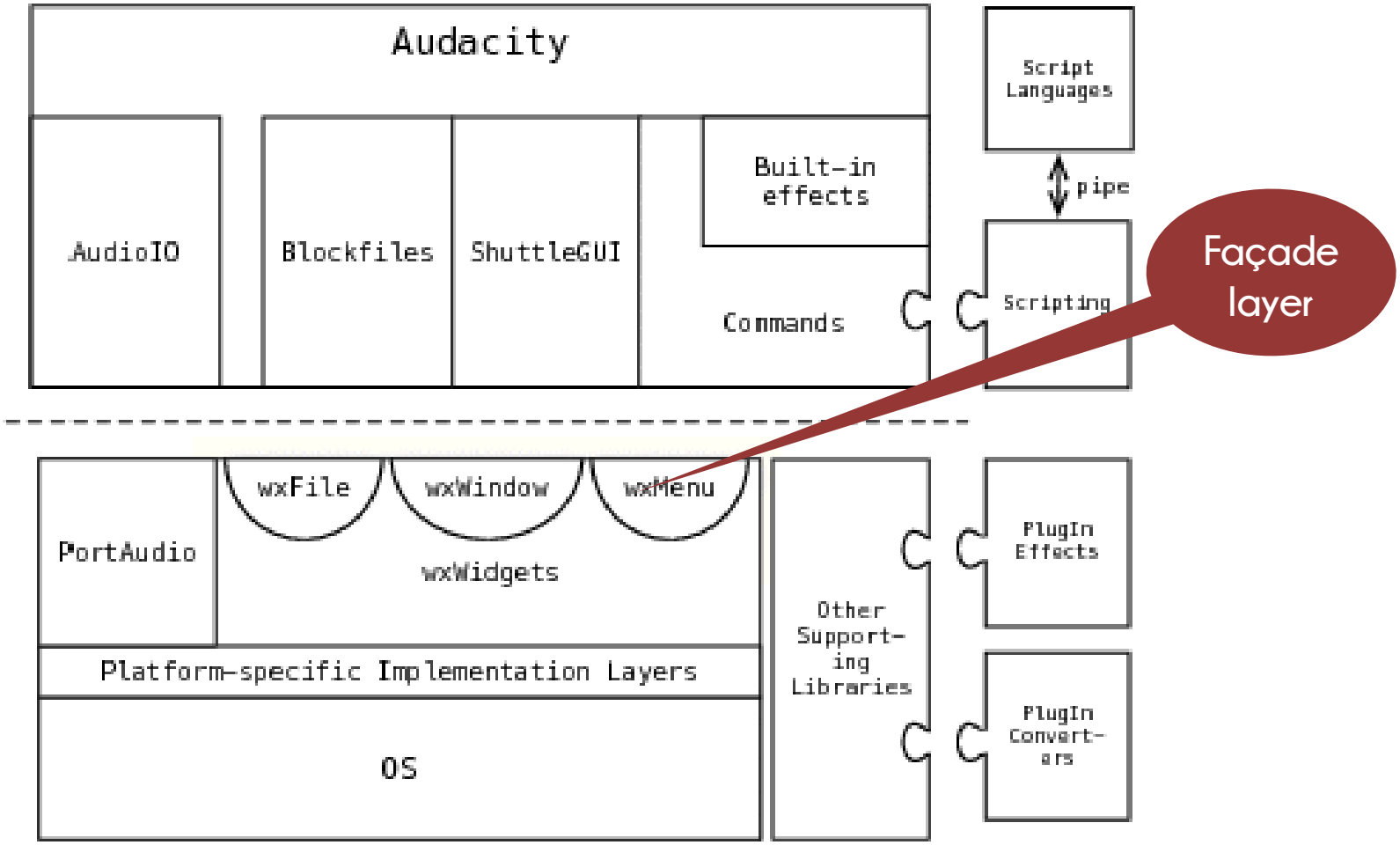
Architecture of Audacity

Functional and Non-Function Requirements

- Audacity should :
  - Be a cross-platform software
  - Overcome license issue of audio coder/decoder
  - Be free of major security loopholes

- Be a cross-platform software
  - Using a cross-platform library: abstractions for file system, UI, localization
- Overcome license issue of audio coder/decoder
  - Users can install coder/decoder as plugins for Audacity
- Be free of major security loopholes
  - Remove all TCP-related functions (less is more!)

Architecture of Audacity

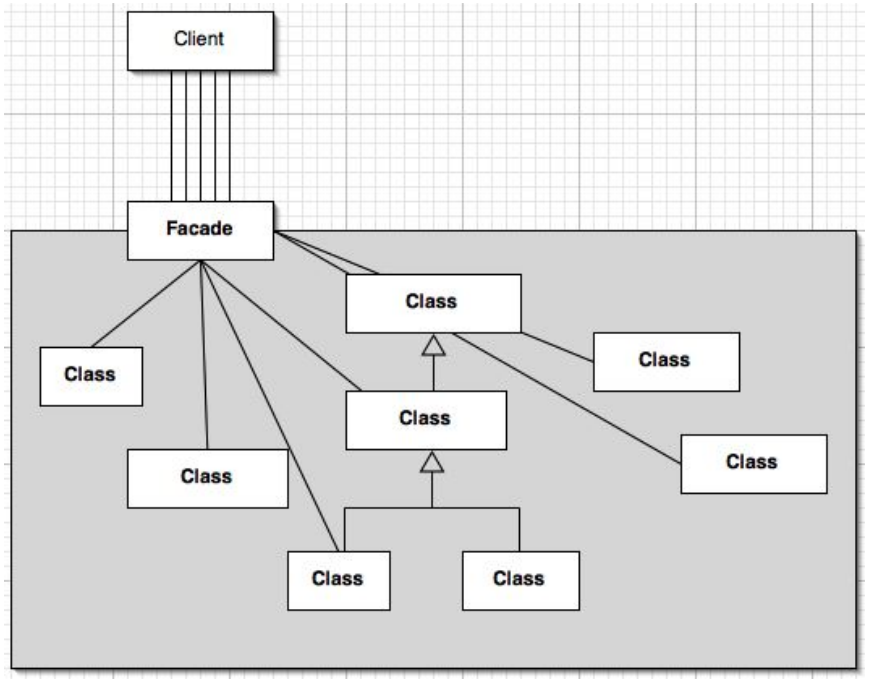


Example of SE architecture writing

"Third party APIs such as PortAudio and wxWidgets have been of huge benefit. They've given us code that works to build on, and abstracted away many platform differences. One price we pay for using them is that we don't get the flexibility to choose the abstractions. We have less than pretty code for playback and recording because we have to handle threading in three different ways."

Example of SE architecture writing

"The API given to us by wxWidgets tempted us into writing some verbose, hard to follow application code. Our solution to that was to add a façade in front of wxWidget to give us the abstractions we wanted and cleaner application code."



- Which architecture style/design best fit your course project?

- Please polish your report for Deliverable #2!

- Please spend time to rehearse your project presentation!