All Eyes AKA ‘ae’ Design Document (in works)

This document explains ‘ae’ architecture, development environment, future enhancements etc. PUT THIS IN A WIKI!!!!

# Overall ae design

Put Todd’s picture.

## Components Descriptions

‘ae’….

‘ae’ monitors

‘aemgr’

‘proxy’

# How to setup development Environment

## ae (Ubuntu Linux environment)

- Need Linux development environment; Makefile is provided. Just type "make" in src/ae directory.

- One can also use Eclipse; will be easier to develop as we are sync'ing with google-code SVN.

- Make sure syslog service is running to see the logs in /var/log/messages.

## ae server

'ae' is a dynamically linked for security. Each Monitor is a plugin at compile time. Monitors are spawned as threads with two parameters i.e. to act in monitor mode/minitor-and-action mode and the lifespan is persistent/volatile across reboots. How a monitor keeps its information in persistent is upto the monitor. But, actions are predefined in the framework. Monitors can't take actions. They can only update the status level.

## Proxy

Proxy is…

## Aemgr

Aemgr is an Android App…

## Future ae Enhancements:

* Instead of spawning threads, run each monitor in its own address space i.e. use fork, instead of spawning threads. NOTE: If you do, you will have to develop IPC mechanism to talk to the monitors.
* Make PERSISTENT/VOLATILE per Monitor based, not ae wide.
* Make 'ae' statically linked for embedded environment.
* Add Mutex to protect the monitor struct.

# How to add Monitor-Plugins to ae

* Add the MONCOMM structure values in ae.c to monarray structure array.
* Add a .c file by our monitor name (EX: selfmon.c) with appropriate functions.
* Add your monitor function with the apprpriate prototype in ae.h.