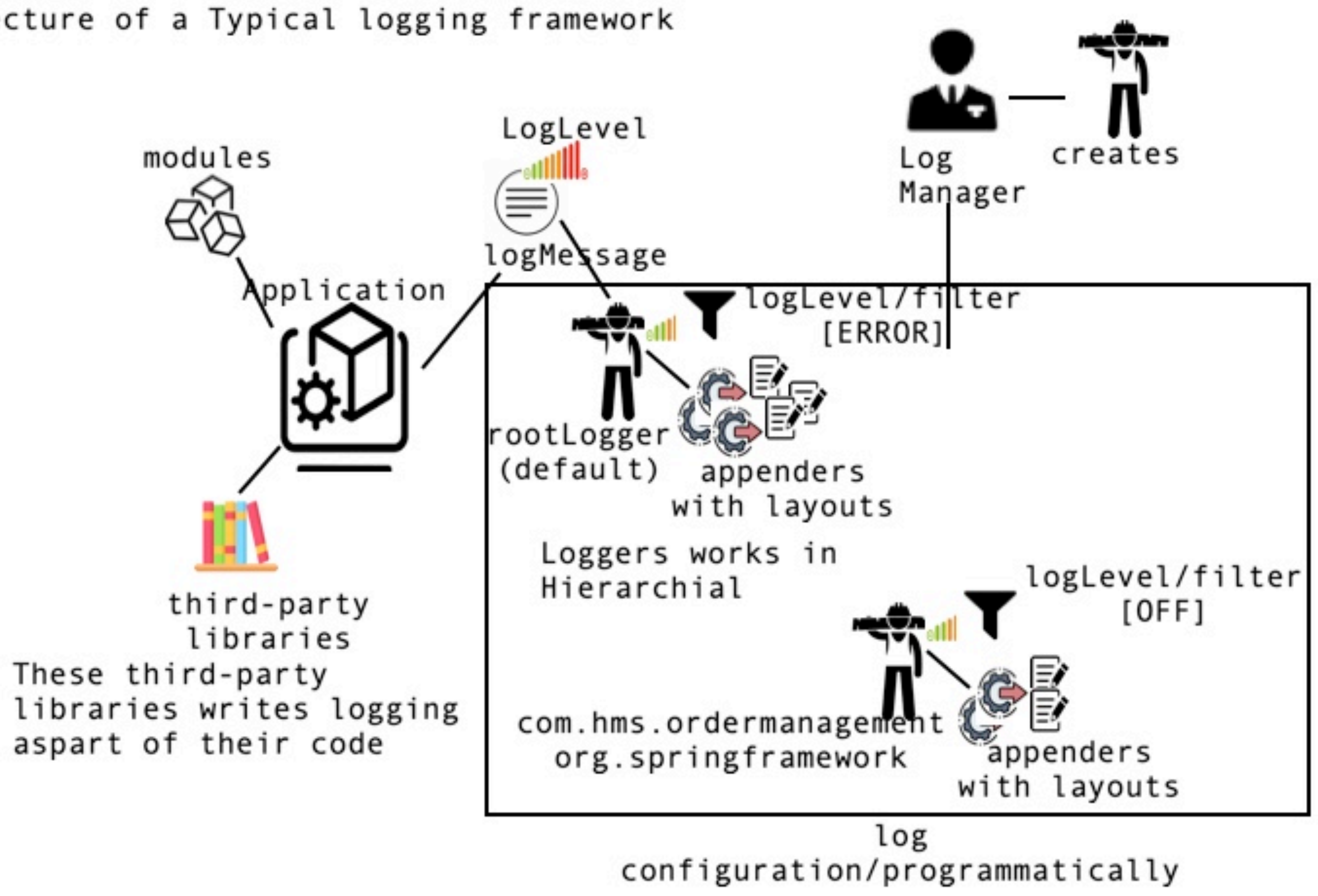
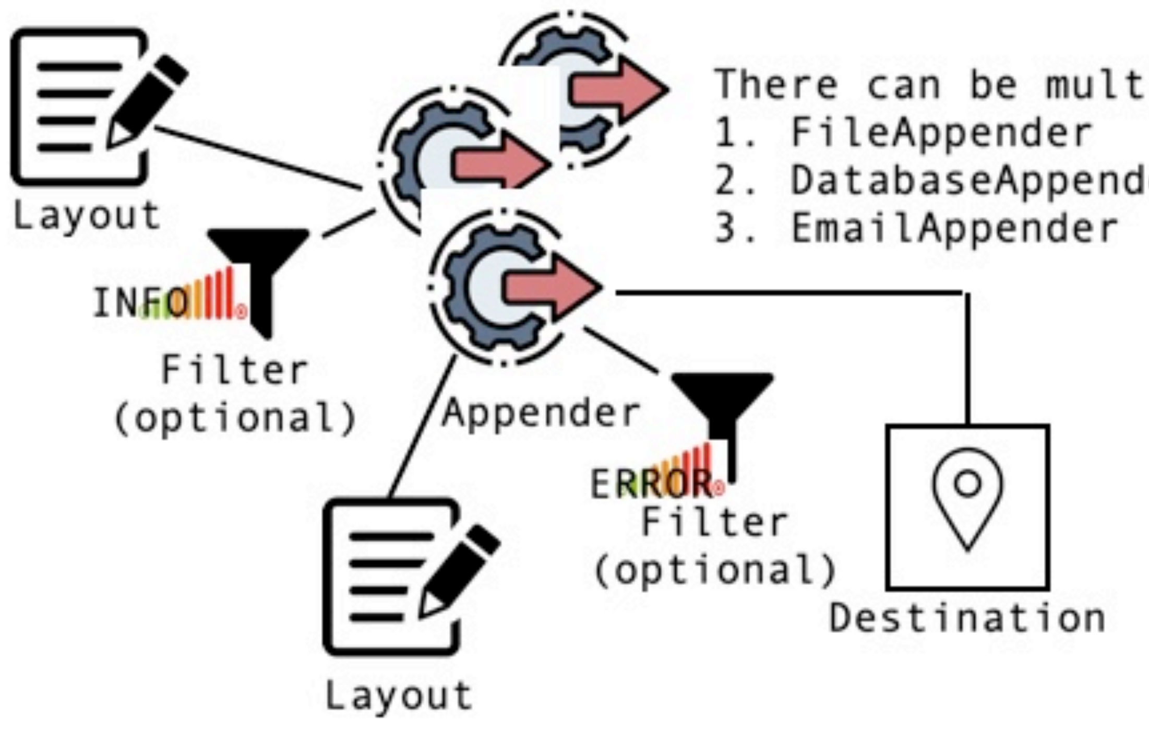
Architecture of a Typical logging framework



These third-party libraries writes logging aspart of their code

log configuration/programmatically

#1. The Appender abstracts us from writing the log messages into an destination
The destination could be: file, database or email etc



There can be multiple types of Appenders can be there like
1. FileAppender
2. DatabaseAppender
3. EmailAppender

LogLevel
The Log Level indicates the logged message provides what level of information about the application.

Typical LogLevels by an Logging framework would be:
1. TRACE = milestone / critical block like entry/exit of methods, conditions, loops etc are logged at TRACE level
2. DEBUG = the intermediate output generated by the code, that helps in debugging the application is logged at DEBUG level
3. INFO = The application/software generates the INFO messages to let the state of the system to the enduser.
4. WARN = The critical information about the application like reporting cpu usage or memory usage or threadcount that needs the attention of the user is being logged in WARN level.
5. ERROR = Application Exceptions are logged at ERROR level
6. FATAL = Failures that might cause the termination of the program are logged at FATAL level
7. OFF = turns-off all the log messages of the application.

The LogLevels are hierarchial in weight number order as below
OFF(7)
FATAL(6)
ERROR(5)
WARN(4)
INFO(3)
DEBUG(2)
TRACE(1)
If we set the LogLevel as INFO means all the messages from FATAL TO INFO will be logged
The lowest logLevel is = TRACE
and the highest is = OFF

#2. The Layout component formats the log messages send by the application before those are written to the destination like adding
1. classname
2. methodname
3. linenumber
4. data/time
5. threadid
6. processid
etc
these attributes to the log message. The Layout extracts these details from the ThreadStack inorder to include them into the log messages.

#3. Filter
Filter can be attached at the logger-level or at the Appender-level
It is used for filtering the messages to be included/excluded to be written onto the destinations.
How does filters knows which messages should be Filtered out/in?
Based on LogLevel.

```
package name convention:(app).(module).(stereotype);

package com.hms.login.servlet;
public class LoginServlet extends HttpServlet {
  private final static Logger logger =
                 LogManager.getLogger(LoginServlet.class); //fqnClass

  public void service(httpReq, httpResp) {
    logger.trace("entered into service method");
  }
}
```