Automotive DLT
Offline Logstorage Extension

AGL All Member Meeting 25.02.2016

Christoph Lipka, Senior Engineer

Advanced Driver Information Technology Corporation

ADIT is a joint venture company of Robert Bosch GmbH/Robert Bosch Car Multimedia GmbH and DENSO Corporation

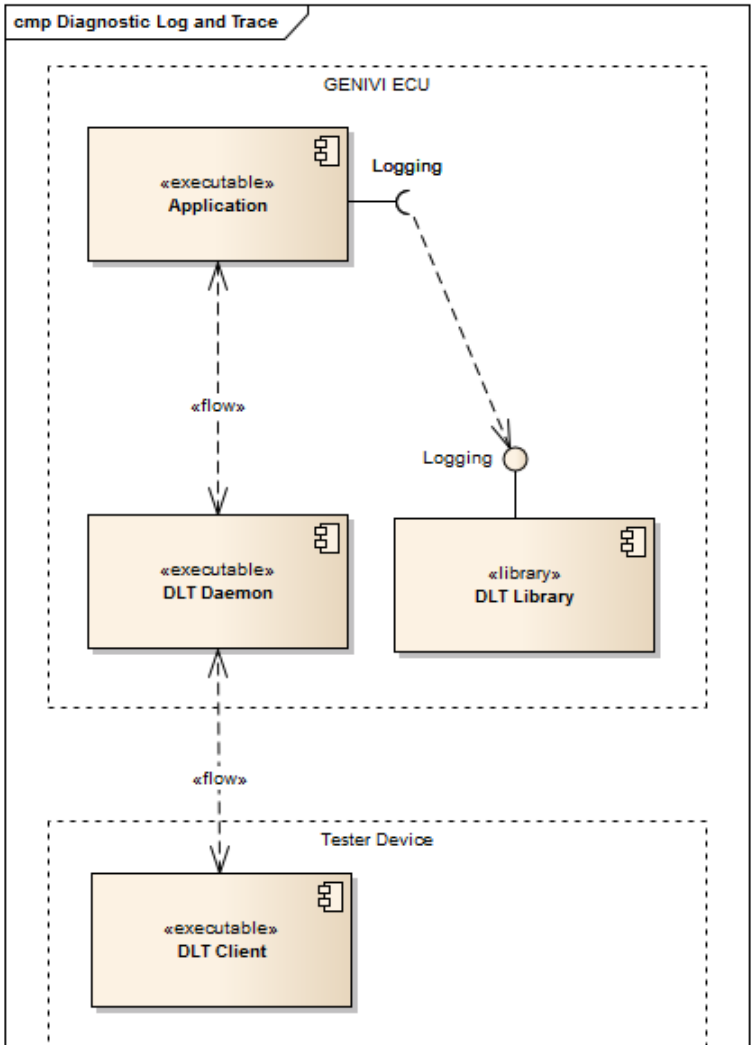
Outline

- Short overview DLT
- Use case and motivation Offline Logstorage
- Design overview Offline Logstorage
- Usage Offline Logstorage

Outline

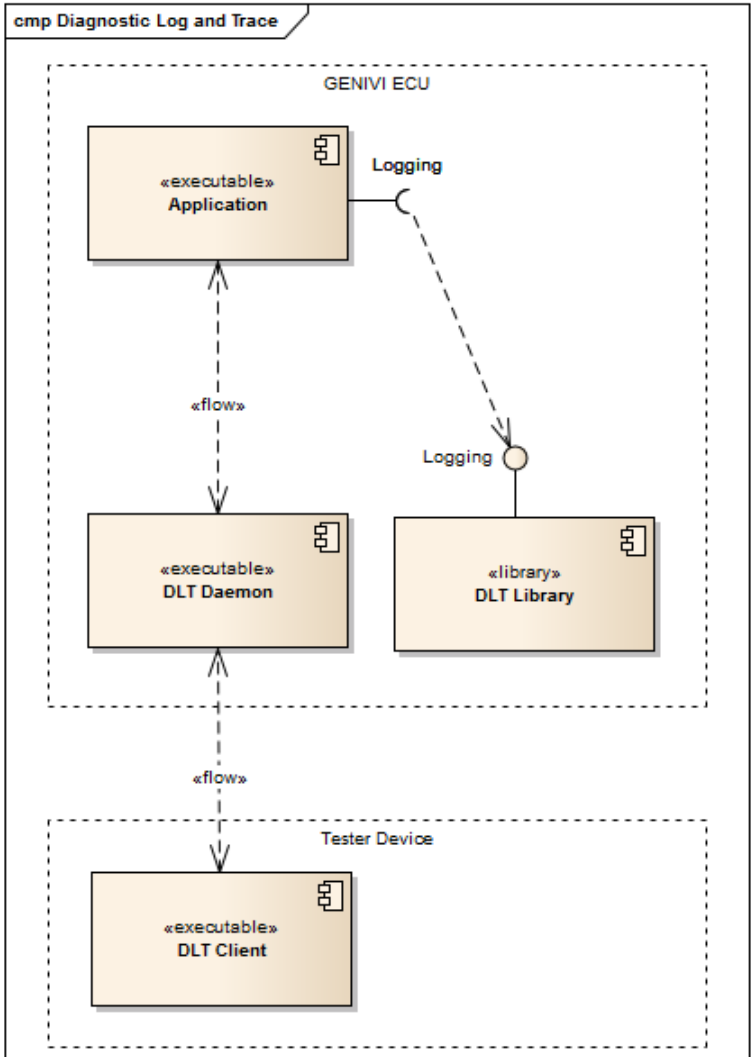
- **Short overview DLT**
- Use case and motivation Offline Logstorage
- Design overview Offline Logstorage
- Usage Offline Logstorage

- **Diagnostic Log and Trace**
  - Component for logging and tracing in ECUs in AUTOSAR 4.0
  - Provides standardized interface for logging and data format for transferring to host
  - Goals
    - Validation during development
    - Analysis of the end product
    - Standardization
- **Status Linux Porting**
  - GENIVI OSS component, current stable release 2.14.1, MPL 2.0 license
  - Based on AUTOSAR 4.0 standard DLT
  - Offine Logstorage introduced in v2.13
  - Already used by DENSO and Bosch
- **Information and Code**
  - *http://projects.genivi.org/diagnostic-log-trace/*



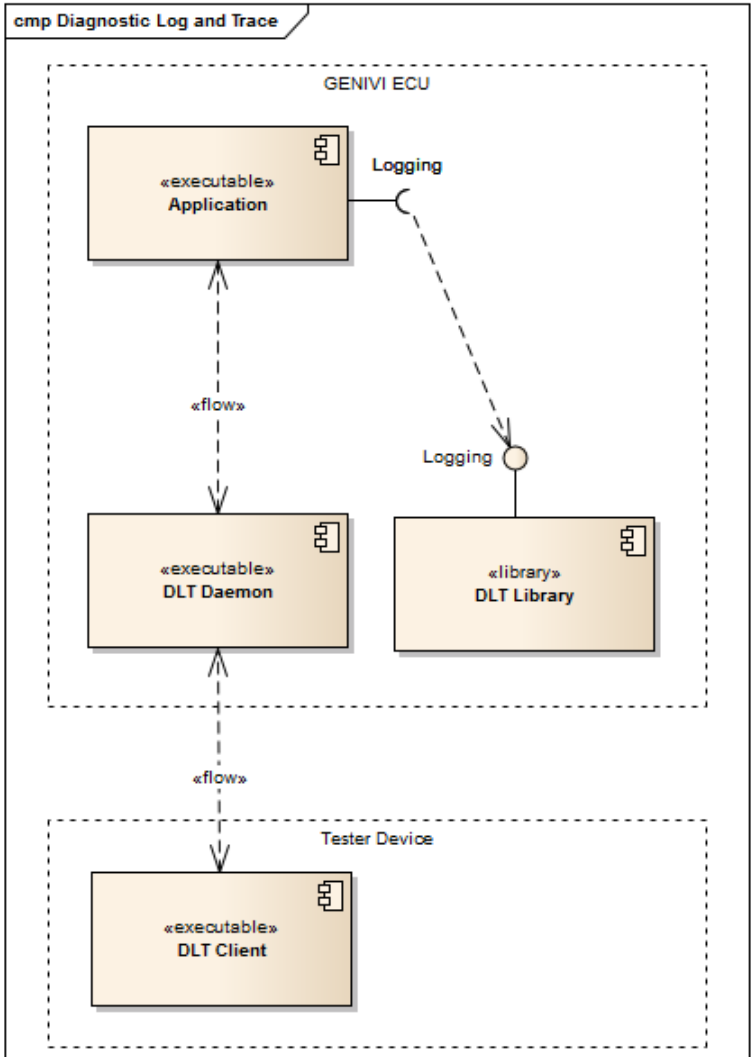
http://projects.genivi.org/diagnostic-log-trace/about

- **Supported Features**
  - Support for multiple applications with multiple contexts
  - Support for different interfaces between daemon and viewer (TCP/IP, Serial)
  - Verbose and Non-verbose mode logging
  - Predefined control messages
  - Message injection callback
  - User library and daemon providing a temporary internal buffer
  - Adapters to connect Linux log facilities like syslog
  - MultiNode support
  - Offline Logging
  - etc.
- **DLT Viewer „ready to use“ host tool to view logs, control the DLT Daemon and applications**



http://projects.genivi.org/diagnostic-log-trace/about

- **Supported Features**
  - Support for multiple applications with multiple contexts
  - Support for different interfaces between daemon and viewer (TCP/IP, Serial)
  - Verbose and Non-verbose mode logging
  - Predefined control messages
  - Message injection callback
  - User library and daemon providing a temporary internal buffer
  - Adapters to connect Linux log facilities like syslog
  - **MultiNode support**
  - **Offline Logging**
  - etc.
- **DLT Viewer „ready to use“ host tool to view logs, control the DLT Daemon and applications**



http://projects.genivi.org/diagnostic-log-trace/about

DLT Example Application

```
#include <stdio.h>
#include <dlt/dlt.h>
DLT_DECLARE_CONTEXT(ctx); /* declare context */

int main()
{
        DLT_REGISTER_APP("MAPP", "Test Application for Logging"); /* register application */

        DLT_REGISTER_CONTEXT(ctx, "TES1", "Test Context 1 for Logging"); /* register context */
        DLT_LOG(ctx, DLT_LOG_ERROR, DLT_INT(5), DLT_STRING("This is a error")); /* Write your logs */

        /* ... */

        DLT_UNREGISTER_CONTEXT(ctx); /* unregister your contexts */
        DLT_UNREGISTER_APP(); /* unregister your application */
        return 0;
}
```

DLT Example Application in DLT Viewer

Project

Id	Description	LogLevel	TraceStatus
ECU online	A new ECU [localhost:3490]	Default: info	Default: off
MAPP	Test Application for Logging		
TES1	Test Context 1 for Logging	info	default

Index	Time	Timestamp	Count	Ecuid	Apid	Ctid	ssionI	Type	Subtype	Mode	#Args	Payload
0	2016/02/23 12:01:25.…	73984.57…	0	ECU	APP	CON	0	contr…	reque…	non…	0	[get_software_version]
1	2016/02/23 12:01:25.…	73984.57…	0	ECU	APP	CON	0	contr…	reque…	non…	0	[set_default_log_level] 04 72 65 6d 6f
2	2016/02/23 12:01:25.…	73984.57…	0	ECU	APP	CON	0	contr…	reque…	non…	0	[set_default_trace_status] 00 72 65 6d 6f
3	2016/02/23 12:01:25.…	73984.57…	0	ECU	APP	CON	0	contr…	reque…	non…	0	[set_verbose_mode] 01
4	2016/02/23 12:01:25.…	73984.57…	0	ECU	APP	CON	0	contr…	reque…	non…	0	[set_timing_packets] 00
5	2016/02/23 12:01:25.…	73984.57…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[connection_info ok] connected
6	2016/02/23 12:01:25	73969.14	0	ECU1	DLTD	INTM	11421	log	info	ver…	1	Daemon launched. Starting to output traces
7	2016/02/23 12:01:25.…	73977.54…	1	ECU1	DLTD	INTM	11421	log	info	ver…	1	ApplicationID 'MAPP' registered for PID 11434, Desc
8	2016/02/23 12:01:25.…	73977.54…	2	ECU1	DLTD	INTM	11421	log	info	ver…	1	ContextID 'TES1' registered for ApplicationID 'MAPP
9	2016/02/23 12:01:25.…	73977.54…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[get_log_info 7] 01 00 4d 41 50 50 01 00 54 45 53 3
10	2016/02/23 12:01:25.…	73977.54…	0	ECU1	MAPP	TES1	11434	log	error	ver…	2	5 This is a error
11	2016/02/23 12:01:25.…	73977.54…	3	ECU1	DLTD	INTM	11421	log	info	ver…	1	Unregistered ContextID 'TES1' for ApplicationID 'MA
12	2016/02/23 12:01:25.…	73977.54…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[unregister_context ok] 4d 41 50 50 54 45 53 31 72
13	2016/02/23 12:01:25.…	73977.54…	4	ECU1	DLTD	INTM	11421	log	info	ver…	1	Unregistered ApplicationID 'MAPP'
14	2016/02/23 12:01:25.…	73984.58…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[get_software_version ok] DLT Package Version: 2.14
15	2016/02/23 12:01:25.…	73984.58…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[set_default_log_level ok]
16	2016/02/23 12:01:25.…	73984.58…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[set_default_trace_status ok]
17	2016/02/23 12:01:25.…	73984.58…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[set_verbose_mode ok]
18	2016/02/23 12:01:25.…	73984.58…	0	ECU1	DA1	DC1	0	contr…	respo…	non…	0	[set_timing_packets ok]

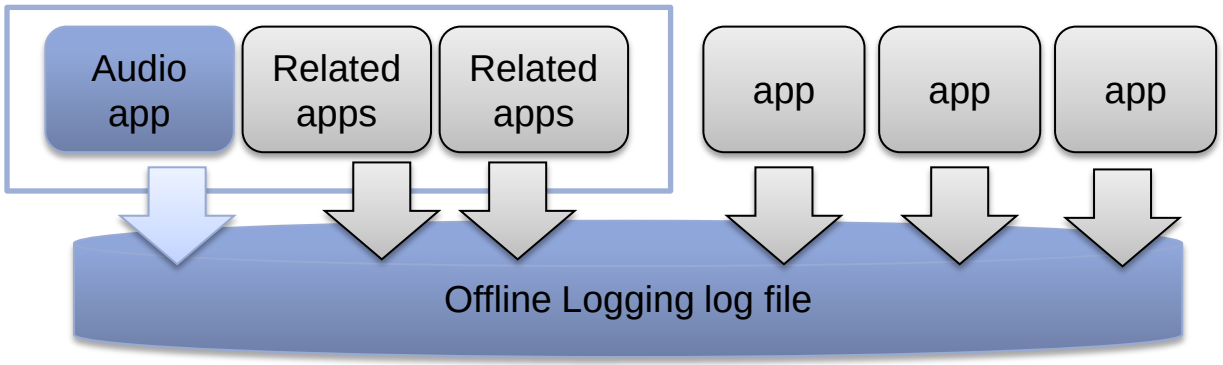
Outline

- Short overview DLT
- **Use case and motivation Offline Logstorage**
- Design overview Offline Logstorage
- Usage Offline Logstorage

Use case and Motivation

Choppy sound

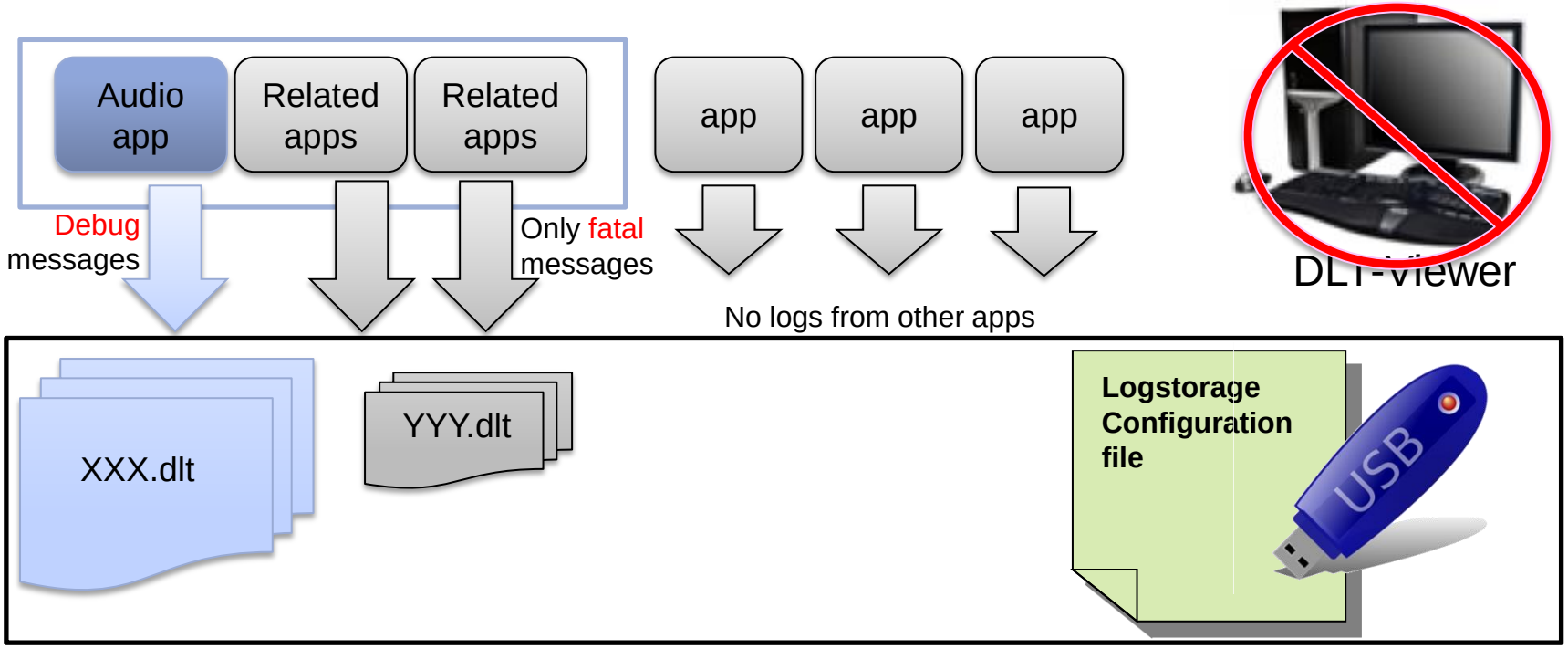
DLT-Viewer



- **Use Cases**
  - **System Test** DLT Viewer might not be available to control logging
  - **Production** Store specific log information over lifetime
- **Hardly possible with original Offline Logging feature due to lack of runtime configuration options**

Supported features

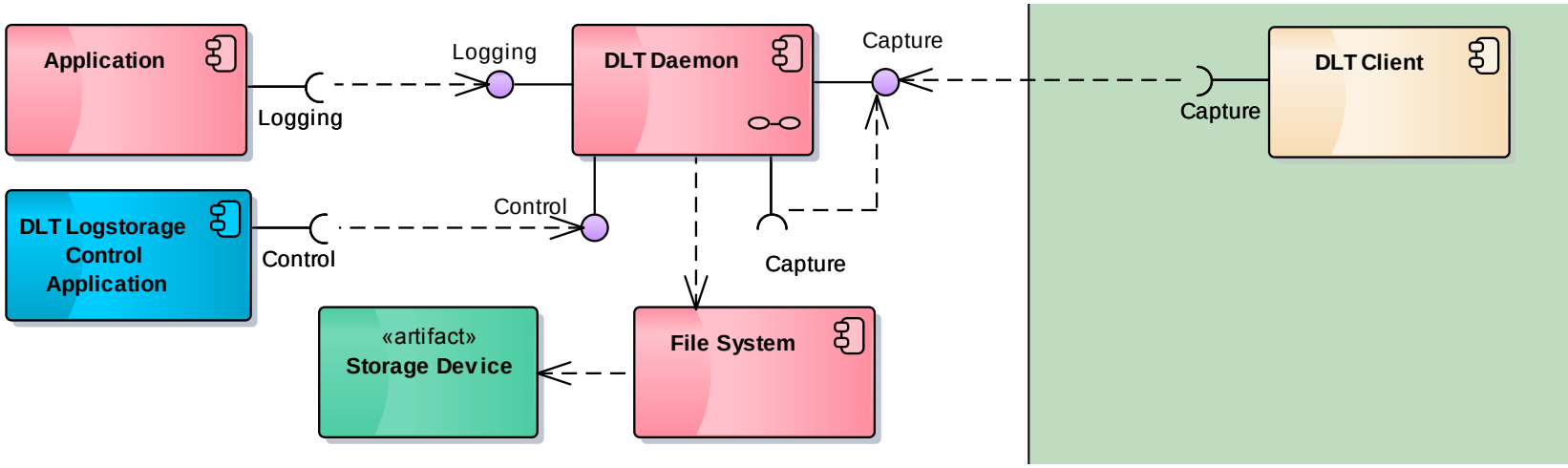
- Store application log messages to different storage devices available in the platform (e.g. USB, SD-Card, FLASH)
- Runtime configuration
  - Start/Stop logging into Logstorage device
  - Many configuration options per filter configuration



Outline

- Short overview DLT
- Use case and motivation Offline Logstorage
- **Design overview Offline Logstorage**
- Usage Offline Logstorage

Design Overview



- **DLT Daemon**
  - Receives logs from applications and forwards them to connected DLT clients or stores them temporary in a buffer
  - Logstorage is designed as DLT client inside DLT Daemon (like Offline Logging)
  - Incoming logs are filtered inside the Logstorage module and stored in DLT log files
  - If a log message passes the filter, it is stored in corresponding log file

[Filter<unique number>]

LogAppName=<Application ID, .* for all>

ContextName=<Context ID, .* for all>

LogLevel=<Log level>

File=<File name>

FileSize=<Maximum file size in bytes>

NOFiles=<Maximum number of files>

SyncBehavior=<Sync strategy>

EcuID=<ECU identifier>

- **Log Level setting**
  - During Logstorage device connect
    - Comparison between requested log level for a specific context (from Logstorage filter configuration) and currently active one
    - Updates the context log level by sending a request to the application
    - Remembers the active log level
  - During Logstorage device disconnect
    - Restore the previous active log level

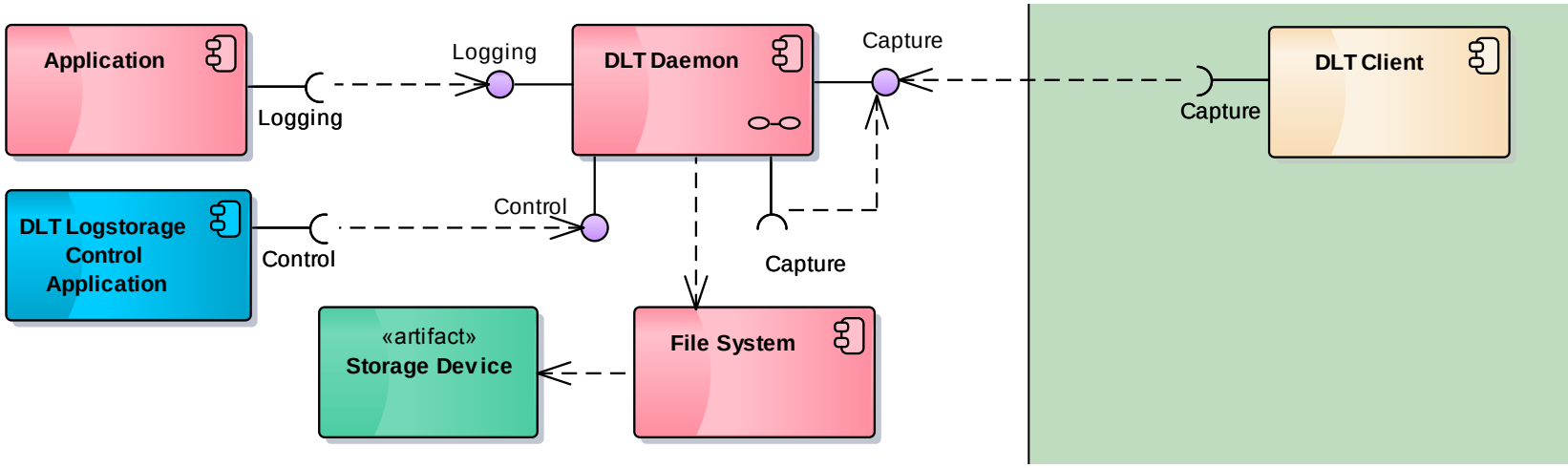
DLT_LOG_VERBOSE
DLT_LOG_DEBUG
DLT_LOG_INFO
DLT_LOG_WARNING
DLT_LOG_ERROR
DLT_LOG_FATAL

DLT_LOG_FATAL (active) AND DLT_LOG_DEBUG (requested) => DLT_LOG_DEBUG (active)

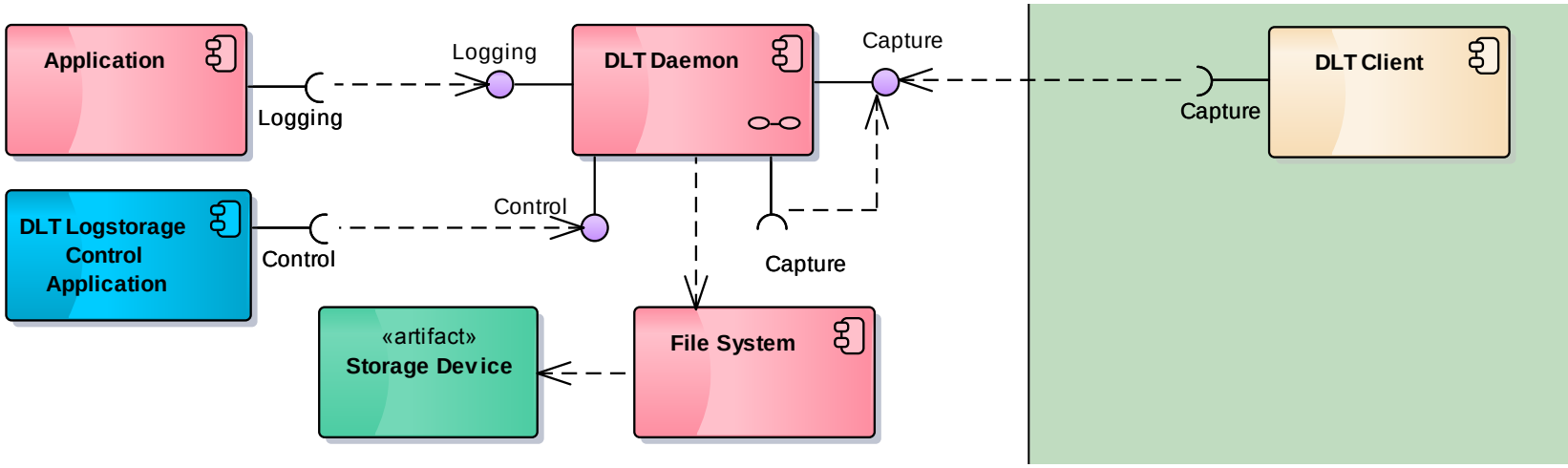
DLT_LOG_INFO (active) AND DLT_LOG_ERROR (requested) => DLT_LOG_INFO (active)

- **Filtering of incoming log messages based on application and context ID**
  - A filter configuration supports wildcards (“**.***”) and lists of application and context IDs
  - Wildcard for application ID and context ID is not allowed (Offline Logging)
- **The following filter configuration cases have to be considered**
  - One application ID (“App1”) and one context ID (“Ctx1”) -> “App1:Ctx1”
  - One application ID (“App1”), wildcard (“**.***”) for context ID -> “App1”
  - Wildcard of application ID (“**.***”), one context ID (“Ctx1”) -> “Ctx1”
  - Wildcard of application ID (“**.***”), list of context IDs (“Ctx1”,”Ctx2”): -> ”Ctx1”,”Ctx2” (same the other way around)
  - List of application (App1, App2) and context IDs (Ctx1, Ctx2): all combinations of application ID and context ID are possible. -> “App1:Ctx1”, “App1:Ctx2”, “App2:Ctx1”, “App2,Ctx2”
- **If an incoming message fits into one of these cases, it is stored into the corresponding filter file**

- **Logstorage on internal memory**
  - In case an internal storage device is used (e.g. eMMC), writing EVERY log message to the device has to be avoided due to limited number of write cycles
  - To overcome this, an internal ring buffer (RAM) can be used and synced to storage device based on a sync strategy
  - The following strategies might be considered, based on the use case
    - Write every log message (ON_MSG; default)
    - Write on daemon exit (ON_DAEMON_EXIT)
    - Write on demand (available soon)
    - Write on file size reached
    - …
  - Logstorage can be easily extended
  - The sync strategy can be set per filter configuration by
    - defining the “SyncBehavior”



- **DLT Logstorage Control Application**
  - Manual Control
    - Direct control of DLT Logstorage by
      - providing the mount point where configuration file is expected ("**-p**")
      - specifying the control action; 1 to connect, 0 to disconnect ("**-c**")
    - Example:
      - `dlt-logstorage-ctrl -p /media/9812-7B1E -c 1` (connect)
      - `dlt-logstorage-ctrl -p /media/9812-7B1E -c 0` (disconnect)



- **DLT Logstorage Control Application**
  - Daemonized Logstorage Control
    - Registration to device events depending on the chosen handler
      - New device/mount events
      - Device removed/unmount events
    - After receiving a mount event, the control application will check for Logstorage configuration file and send a control command to the DLT Daemon
      - `dlt-logstorage-ctrl -d` (Udev based control)
    - The control application can easily adapted to use other handler
      - `dlt-logstorage-ctrl -dprop` (proprietary based control)

Outline

- Short overview DLT
- Use case and motivation Offline Logstorage
- Design overview Offline Logstorage
- **Usage Offline Logstorage**

```
##############################################################################
# Offline logstorage                                                         #
##############################################################################
# Store DLT log messages, if not set offline logstorage is off (Default: off)
# Maximum devices to be used as offline logstorage devices
OfflineLogstorageMaxDevices = 2

# Path to store DLT offline log storage messages (Default: off)
# OfflineLogstorageDirPath = /opt

# File options
# Appends timestamp in log file name, Disable by setting to 0 (Default: 1)
OfflineLogstorageTimestamp = 0

# Appends delimiter in log file name, allowed punctutations only (Default: _)
OfflineLogstorageDelimiter = .

# Wrap around value for log file count in file name (Default: UINT_MAX)
# OfflineLogstorageMaxCounter = 999

# Maximal used memory for Logstorage Cache in KB (Default: 30000 KB)
# OfflineLogstorageCacheSize = 30000
```

Demonstration

- **Setup**
  - ADIT GENIVI platform inside VirtualBox
  - Store logs of 2 applications using Logstorage
    - dlt-example-user (LOG, TEST, DLT_LOG_INFO)
    - dlt-test-logstorage(DLST, TEST, DLT_LOG_WARN…FATAL)
- **Steps to rerun demonstration**
  - On Host
    - Prepare dlt_logstorage.conf configuration file
  - On Target
    - Configure DLT Daemon to enable Logstorage
    - Start DLT Daemon
    - Run Logstorage control application to connect Logstorage device to DLT Daemon
    - Start dlt-example-user
    - Disconnect Logstorage device
  - On Host
    - Open created DLT log file with DLT Viewer

```
[Filter0]
LogAppName=LOG
ContextName=TEST
LogLevel=DLT_LOG_INFO
File=example
FileSize=1000
NOFiles=5
```

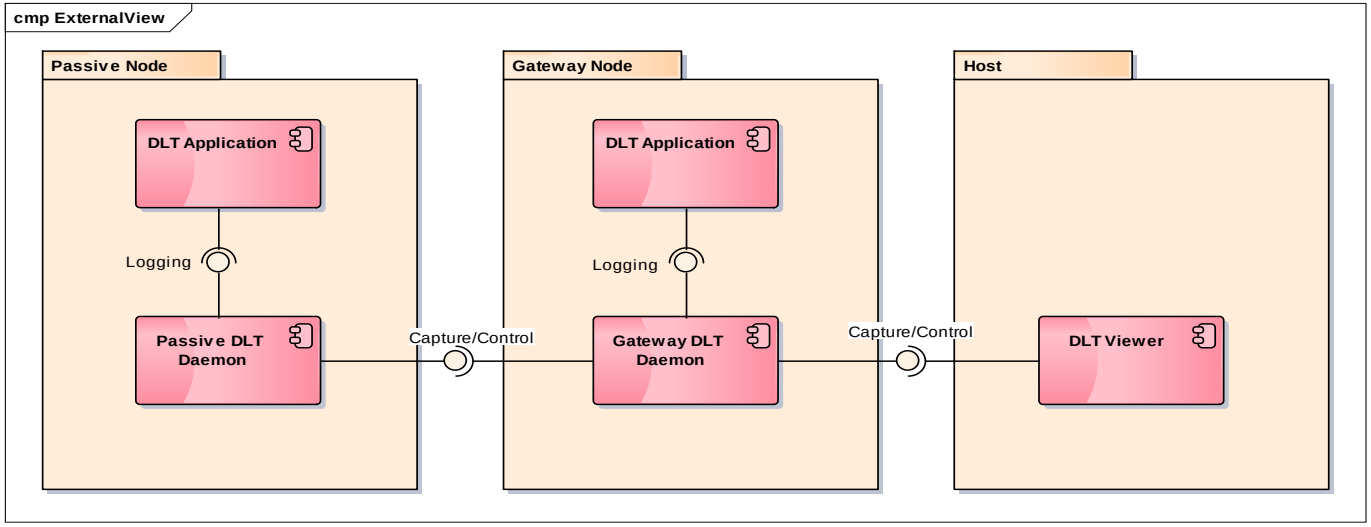
Thanks for listening!

Any questions?

Cmake options

- **Set control application backend**

```
option(WITH_DLT_LOGSTORAGE_CTRL_UDEV "PROTOTYPE! Set to ON to build logstorage control application with udev support" OFF)
option(WITH_DLT_LOGSTORAGE_CTRL_PROP "PROTOTYPE! Set to ON to build logstorage control application with proprietary support" OFF)
```



- **Gateway DLT Daemon**
  - Establishes connection to a passive DLT Daemon as DLT Client (on startup/on demand)
    - Only TCP currently
  - Forwards messages from Passive Nodes to DLT Clients
  - Forwards control messages from DLT Clients to Passive DLT Daemons
- **Forwarding decision is based on specified ECUid in message header**
- **Assumption: Every DLT Daemon is configured with a unique ECUid**

Configuration

- Enable Gateway Mode (dlt.conf)

```
# Enable Gateway mode (Default: 0)
GatewayMode = 1
```

- Passive Node Configuration

dlt_gateway.conf

```
[PassiveNode1]
IPaddress = 192.168.2.32
Port      = 3490
EcuID     = ECU2
Connect   = OnStartup
; timeout in seconds
Timeout   = 10
```