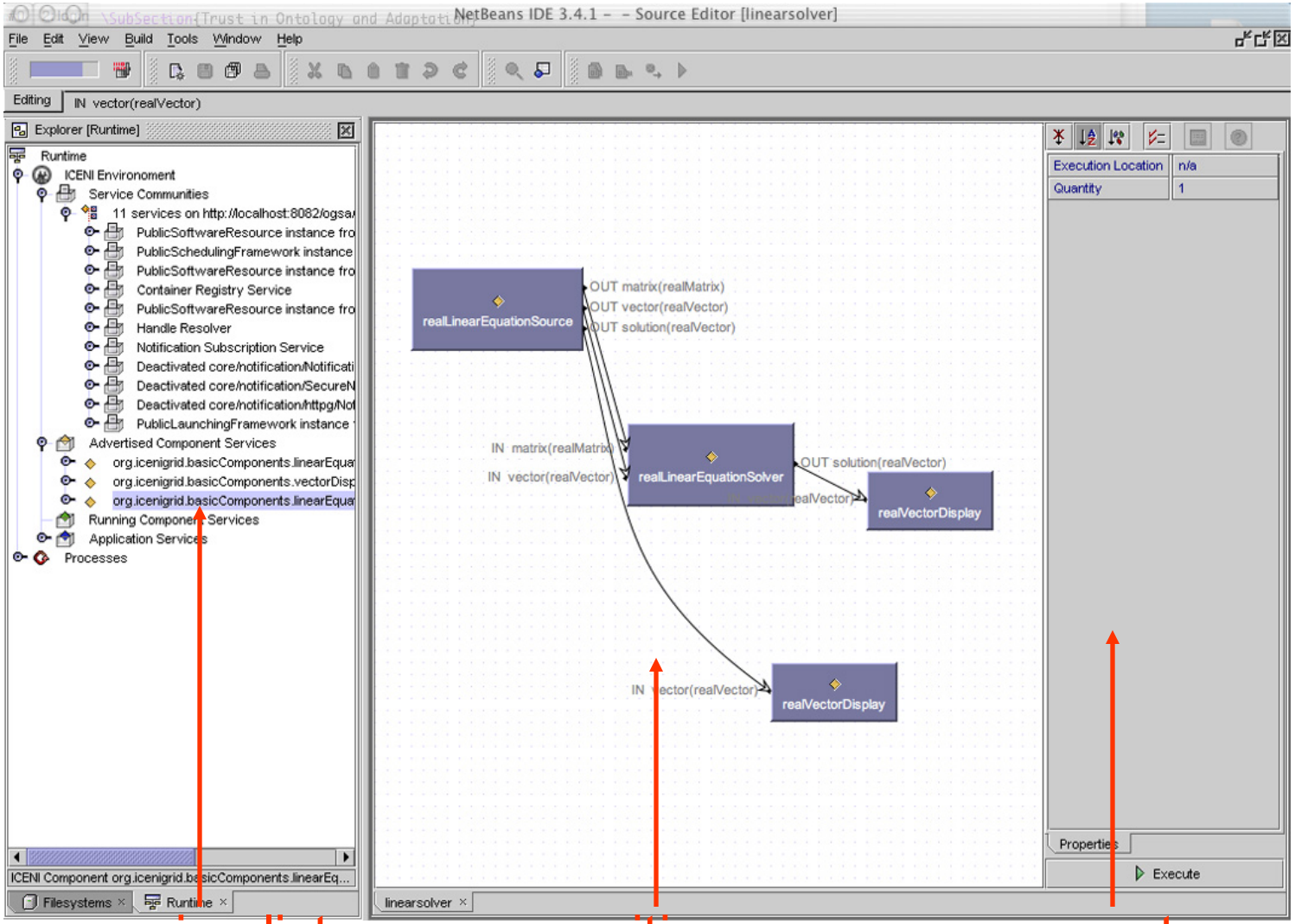
Reservation Service



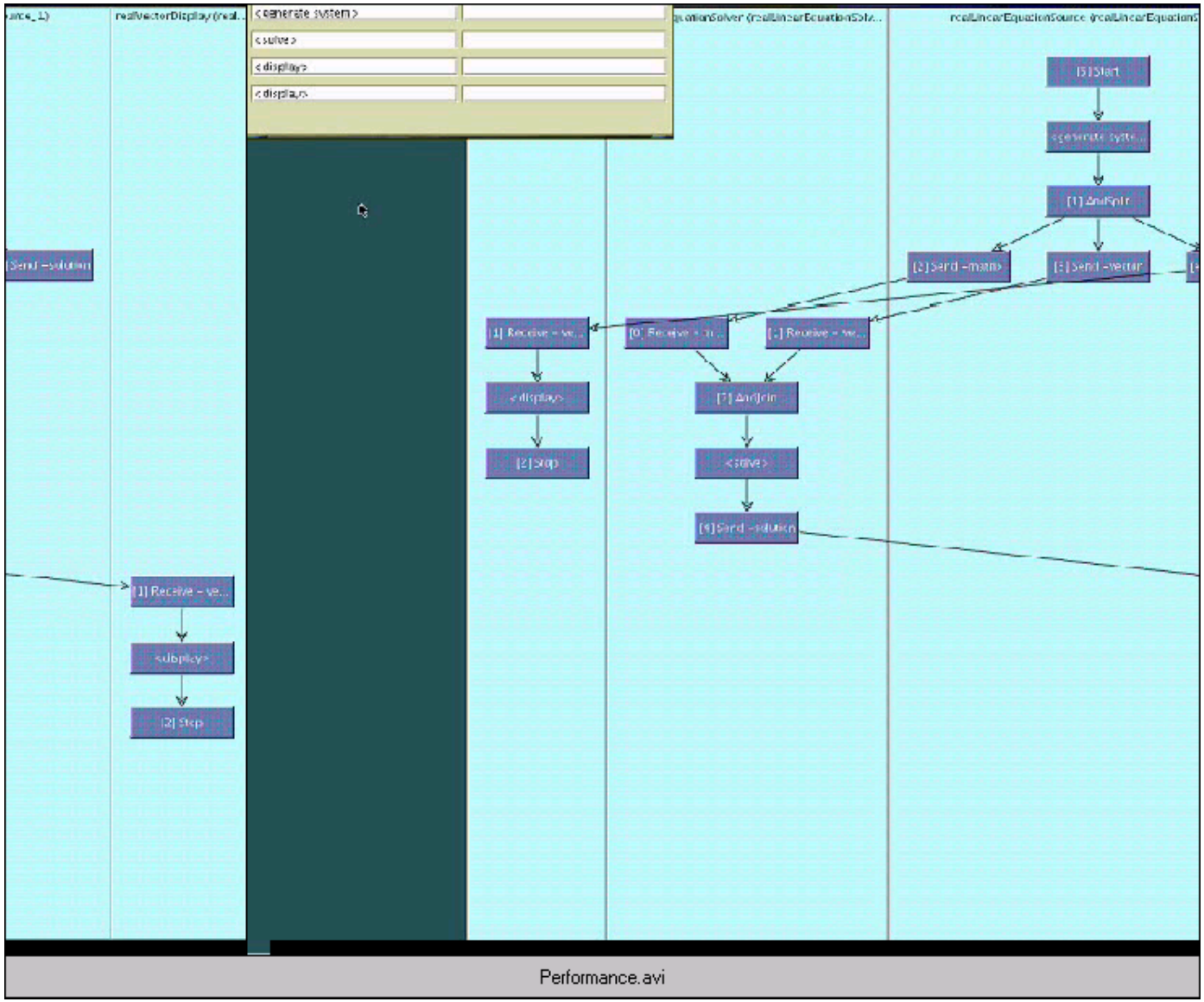
Reservations



Example: Linear Equation Solver



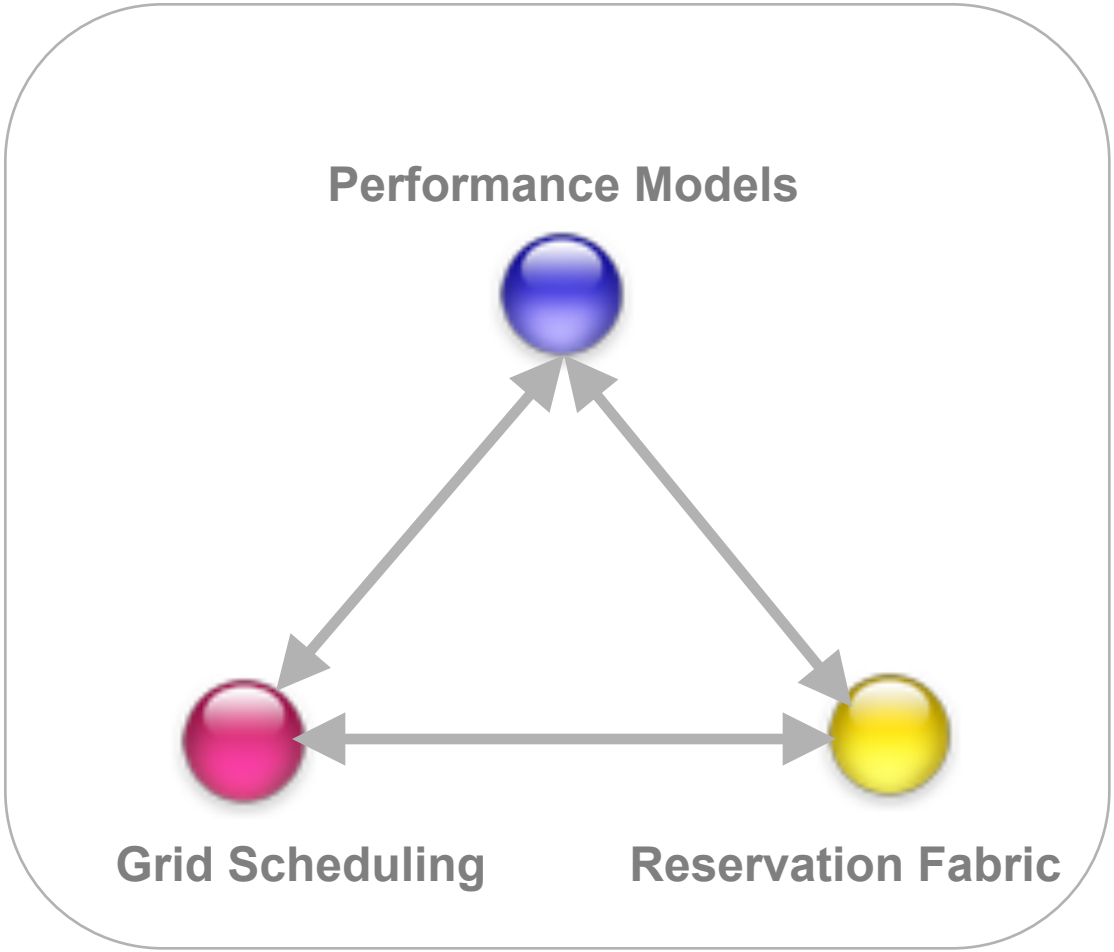
Inferring Workflow from Dataflow



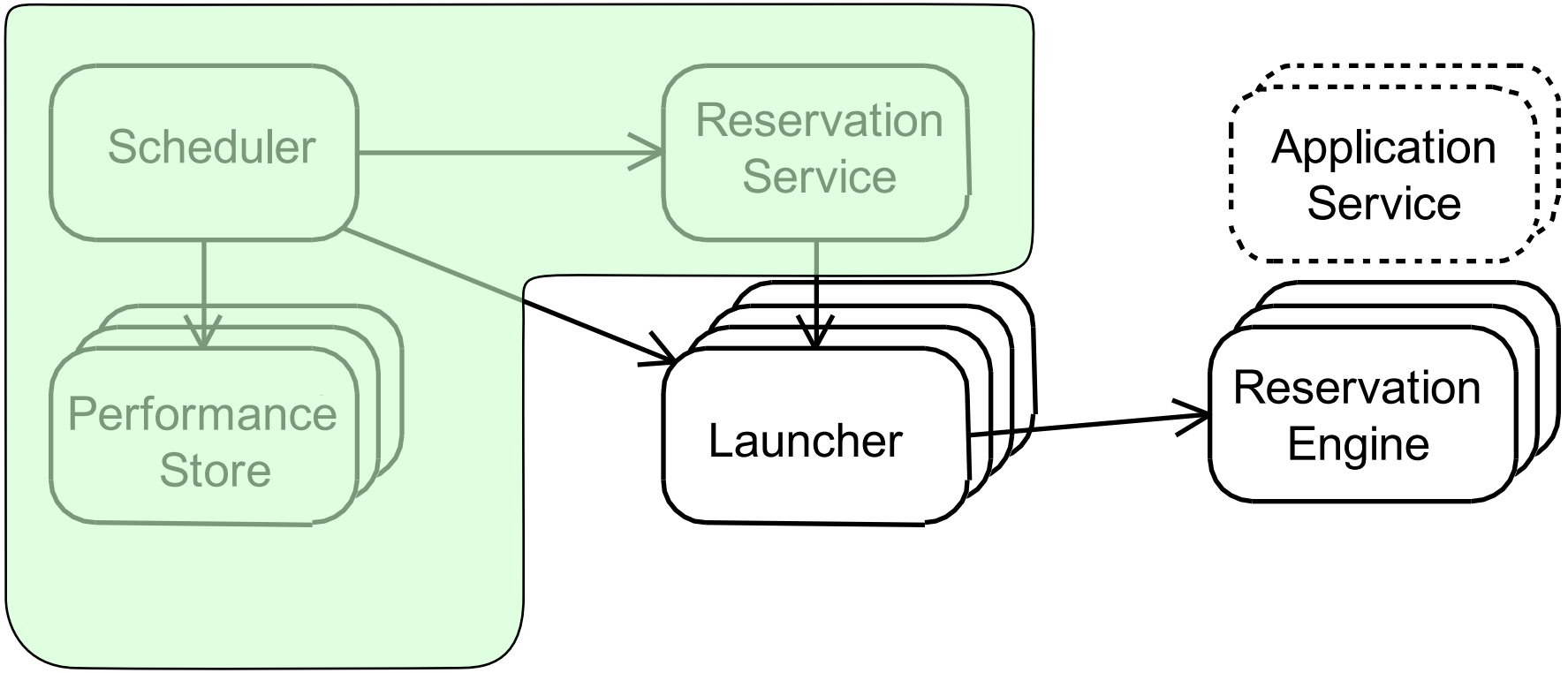
Conclusion

- Better usage of resources.
  - Reservations of resources in the future
  - Determining if co-allocation of components will affect performance
  - Late Enactment of components
- Critical Path analysis – can schedule this appropriately
- Provides a framework for experimentation with
  - Different scheduling algorithms
  - Different performance models
  - Different reservation policies

Performance Trinity



The Architecture: Showing the Trinity



Acknowledgements

- Director: Professor John Darlington
- Research Staff:
  - Nathalie Furmento, Stephen McGough, William Lee
  - Jeremy Cohen, Marko Krznaric, Murtaza Gulamali
  - Asif Saleem, Laurie Young, Jeffrey Hau
  - David McBride, Ali Afzal
- Support Staff:
  - Oliver Jevons, Sue Brookes, Glynn Cunin, Keith Sephton
- Alumni:
  - Steven Newhouse, Yong Xie, Gary Kong
  - James Stanton, Anthony Mayer, Angela O'Brien
- Contact:
  - http://www.lesc.ic.ac.uk/ • e-mail: lesc@ic.ac.uk

Scheduling Architecture in ICENI