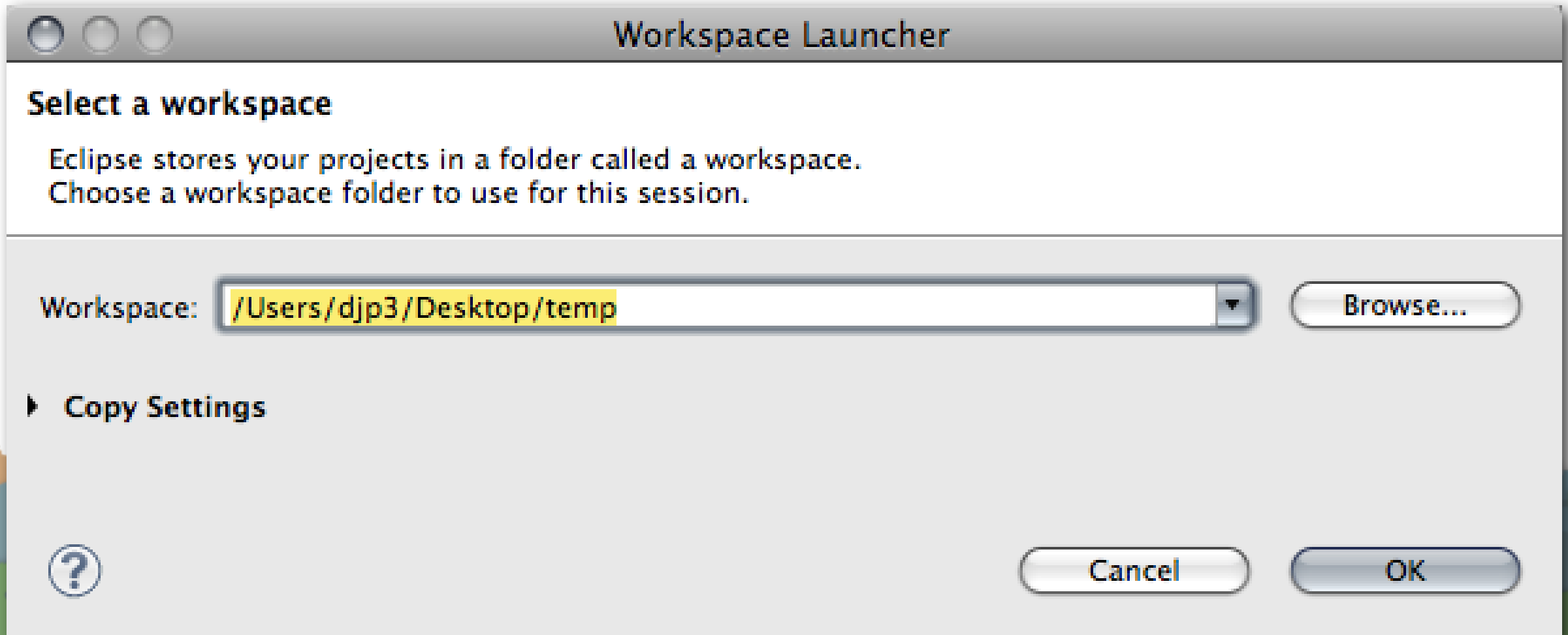# User Interaction: Software for MT

Assoc. Professor Donald J. Patterson
INF 133 Fall 2011
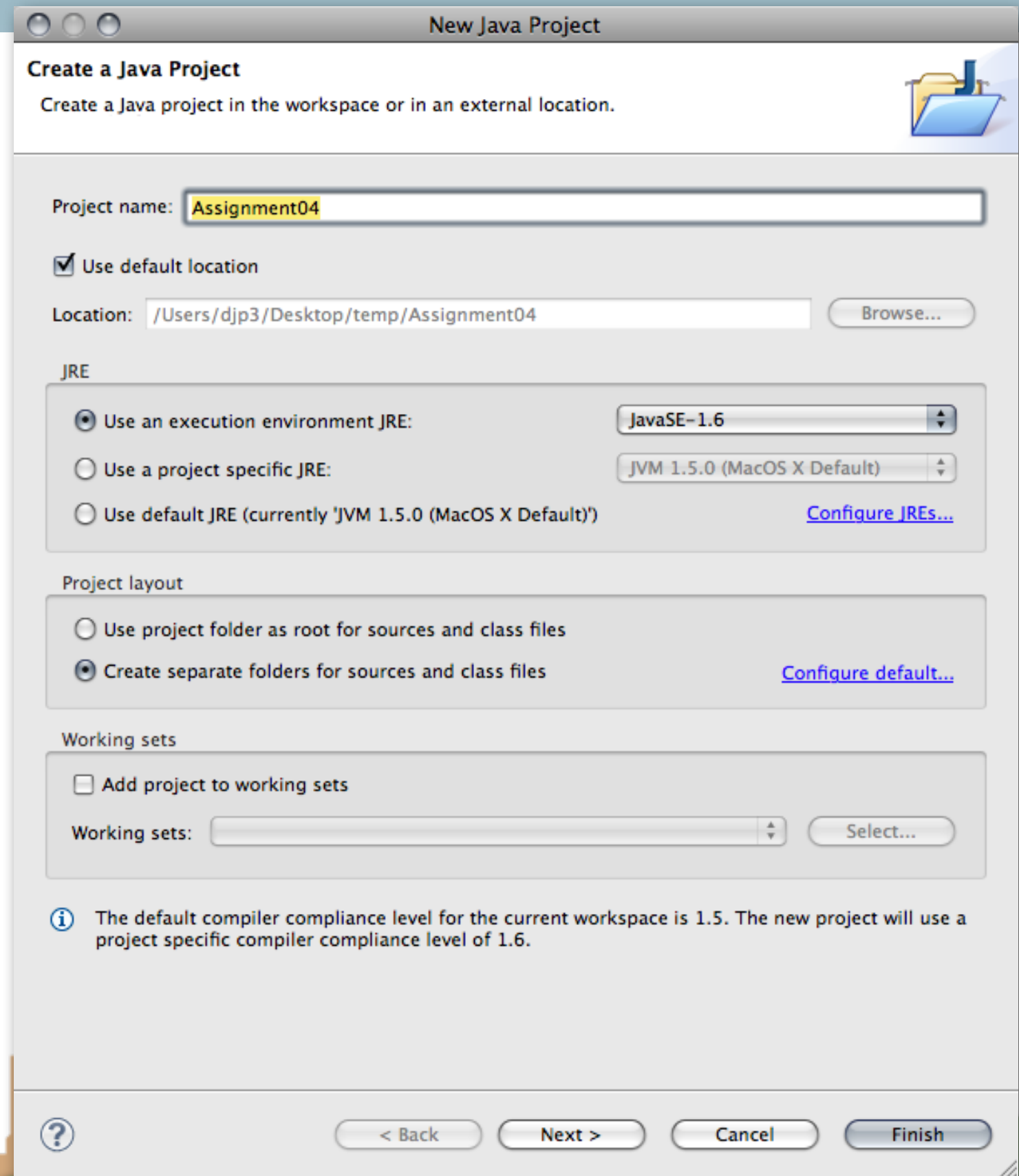
# Getting the software going

- Download and Install Eclipse

  - http://www.eclipse.org/

- Download MT4J

  - http://www.mt4j.org/mediawiki/index.php/Downloads

- Pick a directory and create a new workspace in Eclipse

**Workspace Launcher**

**Select a workspace**

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/djp3/Desktop/temp ▾    Browse...

▸ Copy Settings

Cancel    OK

# Getting the software going

- Create a new Java Project in Eclipse



New Java Project

**Create a Java Project**
Create a Java project in the workspace or in an external location.

Project name: Assignment04

☑ Use default location

Location: /Users/djp3/Desktop/temp/Assignment04     Browse...

**JRE**

◉ Use an execution environment JRE:     JavaSE–1.6

○ Use a project specific JRE:     JVM 1.5.0 (MacOS X Default)

○ Use default JRE (currently 'JVM 1.5.0 (MacOS X Default)')     Configure JREs...

**Project layout**

○ Use project folder as root for sources and class files

◉ Create separate folders for sources and class files     Configure default...

**Working sets**

☐ Add project to working sets

Working sets:     Select...

ⓘ The default compiler compliance level for the current workspace is 1.5. The new project will use a project specific compiler compliance level of 1.6.
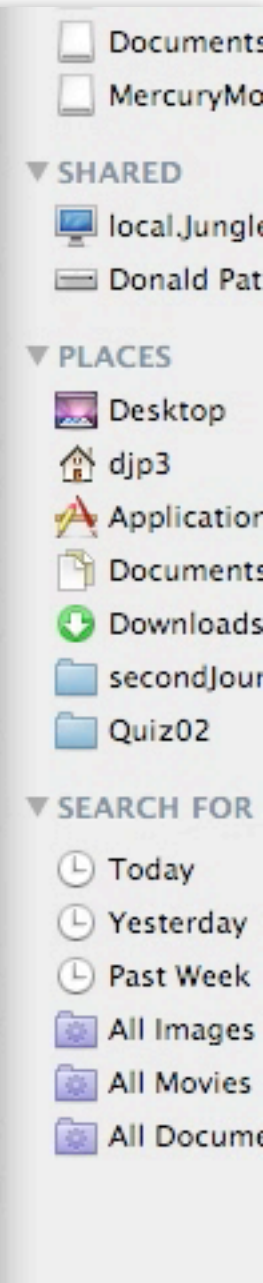
< Back     Next >     Cancel     Finish

# Getting the software going

- Create a new Java Project in Eclipse for MT4J
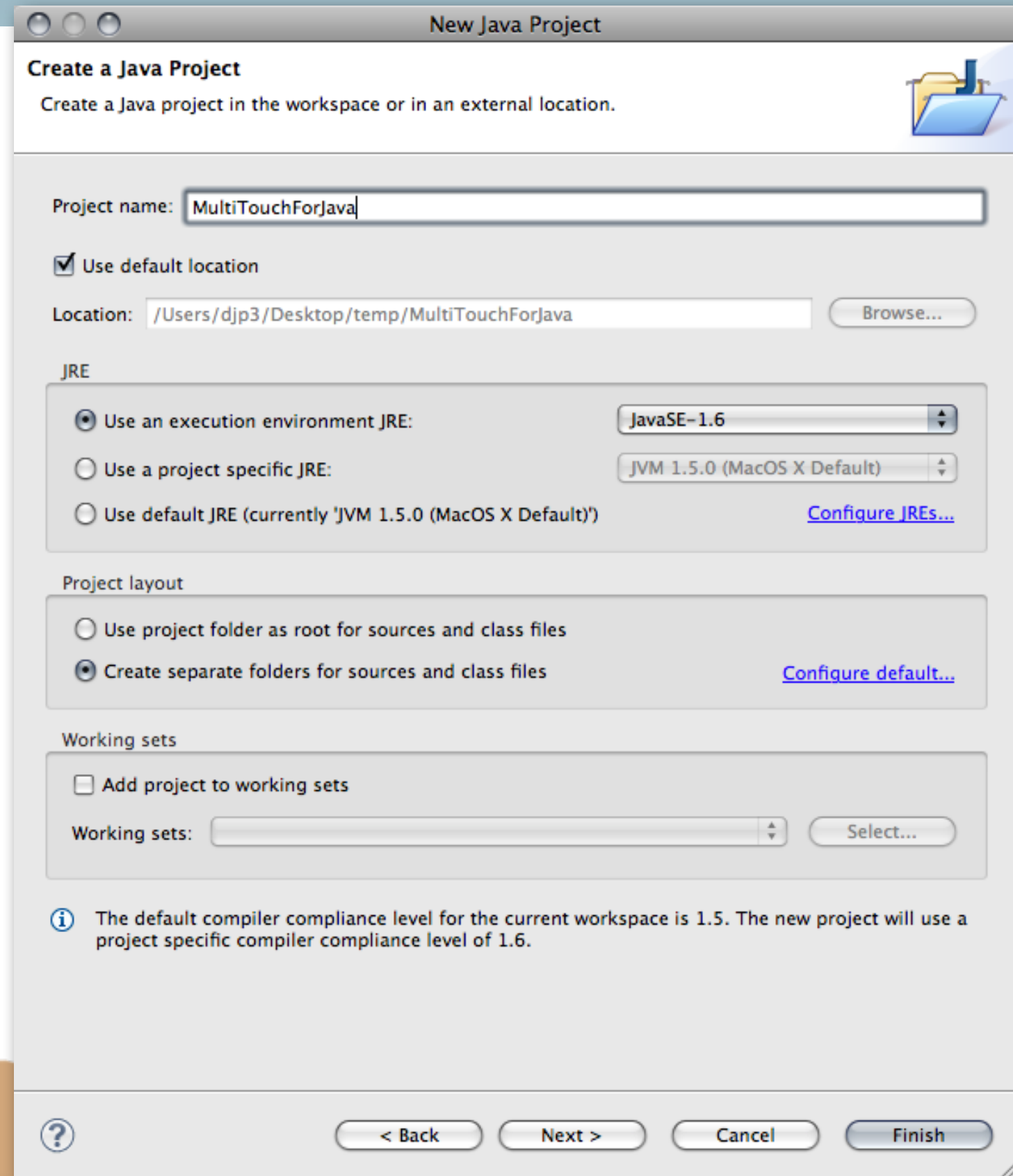
# Getting the software going

- Move MT4J to the new project and refresh in Eclipse
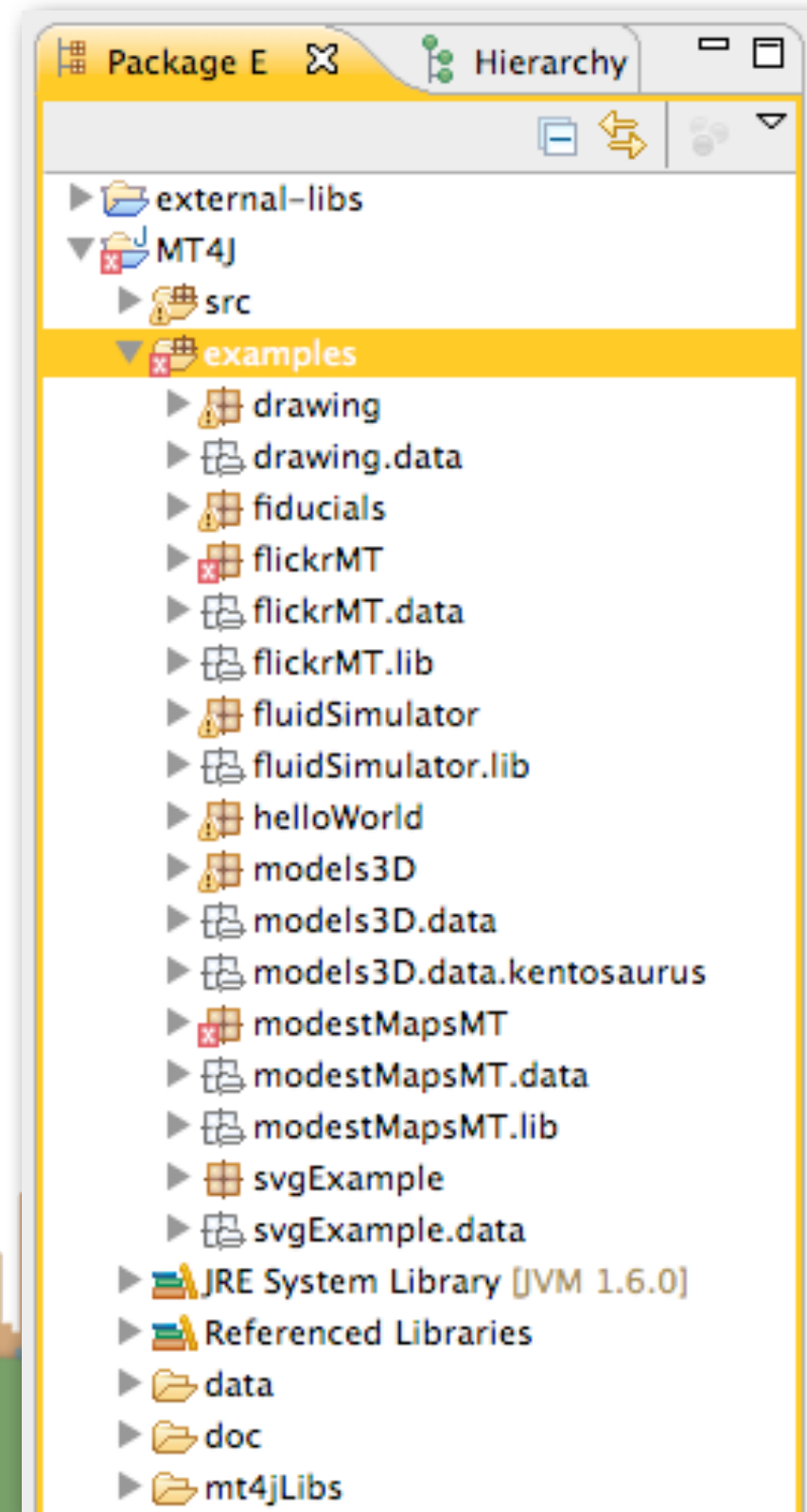
## To get the examples running

- Add all the .jar files in the mt4jLibs directory into the build path for the MT4J project.
  - Right click -> Build path -> "Add to Build Path"
- Make sure "examples" is in your build path as a source directory.  (Look at the project properties)

# Getting the software going

- When you run an MT4J program the TUIO server is built in

  - Hello World

  - Test Scene

  - MTGestures

  - Space Scene

  - Fluid Simulator



Package E ✕ | Hierarchy

▸ 📁 external-libs
▾ 📦 MT4J
  ▸ 📦 src
  ▾ 📦 examples
    ▸ 🗂 drawing
    ▸ 🗂 drawing.data
    ▸ 🗂 fiducials
    ▸ 🗂 flickrMT
    ▸ 🗂 flickrMT.data
    ▸ 🗂 flickrMT.lib
    ▸ 🗂 fluidSimulator
    ▸ 🗂 fluidSimulator.lib
    ▸ 🗂 helloWorld
    ▸ 🗂 models3D
    ▸ 🗂 models3D.data
    ▸ 🗂 models3D.data.kentosaurus
    ▸ 🗂 modestMapsMT
    ▸ 🗂 modestMapsMT.data
    ▸ 🗂 modestMapsMT.lib
    ▸ 🗂 svgExample
    ▸ 🗂 svgExample.data
  ▸ 📚 JRE System Library [JVM 1.6.0]
  ▸ 📚 Referenced Libraries
  ▸ 📁 data
  ▸ 📁 doc
  ▸ 📁 mt4jLibs

# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations

```
┌─────────────────┐
│                 │
│   Application   │
│                 │
└────────┬────────┘
         │
┌────────┴────────┐
│                 │
│      I/O        │
│                 │
└─────────────────┘
```
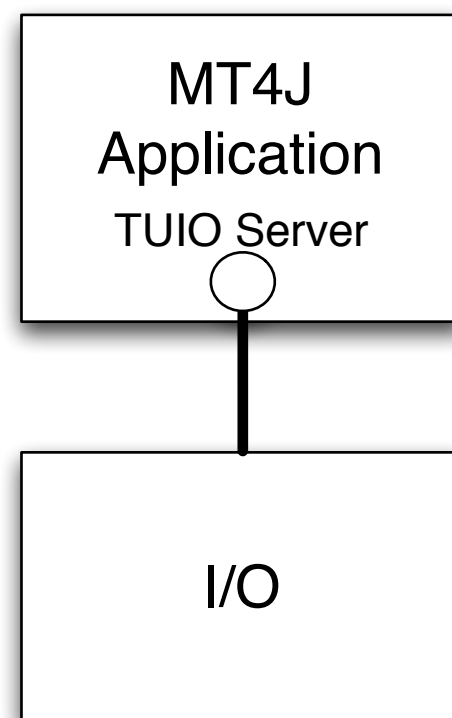
# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations

```
┌─────────────────┐
│     MT4J        │
│   Application   │
│                 │
│   TUIO Server   │
│       ◯         │
└───────┬─────────┘
        │
┌───────┴─────────┐
│                 │
│      I/O        │
│                 │
└─────────────────┘
```
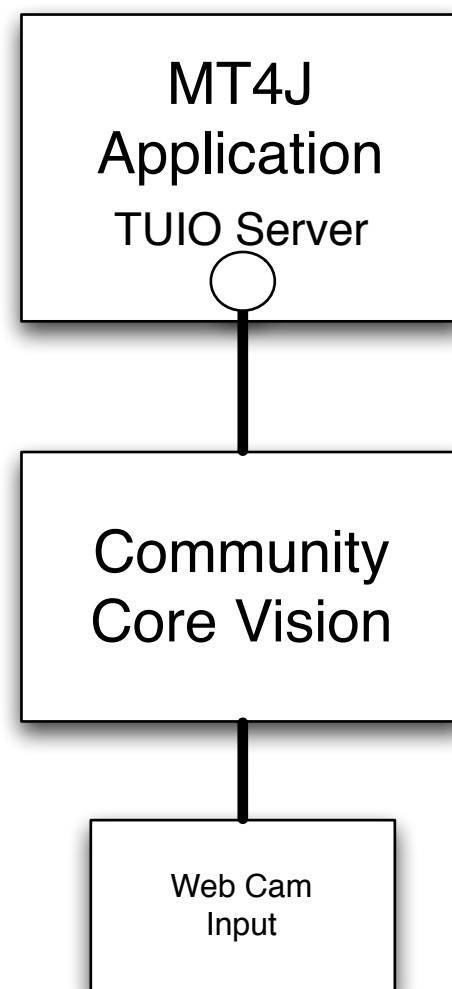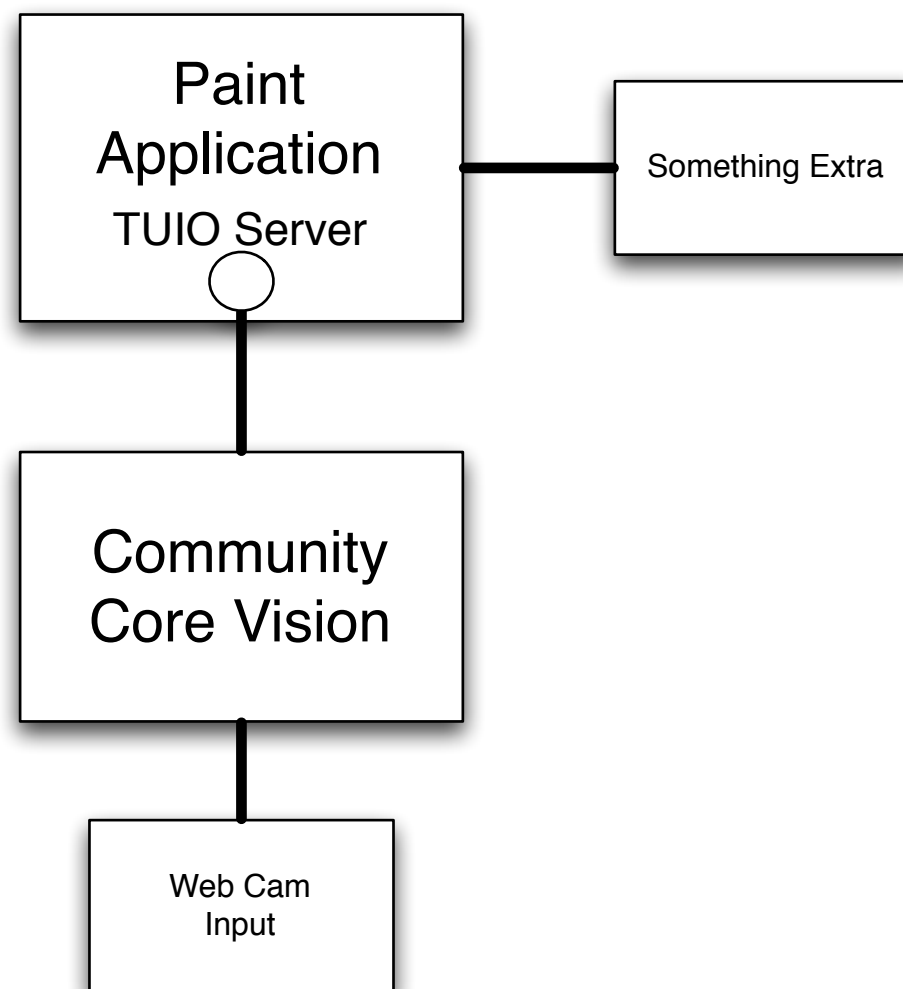
# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations

```
┌─────────────────┐
│      MT4J       │
│   Application   │
│   TUIO Server   │
│       ◯         │
└────────┬────────┘
         │
┌────────┴────────┐
│                 │
│    Community    │
│   Core Vision   │
│                 │
└────────┬────────┘
         │
    ┌────┴────┐
    │ Web Cam │
    │  Input  │
    └─────────┘
```
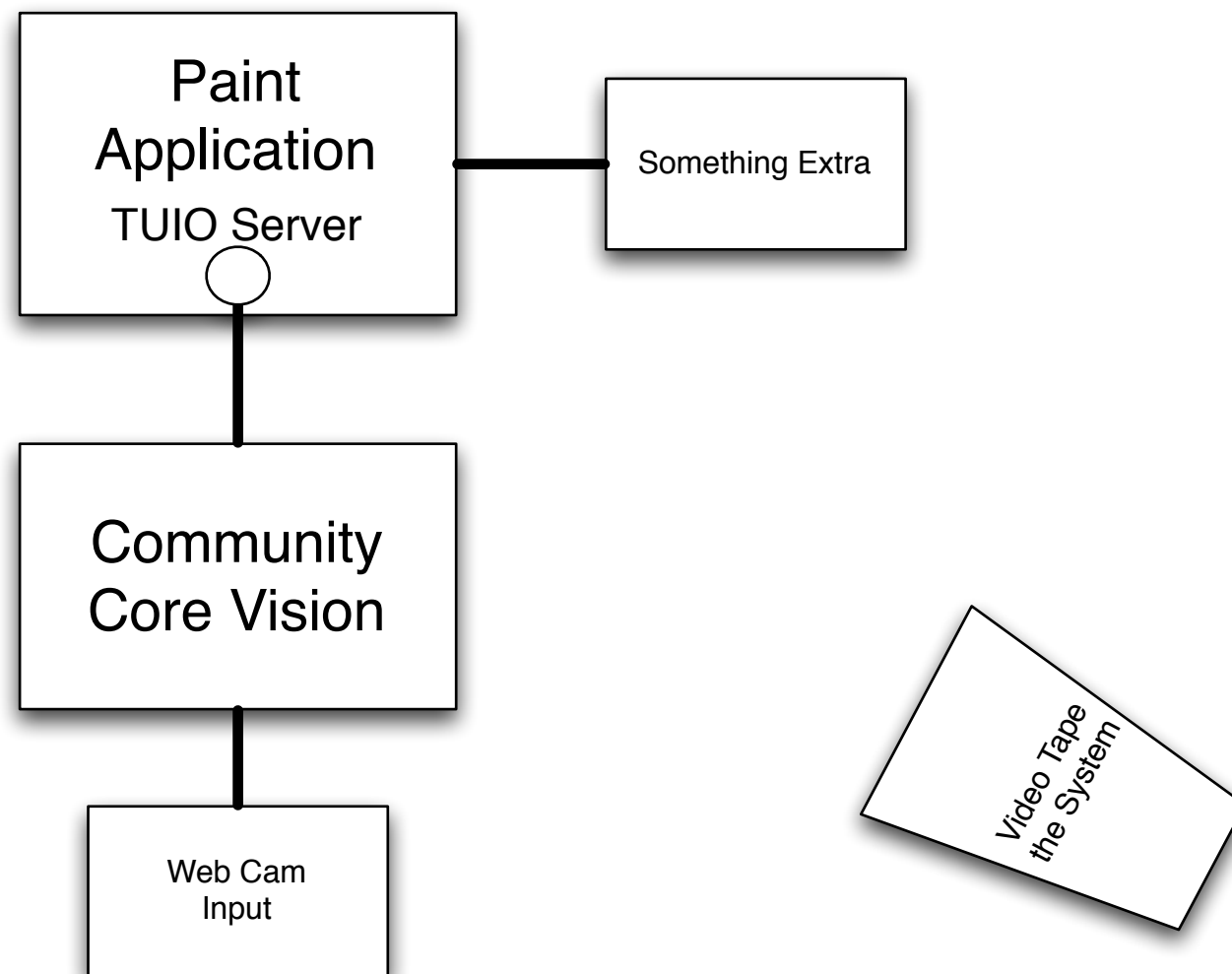
# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations

```
┌─────────────────┐          ┌─────────────────┐
│      Paint       │          │                 │
│   Application    │──────────│ Something Extra  │
│                  │          │                 │
│   TUIO Server    │          └─────────────────┘
│        ○         │
└────────┬─────────┘
         │
┌────────┴─────────┐
│                  │
│    Community     │
│   Core Vision    │
│                  │
└────────┬─────────┘
         │
    ┌────┴─────┐
    │ Web Cam  │
    │  Input   │
    └──────────┘
```
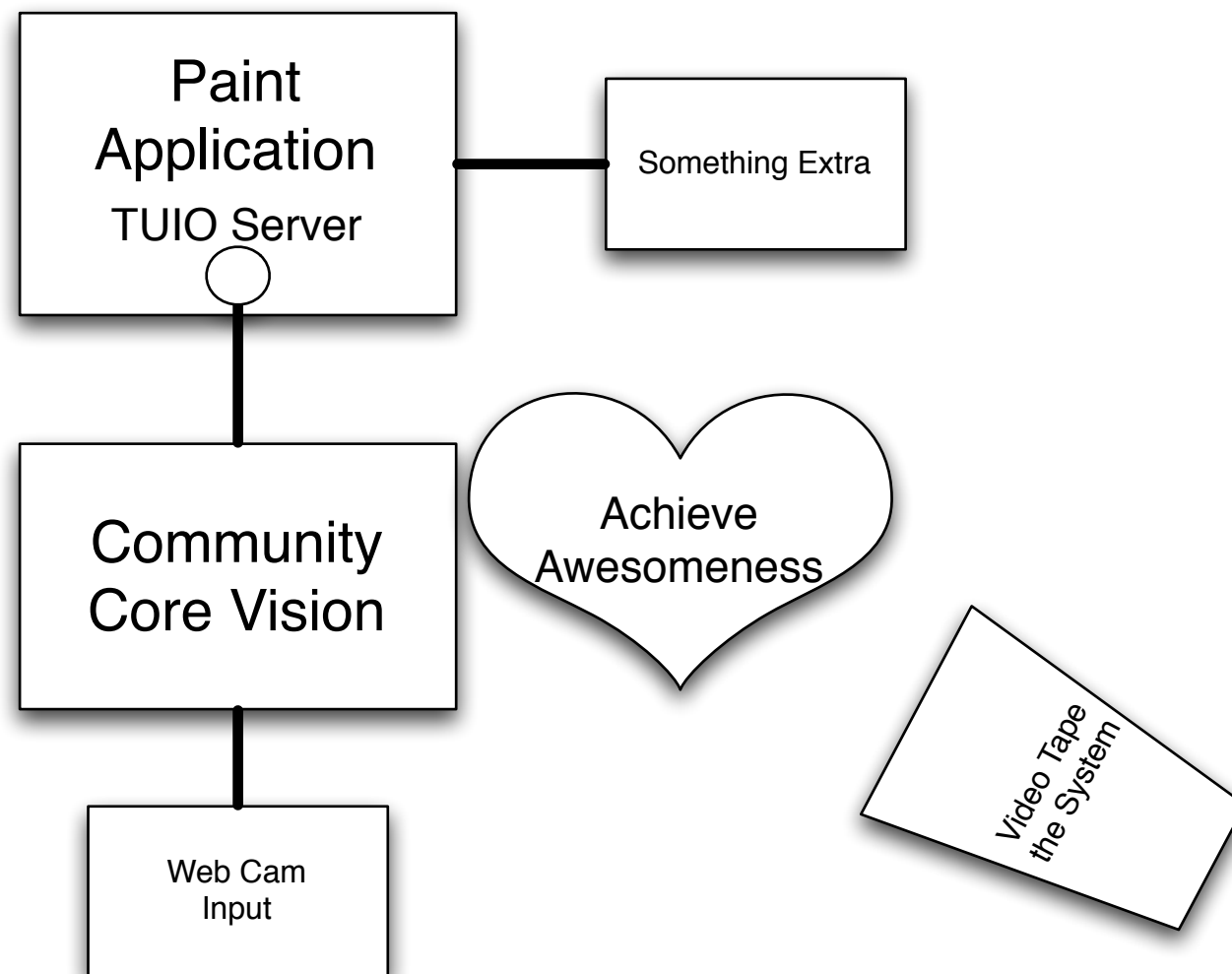
# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations
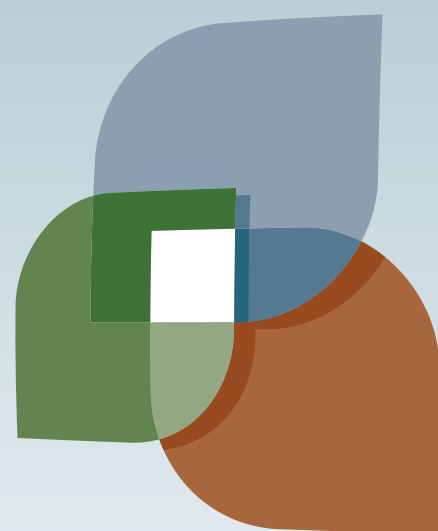
# How to go about doing Assignment #4

- Get an end-to-end demo running with off the shelf stuff

- Then slowly replace the pieces with your implementations

# How to go about doing Assignment #4

- Options for increased awesomeness

  - Do a different application than a paint program

    - Make sure it needs multi-touch (talk to prof.)

  - Do a different input than a webcam

    - Make sure it is more than just a download (talk to prof.)

  - Remember this can be a portfolio piece!

LUCI