Capturing provenance information with Kieker.WorkflowMonitor

Peer C. Brauer
Wilhelm Hasselbring

Software Engineering Group
Christian Albrechts Universität

Agenda

- PubFlow
- An introduction to Kieker
- Collecting provenance data with Kieker

PubFlow

The project group

Software Engineering Group
Library of CAU
Computing Center of CAU

Kiel Datamanagement Team
Library of Geomar
Data and Computing Center Geomar

Leibniz-Informationszentrum
Wirtschaft
Leibniz Information Centre
for Economics

ozean der zukunft
DIE KIELER MEERESWISSENSCHAFTEN

What is PubFlow about?

- **Pub**lication Work**Flow**s

What is PubFlow about?

The goal of the project is it, to create a workflow environment for the work with scientific data based on established business workflow systems like Apache ODE, which increases the degree of automation in the publication process.

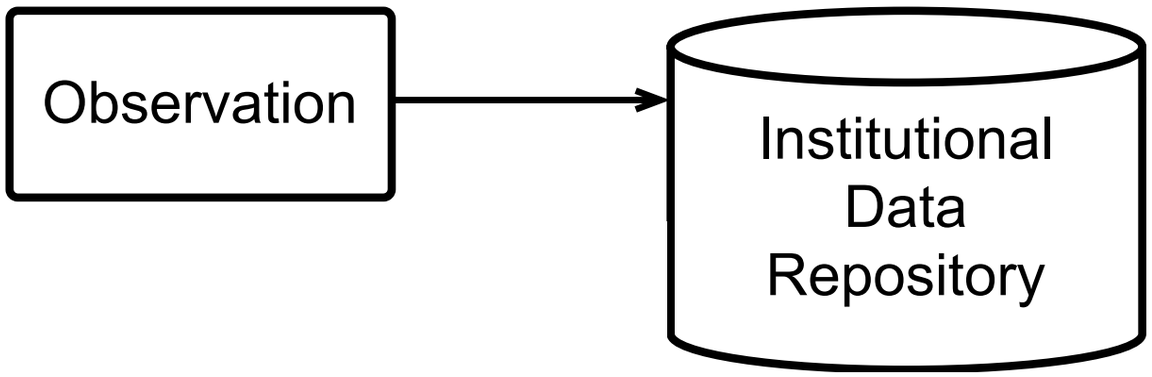
What is PubFlow about?

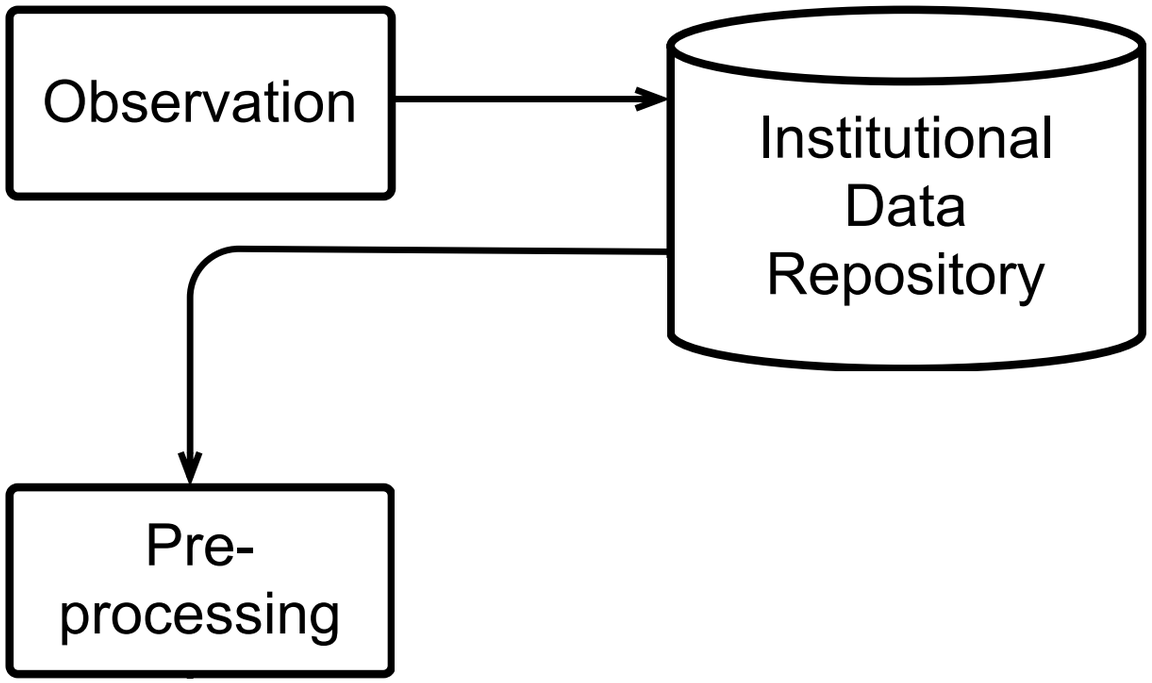
The goal of the project is it, to create a workflow environment for the work with scientific data based on established business workflow systems like Apache ODE, which increases the degree of automation in the publication process.

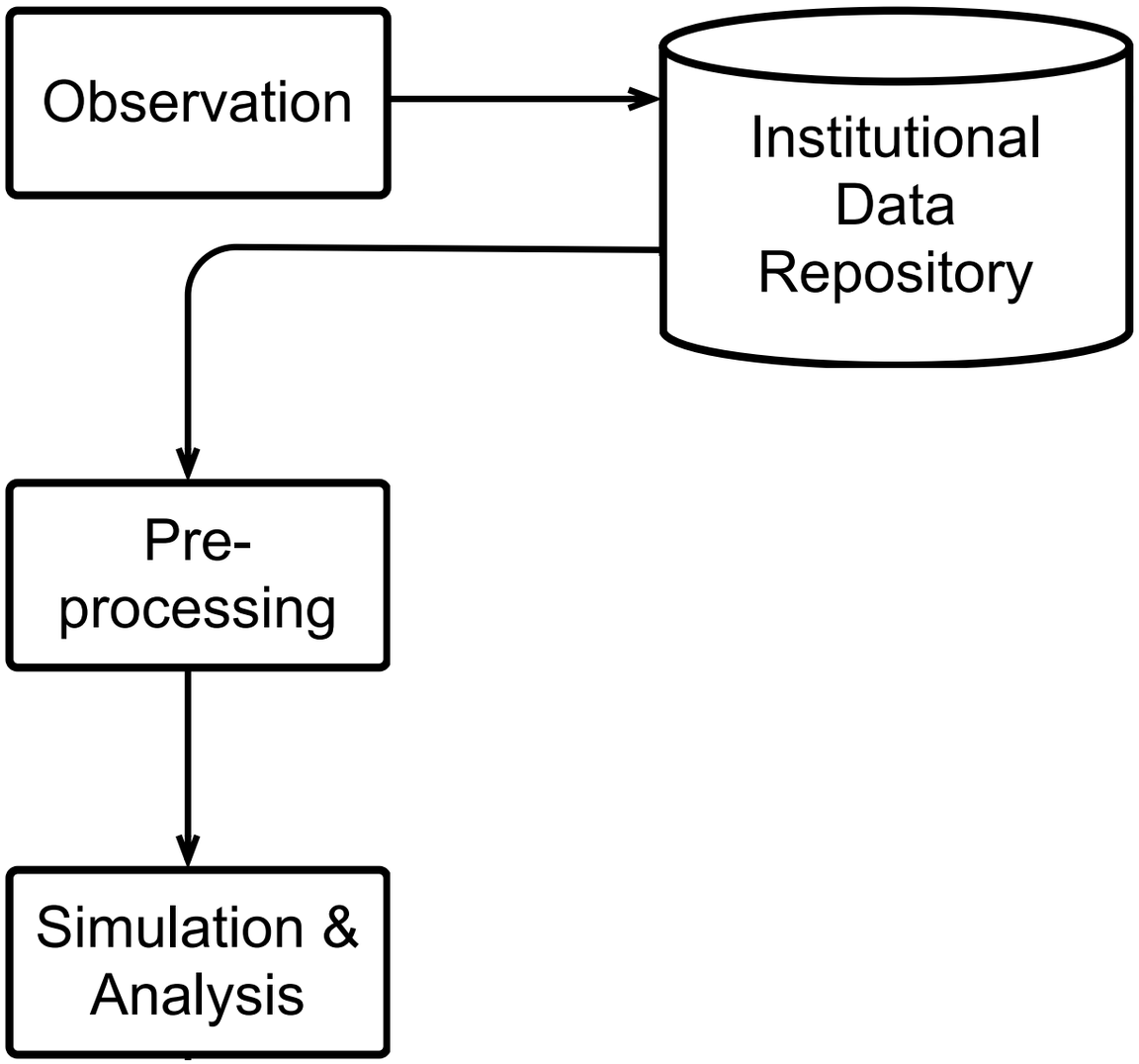
What is PubFlow about?

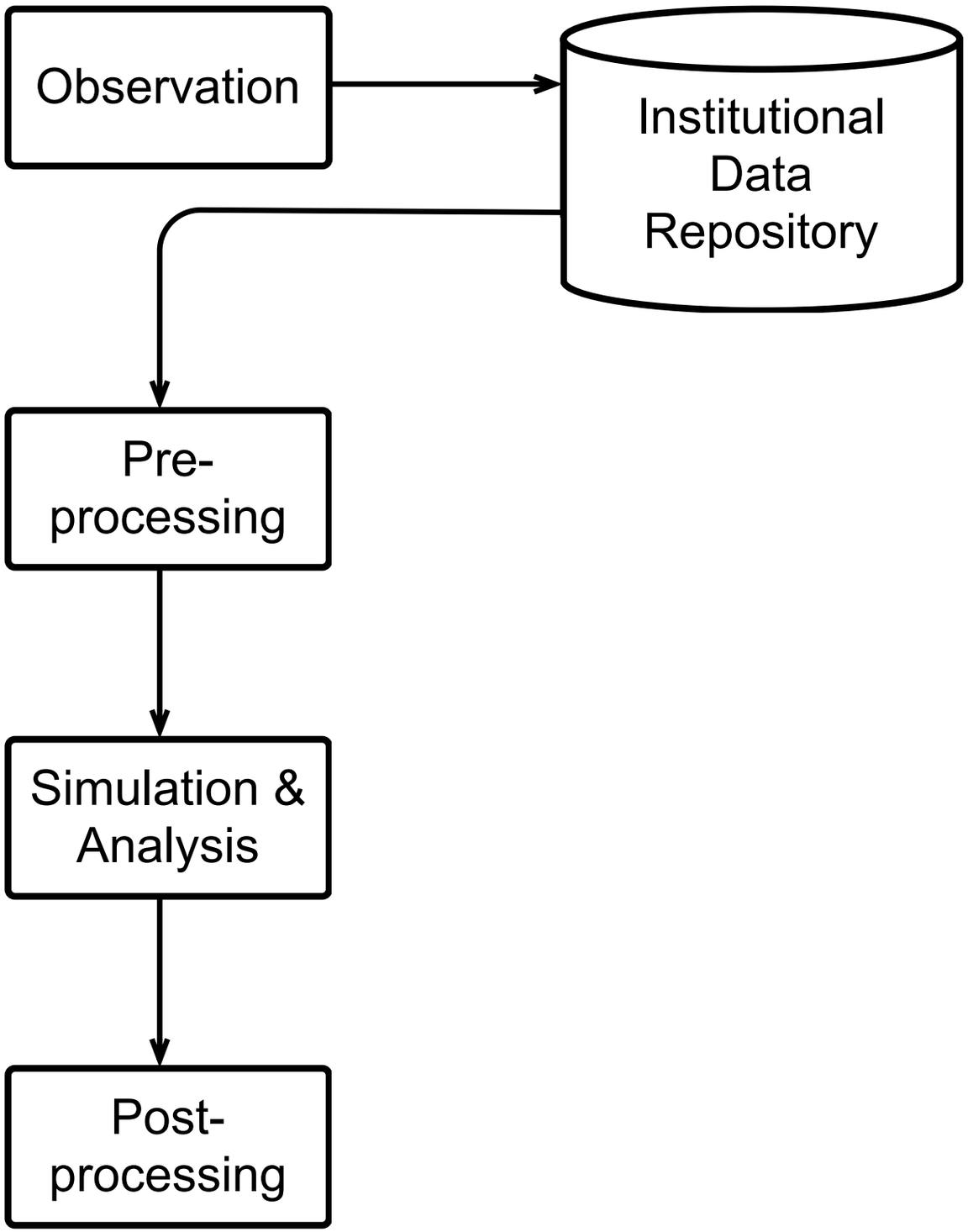
The goal of the project is it, to create a workflow environment for the work with scientific data based on established business workflow systems like Apache ODE, which increases the degree of automation in the publication process.

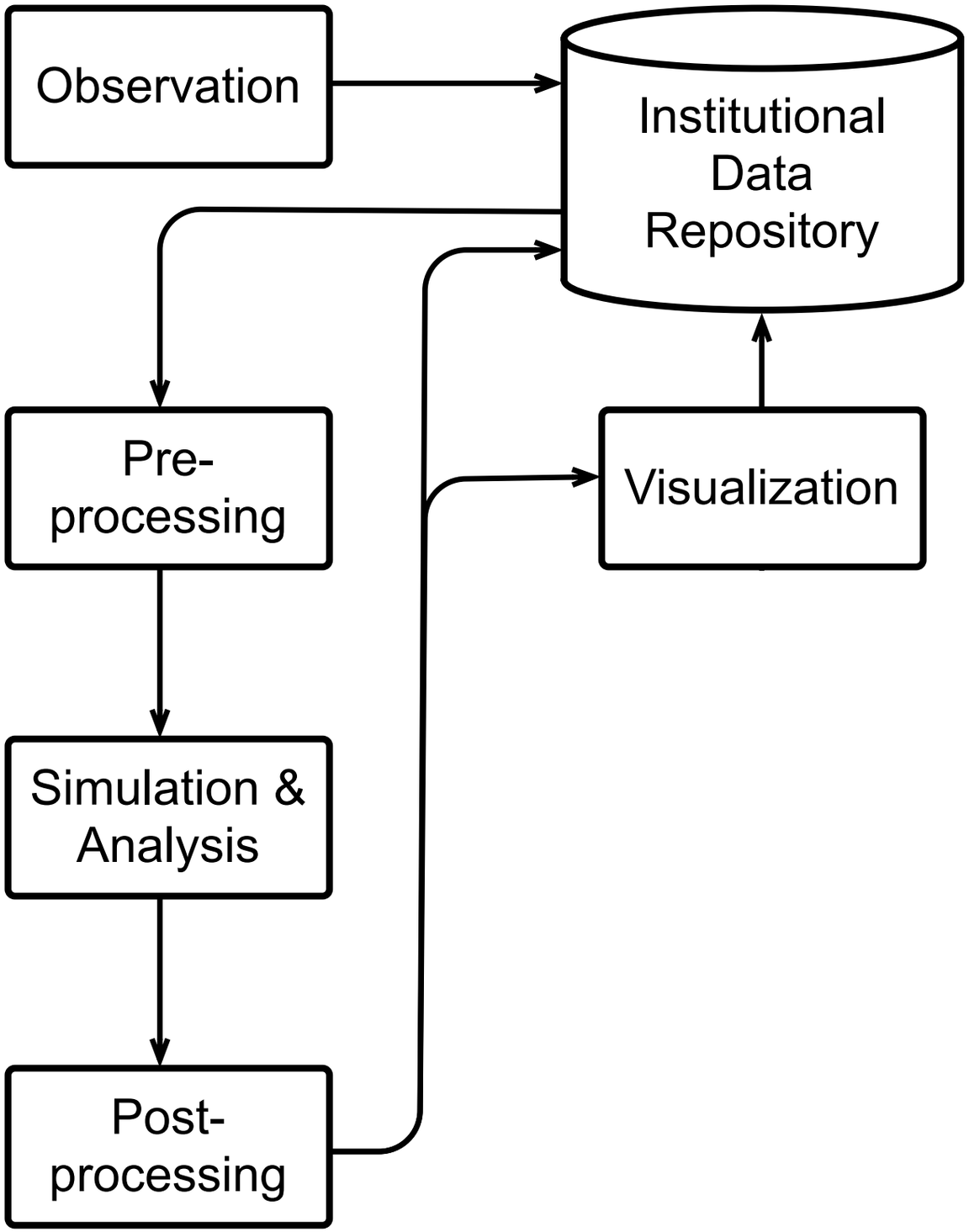
Observation

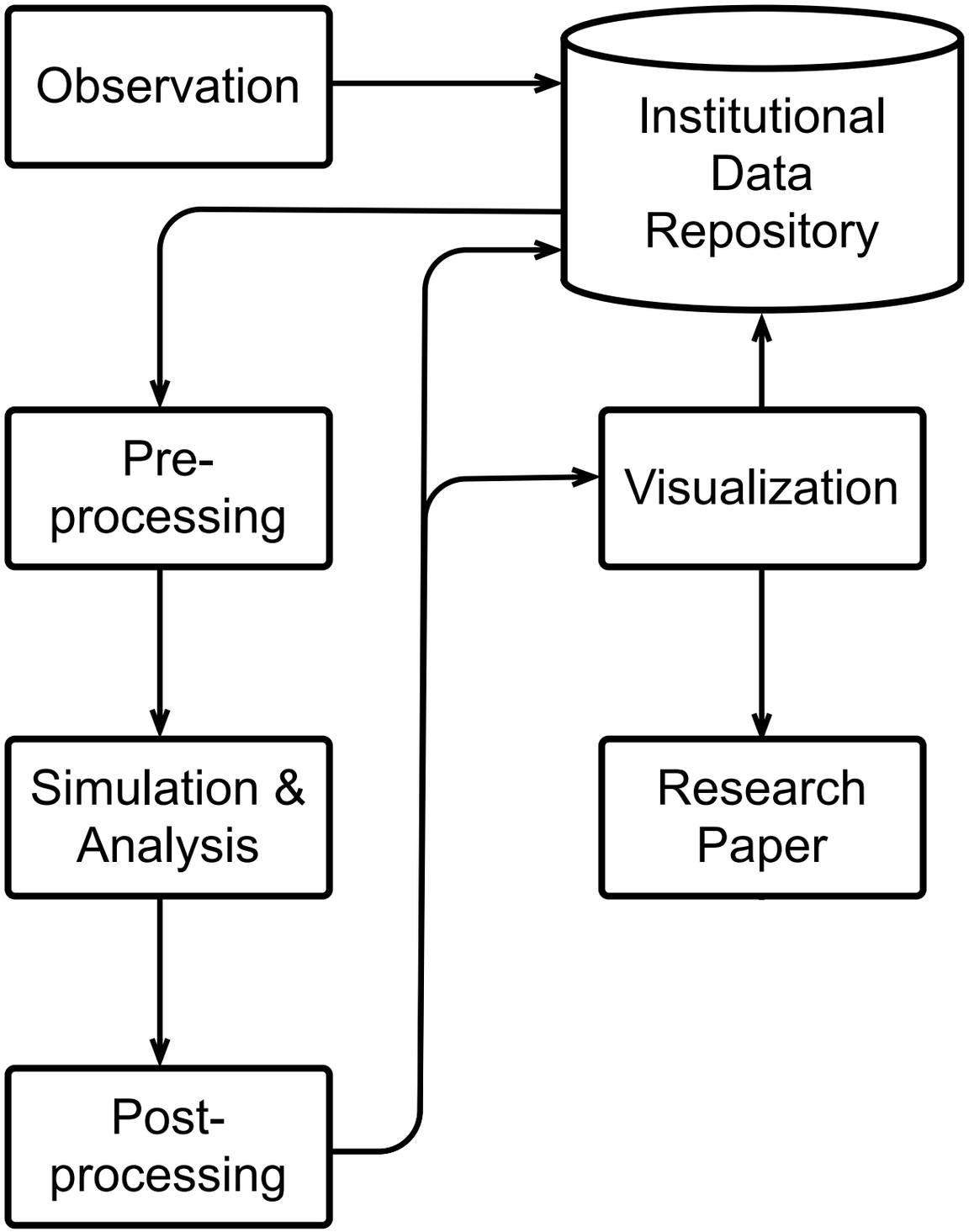

Observation
Institutional
Data
Repository

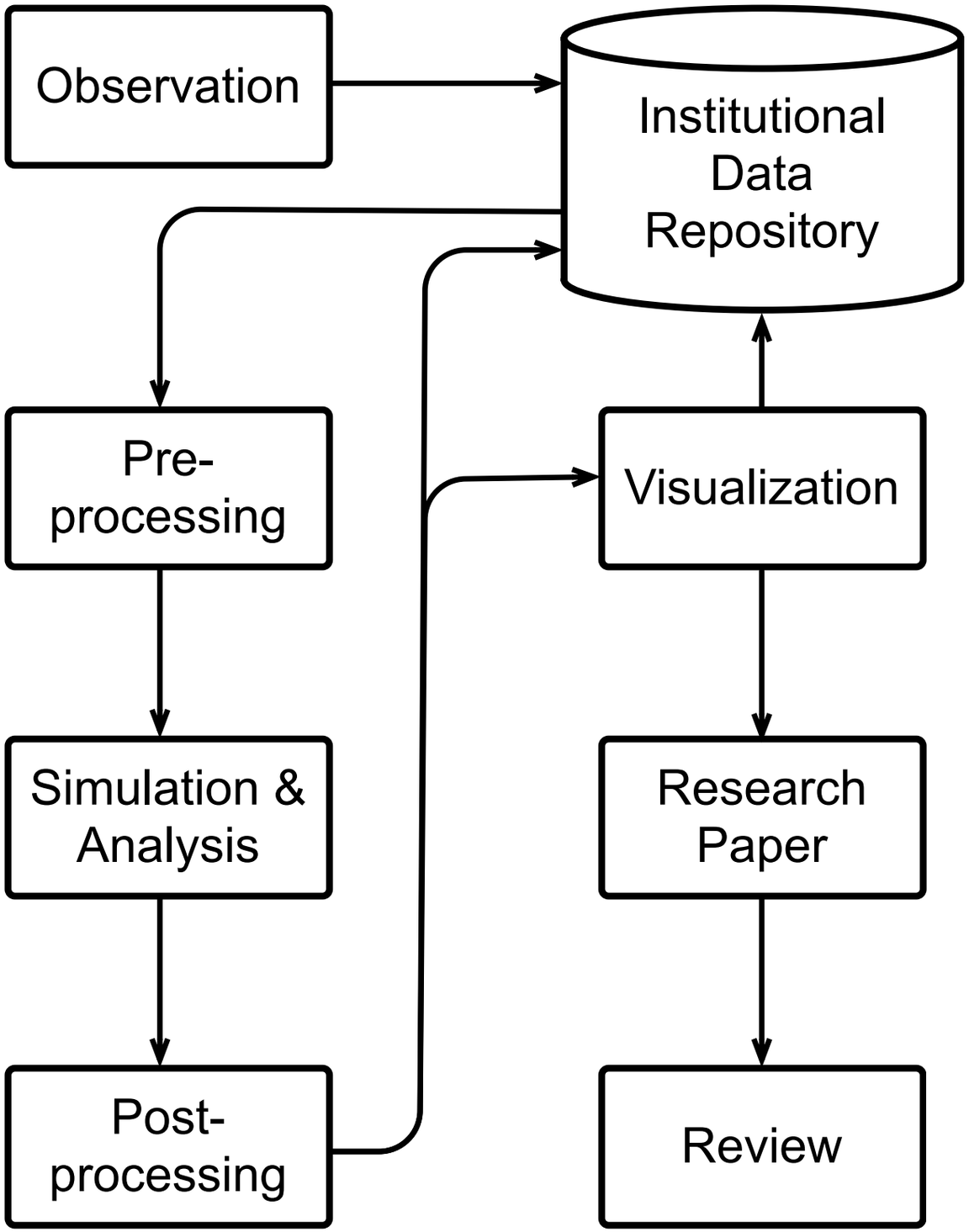

Observation
Institutional
Data
Repository
Pre-
processing

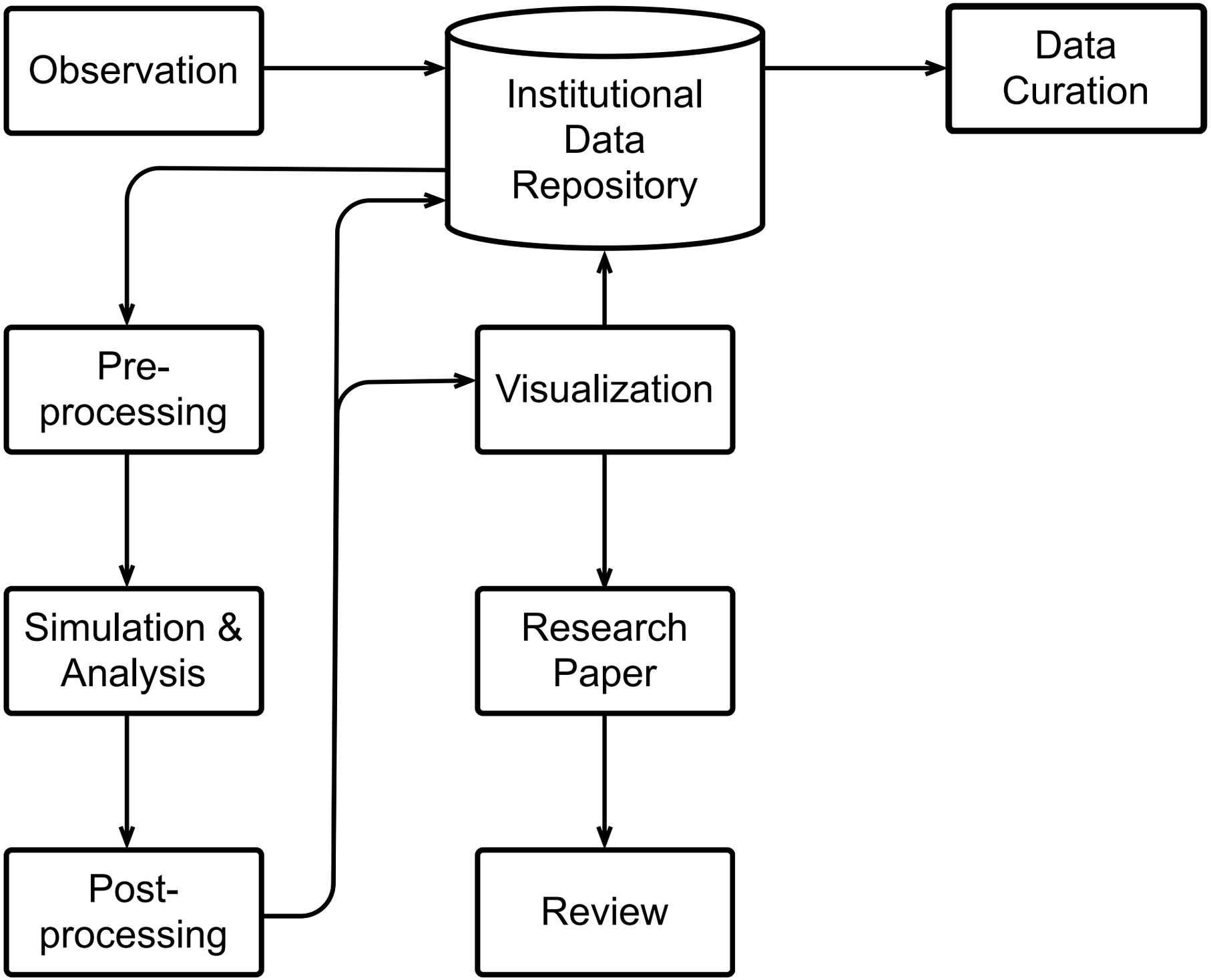

Observation
Institutional Data Repository
Pre-processing
Simulation & Analysis

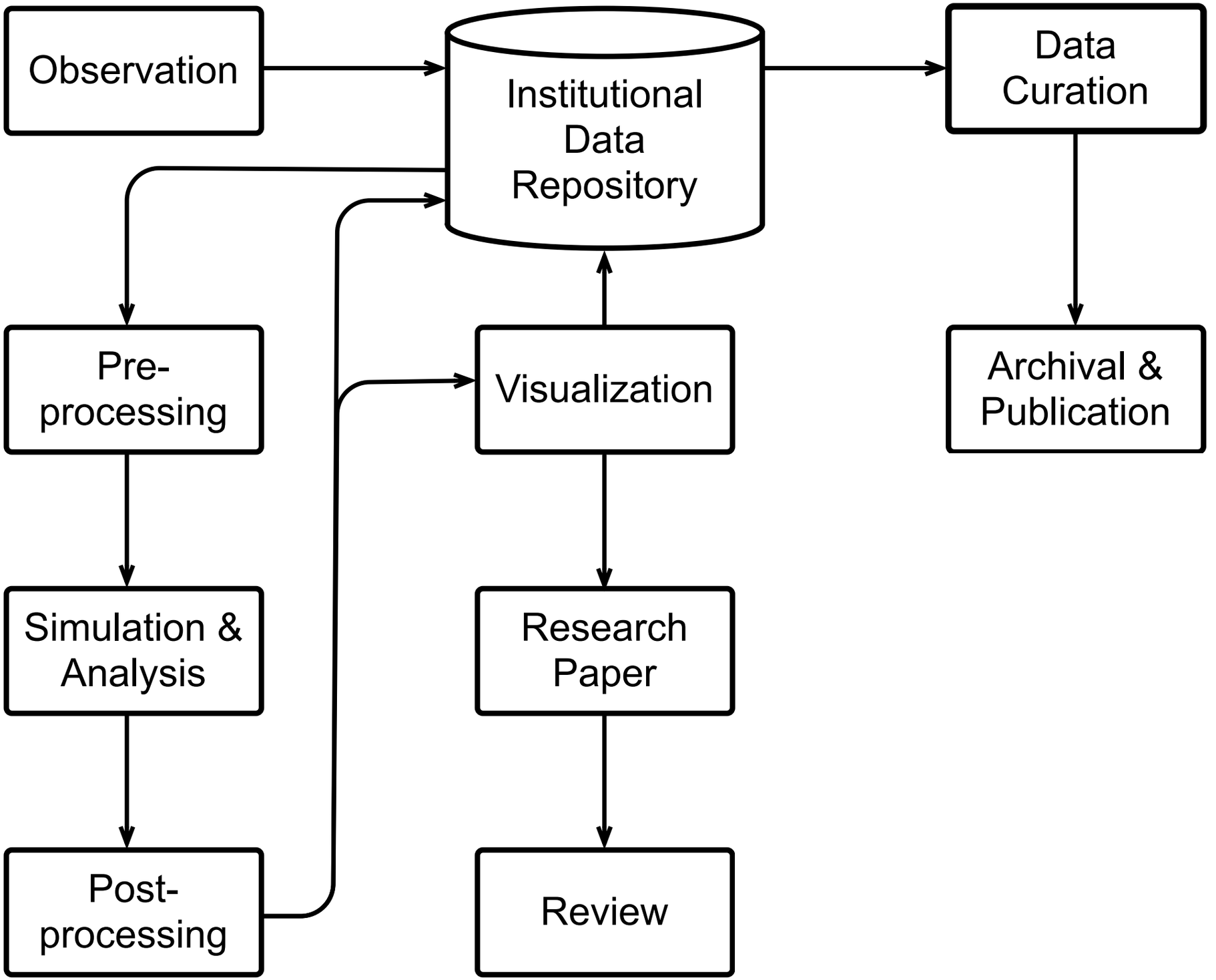

Observation
Institutional Data Repository
Pre-processing
Simulation & Analysis
Post-processing

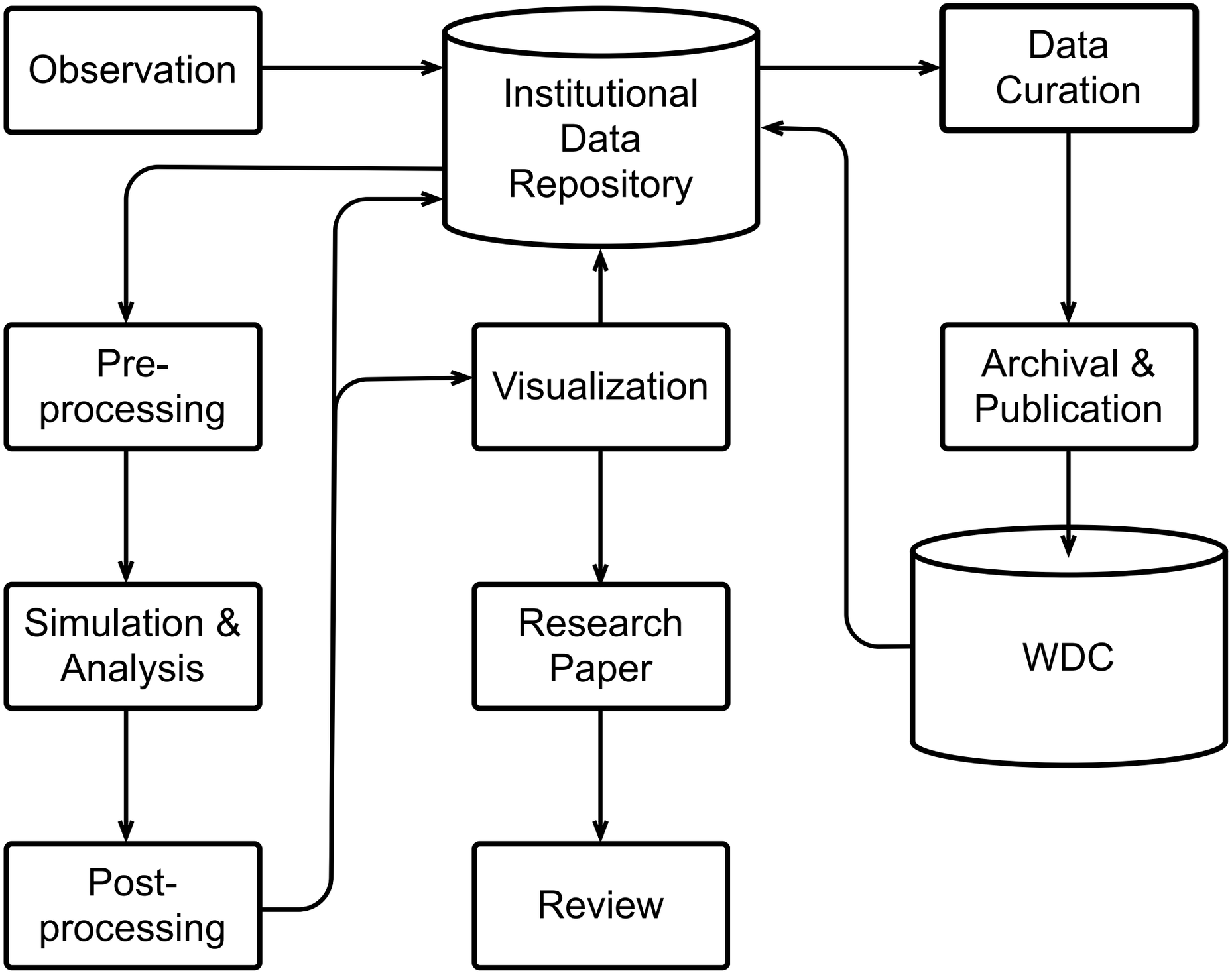

Observation
Institutional Data Repository
Pre-processing
Visualization
Simulation & Analysis
Post-processing

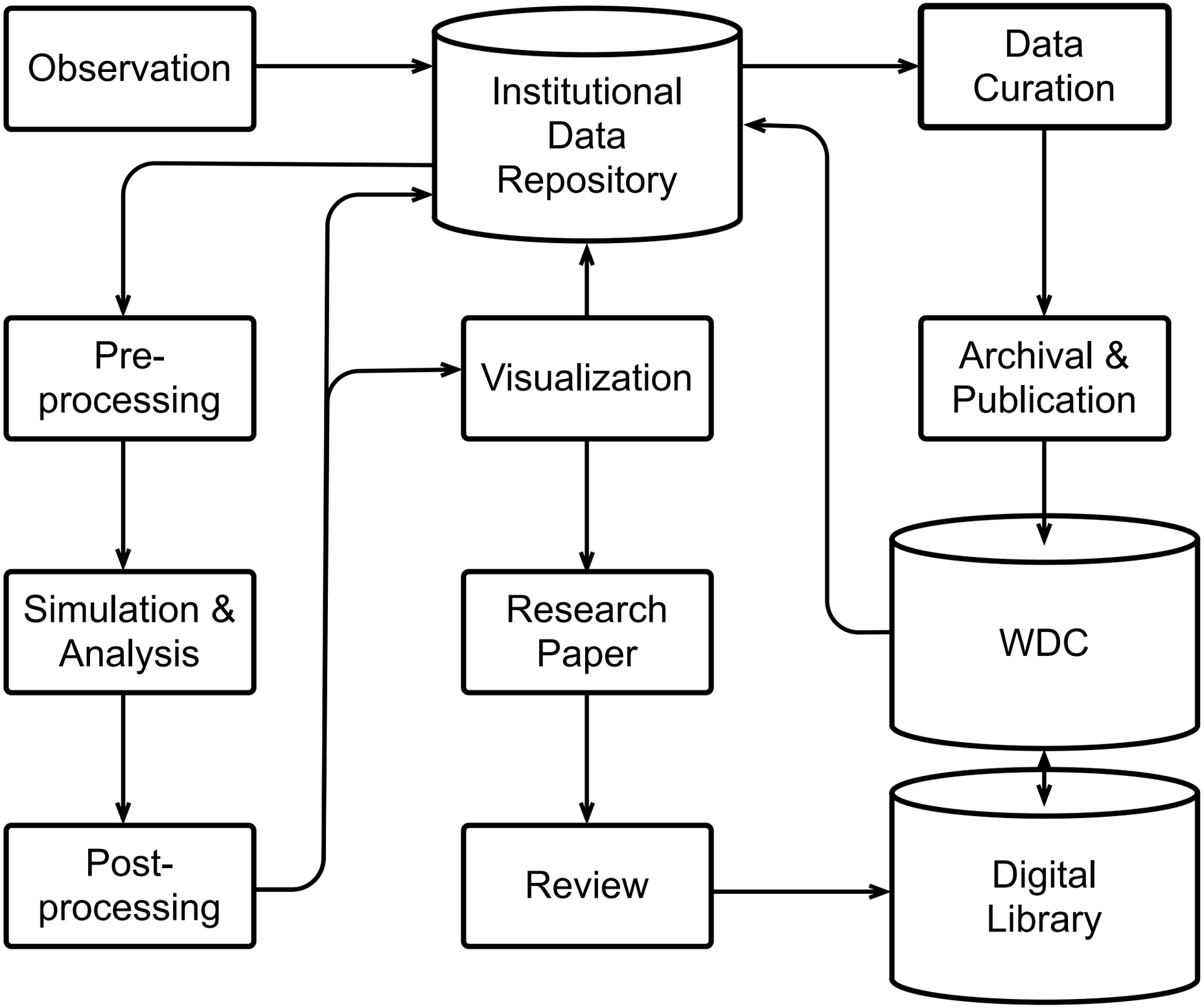

Observation
Institutional Data Repository
Pre-processing
Visualization
Simulation & Analysis
Research Paper
Post-processing

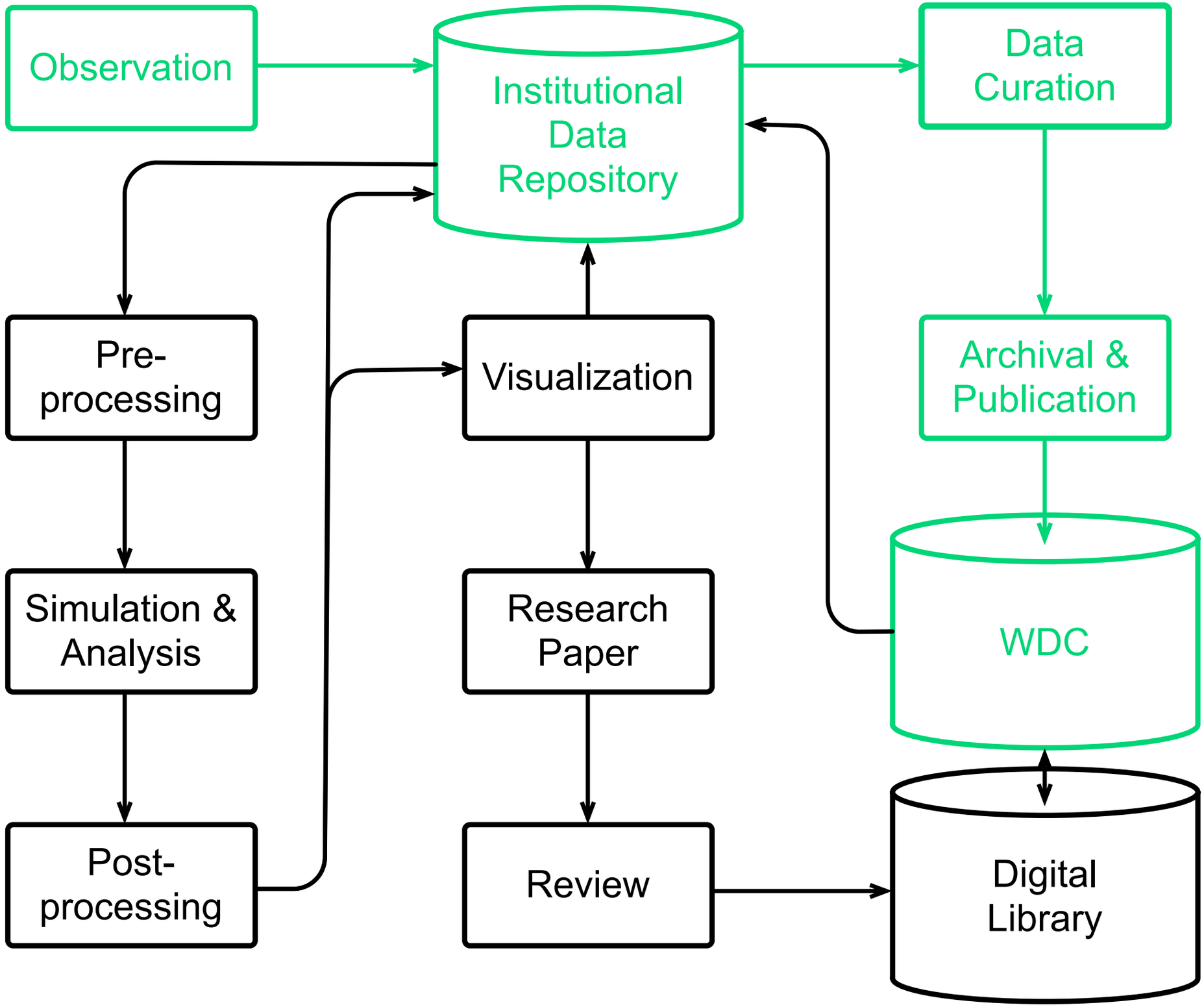

Observation
Institutional Data Repository
Pre-processing
Visualization
Simulation & Analysis
Research Paper
Post-processing
Review


Observation
Institutional Data Repository
Data Curation
Pre-processing
Visualization
Simulation & Analysis
Research Paper
Post-processing
Review


Observation
Institutional Data Repository
Data Curation
Pre-processing
Visualization
Archival & Publication
Simulation & Analysis
Research Paper
Post-processing
Review


Observation
Institutional Data Repository
Data Curation
Pre-processing
Visualization
Archival & Publication
Simulation & Analysis
Research Paper
WDC
Post-processing
Review


Observation
Institutional Data Repository
Data Curation
Pre-processing
Visualization
Archival & Publication
Simulation & Analysis
Research Paper
WDC
Post-processing
Review
Digital Library


Observation
Institutional Data Repository
Data Curation
Pre-processing
Visualization
Archival & Publication
Simulation & Analysis
Research Paper
WDC
Post-processing
Review
Digital Library

the case study

Kieker
a short introduction

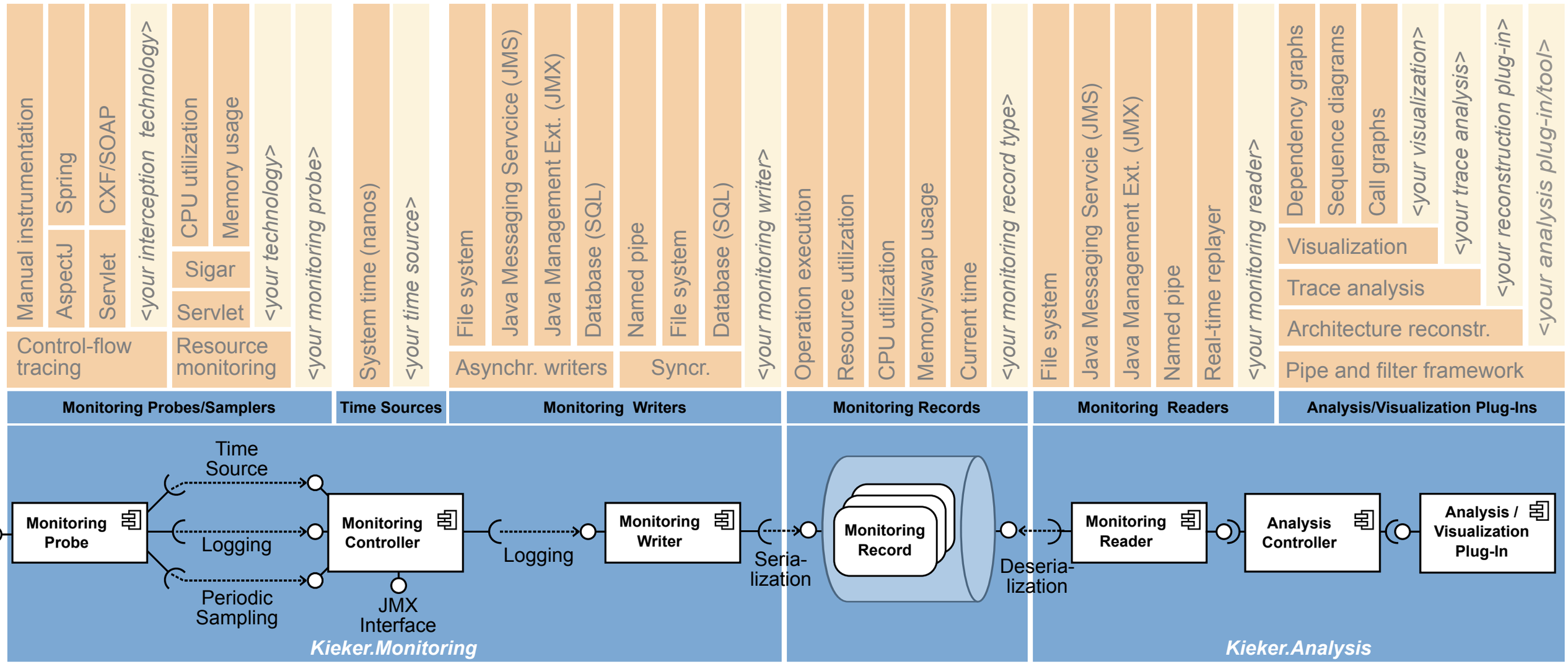
What is Kieker?

- Kieker is a modular monitoring framework
- developed by the Software Engineering Group CAU Kiel
- Kieker is open source (APL 2)

What is Kieker?

Kieker is distributed as part of SPEC® RG's repository of peer-reviewed tools for quantitative system evaluation and analysis

http://research.spec.org/projects/tools.html


Manual instrumentation
AspectJ
Servlet
<your interception technology>
Control-flow tracing
Spring
CXF/SOAP
CPU utilization
Memory usage
Sigar
Servlet
<your technology>
Resource monitoring
<your monitoring probe>
Monitoring Probes/Samplers
System time (nanos)
<your time source>
Time Sources
File system
Java Messaging Servcie (JMS)
Java Management Ext. (JMX)
Database (SQL)
Asynchr. writers
Named pipe
File system
Database (SQL)
Syncr.
<your monitoring writer>
Monitoring Writers
Operation execution
Resource utilization
CPU utilization
Memory/swap usage
Current time
<your monitoring record type>
Monitoring Records
File system
Java Messaging Servcie (JMS)
Java Management Ext. (JMX)
Named pipe
Real-time replayer
<your monitoring reader>
Monitoring Readers
Dependency graphs
Sequence diagrams
Call graphs
<your visualization>
Visualization
Trace analysis
Architecture reconstr.
Pipe and filter framework
<your trace analysis>
<your reconstruction plug-in>
<your analysis plug-in/tool>
Analysis/Visualization Plug-Ins
Monitoring Probe
Time Source
Logging
Periodic Sampling
Monitoring Controller
JMX Interface
Logging
Monitoring Writer
Seria-lization
Monitoring Record
Deseria-lization
Monitoring Reader
Analysis Controller
Analysis / Visualization Plug-In
Kieker.Monitoring
Kieker.Analysis

What is the scope of Kieker?

- Kieker is for application monitoring - not for profiling

What is the scope of Kieker?

- Kieker is for application monitoring - not for profiling

Monitoring of an application during runtime

What is the scope of Kieker?

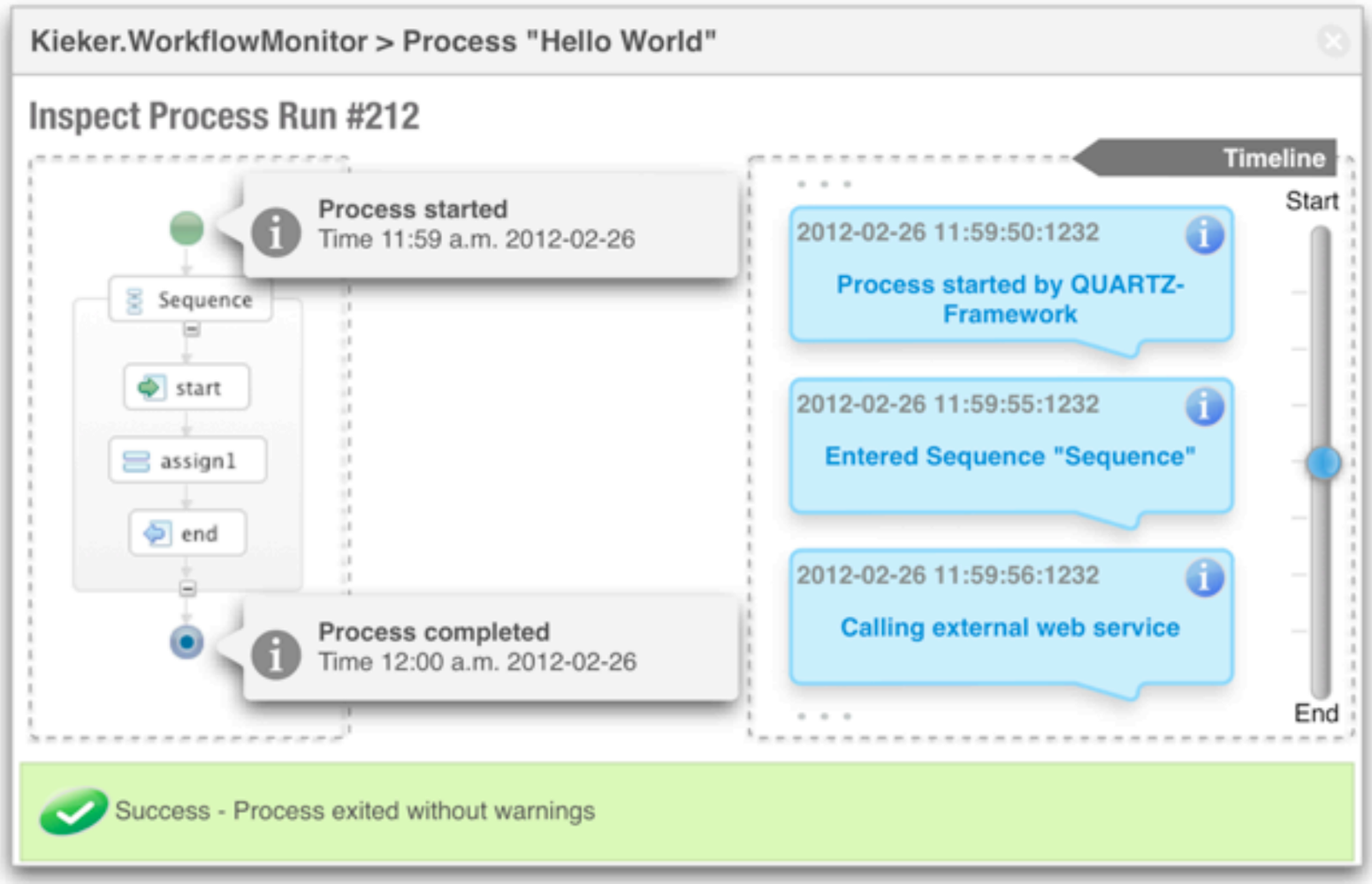
- Kieker is for application monitoring - not for profiling

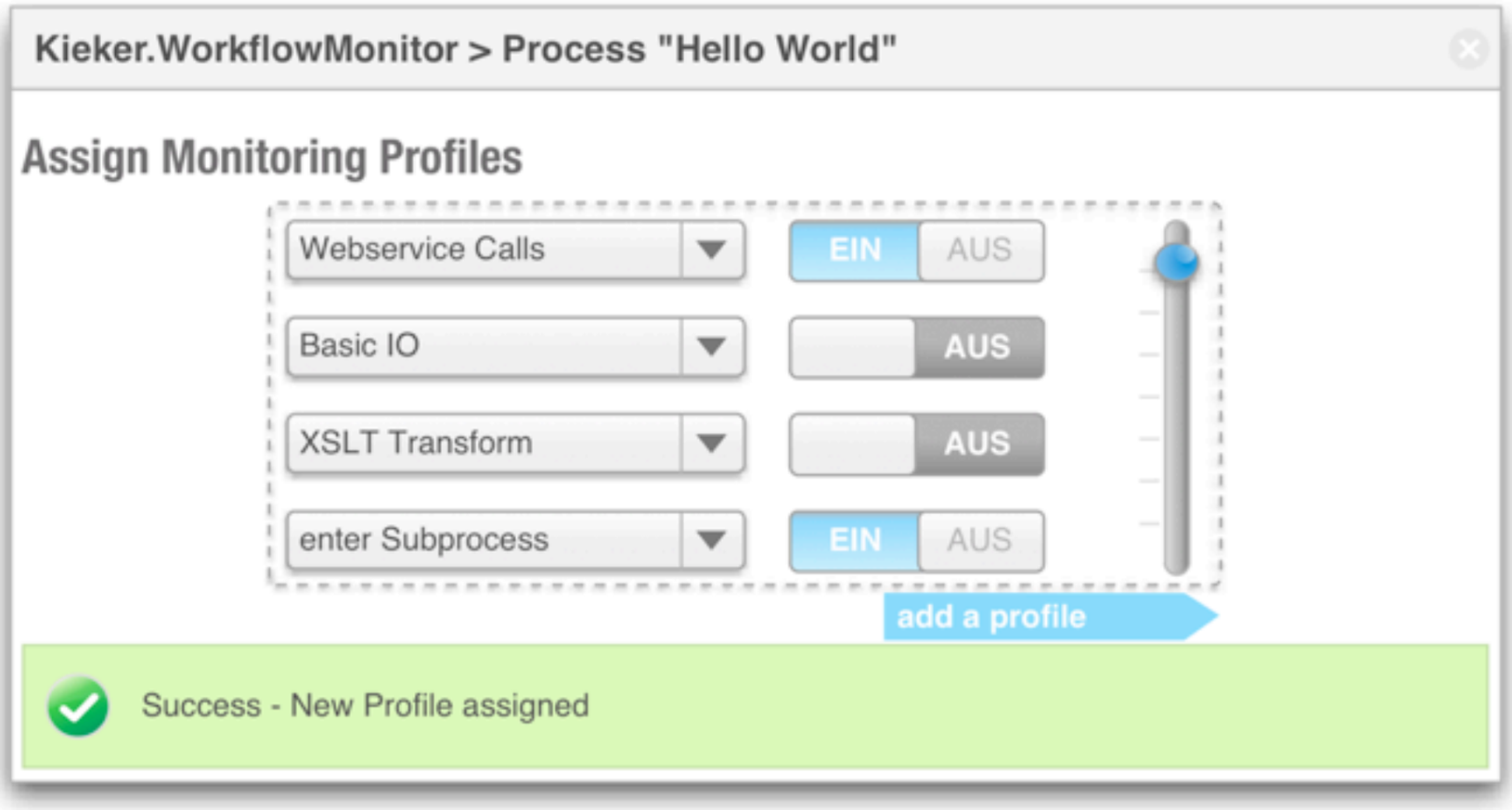
measure application performance during development stage

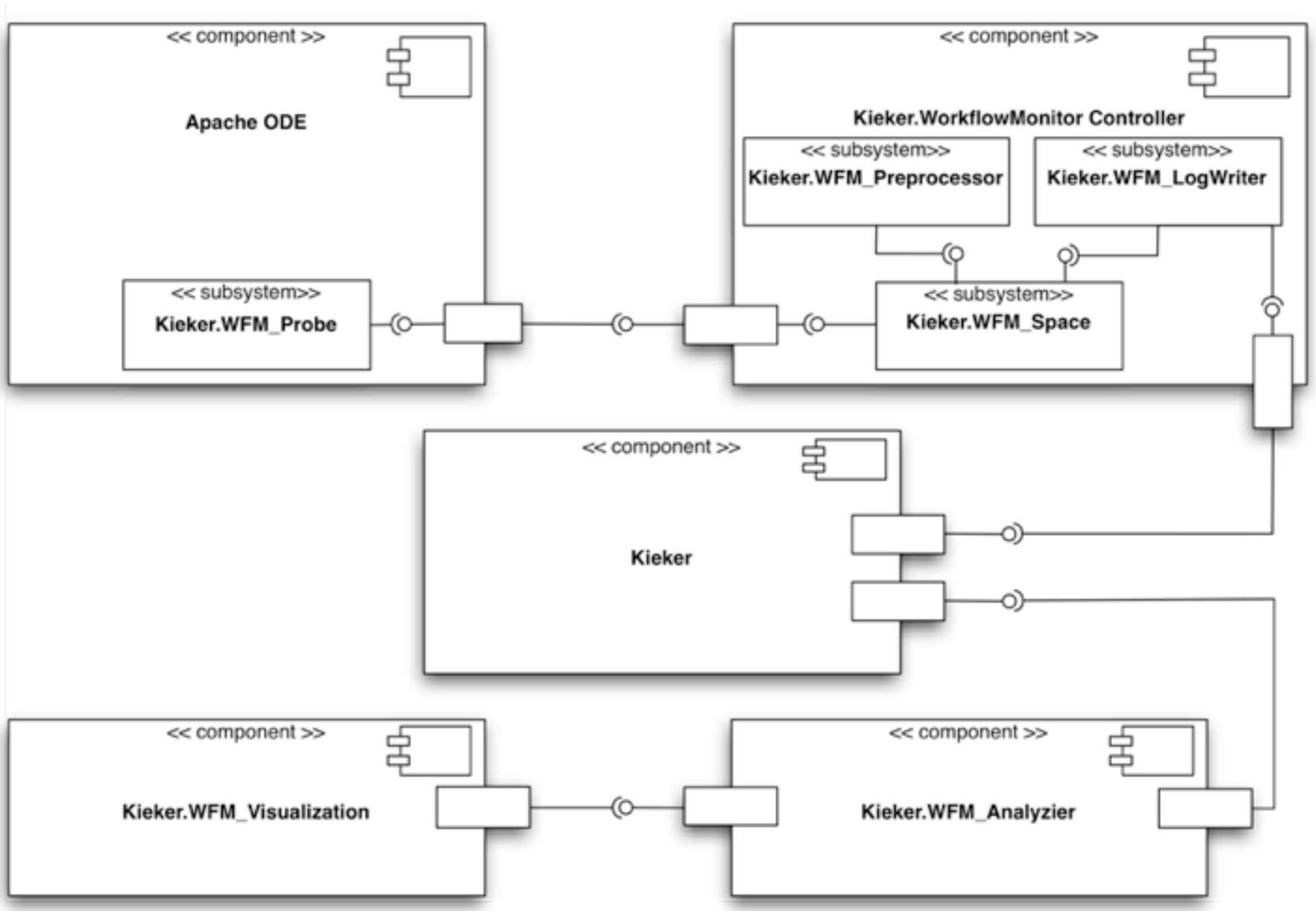
Collecting provenance data with Kieker

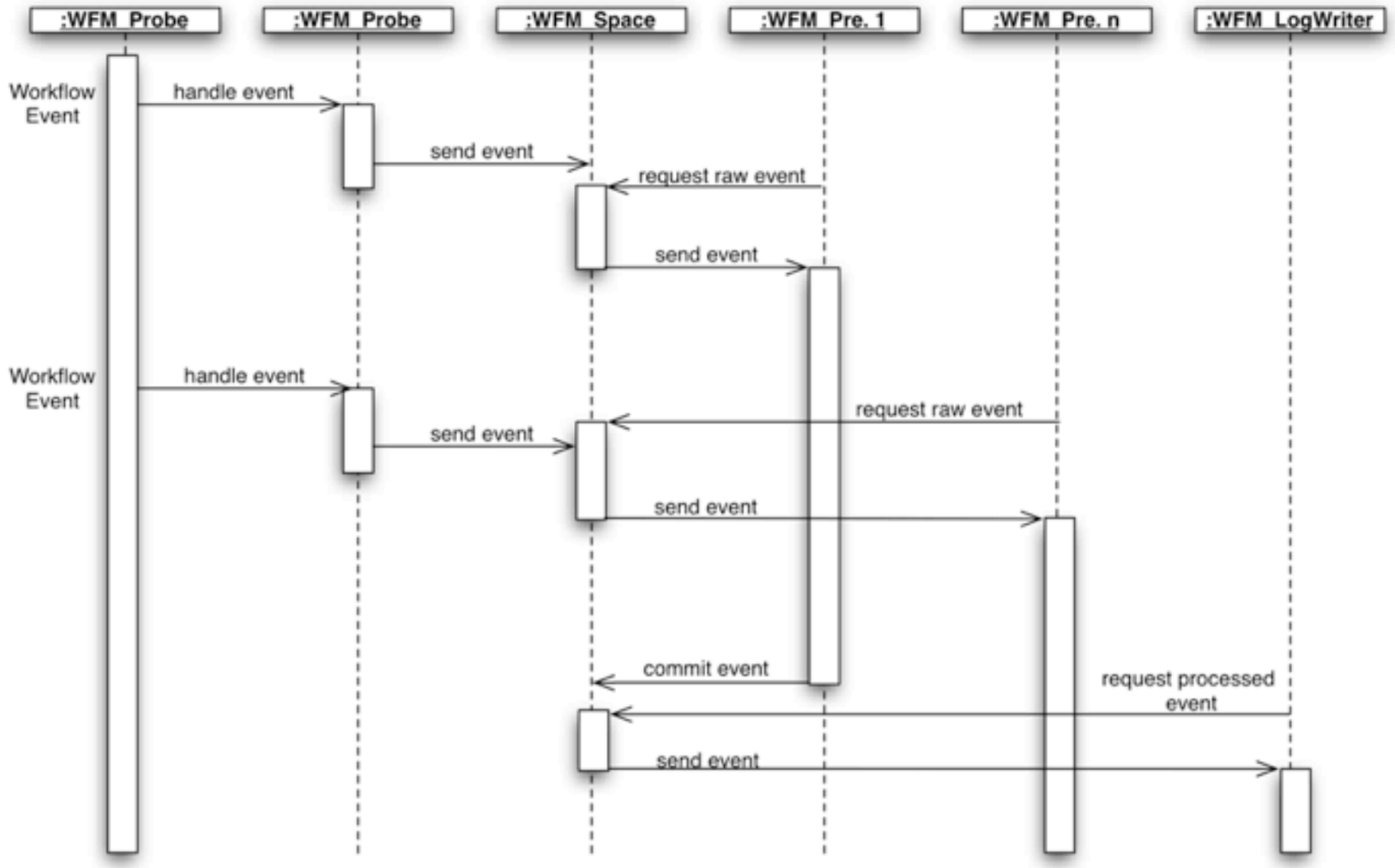
requirements

- …


Kieker.WorkflowMonitor > Process "Hello World"
Inspect Process Run #212
Process started
Time 11:59 a.m. 2012-02-26
Sequence
start
assign1
end
Process completed
Time 12:00 a.m. 2012-02-26
Timeline
Start
2012-02-26 11:59:50:1232
Process started by QUARTZ-Framework
2012-02-26 11:59:55:1232
Entered Sequence "Sequence"
2012-02-26 11:59:56:1232
Calling external web service
End
Success - Process exited without warnings


Kieker.WorkflowMonitor > Process "Hello World"
Assign Monitoring Profiles
Webservice Calls
EIN
AUS
Basic IO
AUS
XSLT Transform
AUS
enter Subprocess
EIN
AUS
add a profile
Success - New Profile assigned


<< component >>
Apache ODE
<< subsystem>>
Kieker.WFM_Probe
<< component >>
Kieker.WorkflowMonitor Controller
<< subsystem>>
Kieker.WFM_Preprocessor
<< subsystem>>
Kieker.WFM_LogWriter
<< subsystem>>
Kieker.WFM_Space
<< component >>
Kieker
<< component >>
Kieker.WFM_Visualization
<< component >>
Kieker.WFM_Analyzier


:WFM_Probe
:WFM_Probe
:WFM_Space
:WFM_Pre. 1
:WFM_Pre. n
:WFM_LogWriter
Workflow Event
handle event
send event
request raw event
send event
Workflow Event
handle event
send event
request raw event
send event
commit event
request processed event
send event

```
<monitor:profile name="provenanceLight„>
  <monitor:targetelement type="activity„ >
    <monitor:eventtype>activityEnabledEvent</monitor:eventtype>
    <monitor:eventtype>activityDisabledEvent</monitor:eventtype>
  </monitor:targetelement>
  ...
  <monitor:targetelement type="variable„ >
    <monitor:eventtype>VariableModificationEvent</monitor:eventtype>
    <monitor:eventtype>VariableReadEvent</monitor:eventtype>
  </monitor:targetelement>
  ...
  <monitor:event type="VariableModificationEvent„>
    <monitor:fokus>varName</monitor:fokus>
    <monitor:fokus>newValue</monitor:fokus>
  </monitor:event>
</monitor:profile>
```