IoT: Client Devices

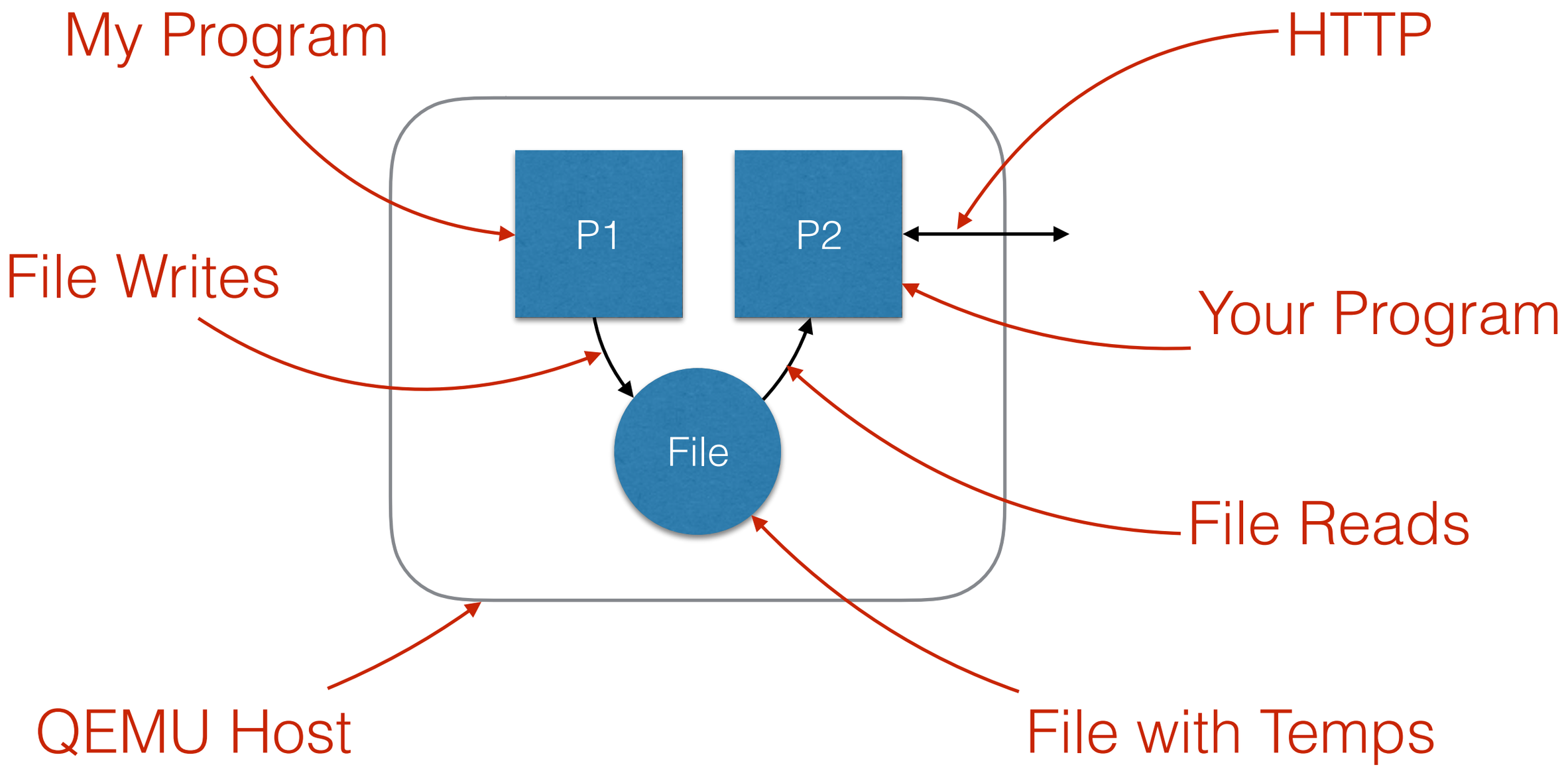
Project (II)

Goal

Emulating a thermostat

- I'll provide a small program to you that writes temperatures to a file (in degrees Celsius)
- You'll pretend that file is an actual thermocouple
- Your thermostat is programmable (no less than 3 different points over a day)
  - Weekends? calendars? weekly programs? extra credit!
- Program remotely via HTTP interface
- Report temperatures and status via HTTP interface
- You'll turn a heater on/off based on the program and reported temperature (you'll write this to a known file with a timestamp)

How will this work?



Requirements

YOUR SYSTEM SHALL

- Read the current temperature from a known file
  - /var/log/temperature
  - Read a single temperature value written to file
  - Float in degrees C
- Turn heat on/off based on program and current temperature
  - /var/log/heater
  - Turn heat on/off by writing to /var/log/heater
  - A single line <action> : <timestamp>
  - action:= <on|off> timstamp:=<posix time of action>

Requirements

Your system shall

- Start a daemon service that can also run from command line
- Process a configuration file
  - Default option, also supplied to program via -c & —config_file flags (e.g. -c <config_file> or —config_file <config_file>)
- Provide a help option (-h or —help)
  - This will print typical help for the application

Requirements

YOUR SYSTEM SHALL

- The configuration file shall configure
  - Service endpoint (e.g. http://<some_host>:8000)
  - Log files (e.g. /my/logfile/here, for program output)
  - Any other config files
- Accept programs via an HTTP interface
  - program up to three different temperatures for a day set at arbitrary times

Requirements

Your system shall

- Report status to an outside process via HTTP
- Report actions to an outside process via HTTP

Your system may:

- Support more extensive programming