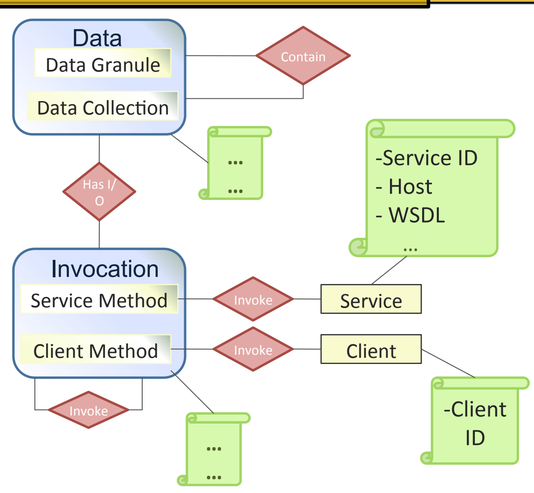
# A Comprehensive Provenance Model

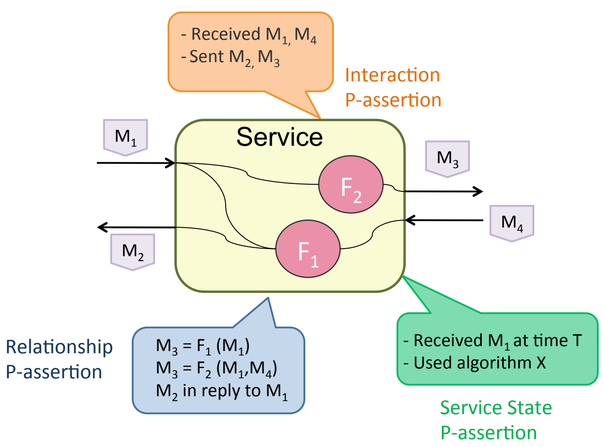
**Salmin Sultana, Elisa Bertino**

**NSF Award Number: 1111512 "Privacy-Enhanced Secure Data Provenance"**

## Provenance Model Observations




**Workflow-based Model**



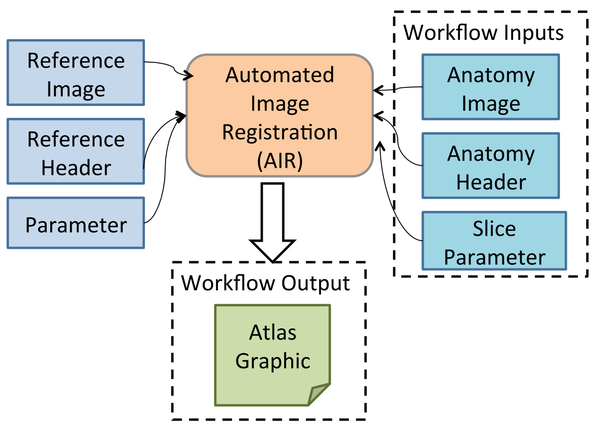

**Process-based Model**

| Process Name |
|---|
| Process ID |
| Argument |
| Kernel Version |
| Kernel Module |
| Environment |
| Input |
| Output |

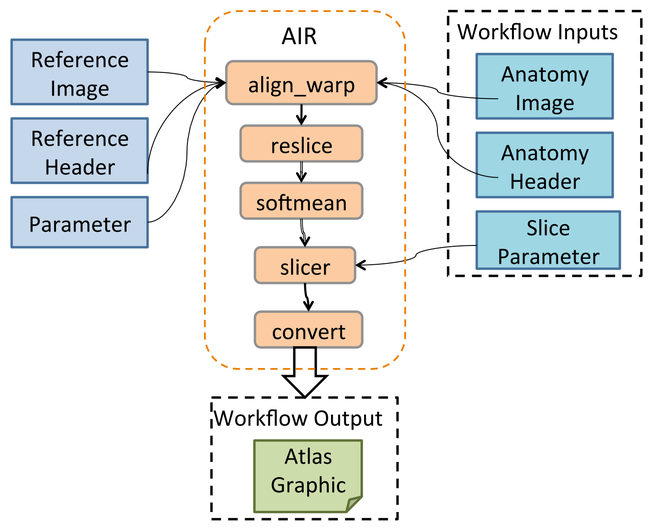
**OS-based Model**

- Workflow based Provenance Model: Mostly specific to a particular workflow
- Process based Provenance Model: Backtracking the log to view provenance
- OS based Provenance Model: No comprehensive model

## Provenance Granularities




Provenance capture at Process level




Provenance capture at Operation level

### Desired Characteristics of a Provenance Model

**Unified Framework**

- A model able to capture and store provenance for any kind of data at any abstraction layer.

**Provenance Security**

- Privacy aware fine grained access control policies

**Granularity**

- User specified policies to customize provenance capture
- Policies to generate desired provenance view

**Efficient Provenance Queries and Views**

- Easy tracking of data dependence and usage
- Efficient filter, summarization or compression of provenance graph

## Proposed Provenance Model

*The provenance of a data object is the documented history of the actors, process, operations, inter-process / operation communications, environment, access control and other user preferences related to the manipulation of the object. The relationships between the entities form a provenance graph (DAG) for the data object.*

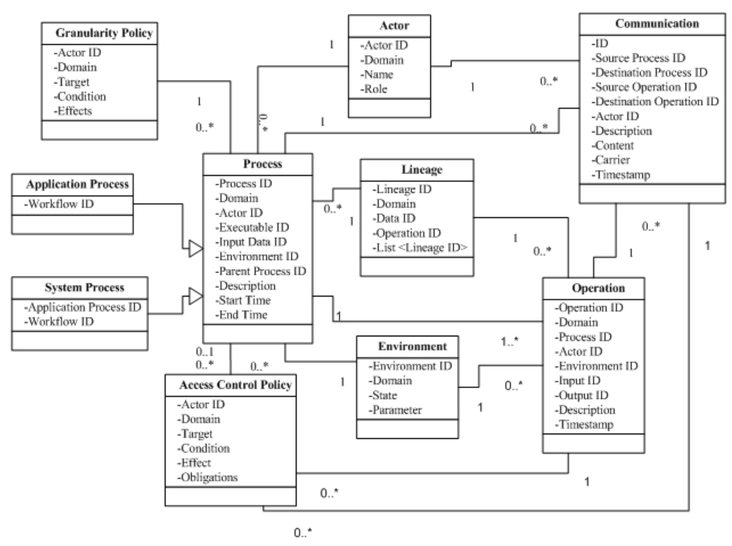
### Entities

- Process
- Operation
- Communication
- Lineage
- Environment
- Actor
- Access Control Policy
- Granularity Policy

### Relationships

- Process —*consists of*→ Operation
- Process —*Communication*→ Process
- Operation —*Communication*→ Operation
- Lineage —*used by/ generated by*→ Process
- Lineage —*used by/ generated by*→ Operation
- Lineage —*derived from*→ Lineage
- Process / Operation —*affected by*→ Environment
- Process / Operation / Communication —*operated by*→ Actor
- A/C Policy —*acts upon*→ Process / Operation / Communication
- Granularity Policy —*acts upon*→ Operation

## Abstract Schema




- ✓ Each provenance record uniquely Identified by an ID
- ✓ *domain* specifies the system performing data manipulations
- ✓ *Process* is a base class
  - the high level process and the system process are differentiated by the two inherited classes.

## Access Control Policy

Controls whether and how other actors may utilize *process*, *operation* and *communication* records

```
<obligations>
  <obligation>
    <operation> inform the actor </operation>
    <temporal constraint> 10 days </temporal constraint>
    <fulfill on> access </fulfill on>
  </obligation>
</obligations>
```

```
<policy ID=1>
 <domain> konark </domain
 <target>
  <subject> anyuser </subject>
  <record> process </record>
  <restriction> anyuser.role == professor AND
                process.timestamp < 1.1.2011
  </restriction>
 </target>
 <condition> system.machineid == hector
             AND purpose == research
 </condition>
 <effect> necessary permit </effect>
</policy>
```

## Granularity Policy

Controls the
- capture and storage of provenance records
- user queries and views of provenance

```
<effects>
  <effect>
    <record> operation </record>
    < action> include </action>
  </effect>
</effects>
```

```
<policy ID=1>
 <domain> konark </domain>
 <actor ID> 3 </actor ID>
 <target>
  <subject> actor </subject>
  <restriction> actor.role == student AND
                process.Executable ID == 2A1
  </restriction>
 </target>
 <condition> system.machineid == hector
 </condition>
 <effects> … … </effects>
</policy>
```

## Queries

| | |
|---|---|
| **Fundamental Queries** | • Generate the sequences of *processes* in a workflow<br>• Find all the *operations* to a *process* |
| **Queries on Invocations** | • Retrieve the *commands* in a particular manipulation<br>• Remove *commands* before/after a specified time |
| Lineage Queries | • Find the *ancestors* of a given data object<br>• Find the data objects that are the results of a *flow pattern* |
| Provenance View | • Compress or summarize the provenance graph |

## Future Works

- Implementation of the provenance model
- Utilization of the model in real world data processing systems
- Investigating how to link multiple provenance of different granularities