University of Pittsburgh

Composing, Reproducing, and Sharing Simulations

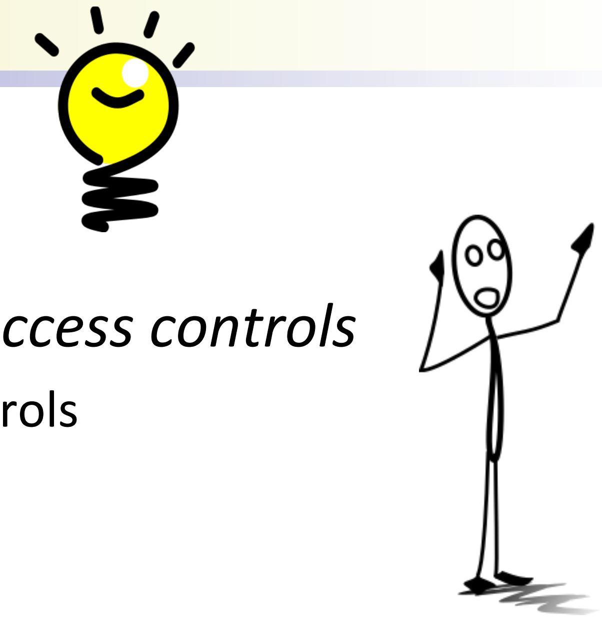
Daniel Mosse
{mosse,childers}@cs.pitt.edu

Debashis Ganguly, William C. Garrison III, David Wilkinson, Bruce R. Childers, Adam J. Lee, and Daniel Mosse’

Dept. of Computer Science, University of Pittsburgh

Thanks NSF and openclipart.org

Starting a new project…



- You've got a new idea:

  PIM encryption engine to support access controls

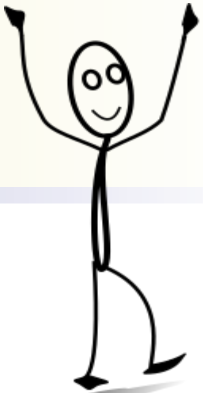
  - Access controls, applications using access controls
  - Memory activity and architecture simulator
- Starting point (The Introspective Scientist)
  - Hmmm. I wonder **what simulators exist**?
  - Oh, person Y made this model! Can I **leverage & reuse it** for my work?
  - Oops. Something is **missing in paper**. What were those sim params???
  - Yikes! Wait! My **results are different**???

Consider this:

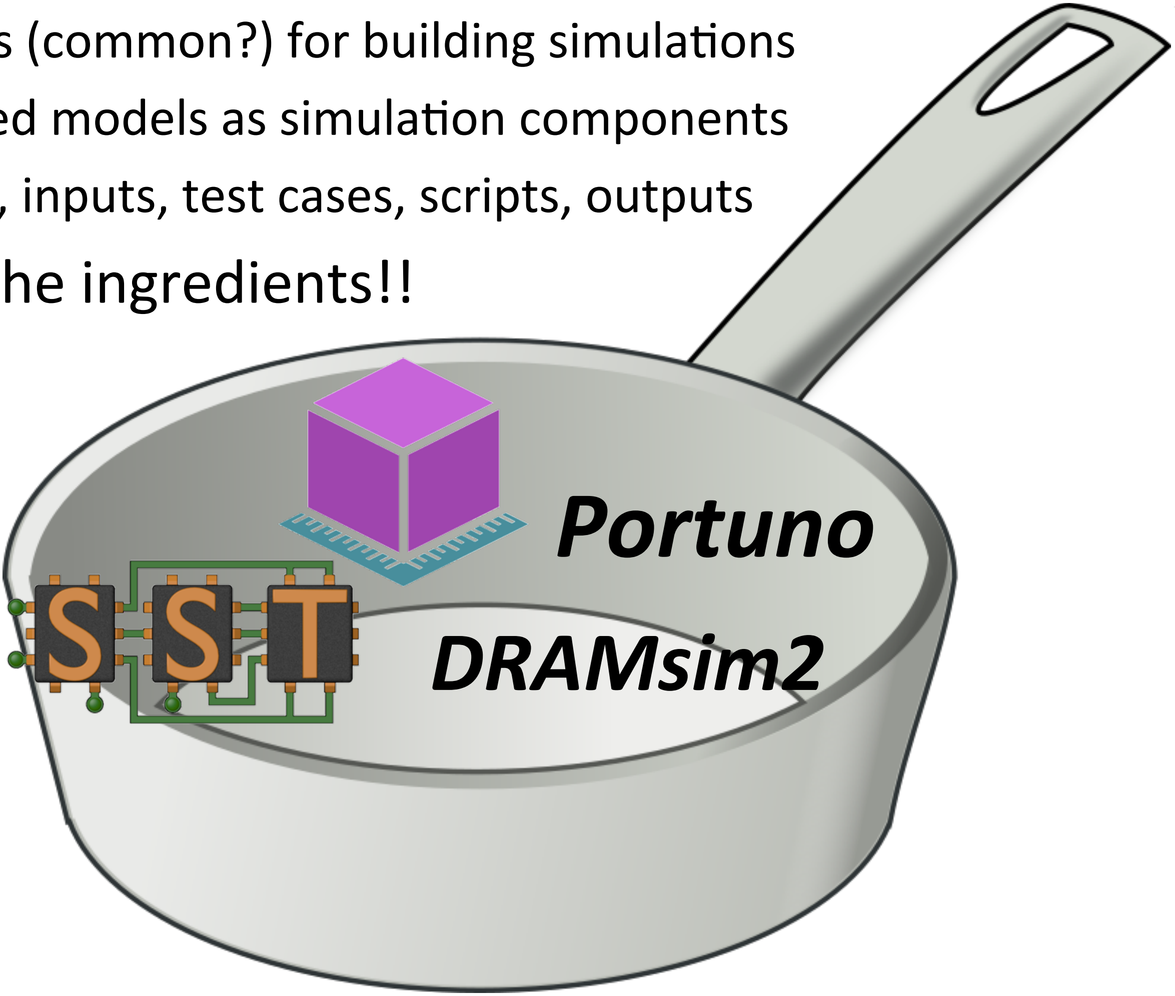
*Nature asked 1,576 scientists … **over 70% said they'd tried and failed to reproduce another group's experiments**.*

-- Scientific American, May 28, 2016

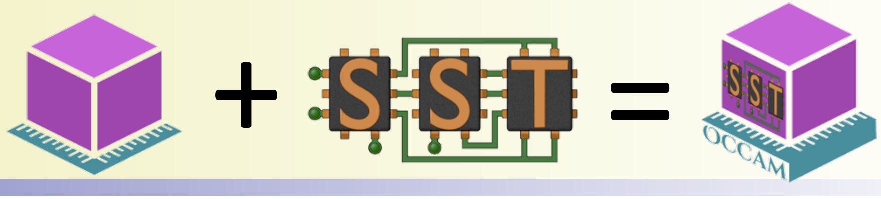
It really shouldn't be that difficult...



- Modeling and simulation has advantages
  - Frameworks (common?) for building simulations
  - Implemented models as simulation components
  - Parameters, inputs, test cases, scripts, outputs
- We have all the ingredients!!



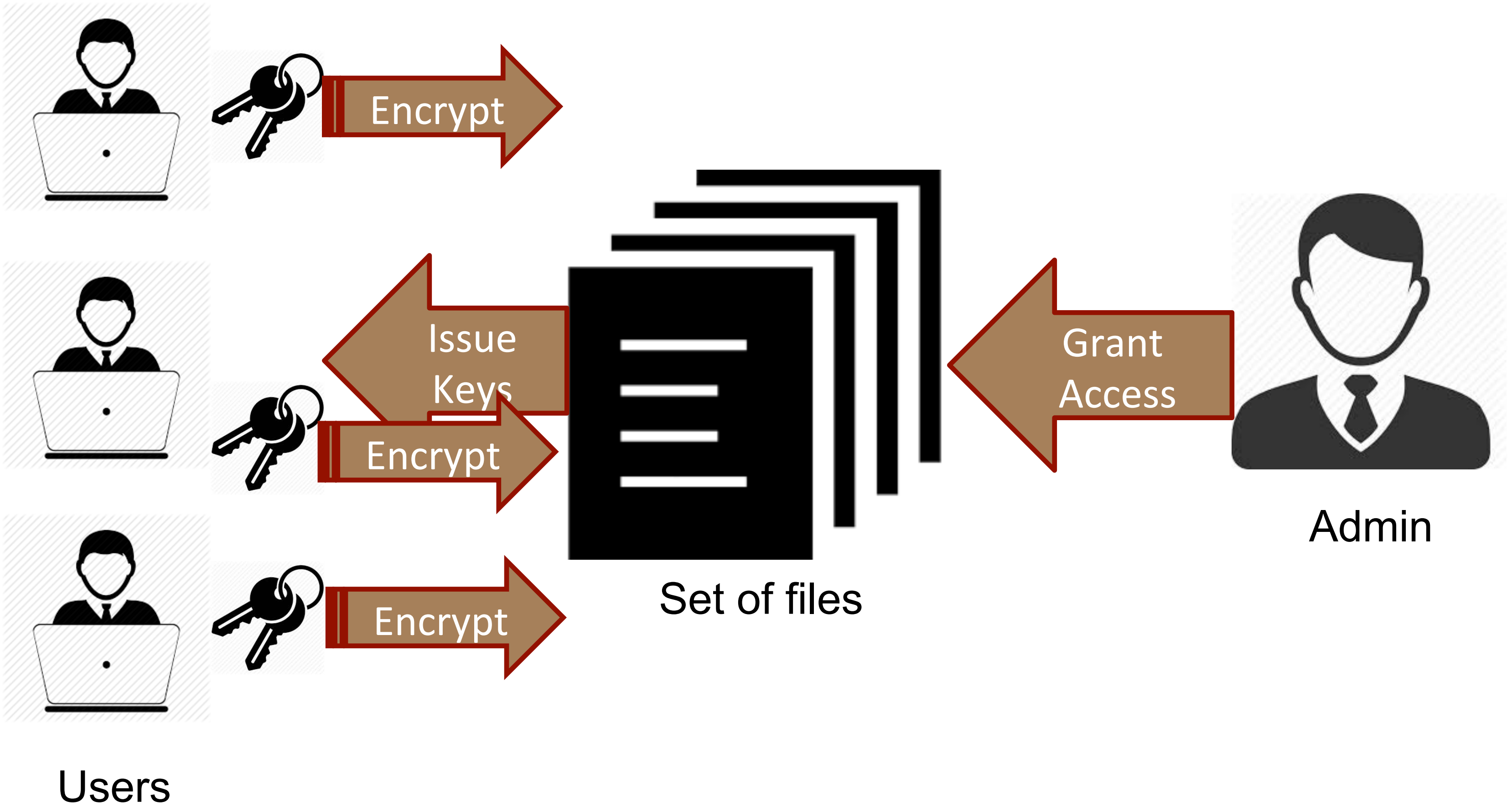
The Ingredients



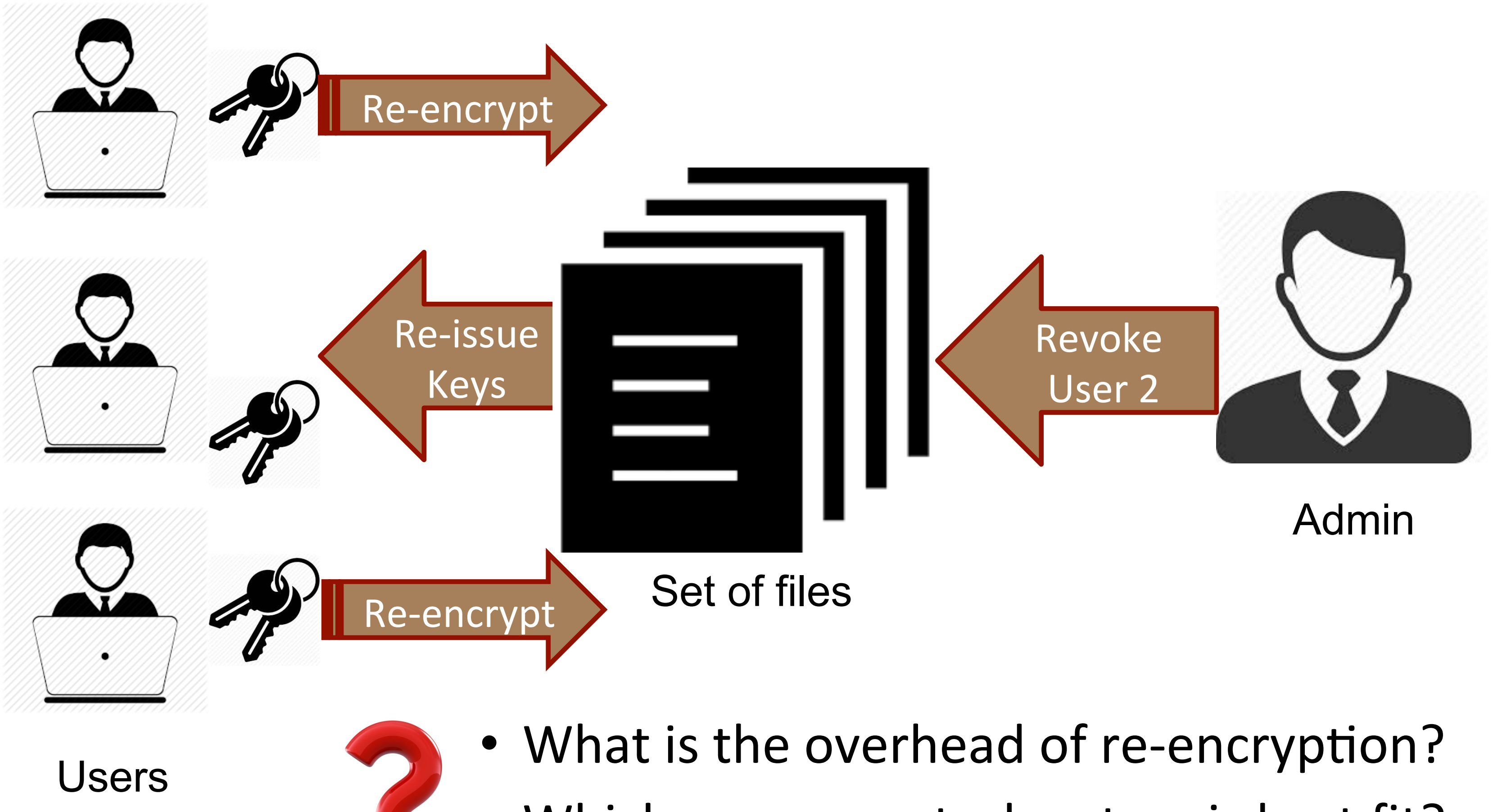
SC16 Tutorial: Solving and Sharing the Puzzle, Sunday, Nov. 13

- **Portuno:** Access control simulator/analyzer
- **SST**: Host Portuno & compose with memory simulator
  - Instantiate a simulator used for experiments
  - ***Share and compose (reuse) simulation models*** within SST
- **OCCAM**: Create, run and share experiments
  - Define workflows of data, tools, results for experiments
  - Use simulators/tools & experiments (w/results and visualizations)
  - ***Share and compose (reuse) tools for experiment workflows***
  - ***Repeatability and modification of experiments***

Portuno: Access Control

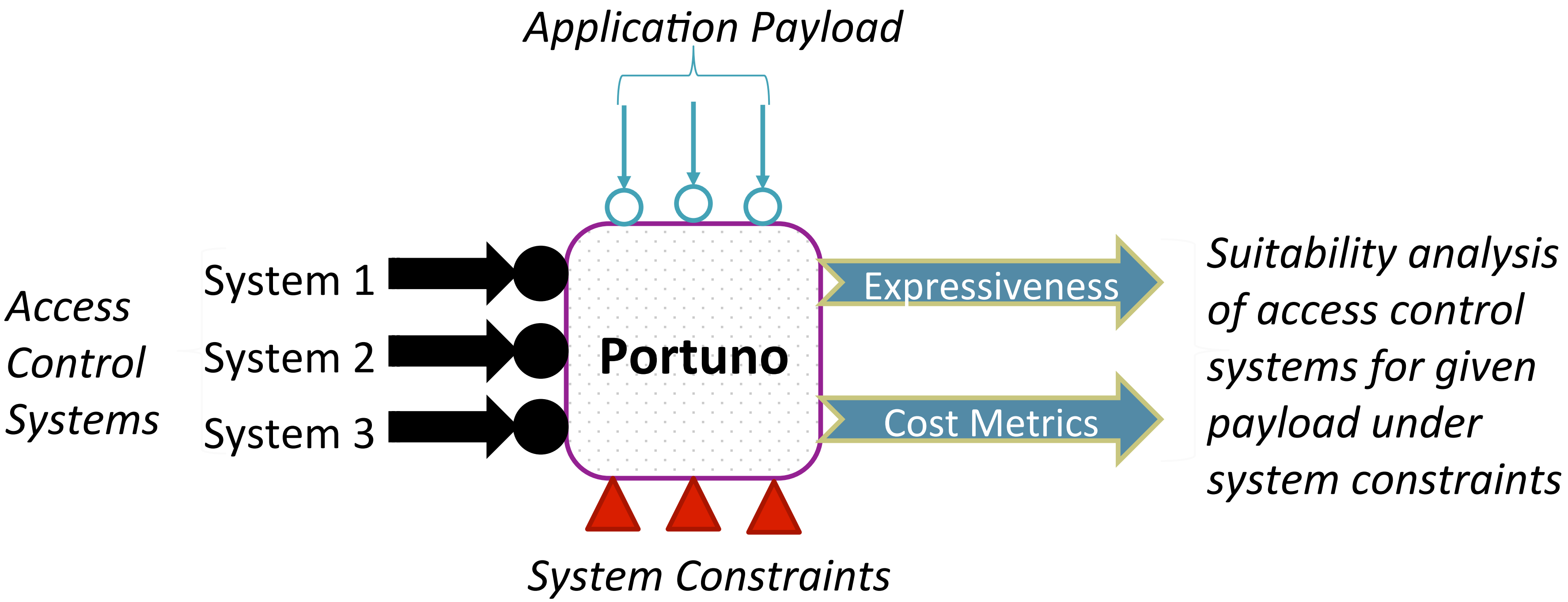


Portuno: Access Control



- What is the overhead of re-encryption?
- Which access control system is best fit?

Portuno: Access Control



- Monte-Carlo simulator
- Inputs characterize likelihood of individual behaviors
- Individual instances independent and parallelizable

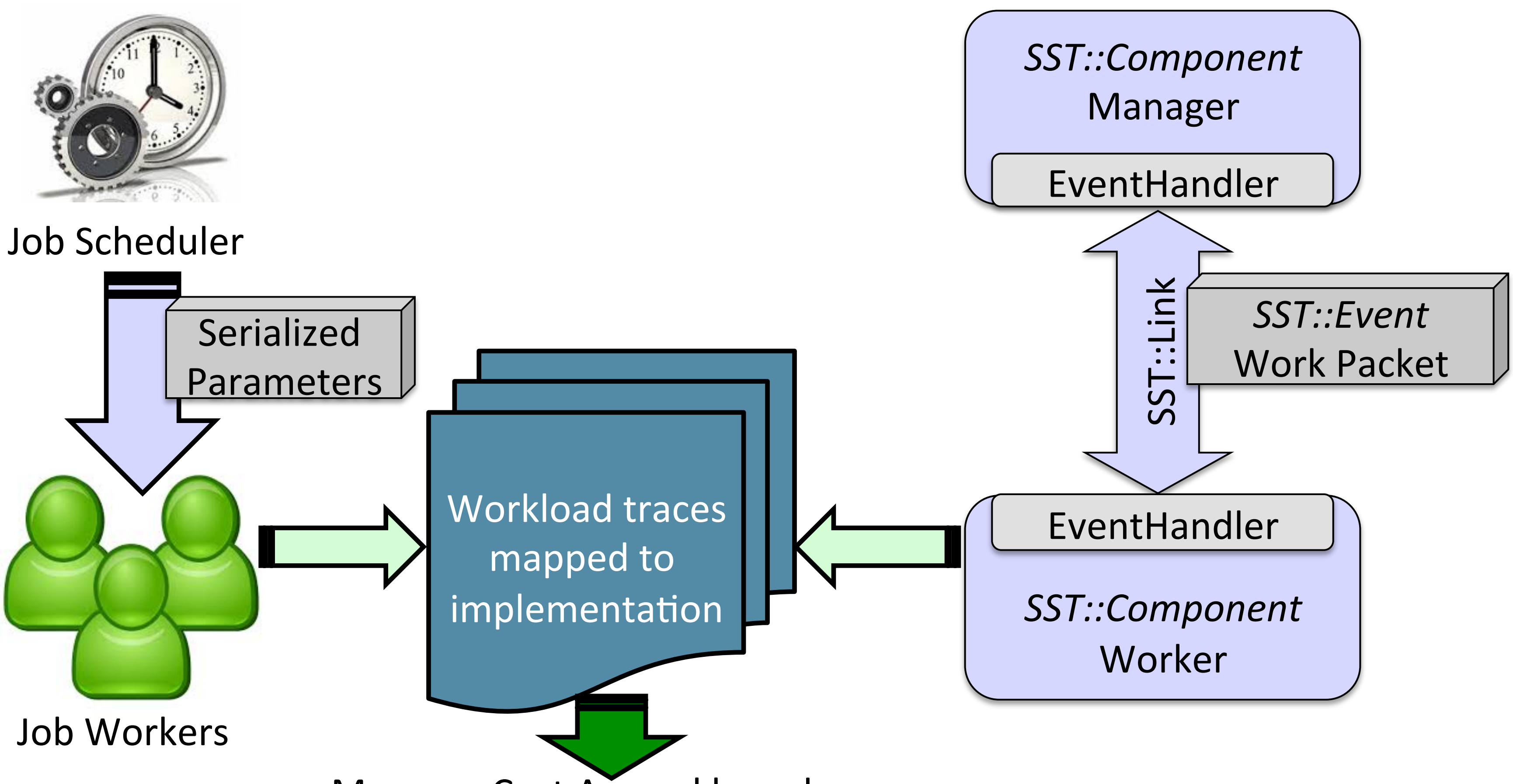
A parallel, discrete-event simulation framework for scalability and flexibility

- Open, multiscale, and scalable parallel execution
- Highly modular framework that is extensible
- APIs to monitor/measure run-time statistics
- Compose and integrate simulators/models

➔ ***Ideal framework to build our co-design simulator***

- Portuno implemented as a Java framework
- Integrate Portuno into SST
- Reuse existing memory simulators (DRAMsim2, or our own, HMMsim)

Portuno integrated with SST



Job Scheduler

Serialized Parameters

Job Workers

Workload traces mapped to implementation

SST::Component Manager

EventHandler

SST::Link

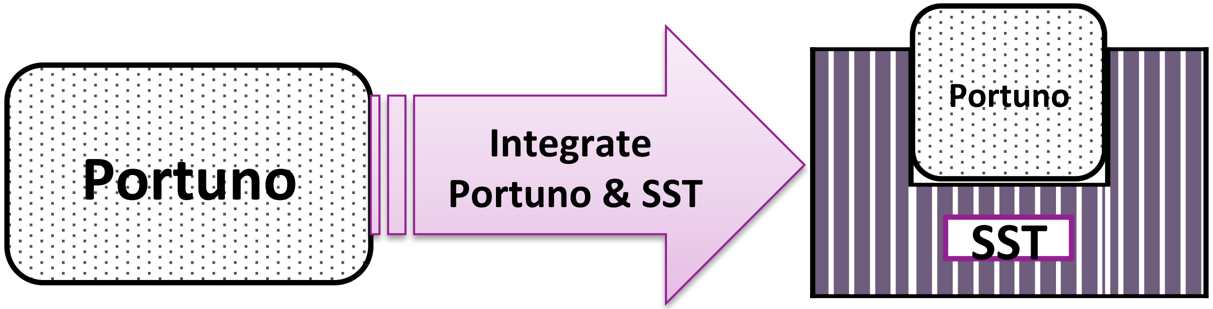
SST::Event Work Packet

EventHandler

SST::Component Worker

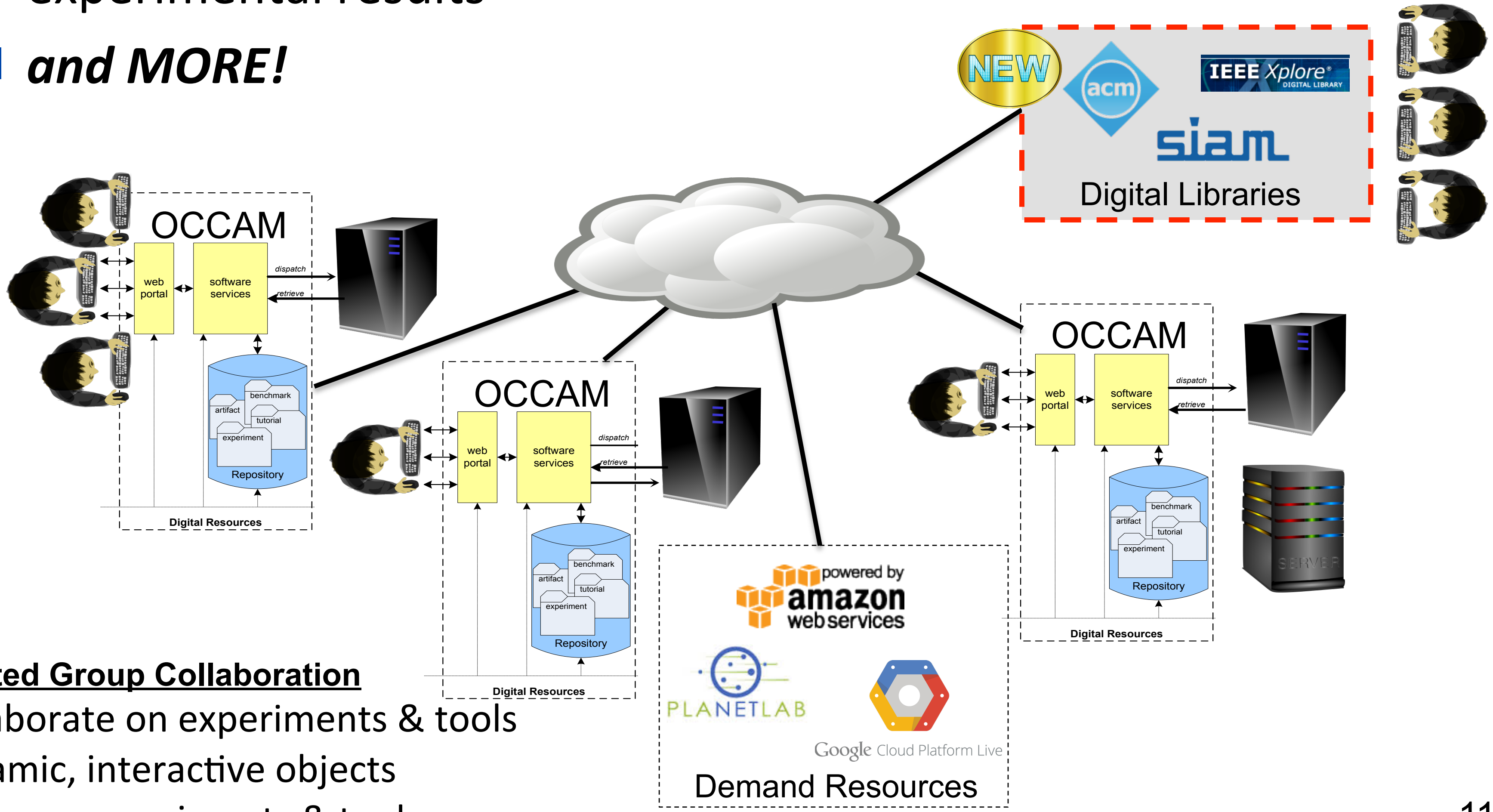
Measure Cost Accrual based on
Cost Functions under constraints

Experiments with Portuno in SST



- How to define, run & share experiments?
- How to maintain provenance?
- How to limit burden of peer review and overcome limitations of page constraints?

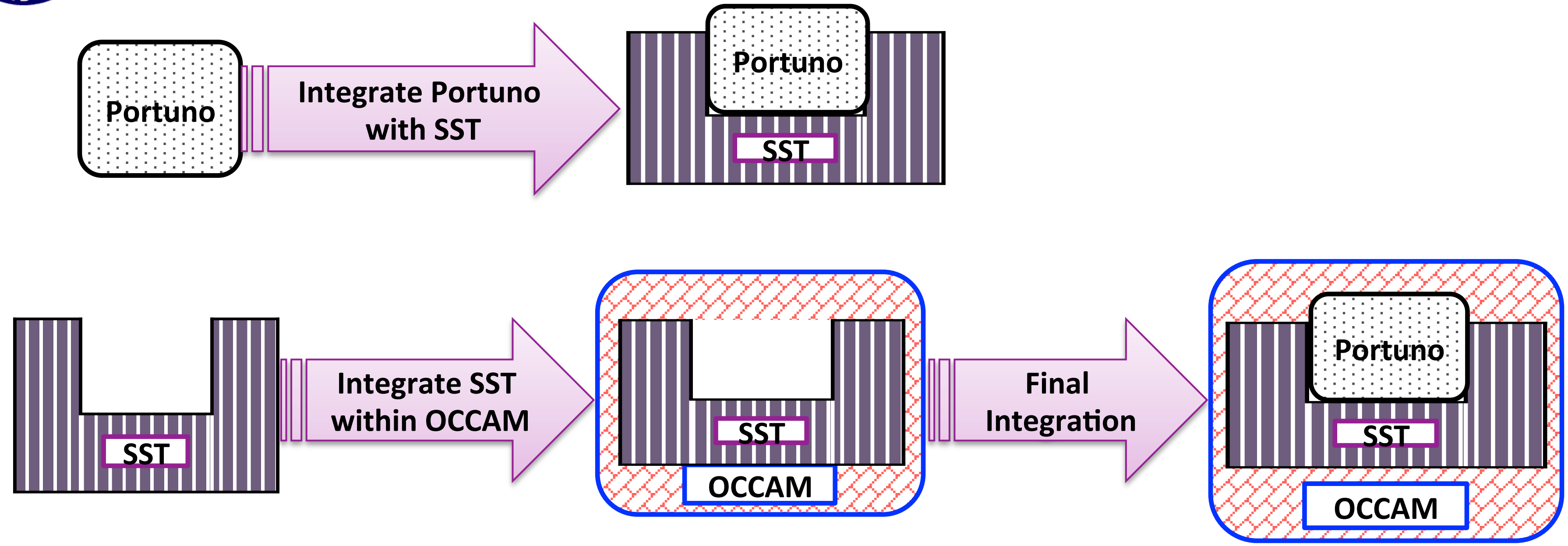
- Community-supported exchange for curation of simulation & experimental results
- ***and MORE!***



Federated Group Collaboration

- Collaborate on experiments & tools
- Dynamic, interactive objects
- Leverage experiments & tools

Complete co-design environment



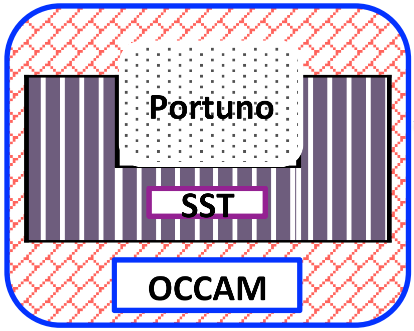
① Create, run and analyze results (through simplified GUI)

② Collaborate and share to conduct experiments

③ Reuse tools and experiments to accelerate research

④ Repeatability with provenance

Observations from current work

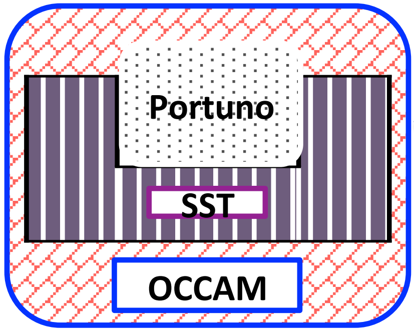


Scientific Observation

Cryptographically-enforced RBAC policies are prohibitive, even in moderate scale

How to decrease the overheads associated with enforcing cryptographic access controls in cloud environments?

Observations from current work



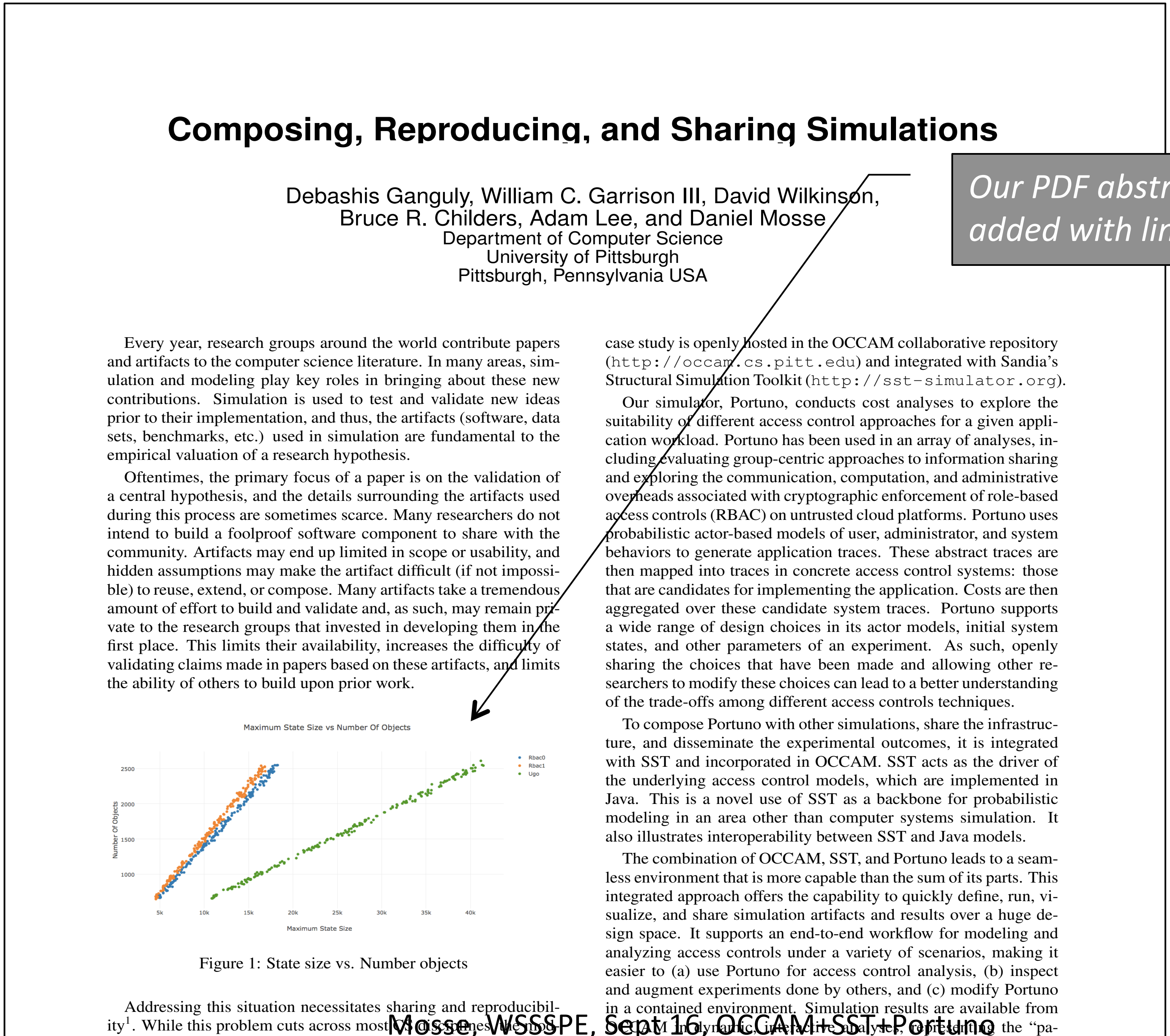
Process Observation

Easier to figure out the scientific observations, given the infrastructure in place with Portuno, DRAMSim2, SST, OCCAM

How to spread the love? How to decrease the barrier to entry of the infrastructure? Why do we want to do it?

Advantage?? Article of the future

Our PDF abstract with a graph added with link to experiment



Composing, Reproducing, and Sharing Simulations

Debashis Ganguly, William C. Garrison III, David Wilkinson,
Bruce R. Childers, Adam Lee, and Daniel Mosse
Department of Computer Science
University of Pittsburgh
Pittsburgh, Pennsylvania USA

Every year, research groups around the world contribute papers and artifacts to the computer science literature. In many areas, simulation and modeling play key roles in bringing about these new contributions. Simulation is used to test and validate new ideas prior to their implementation, and thus, the artifacts (software, data sets, benchmarks, etc.) used in simulation are fundamental to the empirical valuation of a research hypothesis.

Oftentimes, the primary focus of a paper is on the validation of a central hypothesis, and the details surrounding the artifacts used during this process are sometimes scarce. Many researchers do not intend to build a foolproof software component to share with the community. Artifacts may end up limited in scope or usability, and hidden assumptions may make the artifact difficult (if not impossible) to reuse, extend, or compose. Many artifacts take a tremendous amount of effort to build and validate and, as such, may remain private to the research groups that invested in developing them in the first place. This limits their availability, increases the difficulty of validating claims made in papers based on these artifacts, and limits the ability of others to build upon prior work.

Figure 1: State size vs. Number objects

Addressing this situation necessitates sharing and reproducibility[1]. While this problem cuts across most CS disciplines, the mod-

case study is openly hosted in the OCCAM collaborative repository (`http://occam.cs.pitt.edu`) and integrated with Sandia's Structural Simulation Toolkit (`http://sst-simulator.org`).

Our simulator, Portuno, conducts cost analyses to explore the suitability of different access control approaches for a given application workload. Portuno has been used in an array of analyses, including evaluating group-centric approaches to information sharing and exploring the communication, computation, and administrative overheads associated with cryptographic enforcement of role-based access controls (RBAC) on untrusted cloud platforms. Portuno uses probabilistic actor-based models of user, administrator, and system behaviors to generate application traces. These abstract traces are then mapped into traces in concrete access control systems: those that are candidates for implementing the application. Costs are then aggregated over these candidate system traces. Portuno supports a wide range of design choices in its actor models, initial system states, and other parameters of an experiment. As such, openly sharing the choices that have been made and allowing other researchers to modify these choices can lead to a better understanding of the trade-offs among different access controls techniques.

To compose Portuno with other simulations, share the infrastructure, and disseminate the experimental outcomes, it is integrated with SST and incorporated in OCCAM. SST acts as the driver of the underlying access control models, which are implemented in Java. This is a novel use of SST as a backbone for probabilistic modeling in an area other than computer systems simulation. It also illustrates interoperability between SST and Java models.

The combination of OCCAM, SST, and Portuno leads to a seamless environment that is more capable than the sum of its parts. This integrated approach offers the capability to quickly define, run, visualize, and share simulation artifacts and results over a huge design space. It supports an end-to-end workflow for modeling and analyzing access controls under a variety of scenarios, making it easier to (a) use Portuno for access control analysis, (b) inspect and augment experiments done by others, and (c) modify Portuno in a contained environment. Simulation results are available from OCCAM in dynamic, interactive analyses, representing the "pa-

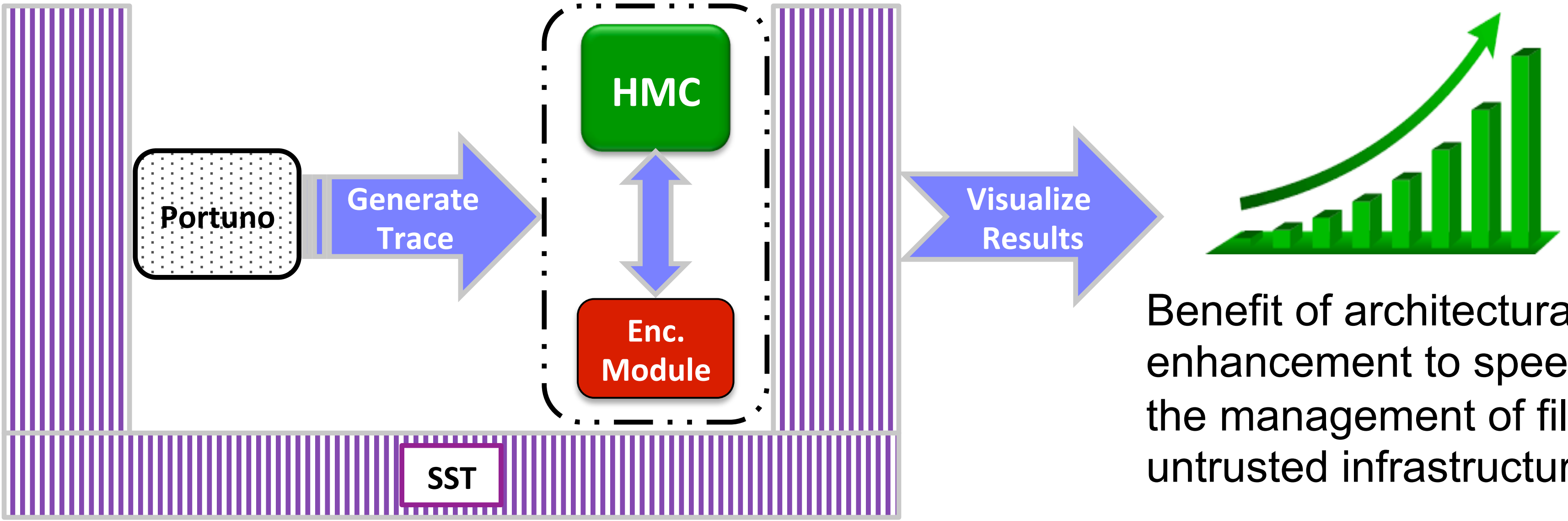
Work in progress

- Push re-encryptio processor to re-en
- Co-design support execution environ

① Composition with common framework

② Accelerate research by sharing

③ Increase impact through access



Benefit of architectural enhancement to speedup the management of files on untrusted infrastructure.