Log Upload

Overview

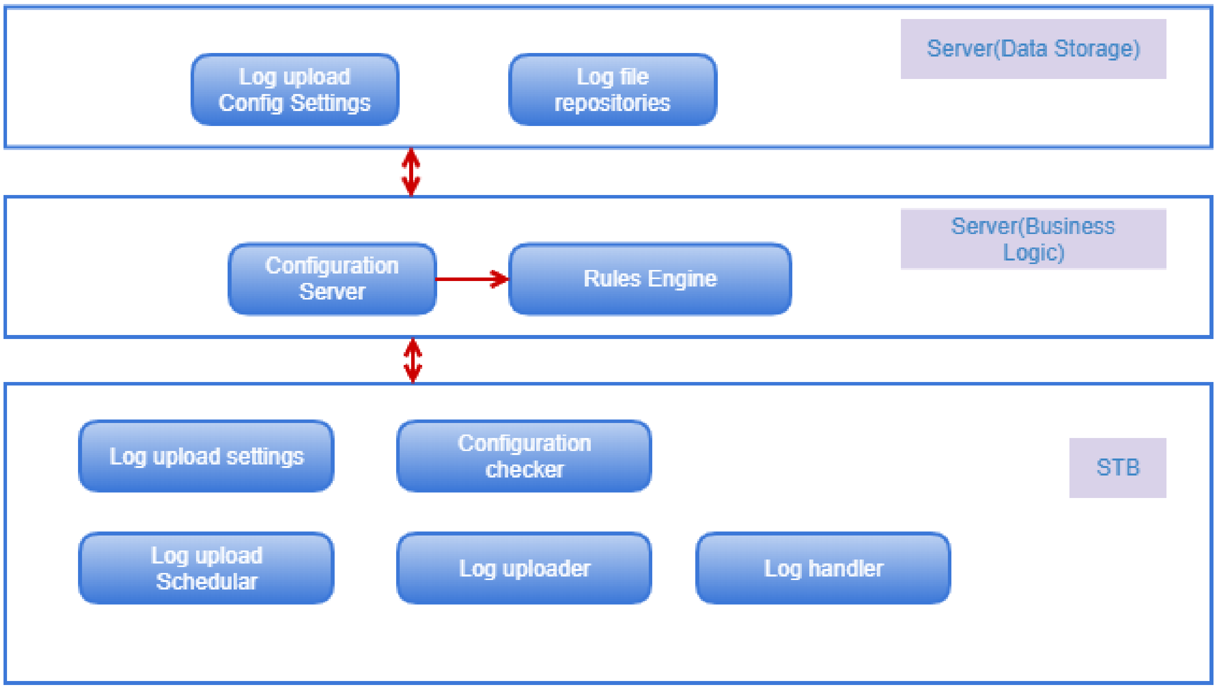
Logging and Log Upload are two critical aspects when considering large scale RDK deployment. STB stores logs for limited number of days (~10 days). Log repository get logs for a week or so which can help in debugging deployed STBs.

Importance of Logging in RDK Deployments

- Logs for deriving field matrices
- Logs for assessment of the criticality of the situation
    - Should I look at the problem at the middle of the night or can wait till next day morning
- Controlled log uploads from 100% to very small numbers (2-3%)based on device maturity
- Configurable upload intervals typically done once in a day
- Right level of abstraction to balance between the available storage space and the logs generated
- Rotate-backup-upload policies

More details on how logging is done in RDK can be found here: **RDK Logger**

RDK Log Upload Mechanism



Responsible for gathering local logs, packaging them, and uploading them to a remote URL. A client-side scheduler would control when it executes, but it could also be activated via immediate triggers from the log upload checker, SNMP MIBs, or other local actions.

The Uploader is responsible for sending STB log files to a remote file store. It performs the following:

- Responds to triggers (i.e. from cron or the SNMP agent)
- Checks local Device Settings for the URL of the file store, the logs it should send, and the repository it should use
- Gets connection credentials
- Makes a connection to the remote server and sends the files

Features

- Uses Xconf as the primary end point
- Ability to configure enable/disable log upload for a device/group of devices (During bootup, box retrieves configuration information from xconf. Based on rules defined in xconf server, we can control whether this particular device needs to do log upload or not.)
- Allows sampling of devices from a bigger population

- Configuration server is responsible for retrieving and delivering settings to any client
- Configurable upload methods (HTTP/SCP)

How STBs Upload Logs and Metrics

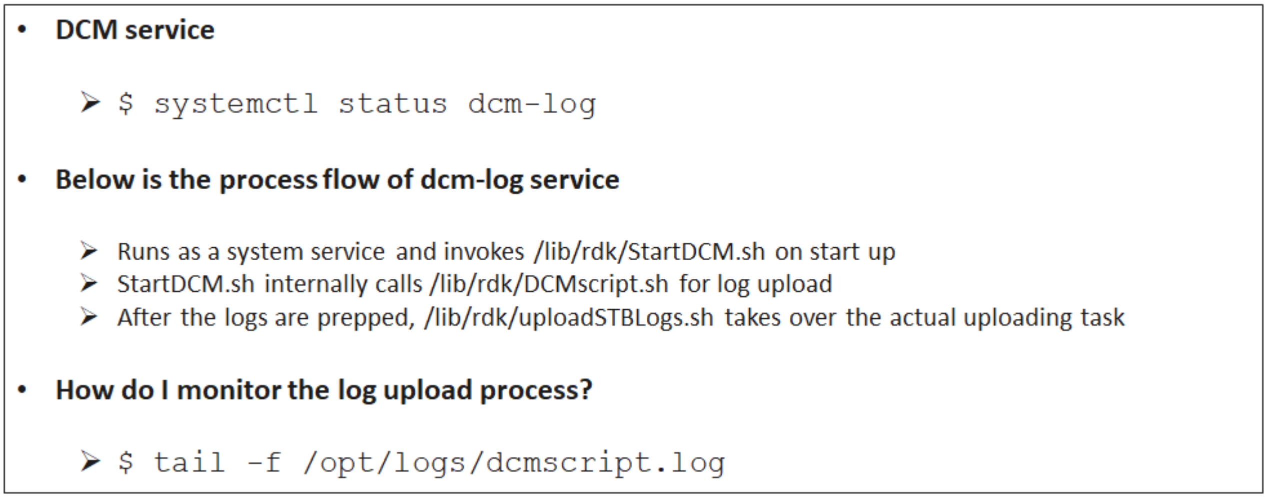
STBs check in with Xconf with the query : "Give me my logging and telemetry configuration." Part of their query string includes several (about four) parameters of this particular box. Things like MAC address, model, current firmware version, and environment.

Xconf server looks at these parameters and runs them through a rules engine that finds the closest **targeting rule** match to the box's profile. Once it finds a match, it retrieves the configuration for that rule.

The configuration includes things like upload schedule, repository URL, upload protocol, path to the file server, a protocol indicator, upload time smear and more.

Once the box gets the response, it parses it. It sets the specified random schedule that delays uploading the files. At that time, the box packages up all of its log files (up to twelve of them), compresses the bundle, and uploads the archive to the destination.

Service and scripts responsible for LogUpload in STB



- **DCM service**

  ➢ `$ systemctl status dcm-log`

- **Below is the process flow of dcm-log service**

  ➢ Runs as a system service and invokes /lib/rdk/StartDCM.sh on start up
  ➢ StartDCM.sh internally calls /lib/rdk/DCMscript.sh for log upload
  ➢ After the logs are prepped, /lib/rdk/uploadSTBLogs.sh takes over the actual uploading task

- **How do I monitor the log upload process?**

  ➢ `$ tail -f /opt/logs/dcmscript.log`