OWTF: One Wire Temperature Finder

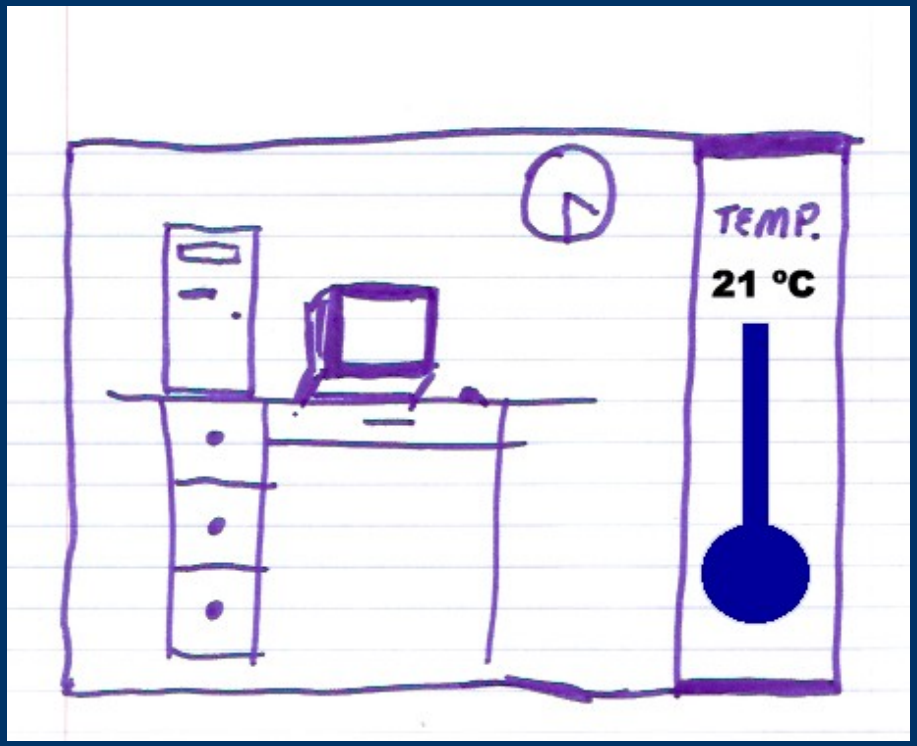
Why is temperature important?

- Temperature extremes can cause a variety of problems, including premature aging and failure of chips or mechanical failure of devices.
- Heat generated by electronic circuitry must be dissipated to prevent immediate failure and improve long term reliability. If it gets too hot, heat can't dissipate.

Why is humidity important?

- High humidity conditions can cause moisture migration and penetration into the system.
- Moisture can cause corrosion of internal components and degradation of properties such as electrical resistance, thermal conductivity, physical strength, and size. This is bad!
- Extremely low-humidity conditions promote the generation of significant static electrical charges. These charges can destroy sensitive electronics.

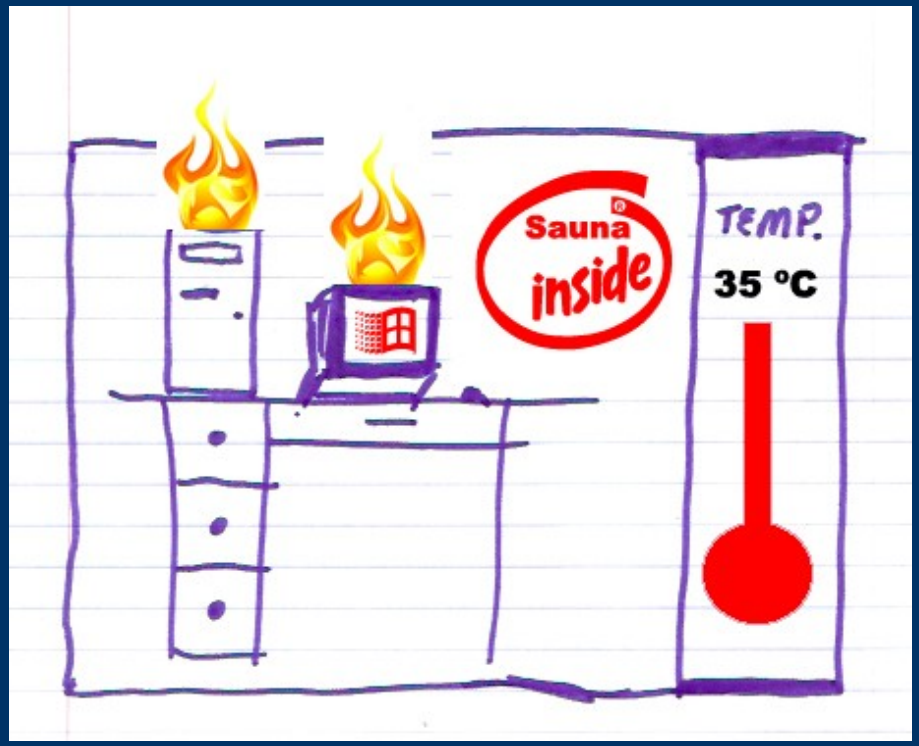
Ideal Conditions



For Computers:

- Temperature Range:
  - 5 ˚C (41 ˚F) minimum
  - 35 ˚C (95 ˚F) maximum
- Humidity Range:
  - 10% - 85% Humidity
- Air Quality:
  - Low Dust / Particles
  - Air Should Circulate
  - Shouldn't be explosive!

Not So Ideal Conditions



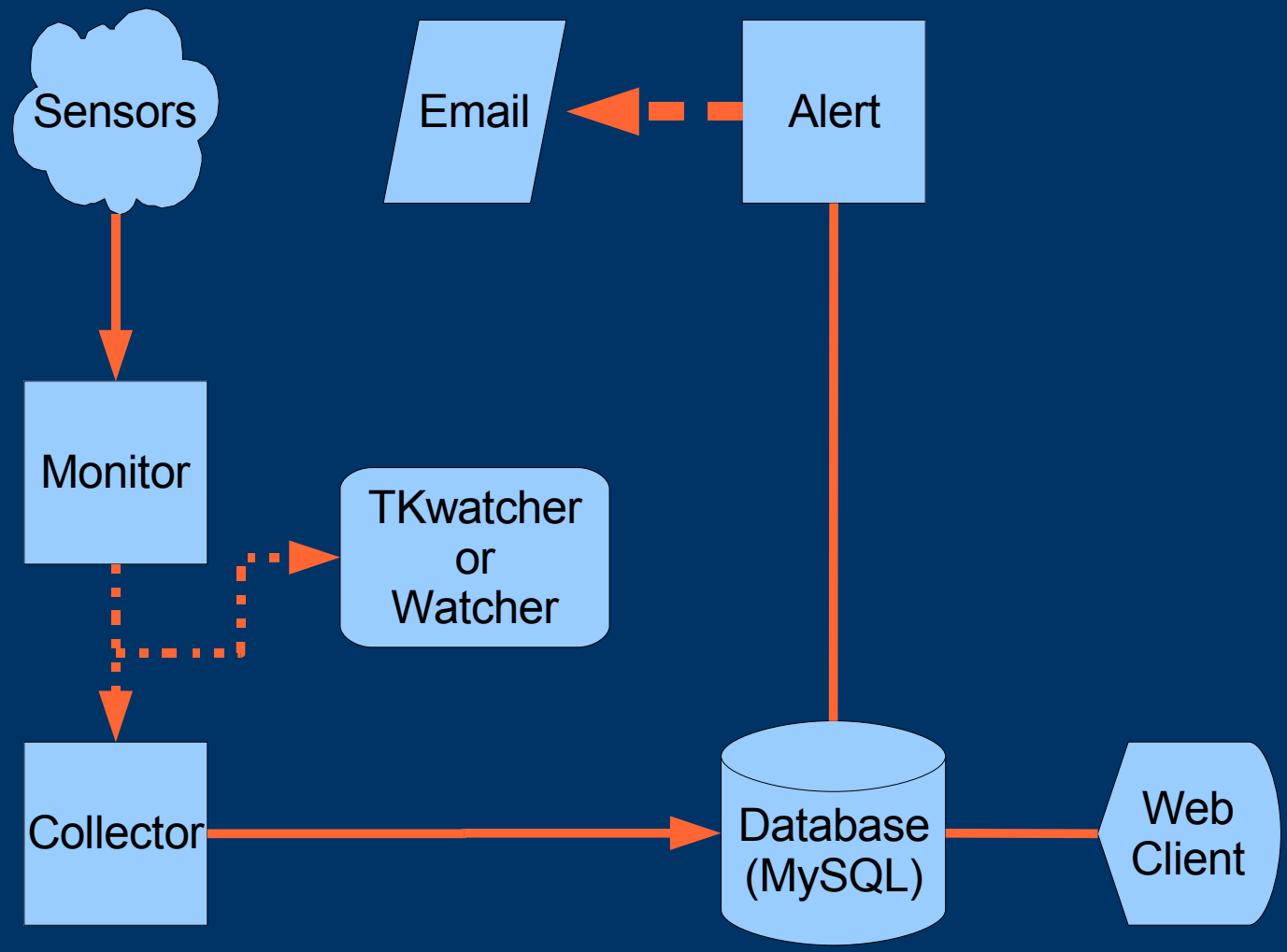
Can Cause:

- Chips to become loose in their sockets.
- Disk read or write data errors.
- Equipment to overheat and fail (i.e. “fry cook”, “heat death”).

Introducing... OWTF!

- OWTF is a distributed environmental monitoring and alert system.
- OWTF is written entirely in Python and is for Linux / Unix only.
- OWTF can be extended by the end user to monitor more than just simple temperature and humidity data. Users are only limited by the available One Wire sensors.

OWTF System Overview



Reading Objects

- Container for:
  - Monitor ID (mid)
  - Sensor ID (sid)
  - Sensor location (location)
  - Datetime [YYYY-MM-DD HH:MM:SS] (cron)
  - Temperature (temp)
  - Humidity (humid)
  - Voltage 1 / extra data slot (V1)
  - Voltage 2 / extra data slot (V2)
- Object used for transmission of sensor data from Monitor(s) to Collector over the network via UDP.

OWTF Main Components

- Monitor.py
  - Gets data from sensors and sends it to the collector.
- Collector.py
  - Collects monitor data and sends it to a central database.
- Alert.py
  - Deletes old (32+ days old) sensor data.
  - Sends an email alert if a reading is outside allowable parameters.
- Alertables.py
  - Allows the end user(s) to set custom alerting rules for each sensor.

OWTF Extended Components

- Sensorgen.py
  - Outputs all One Wire sensors which can be read by a monitor.
  - Program used to get listing of sensors. Allows user to give a sensor a “location”.
- Watcher.py
  - Command line utility which outputs all OWTF traffic sent over UDP Multicast port 50000.
- TkWatcher.py
  - Graphical version of Watcher.py.
  - Can also serve as a simple text editor.