

# DOMAIN: TAMING HETEROGENEOUS COMPUTING COMPLEXITY WITH FULL-STACK GOVERNANCE OF DOMAIN-SPECIFIC LANGUAGES

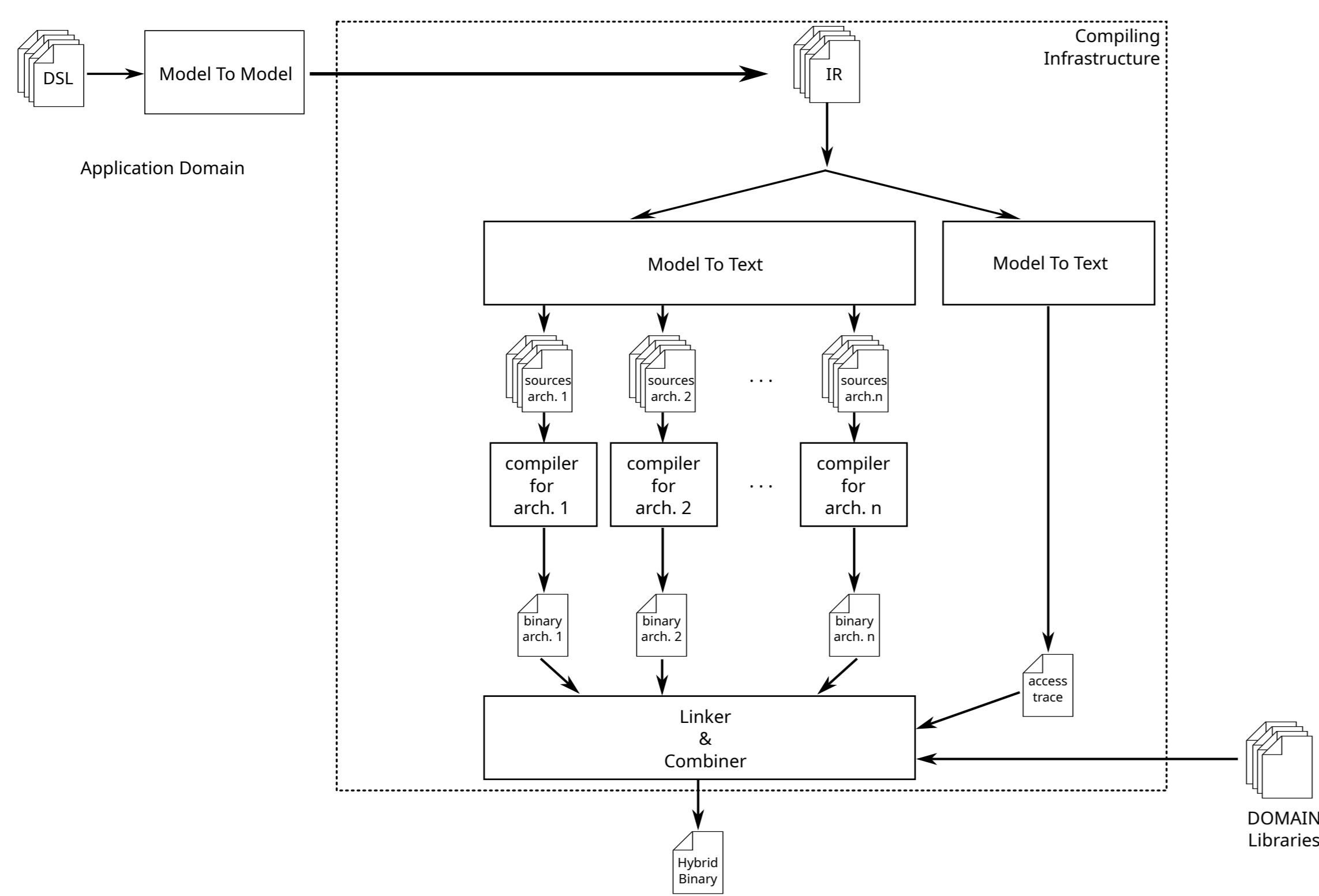
Alessandro Pellegrini  
a.pellegrini@ing.uniroma2.it  
Tor Vergata University of Rome

## PROJECT GOALS

- ▶ Programming for heterogeneous architectures is hard
- ▶ The greatest benefits are for non-IT domain experts
- ▶ These scientists cannot be exposed to the complexity of heterogeneous architectures
- ▶ Our solution: using DSLs to run task-based applications on heterogeneous architectures

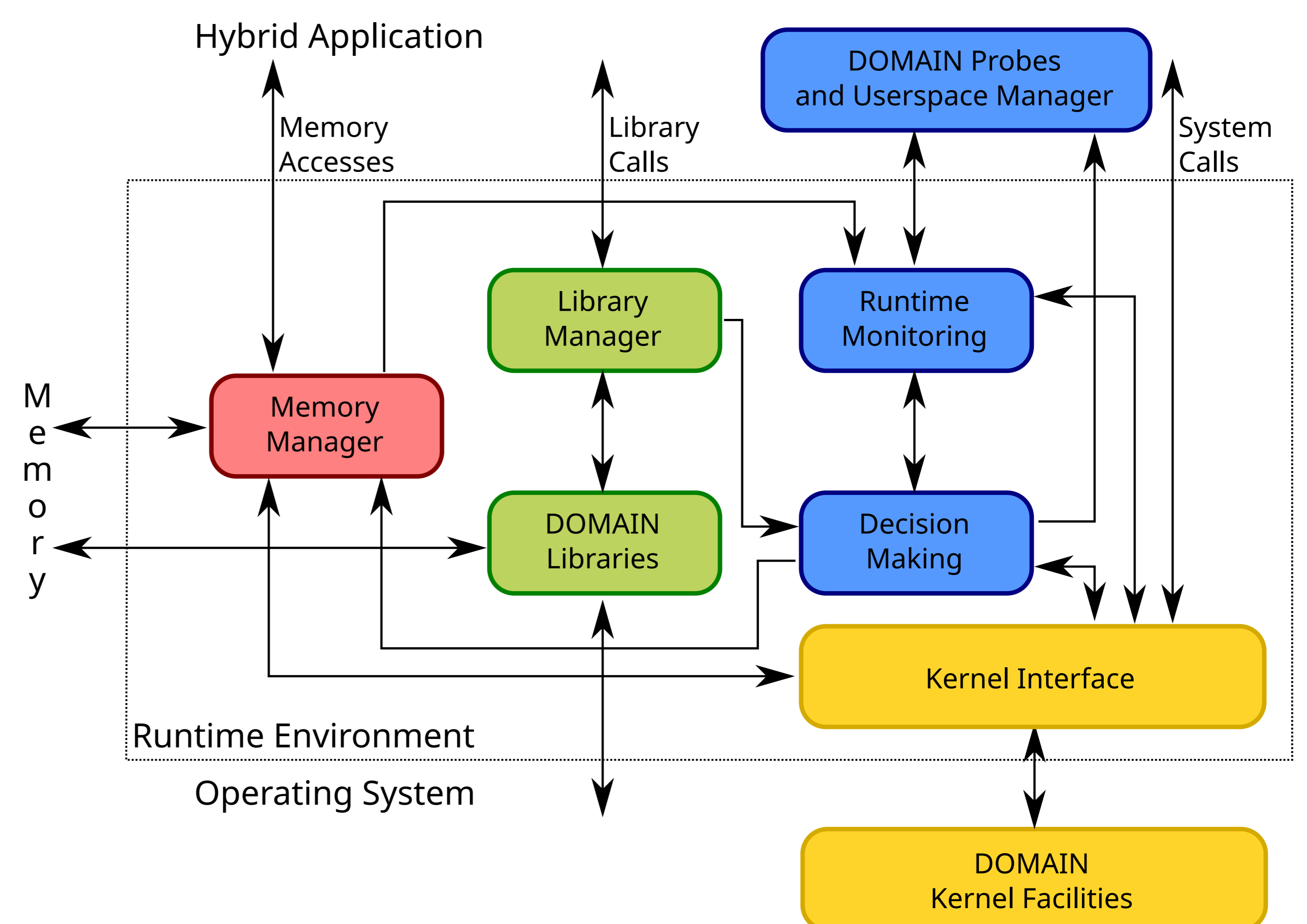
## LANGUAGE DEVELOPMENT

- ▶ We use model-driven engineering techniques to support DSL development
- ▶ The creation of a new DSL requires the creation of a Model To Text transformation to an intermediate representation (based on Actor Model and Mbedder)
- ▶ We provide Model To Text transformations for CPU, GPU, FPGA code
- ▶ Standard compilation toolchains are used to generate a *hybrid binary*



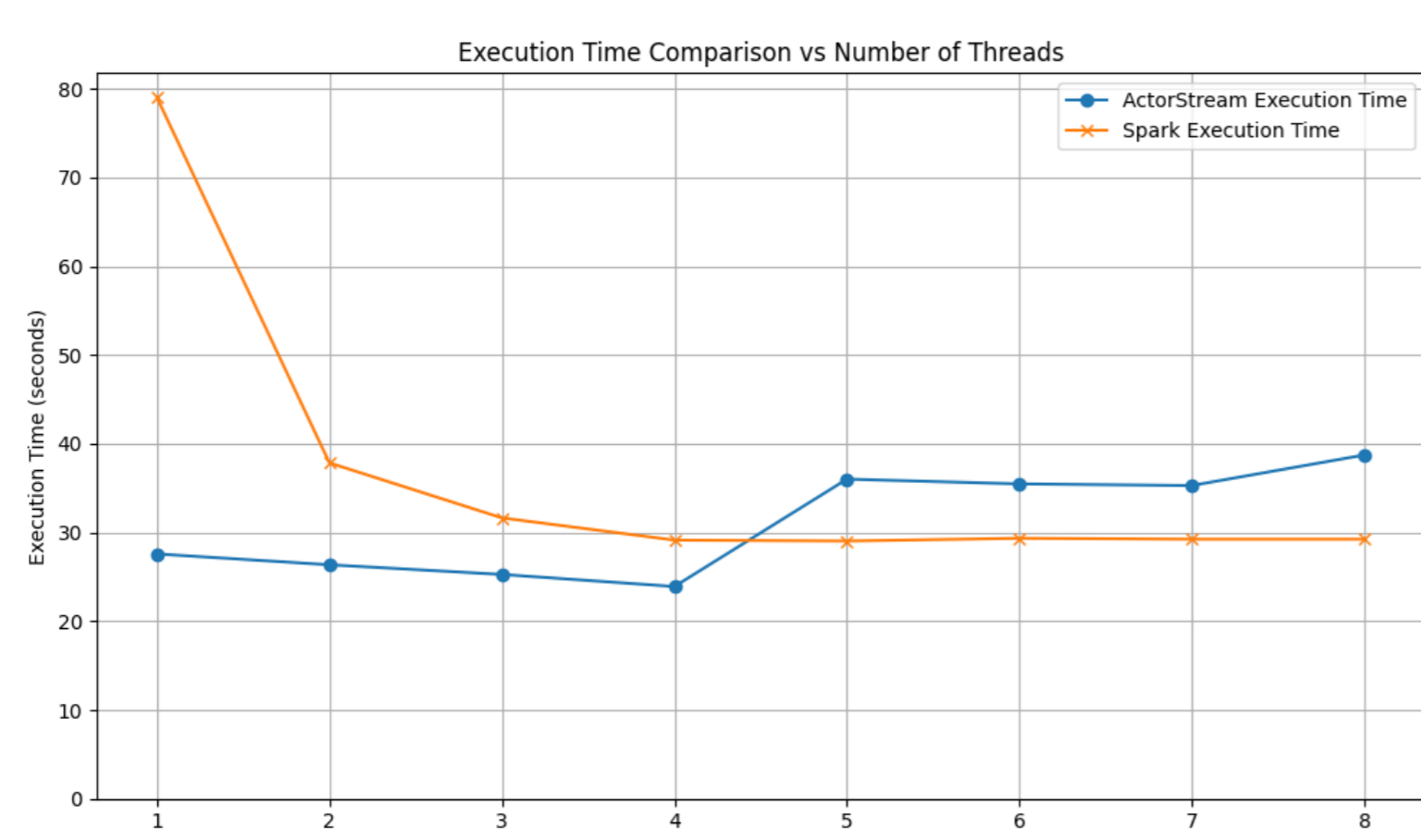
## RUNTIME ENVIRONMENT

- ▶ Hybrid binaries are executed in sandboxes thanks to a dedicated *runtime environment*
- ▶ Dynamic memory is tracked to monitor the state of each actor
- ▶ Application execution is monitored to profile its behaviour at runtime
- ▶ A decision making module determines the optimal placement on the available heterogeneous hardware at runtime
  - ▶ The search for optimal task allocation can be configured according to different objectives: throughput, energy efficiency, power capping, ...
  - ▶ Optimisation based on *Answer Set Programming*
- ▶ During execution, *reconciliation points* are identified where tasks can be migrated between different devices
  - ▶ The state of actors is serialized and transferred across devices
  - ▶ The routing of tasks is managed by the runtime environment and delivered to the correct device



## PRELIMINARY RESULTS

- ▶ Stream processing pipeline implemented based on Streaming SQL
- ▶ NY Taxi and Limousine trip record data
- ▶ Compared against Apache Spark



## CURRENTLY ONGOING

We are currently targeting the following domains:

- ▶ Discrete-Event Simulation
- ▶ Biological Reactions
- ▶ Demography

