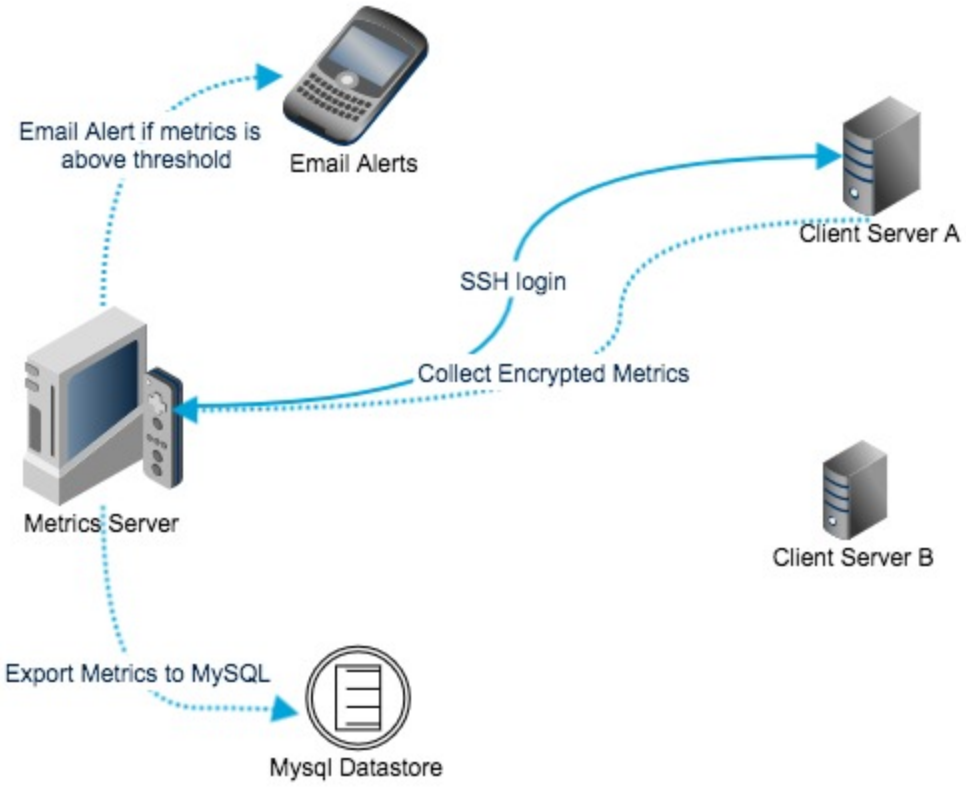
Design Document

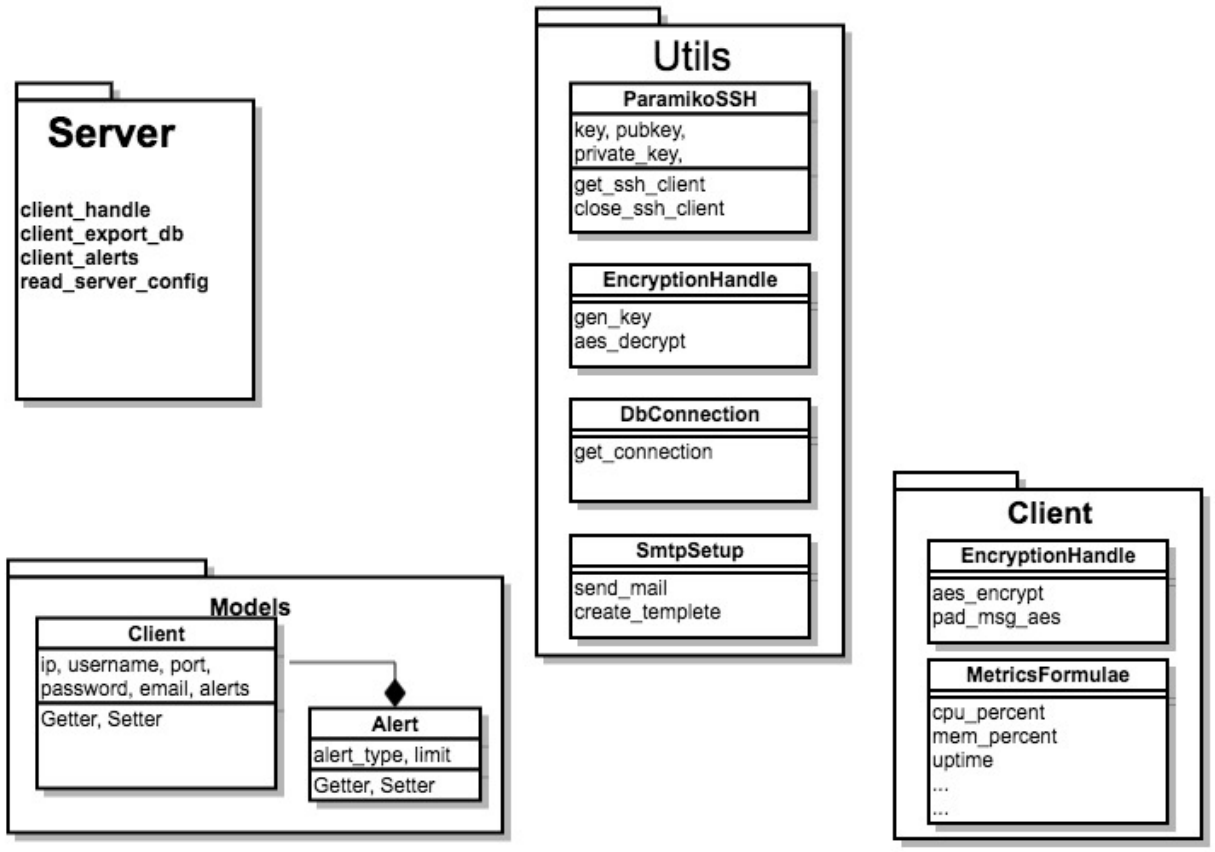
Packages

SSH login: paramiko
Cryptography: pycrypto
Cross platform metrics collection: psutil
Db connection: MySQL-python

High level requirement analysis and Architecture



Functional components



Workflow:

1. Read server_config.xml for latest hosts and alerts
2. Every host can be configured to have different types of alerts [eg. for host A (uptime, cpu) for host B (mem, cpu)]
3. Read xml clients and alerts in models class
4. For each client start a parallel thread for metrics pull
5. Initialise paramiko and get ssh,sftp connection for client
6. Create remote directory ‘/tmp/remotedir/’ and copy client script by sftp
7. Prepare remote python environment and install dependencies for client script (this is a one time overhead needed because most servers do not allow installing new python packages in default python env if user is not root, next execution will use already created virtualenv)
8. Execute client script with following parameters:

    a. Metrics required for this client as configured in server_config
    b. AES key to encrypt metrics from client to server (In this case the channel is already encrypted by secure shell but, AES encryption can be useful if metrics is sent to any listening socket at server)
9. Collect and decrypt returned metrics from the client
10. Write results to a MySQL db with IP, timestamp of the metrics when collected and metrics data (db_model.sql)
11. If alert_type has value more than threshold send mail to the configured user (have used AWS ses SMTP server)