Logging Library Low Level Design

UsesCases: 1) Multiple severity levels of the message e.g. Info,Error, Debug etc.

2) Logging severity level configured through config properties file. Only the messages whose severity is greater or equal to defined severity should be logged.

3) Multiple logging sinks. Each severity level can have its own sink defined, E.g. Info messages to be printed in text file, Error messages to be printed on console etc.

Observations: 1) Since logging library is used at multiple places, it becomes performance overhead if new instance has to be created every time. Therefore only single instance should be used throughout the application. *Singleton* instance is good option.

2) The application code should only be responsible to push the log messages. Handling the log messages should be decoupled from the application. *Pub-Sub* model of Java which is similar to observer pattern can be used. This has added advantage that consumer can be made low priority thread which does not impact application performance.

3) Since log level messages follow a hierarchy, using *Chain of responsibility* pattern is good option. Each message type has its own handler. If the passed message to handler is not of its type then simply redirect it to the next handler in line.

4) Choosing the sink is a *Strategy* which is decided by config properties.