

Georgia Tech Gesture Toolkit:

Supporting Experiment in Gesture Recognition

Kent Lyons
Helene Brashear
Tracy Westeyn
Jung Soo Kim
Thad Starner

Contextual Computing Group
College of Computing
Georgia Institute of Technology
Atlanta, GA USA

gt²k

Motivation

- Gesture is becoming more common for interfaces
- Building gesture recognition systems is very difficult

gt²k is...

- A user interface toolkit designed to enable the development of gesture-based applications.
- Written in Java for cross-platform use and easy integration into graphical user interface development tools.

When to use gt²k

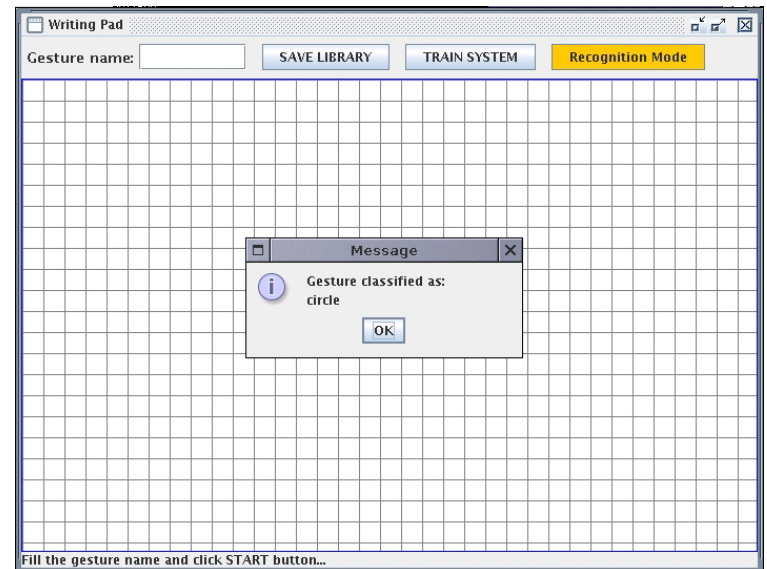
- Applications using symbolic/iconic gesture
 - Sign language
 - handwriting
- Applicable to research in many fields
 - Human-Computer Interaction
 - Assistive Technologies
 - Robotics
 - Brain Research
 - etc.

Outline

- Example applications
 - WritingPad
 - Accelerometer
 - Camera
- gt²k architecture
 - Overview
 - Sensor
 - Library
 - Machine Learning
 - Recognition Process
 - Data collection
 - Training
 - Recognition
- In-depth sample example
 - How to make your own application

Application : WritingPad

- Sensor : mouse sensor
- Allows a user to draw a gesture with a mouse



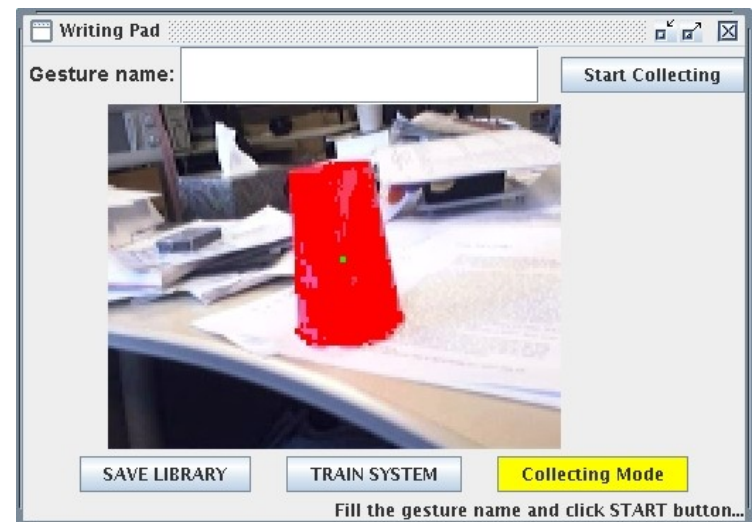
Application : Accelerometer

- Sensor : Accelerometer
- Collects information about users' physical movement

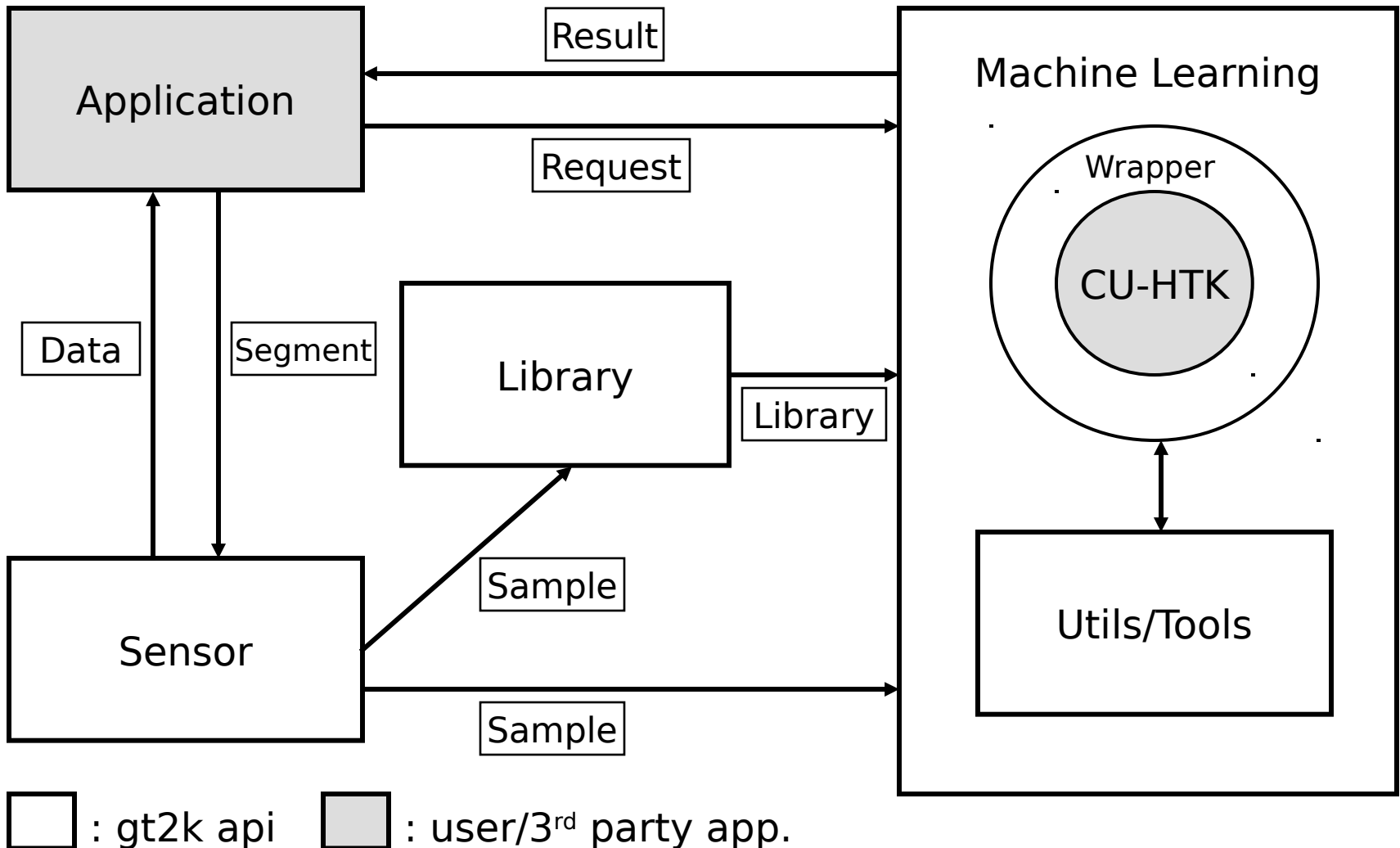


Application : Vision Tracker

- Sensor : Image Sensor
- Tracks the movement of objects

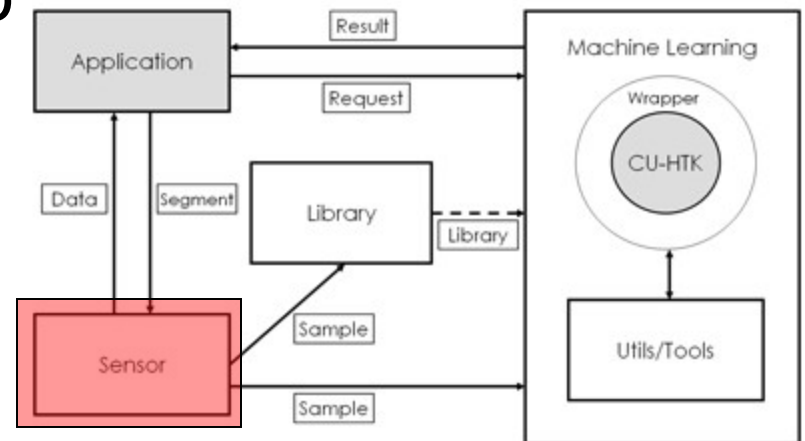


Architecture overview



Sensor

- Interfaces with the hardware and collect data.
- Provides parsing or post-processing of the data.
- Designed around an event-based architecture.
- Allows for both synchronous or asynchronous reading of sensors.

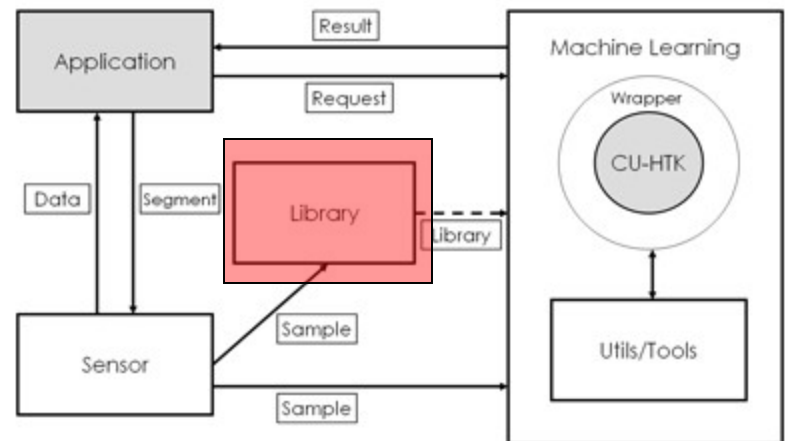


Library

- Responsible for storing and organizing data.
- Composed of a collection of samples.

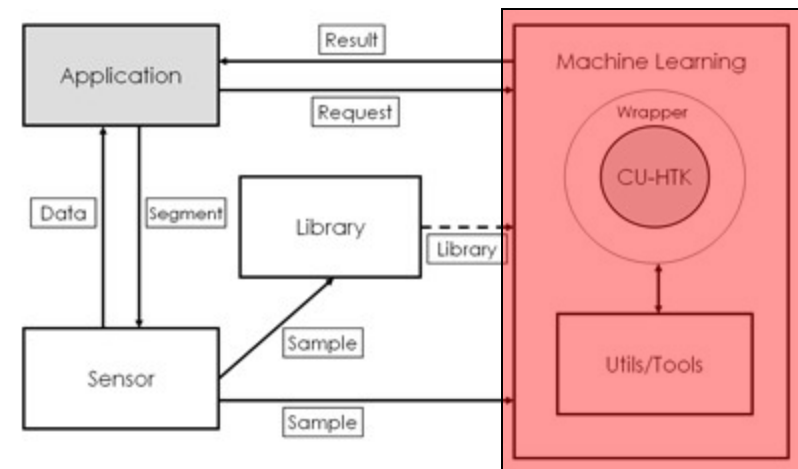
```
- <?xml version="1.0" encoding="UTF-8"?>
  <library name="WritingPadLib">
    <samples>
      <sample id="1">
        <tag end="144" label="star" start="0"/>
        <fvectors>
          <v>0.0 0.0</v>
          <v>-0.32175055 3.1622777</v>
```

```
:
    <v>-1.815775 4.1231055</v>
  </fvectors>
</sample>
</samples>
</library>
```

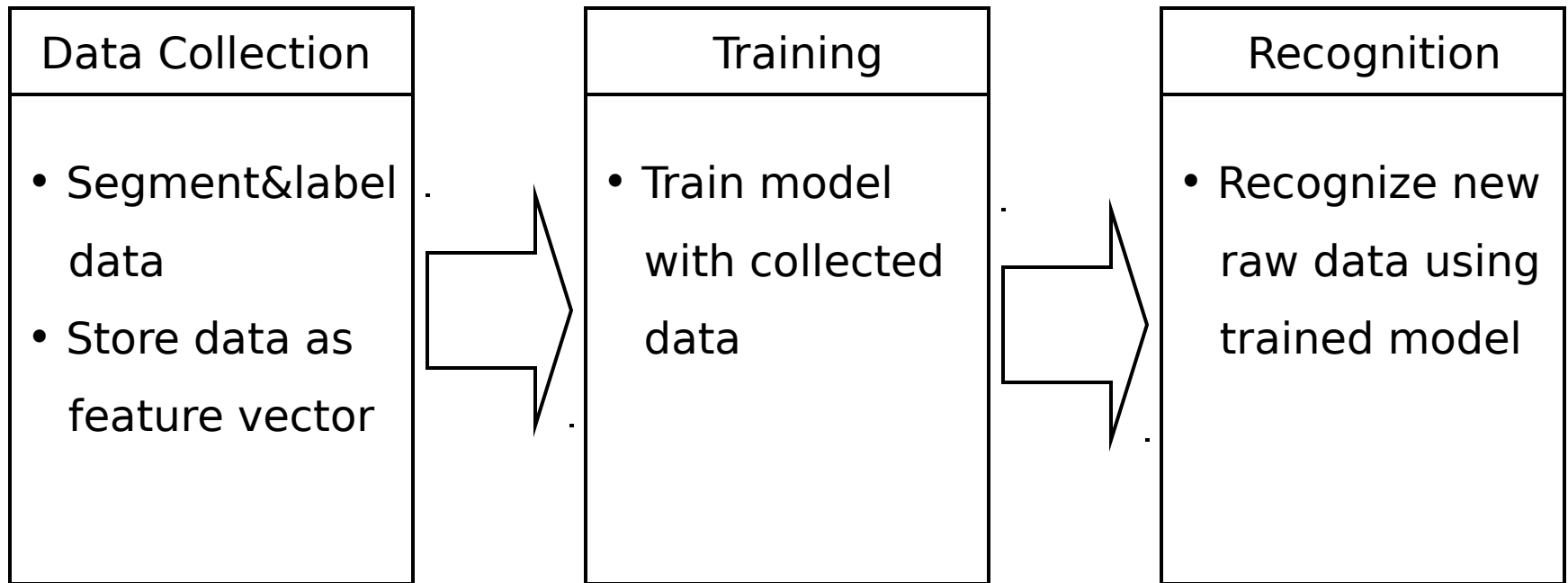


Machine Learning

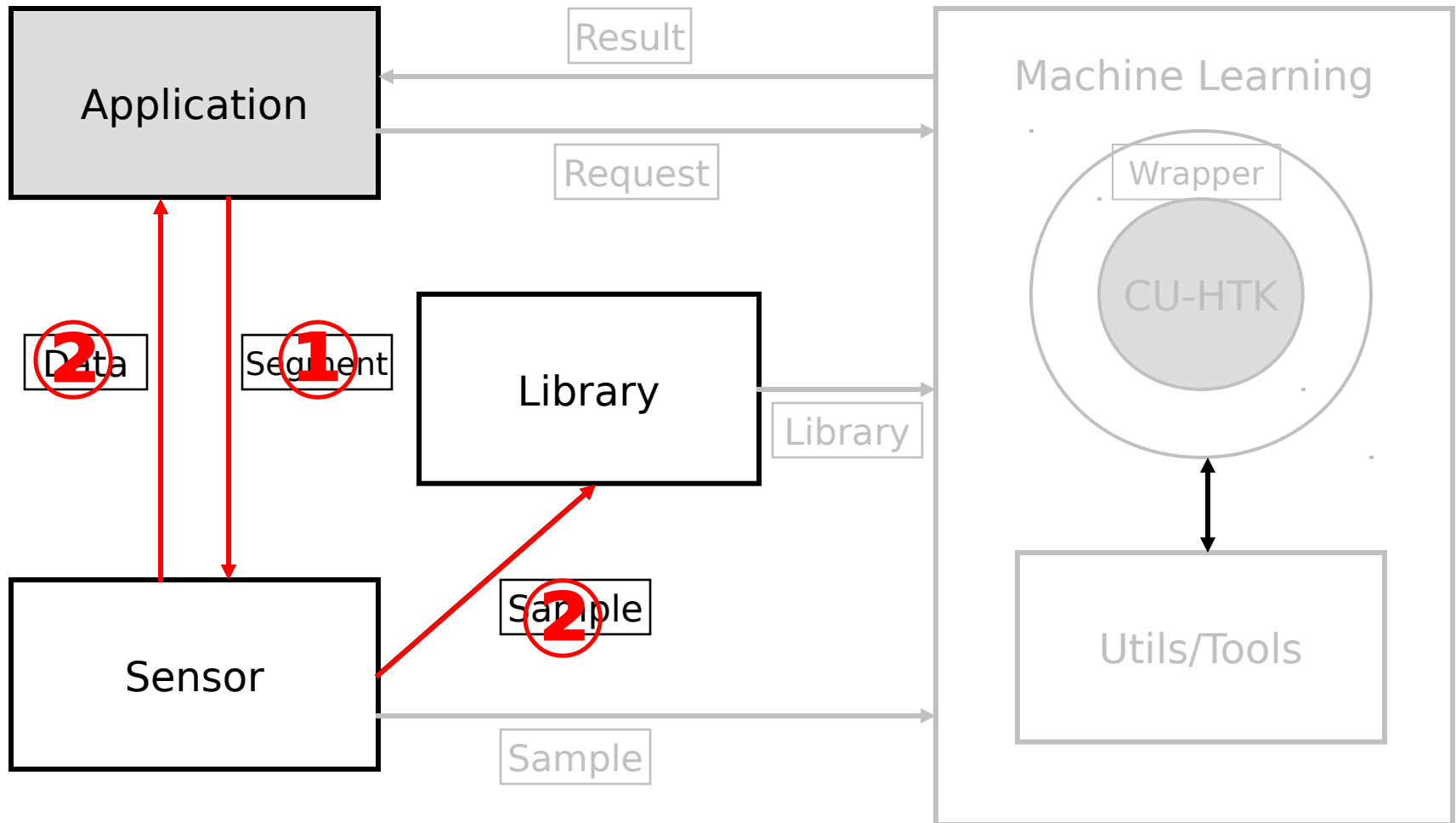
- Provides the toolkit's abstraction for the machine learning algorithms.
- Used for modeling data samples (training) and recognizing gesture samples.
- Utilizes Cambridge University's Hidden Markov Model Toolkit (CU-HTK).



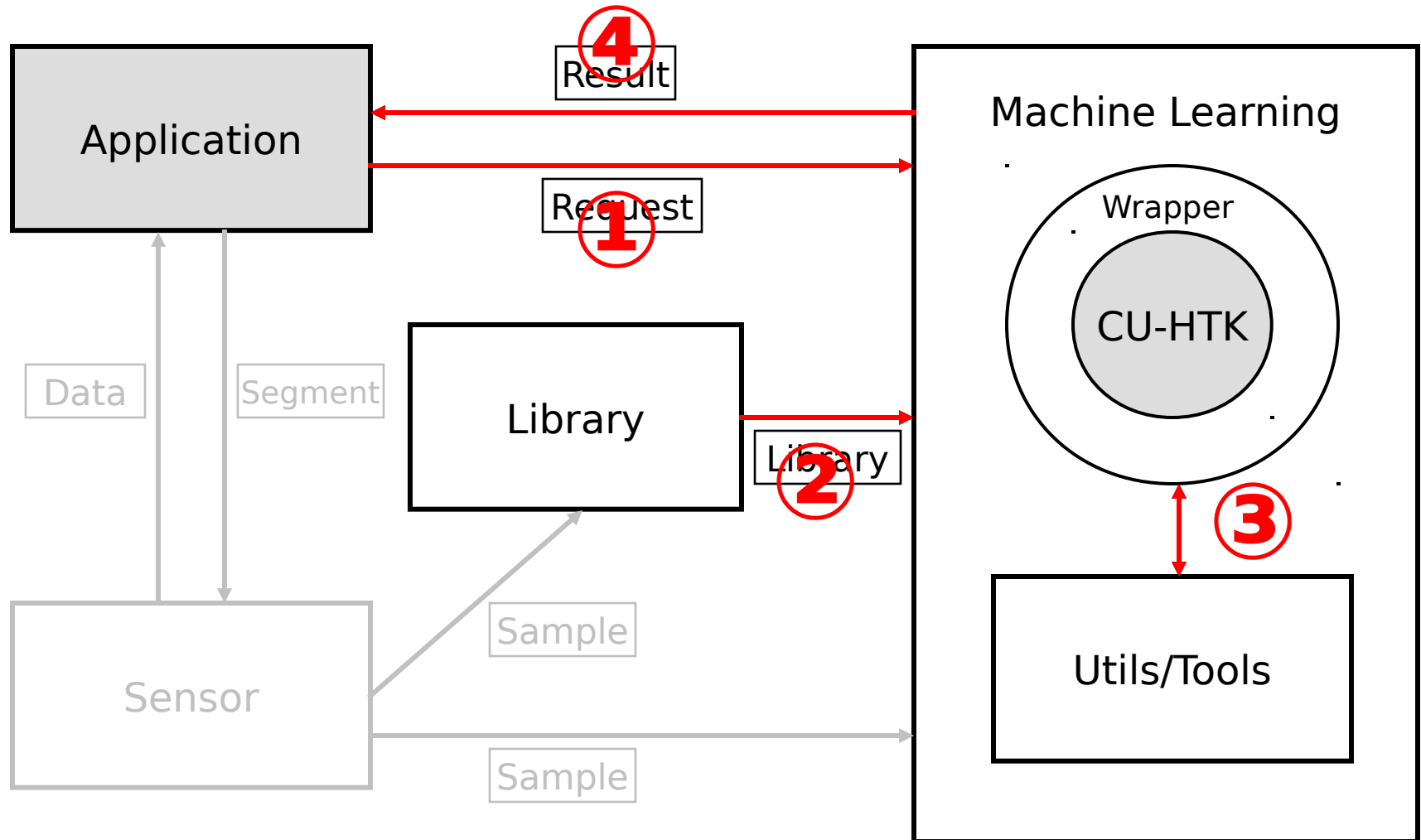
Gesture Recognition Process



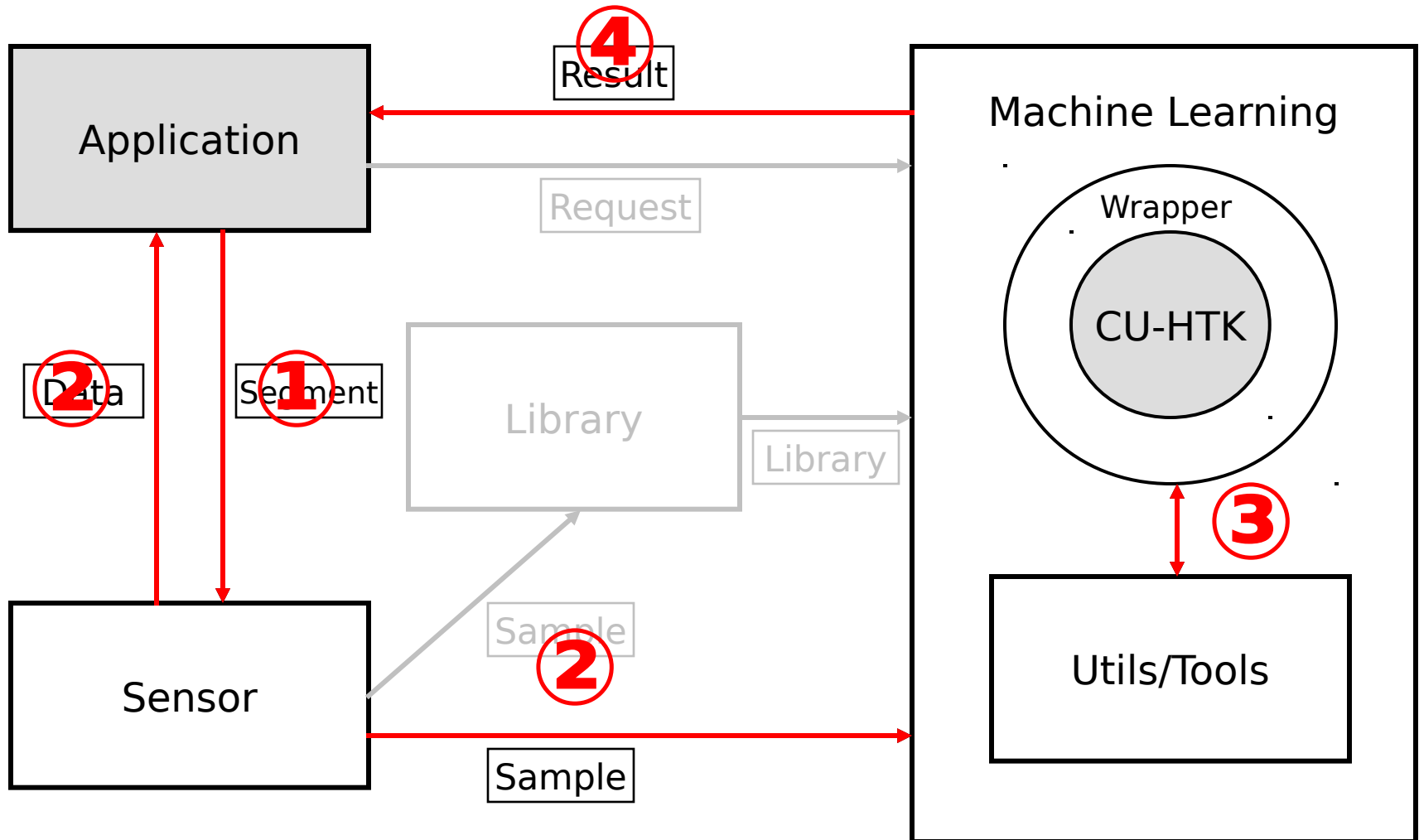
Data Collection



Training



Recognition



How to make your own application?

- Considerations:
 - What kind of sensor will you use?
 - Is it already in the gt²k package or should I build new sensor?
 - How are you going to segment data?
 - Mouse Sensor segments data by clicking (start) and releasing (stop) mouse button
 - How big is my feature vector?
 - Vector size of mouse sensor : 2 (X and Y)
 - Vector size of accelerometer : 3 (X, Y and Z)
 - What if I have two accelerometer sensors?

How to make your own application?

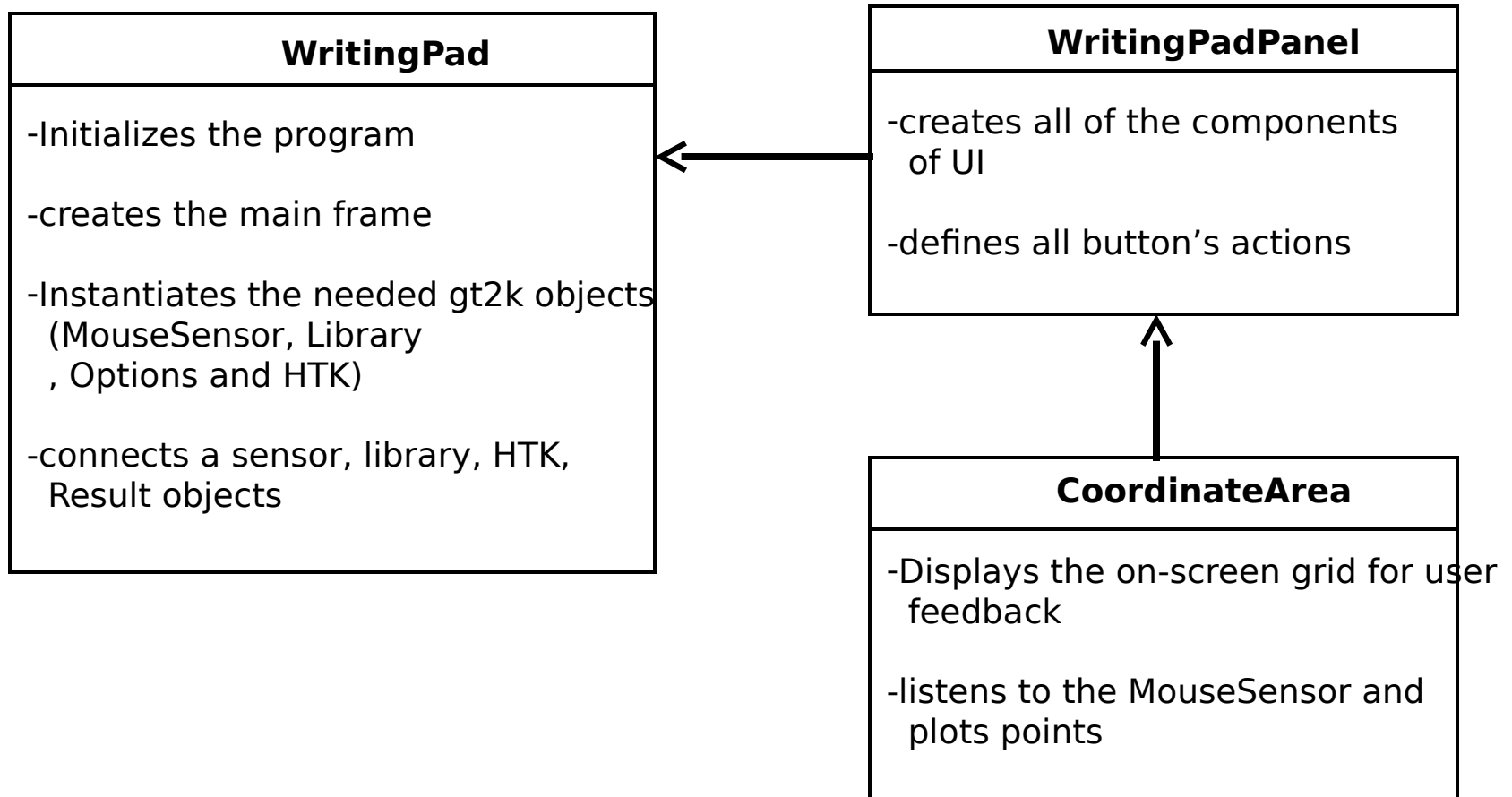
- Create a new set of options :
 - `GT2kOptions myOpts = GT2kOptions();`
`myOpts.setVectorSize(2);`
- Create or load a library object :
 - `try { library=Library.load("MyLibrary.xml");`
`} catch (Exception e) {`
`library=new Library("MyLibrary");`
`}`
- Initializes the machine learning component :
 - `HTK htk = new HTK(myOpts);`
- Create a new sensor:
 - `MouseDragVectorSensor mySensor = new`
`MouseDragVectorSensor();`

How to make your own application?

- For data collection
 - Connect the sensor to the library so it can save the samples
 - `mySensor.addSensorSampleListener(library);`
- For recognition
 - Connect the sensor to the HTK object
 - `mySensor.addSensorSampleListener(htk);`
 - Connect HTK object to application to get a result
 - `htk.addResultListener(myApplication);`

How to make your own application?

- WritingPad application structure



How to make your own sensor?

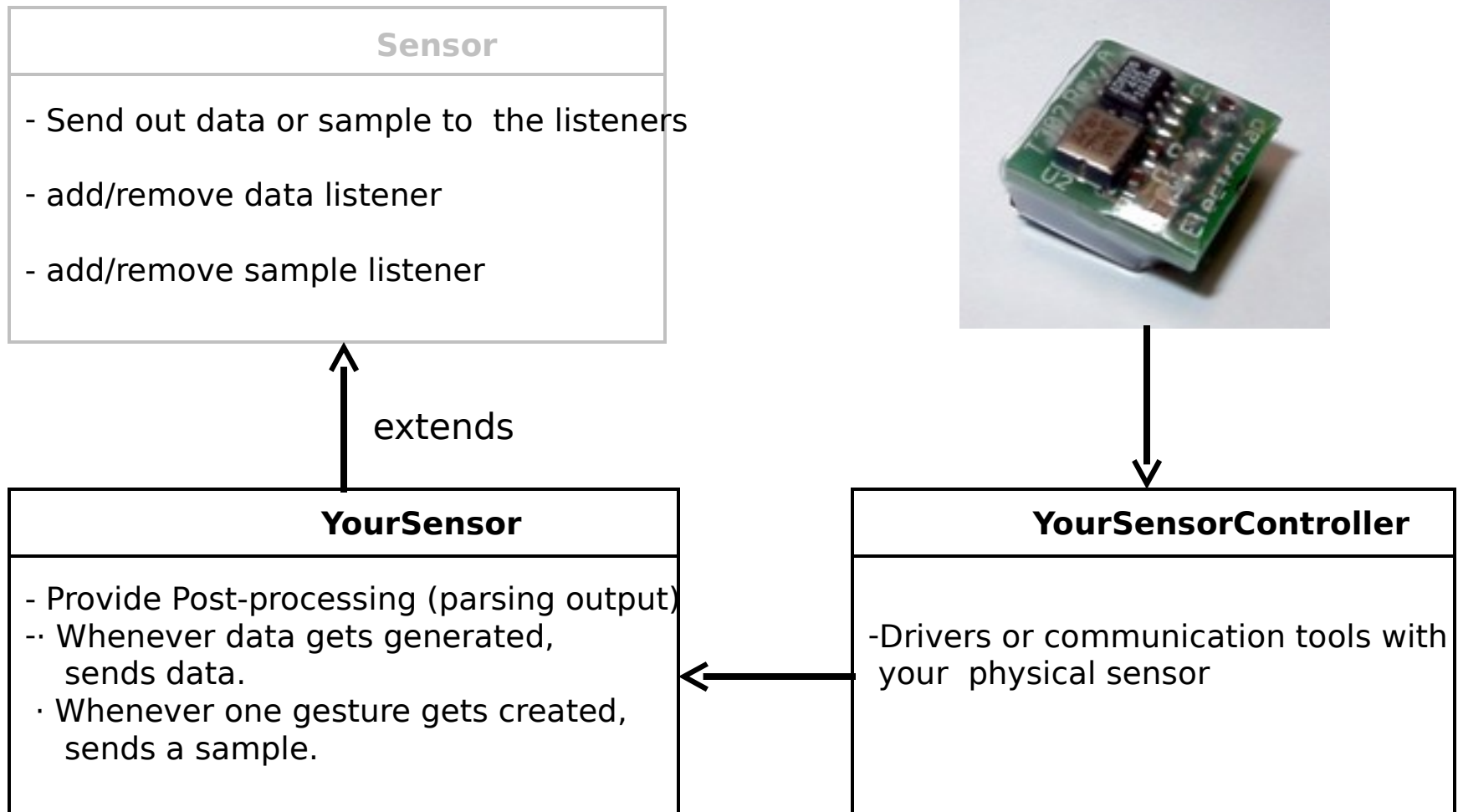
- Considerations:
 - Make sensor Runnable?
 - Mouse sensor implements mouse listeners
 - Accelerometer sensor implements Runnable
 - Start or stop from application
 - Drivers and Java library for your sensor
 - May not work under Windows but Linux

How to make your own sensor?

- Things to do/write
 - Communication with physical sensor
 - Parses output (float type)
 - Connects to the gt²k sensor infrastructure

How to make your own sensor?

- Structure



-
- Questions?

- For more information
visit:

<http://ccg.cc.gt.atl.ga.us/bliki/prj:gt2k:gt2kmanual>

email:

Kent Lyons : kent@cc.gatech.edu

Helene Brashear : brashear@cc.gatech.edu

Tracy Westeyn : turtle@cc.gatech.edu

Jung Soo Kim : jszzang@cc.gatech.edu