Deliverable **D5.2** /

Guidelines and Lessons Learned on Pilot Tools and Data

Version: 1.0 Final

Dissemination level: PU

Lead contractor: FORD

Due date: 31.08.2021

Version date: 06.10.2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723051.

Document information

Authors

Sami Koskinen (VTT)
Frederic Christen (FORD)
Johannes Hiller (ika)
Natalia Neila (JLR)
Erik Svanberg (SAFER)
Markus Kremer (FEV)

Coordinator

Aria Etemad
Volkswagen Group Innovation
Hermann-Münch-Str. 1
38440 Wolfsburg
Germany

Phone: +49-5361-9-13654
Email: aria.etemad@volkswagen.de

Project funding

Horizon 2020
ART-02-2016 – Automation pilots for passenger cars
Contract number 723051
www.L3Pilot.eu

Legal Disclaimer

Table of contents

List of figures

Executive summary

L3Pilot project enabled 750 test subjects in seven countries to try out the latest automated driving functions of prototype vehicles. The overall project objective was to test and study the viability of automated driving as a safe and efficient method of transportation.

This deliverable compiles lessons learned from test data management and development of data post-processing tools. It also serves as an input for upcoming large-scale tests of vehicle automation.

L3Pilot adapted the widely-used FESTA evaluation methodology and its Data Sharing Framework to Connected and Automated Driving (CAD) research. CAD is currently a very competitive field and companies protect technical details of their prototypes. Even though FESTA does not target to evaluate such technical details but high-level impacts on traffic system and the society, instead, sensor data logged from vehicles could still reveal some intellectual property (IP) of a company. For this reason, L3Pilot introduced a stepwise evaluation concept, where pilot leaders together with pilot data processing partners first process the detailed data into indicators and events. The analysis is continued at a consortium level based on the processed intermediate results, which were deemed possible to share.

Data analysis started by converting all logged data into a Common Data Format (CDF) – a format that L3Pilot created and promotes for future collaboration. The CDF is a HDF5-based file format, where each trip is saved as one file, including metadata that provides further information of the recording. The minimum required CDF signal list caters for various impact assessment areas, mainly: driving behaviour, user experience, mobility, safety, efficiency, environment, and socio-economics.

Creating conversion scripts from original logs into harmonized CDF required considerable efforts, as test vehicles had varying data collection setups. However, the alternative of using tailored analysis scripts for each different log format and vehicle type would have been very challenging in such a large research project with several vehicle manufacturers.

The common format enabled development of open-sourced analysis scripts that include indicator calculation, event and driving situation detection, and support for video annotation. The shared calculation framework ensures that fully comparable indicators, distributions and event lists can be extracted from each test site. The resulting indicators and data distributions were used as an input for impact assessment, where results from many test sites are statistically combined. Combining results from different test sites enables using a larger dataset for statistical work.

When selected indicator data and other input for statistical calculations were shared with project consortium for evaluation purposes, this data was considered to no longer contain key navigation software IP. Links to personal data were also severed using driver and vehicle ID pseudonymisation techniques such as hashing, and by sharing only video annotations, e.g. a numeric classification of the situation, not the video footage itself.

Besides protection of company IP, the project saw that also GDPR caused concerns on what data could be collected and shared. Most initial concerns were not warranted, as GDPR is not to hinder research and has related exemptions. However, the legal situation regarding sharing video even between consortium partners seemed to be a moving target with various views. Therefore, research projects would welcome EU-level practices and legal templates for sharing e.g. the front view video from a car within a research consortium, to speed up work and to avoid lengthy legal preparations. The current best practice is to add video data sharing clauses to the consortium agreement, to reduce the need for bilateral agreements. Video sharing also needs to be agreed with test users in their consent forms.

1 Introduction

1.1 L3Pilot objectives

Over the years, numerous projects have paved the way for automated driving (AD). The technology is rapidly advancing and today is at a stage that justifies large-scale testing. The overall objective of the L3Pilot project was to test and study the viability of automated driving as a safe and efficient means of transportation and explore and promote new service concepts to provide inclusive mobility.

Automation is not solved simply by integrating more and better technology. This topic needs above all a focus on user behaviour with automated driving systems. The interaction between the user and the vehicle, and user acceptance are keys to the success of AD on the market as well as understanding of the legal restrictions that need to be discussed and solved first on a broad level.

The project piloted SAE Level 3 and 4 functions exposed to different users, mixed traffic environments, including conventional vehicles and vulnerable road users (VRUs), along different road networks. The data collected in these extensive pilots supported the main aims of the project to:

- Ensure safe and sustainable introduction of higher-level automated vehicles on the roads.
- Lay the foundation for the design of future, user-accepted, L3 and L4 systems, to ensure their commercial success. This will be achieved by assessing user reactions, experiences and preferences of the AD systems' functionalities.
- Enable non-automotive stakeholders, such as authorities and certification bodies, to prepare measures that will support the uptake of AD, including updated regulations for the certification of vehicle functions with a higher degree of automation, as well as incentives for the user.
- Create unified de-facto standardised methods to ensure further development of AD applications (Code of Practice).
- Create a large databank to enable simulation studies of the performance of AD over time which can't be investigated in road tests, due to the time and effort needed. The data will be one product of the pilots.

The L3Pilot consortium brought together stakeholders from the whole value chain, including OEMs, suppliers, academic institutes, research institutes, infrastructure operators, governmental agencies, the insurance sector and user groups. More than 750 users tested 70 vehicles across Europe with bases in 7 Countries, including: Belgium, France, Germany, Italy, Luxembourg, Sweden, and the United Kingdom, as shown in Figure 1.1. The project lasted for 48 months, and road tests started in April 2019 and piloting on variable road conditions lasted for two years.

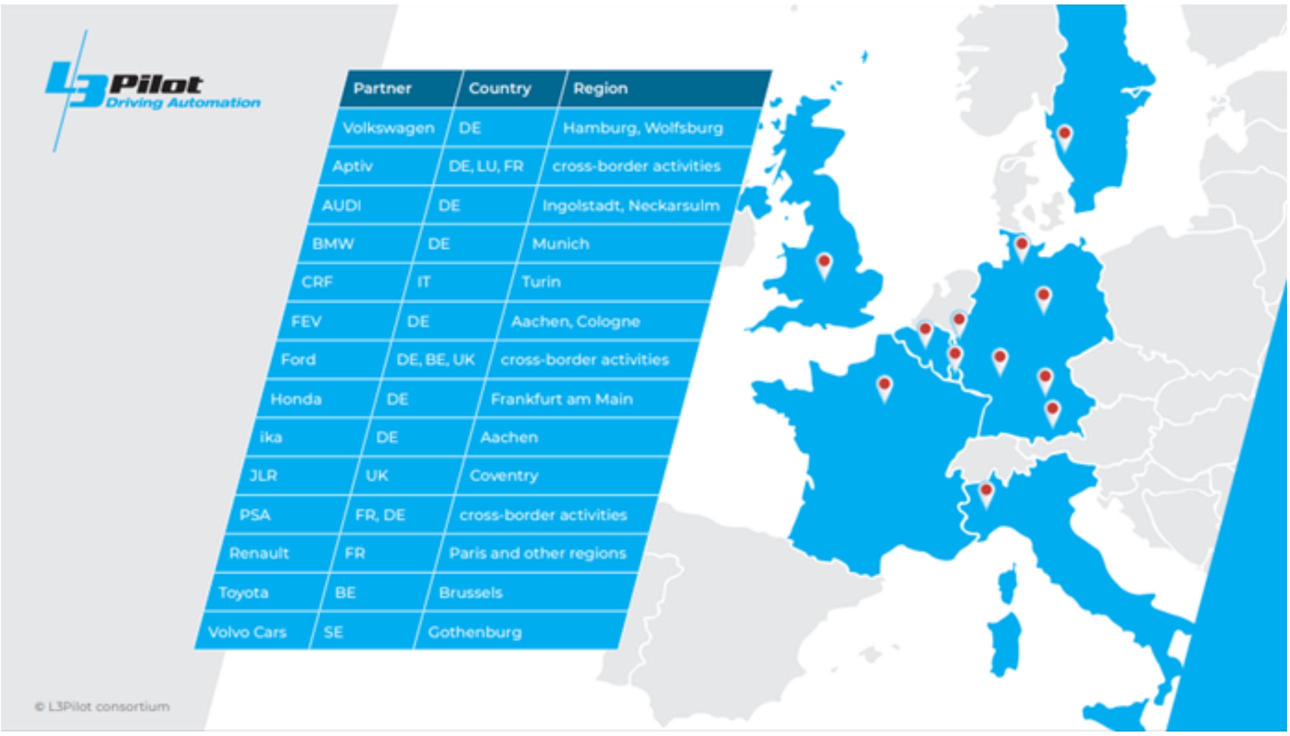

Figure 1.1: L3Pilot testing areas.

1.2 FESTA Methodology

The project follows the FESTA V-process methodology [1] of setting up and implementing tests. FESTA was an EU support action that created a testing methodology to be used in Field Operational Tests (FOTs). L3Pilot has adapted the methodology for ADF testing.

The FESTA methodology has been used in practically all EU-funded FOTs during the past thirteen years. The main handbook has been periodically updated by the FOT-Net community to capture lessons learned and to document further best practices. The current EU support action updating FESTA is the ARCADE project. Through its networking activities, ARCADE has been collecting feedback on applying FESTA to automated driving tests. L3Pilot has been strongly involved in these networking activities via its partners.

1.3 Purpose of this document

This document compiles the lessons learned and main guidelines from L3Pilot's subproject 5 (SP5), Pilot Tools and Data. The purpose is to offer experiences and recommendations to future pilot projects and field operational tests with a specific focus on data handling aspects. It also reflects on FESTA Handbook guidelines, especially the Data Sharing Framework [2].

The lessons have been collected mainly during the first two years of the project, when the data management tool development took place. The main workshop concentrating on lessons was held as a main part of an SP5 meeting in Offenbach, 28.–29.5.2019. At that time, pre-tests and preparation for pilot tests were ongoing. Further feedback regarding the use of tools during the pilot tests, and with larger datasets was consolidated in project general assemblies, in 2019 and 2020. In 2021, final lessons learned have been collected to update this document. In order to provide better insights of the process, the lessons learned are discussed in textual form instead of just presenting a bullet point list.

Since L3Pilot partners have previously taken part in several FOTs, e.g. euroFOT and DRIVE C2X, this document also considers the lessons learned in such past large-scale testing campaigns. The lessons learned presented here are focussing the data collection and analysis. For lessons learned with respect to the Piloting phase or which have been collected by the evaluation partners during the various analyses, the reader is referred to L3Pilot deliverables D6.5 [7] and D7.3 [8] respectively.

The lessons learned are divided in three main chapters: Data Logging, Data Management and Data Analysis. Data logging, management and analysis are based on the concept of Pilot Leader with associated Pilot Data Processing Partners. Figure 1.2 shows this concept and the related data flow.

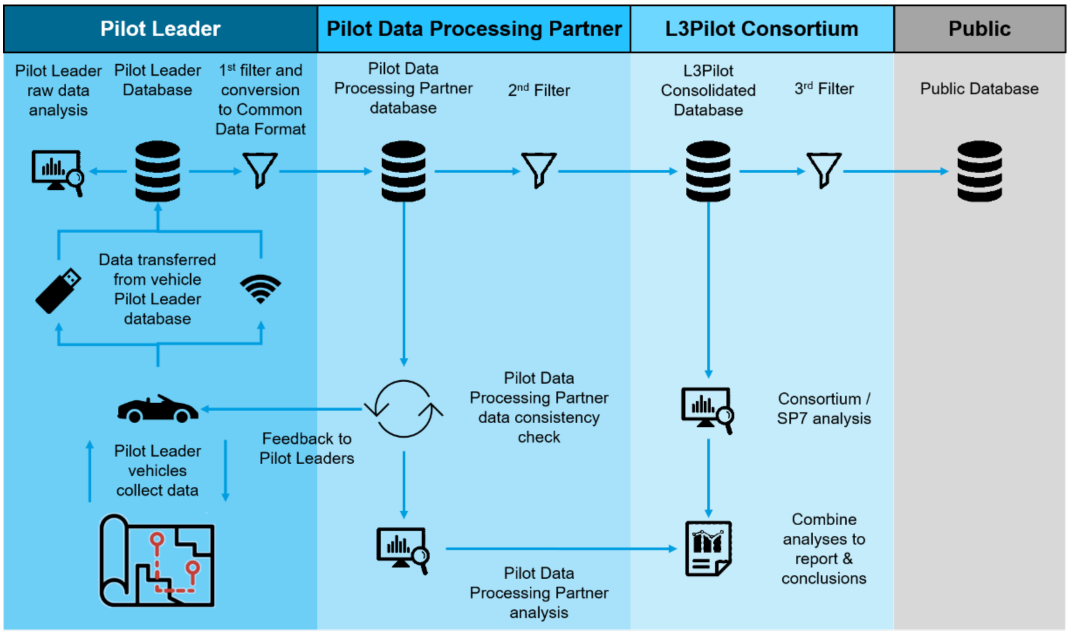

Figure 1.2: Data flow within L3Pilot.

2 Data Logging

2.1 Discussion on test protocol

Every test project should start with a discussion on **research questions** and a definition of their priorities. Prioritisation of evaluation goals brings clarity to data collection and test planning. Moreover, a clear focus helps to achieve understandable and well-justified results.

International testing campaigns can certainly afford to address numerous research questions and they usually set out to make a wide assessment on societal impacts of the new tested technology. They generally also have the resources to collect and manage large datasets. As a result, their datasets capture a lot of detailed driving data, enabling research and re-use also outside the project's main goals. Still, unless specifically targeting naturalistic and long-time driving data collection, projects must focus their data collection and evaluation resources on prioritised aspects of driving.

In L3Pilot, automated driving functions (ADFs) were tested on specific routes, using prototype vehicles which had previously undergone validation steps to ensure safe and smooth operations. Both pre-pilot and pilot test data was collected from these test areas and routes.

Using a **rigorous testing protocol** was the first key aspect to ensure that effects of the functions can be measured as clearly as possible when compared to baseline driving. Test protocols may define, for example, that a test route is always the same, or that the vehicle has to be started in the same location, to simplify data analysis. Considering the prototype nature of the L3Pilot functions, the test protocol is also to ensure safety of all participants and other road users. The prototyping phase of automated vehicle development sets a few understandable limitations: initial operation areas are small, vehicles generally cannot be used in everyday driving, development-phase data has high confidentiality regarding companies' plans for upcoming products, and the functionality of the vehicles continues to improve during the project. These factors also mean that, from evaluation perspectives, some or even most of the data becomes outdated in a few years after the project has ended. Even during the project, vehicle changes have to be managed and kept to minimum, if they affect evaluation.

L3Pilot captured a snapshot of how automated driving technology looks like today. The project scales up the detailed findings from log and survey data to societal impacts with various methods, e.g. macroscopic simulations and transparent stepwise calculations, to estimate impacts of automated driving on society. The project is among the first to collect scientific evidence on near-future impacts.

2.2 Data requirements

In order to compile a list of signals that need to be recorded in the test vehicles, the first step is to analyse the automated functions (i.e. their abilities and their Operational Design Domain, ODD, cf. [5]) and use cases from evaluation perspectives. In L3Pilot, there is a major focus on motorways.

In a further step, research questions are derived from the specified use cases (cf. [6]). Accompanying the research questions are various hypotheses that are to be investigated.

For each hypothesis, specific data can be defined to confirm or reject it. This gives a first idea of **signals that need to be recorded** within the project. One part concerns subjective data specifying the acceptance or feelings of the driver or passenger. This data mostly originates from questionnaires and user evaluations. A second part is the objective data recorded in the vehicles, particularly interesting from a data processing point of view and therefore relevant for the data format.

Based on the hypotheses and the required data, **derived measures and performance indicators** are defined. In this context, derived measures are quantities that are directly calculated from source signal time-series data. These can be vehicular signals or information about the environment delivered by car sensors. Performance indicators, on the other hand, are no longer time-series data. They take different forms depending on the indicators. They can be single values or e.g. averages in a recording or in a specific scenario. However, they can also form histograms over multiple occurrences of a driving scenario. Using this collection of derived measures and performance indicators, their calculations are considered and a list of signals that needs to be recorded within the vehicle can be derived.

Once that a general signal list is available, it needs to be agreed with the pilot leaders, who have the responsibility over the testing campaigns. This initial list of data requirements is then converted into a well-defined specification for the commonly available vehicle signals (with the required frequency, units, resolution, etc.). In an iterative process, this list is further refined based on the feedback from all pilot leaders. Examples of logging requirements were:

- Ego vehicle signals (speed, acceleration, brakes, steering)
- Object data (number of objects, their positions and classifications)
- Lane data (number of lanes, lane position, ego vehicle position with respect to lanes)
- Cameras.

The next important phase involved the transfer of this list into the L3Pilot **Common Data Format** (see [3]), supporting a wide range of automated driving research questions. The format is discussed in more detail in section 3.1. In order to store the signals, partners came up with a hierarchical structure ordering the signals by their coarse usage or origin:

- egoVehicle: Signals relating directly to the vehicle itself, e.g. speed or acceleration
- objects: Object list of surrounding objects such as other vehicles
- laneLines: List of lane markings to the left and right of the ego vehicle
- positioning: Signals related to the positioning system, i.e. GNSS

Additionally, metadata was added to identify the trip and simplify the processing of multiple trips. A discussion on formats and metadata is presented in the following chapter 3.

2.3 Sensors and equipment

2.3.1 SSD storage

Hardware durability is always a topic in vehicle tests. The equipment will have to endure vibrations, heat, and sometimes fluctuations in power supply.

In L3Pilot, most pilots used SSD (solid state drives) storage devices to store data. In comparison with electromechanical drives, SSDs lack the moving parts and are more resistant to vibrations. They also read and write data faster and, still today, are more expensive.

One pilot leader found that not all SSDs would fit their purposes; some had too low data rates, at least temporarily. Another pilot used SSDs in RAID 0[1] configuration to further speed up write performance, writing simultaneously on at least two disks.

Much of the high-speed requirements came from logging raw vehicle data and not directly in the common data format, which is more compact. For development and in-house evaluation purposes, many experiments also logged 360-degree video with high resolution, even if the common data format required only low-resolution videos.

2.3.2 USB cameras

Regarding camera set-ups, a few companies struggled with using multiple (>3) USB video cameras at the same time, connected to a same computer. They reported driver issues, limitations by the USB controllers and software crashes. The solution generally was to use separate video modules in between, i.e. products specially designed for such a purpose. A few hardware suppliers support up to four cameras (Figure 2.1) were found to work well. Each camera could use a different resolution and frame rate.

Figure 2.1: Setup of two Axis F44 base units to collect USB camera video feeds.

Regarding the needs to adjust resolution for each camera, one pilot used a grayscale camera for the driver's feet, and a high-resolution front camera to be able to read road signs.

[1] Wikipedia.org: RAID 0 splits data evenly across two or more disks. This configuration is typically implemented having speed as the intended goal. The drawback is that RAID 0 provides no fault tolerance or redundancy. A failure will result in total data loss.

2.3.3 Time synchronisation

Time synchronisation of data coming from different sensors is a common discussion topic in vehicle tests. Data is often collected with different computers, which basically are all in different time, and still data must be merged in post-processing. Data from a single sensor may go through pre-processing steps before being saved on a computer, thereby causing delay. Occasionally, video frames may be dropped due to connection problems or file saving delayed due to heavy CPU load. What can make synchronisation even more difficult is the case when some data is transmitted over communication links.

Generally, the different logging computers should be kept accurately in time. Today, there are various tools for achieving this over internet connections, as well as GPS hardware options. Time stamping should be done close to the sensors, to avoid delays caused by data processing or communication. One pilot commented that their setup in L3Pilot, using time stamping close-to-sensors, was a clear improvement from their previous projects.

L3Pilot also tried to manage the expectations for time synchronisation by not requiring higher than 0.1 second accuracy for the CDF. After all, usually only a few technical measurements need a higher accuracy and those were up to each test site to consider.

2.4 Reference data logging

The first step of the data flow is the collection of data from the vehicles. The **user-friendliness of a data logger** solution is very important, as the effort for using the data logger system should be as low as possible. The best is a configurable system that can be used by any driver (trained or untrained) and without user interaction.

Also, of great interest is how intuitively the user interface of logger hardware (remote control) and software is implemented.

- Is it necessary to start the measurement manually at the data logger itself (e.g. by pressing a button)?
- Can the measurement be started or stopped by configurable triggers (e.g. CAN signal, remote control button)?
- Is it possible to install the data logger in the vehicle without any user interaction (e.g. automatic start/stop by ignition ON/OFF or any trigger condition)?

The data handling of the data logger is another important point for the daily work of data acquisition and the subsequent analysis and evaluation since the amount of data from video logging is expected to be very high. The data format plays an important role here. If this is a standardized data format, further analysis will be possible with many software products rather than if it is a proprietary solution from the data logger provider.

This quickly raises the questions:

- In which file format is the logging data stored?

- What export options are available in other data formats (i.e. CAN .asc, Video .avi/.mp4) for third-party software?
- What analysis options are available with the supplied software?
- How easy is the data exchange (e.g. USB-HDD, removable SSD, Copy-Station) between the data logger and the vehicle owner's database to minimize the vehicle standstill time?
- How fast is the data transfer (i.e. GbE, 10GbE, USB3, Firewire) from the data logger to the data centre / Cloud storage?

For the logging of data in future related projects, it would be interesting that the logging can be divided into two different types, according to the following considerations.

The information to be logged is quite different when considering the needs of the OEMs and the needs of project partners who want to analyse the behaviour of drivers, vehicles and the interaction between vehicles and environment.

Since OEMs are interested to log the raw data from the vehicles for further improvement of the functionalities, they need to collect the raw data for reingestion needs. On the contrary, the needs in the project for a more general evaluation are not the real-time raw data, but some lower resolution data that can be logged separately.

Regarding the data collection environment for OEMs, all of them need an environment which is capable of pre-analysing and pre-selecting data and use it for different needs.

On the one hand the data which is collected will be used for reingestion into Hardware-in-the-Loop, Software-in-the-Loop or Model-in-the-Loop environments for verification purposes. On the other hand, the OEMs also want to use automatic data analysis for validation needs during the development phases. This cannot be done manually but needs an automated environment.

In case there is some free processing capacity in the logging system, it might be interesting to include a part of pre-processing already there. With the help of this approach, the effort of processing the collected data after the test-drives can be significantly reduced.

Another option for future projects could be to transfer data, which is needed for project relevant analysis and not for reingestion and development of functions, from the vehicles to a cloud system already during the test drives. Then it is possible to start the analysis as soon as the first data is available. That might lead to much lower effort for the test sites.

Also, it could be beneficial that there is a centralized database for the project specific data, which can be filled earlier without additional effort from the pilot sites.

The following part of the lessons learned is about introducing an efficient way to transmit and handle the collected data to optimize the data transmission as well as the following storage and analysis. Some key elements of the framework, such as data generation, private or open cloud backend, data format conversion and in-depth analysis are addressed with practical examples from real world testing.

The vehicle owner is responsible for collecting all the raw data coming from its vehicle including sensors and actuators. In addition, external data such as specific road or weather conditions can be included. For this purpose, a combination of two loggers proved to assure a smooth process, providing the best efficiency in data collection and data handling. One logger (called hereafter ICU) is connected to cloud servers through the mobile network and streams specifically selected data continuously from the vehicle. Another logger is collecting all raw and object data from sensors like cameras, radars, lidar and GPS as well as actuators like the brake system and the steering control system and stores this data on a hard disk inside of the car.

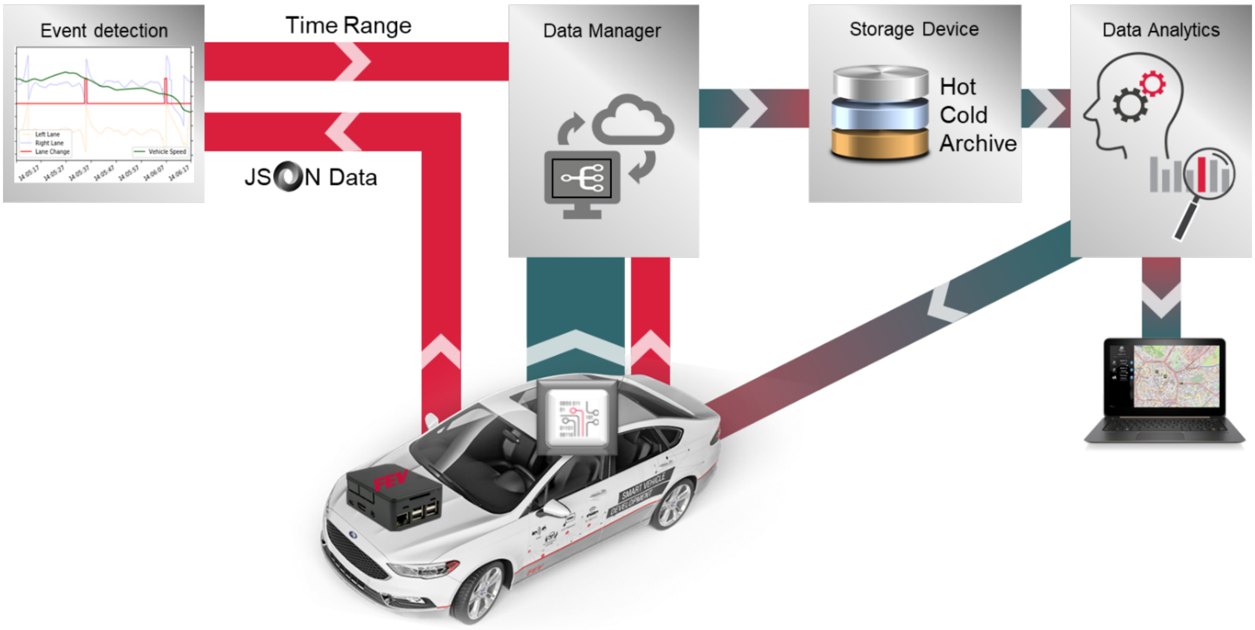

Figure 2.2: Data logging environment with event detection and postprocessing capabilities.

The analysis of the data begins as soon as it is delivered through mobile network to the cloud via the ICU. As soon as the data is available on the server, the data is pre-processed in a way that it is synchronized to the same time steps for all CAN bus data.

Among other evaluation steps, scripts are running which are used to perform automatic detection of certain scenarios defined in the L3Pilot project. When such a scenario is detected then the corresponding data inside the database is automatically annotated.

The aim of this scenario-based assessment is to analyse the behaviour of the automated driving system from a technical as well as from a traffic point of view.

To ease the analysis of pre-defined Performance Indicators (PI), a web-based dashboard could be used for future projects that represents how the automated driving system performs against the selected PIs. The dashboard is exemplary shown in Figure 2.3.

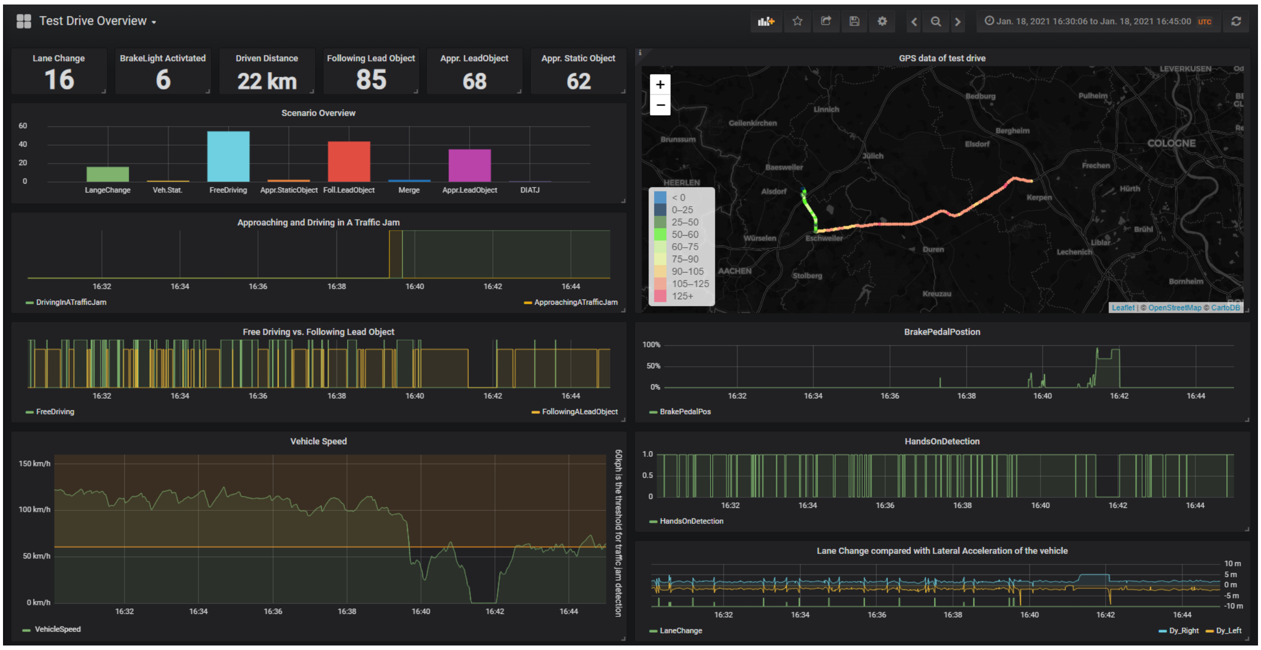

Figure 2.3: Overview of dashboard for one test drive.

Some data collection environment like described above can be easily used for enhancing the usage and preparation of the overall collected data and is **a suggestion on how to improve for future needs in next generation projects** for data logging environments.

3 Data Management

A significant effort was invested in data management in L3Pilot, specifically in **handling large and complex raw data** and converting it to a common data format. The ambition of having a common data processing suite creating derived measures, scenarios, and performance indicators, implied developing and applying a common data format. The requirements for this format were collected following the FESTA methodology as adapted in L3Pilot, to ensure that all relevant measures were included answering the hypotheses. In addition, cooperation with the pilot sites ensured that the relevant signals were available and could be shared. The definitions of each measure included naming, units, data types, resolutions and required frequencies. There were also specifications for the range of values (eliminating incorrect data points or test codes), description of enumerations, codes for unavailable data, and how to deal with data drops and methods for interpolation, all with the aim of creating coherent and harmonized data.

Another important aspect was **data quality**. A data quality checklist was provided to the pilot sites, the pilot data processing partners, and the evaluation partners.

The legal perspective was also addressed, and in particular measures were introduced to **safeguard intellectual property rights** of the companies as well as de-identification of participants in accordance with the rules of GDPR.

The application of all these procedures, after a number of suitable processing steps, allowed to upload the results of the pilot tests to a common consolidated database.

These key subjects are discussed with more detail in the following subchapters.

3.1 Harmonization of collected data

In a project with several test sites, the evaluation methods and results could be very different if the test sites would work independently. Scattered results are difficult to combine if the goal is to assess general effects at international level. Whereas, if the **testing and evaluation is harmonized,** there is a greater chance to pool results together and to consider findings from a bigger set of measurements and larger user groups. The more similar functionalities, methodologies and data collection methods the different test sites have, the easier it becomes to have statistically valid results.

Collaborative work methods, e.g. open source development, can provide outstanding results if the partners agree to share the code and understand the need for collaboration. In automotive testing projects, the vehicle features/functions are usually developed separately (although there have been exceptions) and the collaboration is in test data collection and evaluation. Use of common data logging equipment would simplify a project greatly but combining data from different sites requires sharing at least requirements to collect the same content. In L3Pilot, a common data logger was not practically possible since many Pilot leaders had logging and data management processes already established.

Harmonized data post-processing can produce the same key performance indicators (KPIs, e.g. the average speed on a road that has a 60 km/h speed limit) and detect events (e.g. situations where a car is overtaking another) from different tests, even if the original data formats differ. In that sense, harmonizing the data *format* is not as important as harmonizing the actual data *content*. For example, as long as GPS speed is logged, the file format for how that information is saved can be handled in post-processing, so that common indicators can be calculated.

However, given that the data collected in L3Pilot was much more diverse than plain GPS data, the project found necessary to harmonize both the data format as well as the content. When aiming for a common data processing suite, a common format and definitions are mandatory.

L3Pilot created a **Common Data Format**, CDF [3]. Besides being able to read in any test data sample, the main benefit from using a fixed format is that it enables developing a wide set of tools on top of it. The CDF tools developed in L3Pilot include a data viewer and video annotation tool, harmonized derived measures, numerous driving scenario detection scripts, key performance indicator (KPI) calculation scripts, data distribution analyses, etc. For example, L3Pilot developed scripts for detecting almost 30 different types of driving scenarios from sensor and map data, including amongst others: following a lead object, cut-ins, merging, crossing with a laterally moving object, turning with a lead object, and approaching a traffic jam.

In brief, the CDF definitions contain a large selection of signals to record during the tests and the required sampling frequencies of these signals. A single trip is saved as one HDF5 file. Depending on the specific data acquisition system and pilot test design, some pilot sites although activated data logging when entering the test area and deactivated when leaving.

The CDF was built on HDF5 file format that can be read with tools such as MATLAB and various programming languages (Python, Java, C++). Basic metadata such as the type of the trip (baseline or treatment phase of testing) and the driver types (ordinary drivers, professional drivers, etc.) and IDs are stored in the file's metadata table. The different datasets within the file had either the Pilot Leader or the Pilot Data Processing Partner responsible for providing its content.

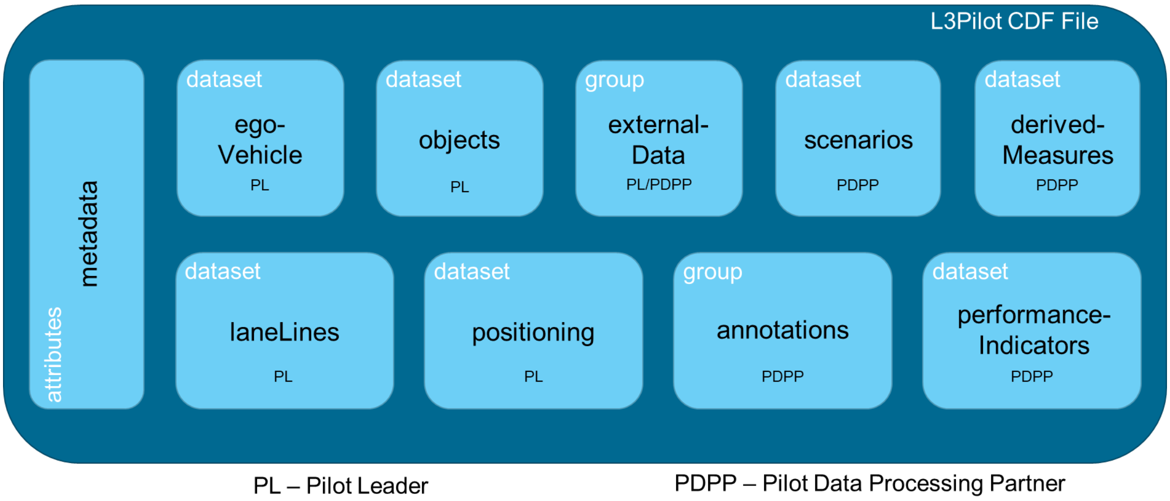

Figure 3.1: L3Pilot CDF File.

The project has made efforts to promote the common data format for future projects as a collaboration format. The CDF is available as open source, at https://github.com/l3pilot/l3pilot-cdf.

Using a common format between OEMs and academia is useful. However, it must be kept in mind that these companies have departments with different logging needs and are already collecting data with existing tools.

The development of software for data conversion was a significant task and it had to be implemented by all pilot sites individually. The project held a series of supporting teleconferences for pilot leaders to discuss conversion issues, sample code and solutions. In many cases, vehicle data collection systems had to be modified simultaneously to match signal collection requirements. If a L3Pilot definition was changed, or if changes to the vehicle or data acquisition system arise, the conversion script had to be updated as well.

Some important attributes, not foreseen before data collection started, could have been part of the common definition. E.g. driver interventions (supporting the AD function in steering or braking) were available in some vehicles, but not included in the common format. Also, continuous data quality attributes (e.g. lane detection confidences, or the distance of how far the lane line could be measured) could have improved the scenario predictions.

A suggestion is also to indicate traffic disturbances (accidents, temporary road works) from manual annotations or time-based geofences[2] to explain that certain behaviour differentiated between drives. In the end, partners recognised that harmonization and conversion of data to the project's common format were worth the efforts.

The main drawback of the HDF5-based format the CDF is built upon, was that the open source HDF5 itself still lacks wide source code examples and some documentation. Additionally, HDF5 Java support was also only partial and reading in the L3Pilot files required implementing some

[2] Wikipedia.org: A geofence is a virtual perimeter for a real-world geographic area.

missing code parts; for example, handling complex variables, which entailed a table within a table. However, as HDF5 is open source, L3Pilot was able to contribute new code to improve the Java support of the format.

3.2 Metadata

When collecting sensor data for research, it is always necessary to write down or generate the context the measurement was intended for. Thus, **metadata** have to be carefully addressed. This “data about data” is called metadata.

There can be many types of metadata. For example, the Data Sharing Framework (DSF), currently maintained by the ARCADE project (https://www.connectedautomateddriving.eu/data-sharing/), proposes different categories of metadata: administrative metadata on who can access the file, descriptive metadata on describing measures and their origin, and structural metadata on describing a file format. Sometimes people are referring to a text document or a spreadsheet to provide context – in which case test documentation might be a more suitable term, but metadata has a wide meaning. However, metadata is something that would be slow or difficult to deduct from the sensor data itself and it is needed for efficiently searching and using the data.

In L3Pilot, CDF contains a Metadata table, where test operators mark down e.g. driver ID (pseudonym number) and type, number of occupants, whether the file/trip belongs to baseline or treatment phase of the study, and if the trip was made on a test track or on public roads, etc. Also, some weather attributes could be filled in manually, if this information is neither available from vehicle sensors nor generated in other ways.

Labelling data manually during driving is a common approach in small-scale tests, supervised tests and system development, whereas in naturalistic long-time testing it is rather important to automate metadata generation. There are **different levels of annotations/labelling**: those that are marked only once for the file, e.g. a file name, and more continuous observations of what happened.

Human observations can be used to mark interesting cases for detailed evaluation and often they can be cases which computers have missed to recognise. Still, human real-time labelling has its limitations, if many attributes would be required. In many test projects, such detailed annotations are done in post-processing, running videos back and forth and using special video annotation tools.

Many L3Pilot pilots saw it necessary to label data manually during driving. It was made to clarify what files belong to which category. Also, interesting cases could be singled out. More detailed annotations were left for video annotation during post-processing (see section 4.2). L3Pilot also used automated scenario detection based on sensor data.

Previous FOT/NDS projects propose an anomaly button, indicating an event of interest, and this approach can also be important in several phases of testing. This was implemented by some pilot leaders indicating an event of interest either for development purposes or analysis phase.

Some partners used tablet computers to save metadata during tests (Figure 3.2, Figure 3.3). Also, real-time annotations were implemented in some cases to classify weather and traffic density. They also used the same tablets to run post-drive questionnaire for participants.

Most partners implemented methods to quickly monitor the status of different systems and data logging. In prototype vehicles, it is useful to have a display showing which subsystems are online and working properly.

Related to GDPR, some pilot leaders implemented the process of participant consent using an app on a tablet. This can offer an advantage over having paper versions of consents. Apps could also be used for storing driver ID.

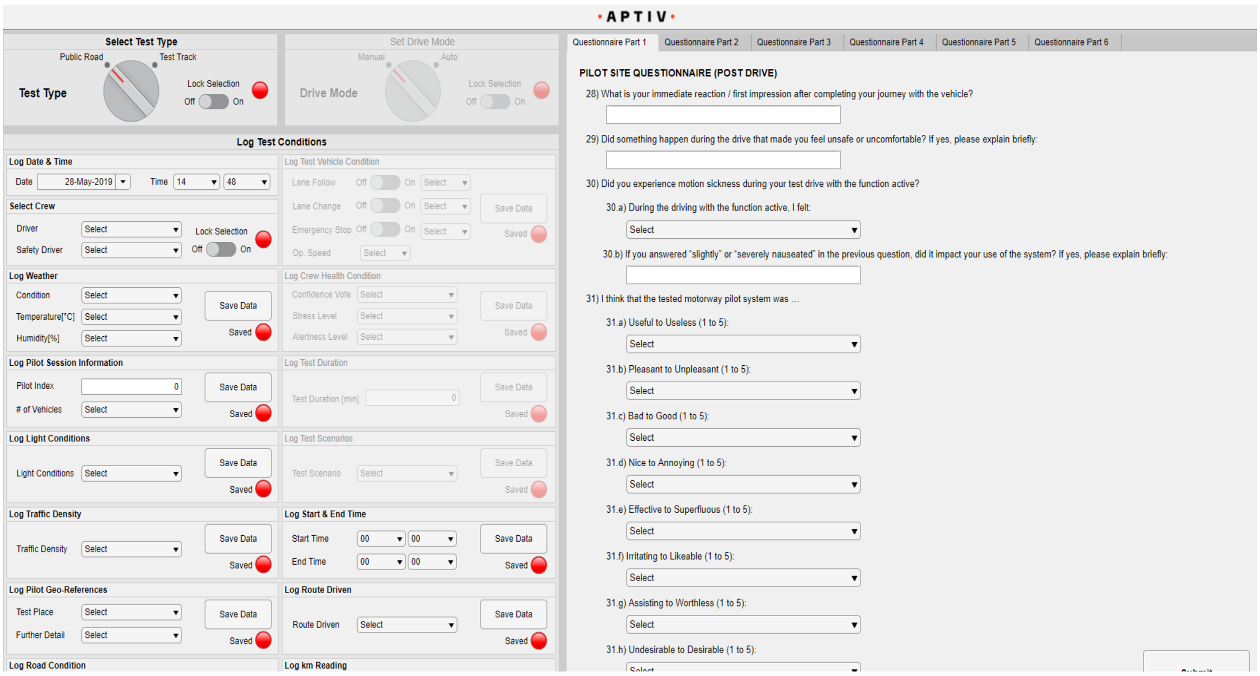

Figure 3.2: Metadata capture along with post-drive questionnaire.

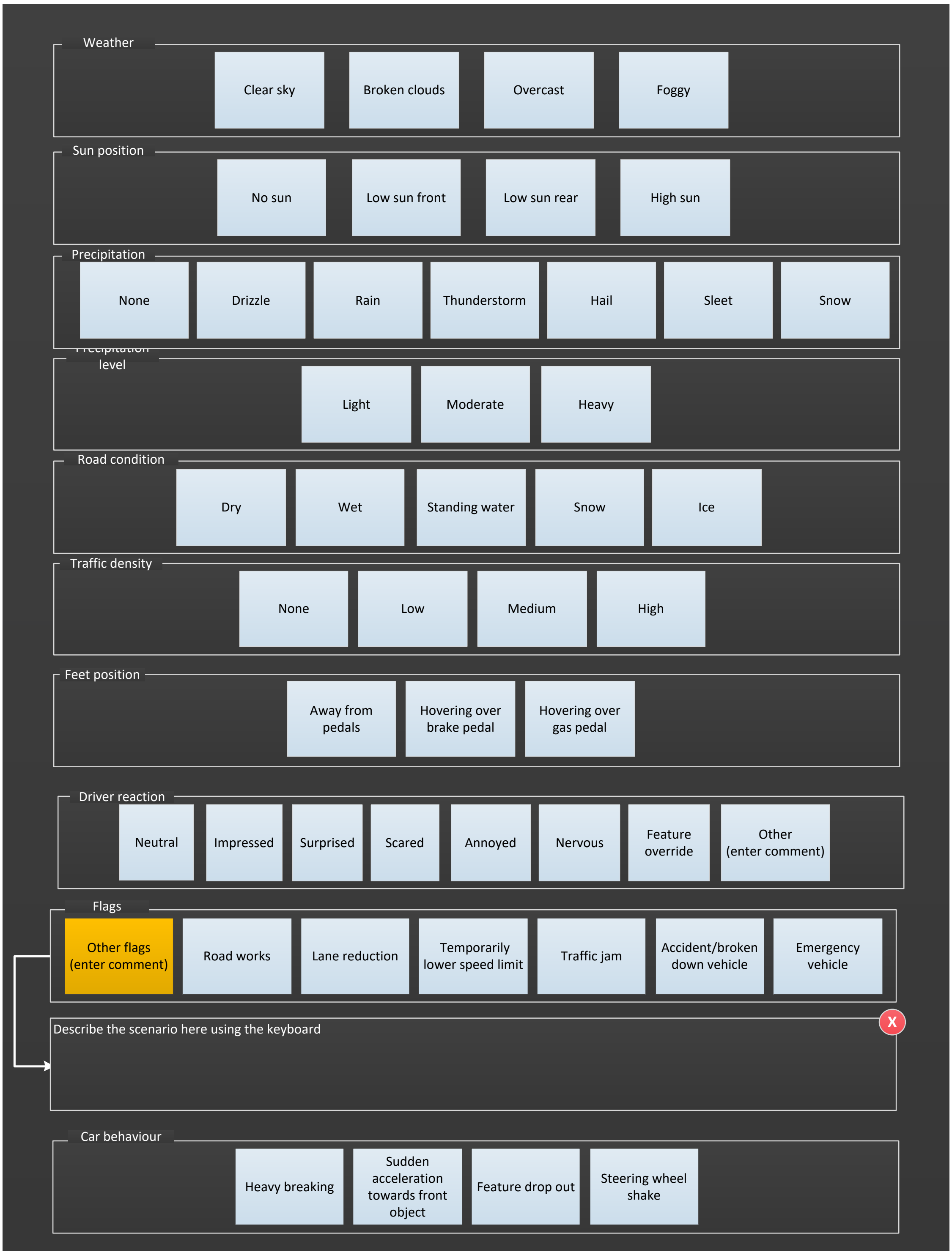

Figure 3.3: Metadata capture and logging status screen.

3.3 Software and data versioning

The efforts in L3Pilot, developing a common data processing suite, led to **multiple software versions.** In addition, the data processing workflow had many steps executed by different actors. In the pilot sites there were updates to the data logging in the vehicle and updates to the AD function itself.

Every time there were changes in vehicle data collection (e.g. a sensor interface), this might have an effect on the scripts to convert the logs into the CDF. Even the log quality would change with a sensor change.

Modifications to the CDF required minor updates to the pilots' conversion scripts. Some of the pilot leaders automated the whole conversion process, running through log files when a change was introduced in the data definitions. Pre-testing activities solved many issues, but much work had to be done to complete the data processing chain.

Next, multiple updates to the data processing tools that build on the CDF – the DMPI framework and the aggregation tools – led to frequent re-processing activities by the pilot data processing partners.

As a consequence, an error detected in the consolidated database could have been introduced in any of the steps. In many cases, the issue was not easy to identify (more a suspicion that something is incorrect). The project would have benefited from improved versioning of the data, including dates for processing and the different versions of the source code generating the data as part of the metadata for each HDF5 file.

Data versioning could also be a part of the data processing suite, in the DMPI framework, aggregation tools and in the consolidated database. Many pilot data processing partners did have this as a manual step for the output of the DMPI framework and the aggregation tools. Exports of the full consolidated database were also performed manually to let the analysis partner work on the same database. It is proposed to take this into account prior to pilot data processing since updates to the software are likely to happen.

As so many large-scale testing projects done in the past, L3Pilot again highlights the importance of pre-testing and recognizes the long time it takes to properly complete this process.

3.4 Quality and validation

Many factors contribute to **overall data quality**, for example:

- Peer-reviewed and focused test setup using thoroughly tested functionality in the test area, as explained in Chapter 2.1
- Data collection where sensor output has been tested against ground truth (or accurate reference sensors) and time synchronisation has been tested
- Harmonized data collection content over different tests, targeting the same key performance indicators

- Frequent monitoring of systems' status to notice broken or disabled systems
- Monitoring of collected data regarding the number of detected interesting events and e.g. kilometres driven
- Evaluation of sample data before actual tests begin, to ensure that results can be calculated from the collected data.

Achieving high quality in pilot and FOT testing requires processes to be set up at the test site, e.g. appointing a data manager to periodically sift through collected data. The larger the tests, the more the focus shifts towards automated testing.

Some main goals of quality checking are to

1. Ensure that the data is usable for evaluation purposes and selected KPIs can be calculated
2. Frequently monitor data collection process so that as little as possible testing would have to be repeated due to bad or missing data.

In L3Pilot, at first, quality was the responsibility of the pilot leaders, ensuring correct logging in the sensors installed in the vehicles. This included alignment of sensors, analysing distance metrics and synchronization between different sensors. One lesson learned was to have a "reference trip", testing the vehicle functions and manually coding the time when doing it. This trip should cover as many conditions as possible, also testing signals not frequently used (e.g. wind shield wipers). For specific checks, like harsh braking or warnings from a safety system, these tests might have to be performed on a test track. The annotated trip should be shared with a pilot data processing partner to be used for checking the scenario creation scripts.

The next step was ensuring correct converted data format and full content of sample log files: that they were according to the CDF specifications and fulfilled the minimum requirements for logged signals. For this purpose, L3Pilot offered a Java-based tool, L3Q, for the Pilot leaders to check the basic contents of log files and output web reports per log file. As the software was able to process data quickly, it could also offer a quick check after a test day (but only after CDF conversion).

Further, the pilot leaders (with the support of pilot data processing partners) analysed the collected data in terms of semantic checks e.g. analysing:

- if hard braking could be detected
- if lane markings could produce trustworthy lanes by comparing to forward video
- time stamp consistency from data synchronization by validating e.g. vehicle speed against forward video when starting and stopping the vehicle
- raw and re-sampled data to ensure that anti-aliasing or filters applied give artefacts.

Moreover, analysts tested if the DMPI-framework scripts run without errors on the files – all the MATLAB scripts that L3Pilot used to calculate derived measures, performance indicators and scenarios. Final phase of quality checking is if the logged data at all contains interesting events for further evaluation – for example, if the test was about overtaking, did the data contain any overtaking events.

In the first versions of the scenario (here: prototypical driving situations) detection scripts many strange cases were detected. Some of them happened outside the test routes. Certain test sites did not have this problem at all since they were starting logging when entering the test area. For the other pilot sites, this was solved when adding a measure describing if the vehicle were within the test area and generating scenarios only then.

One pilot leader highlighted that they did some event detections already during test drives and transmitted event-related summary data over mobile phone networks to a test controller. This enabled them, when a car returns from a test, to already roughly know what data and how many events were collected. Such monitoring has also been considered a good practice in past FOT projects. It requires additions in vehicle processing and a data link to be set-up. In large-scale tests, such monitoring is nearly mandatory to ensure successful data collection.

As a whole, the data quality process was challenging. A detailed quality check checklist was provided for the pilot sites, containing more than 30 data quality checks. The data quality checklist gave the partners a lead on what to do to ensure the quality of the data. The project could though have defined more tools than the L3Q. One proposal is to have a set of semantic checks could have been included, giving warnings of data suspected to be incorrect. This can include simple checks, e.g. short or long state events (based on the scenario definition) or extreme values, but also combination of states and measures that would disqualify some sections not making sense.

It should though be noted that even these checks would not solve an error detected in a script creating “cut-ins” (a vehicle entering the ego vehicle lane): A minor bug of what was the left and right lane caused the script to detect “cut-outs” from a lane. The overall data looked ok, often there were a cut-in (although not perfectly synched since this happened prior to the cut-out). This was later detected in manual video analysis but the number of places this could have been introduced was many. The aggregate values associated to “cut-outs” were of no extremes and even semantic checks would most likely not detect the error. This is a reminder of the value a ground truth forward video camera view. If having semantic checks, this could have been shared among the partners building a repository of checks for the scenarios created in L3Pilot.

As a special case related to environmental sensing, L3Pilot had to agree on how to handle temporary losses of objects. Many partners had issues of ghost objects (e.g. reflections from tunnels or larger vehicles), temporary loss of lane tracking, different identification numbers of the objects, and inconsistent size of the objects. Future sensing systems might solve this within the camera or lidar, or better algorithms should be applied in post processing. In the project each measure had a maximum time for data losses. If data loss is higher than this value, the values should be changed to unknown or unavailable. In addition, it can be useful to set a minimum time for a measure, e.g. to filter objects that lived shorter than 0.3 seconds.

3.5 Privacy and GDPR

In the beginning of L3Pilot, it was clearly seen how companies were extra careful about collecting or sharing logs containing personal data. **GDPR** (General Data Protection Regulation, EU 2016/679) was introduced in May 2018, and there were uncertainty from all actors on how the new law would affect our domain; what can, and cannot, be shared. In addition, they wished to **protect their company IP**, i.e. ADF technology and future product plans.

Regarding research involving tests with users, GDPR clarified former best practices of managing user data.

L3Pilot addressed personal integrity of the study participants and data protection early in the project and best practices from past projects were used to set a common ground (e.g. FESTA handbook, CAD “Data sharing framework”, Drive C2X deliverable D36.1 “DRIVE C2X fleet/user management to test sites”). Still, each partner respectively was responsible and in-charge of for implementing the necessary routines and processes to ensure data protection of the study participants. For research and development, the key has always been about having a valid purpose for collecting and managing data. If the collected dataset contains personal data, it must be properly protected against leaks and misuse. The data collection purpose and targeted use are what the test subjects accept in their consent forms when taking part in a study. The agreement must be clear and understood by both parties. In many cases, an introductory session is needed before entering the tests, to ensure both that the agreement is well understood and to explain safety precautions.

These principles have always been there, but in the past 20 years more and more people have become aware of the legal side. Between the GDPR (2016, implementation in 2018) and its predecessor Directive 95/46/EC, 24 October 1995, there were many technological changes in data collection from vehicle tests. When the first field operational tests (FOTs) started in Europe around the year 2008, the GPS location data of a car, for example, was not considered to be especially sensitive, by most. After all, the data would not even contain factual information on who was driving on a given day.

Since then, more and more data loggers have begun to identify the driver and store video. It has become widely understood that location data can be used to detect e.g. speeding violations, when comparing GPS log data against digital maps. Location traces from long-time use can also point out frequently visited locations (mostly home, work and grocery store), unless post-processed to improve anonymity. Coordinate logs can also be combined with other application logs, e.g. Facebook activations, maybe indirectly revealing someone’s identity. Today, more and more test subjects expect to see serious data protection and data anonymization steps taken, if they are to take part in a study.

The right to data deletion was already in the Directive 95, but GDPR has in many ways underlined it and test subjects have also been educated by the media to expect that they have rights to see what personal data has been collected and, if they wish, delete it. It should be noted that small research projects have it easier – GDPR has certain research exemptions – and a project might consider a deletion request as too resource-consuming to be possible.

Today, companies regularly discuss with lawyers when exchanging data with other companies and when building functionality, where personal data is involved.

Currently, from L3Pilot's vehicle Information and Communications Technology (ICT) research perspective, the most intense discussions are with respect to traffic video and imaging. Exchanging even front view video (footage from public roads) from a test vehicle between organisations requires agreements where the responsibilities of the organisation being a data controller or a data processor must be stated and how data protection and privacy-by-design is implemented.

In research, where the test user consent is given and the purposes for use is clear, unnecessary delays can be avoided. What can help practical work are the consortium agreements between partners that include non-disclosure clauses and ensure a basic level of protection for confidential data. As a future best practice, FOT-Net's Data Sharing Framework recommends considering data sharing details as early as possible in the project phase to avoid delays due to administrative tasks.

Meanwhile, there are countless internet video cameras in Europe capturing traffic footage. Some of them have a purpose for the data capture: traffic situation analysis, weather cameras, and security cameras. However, discussions are ongoing in numerous European cities about adjusting the resolution of official traffic cameras to match the intended purpose: to avoid recording individuals and blur parts of the view (windows from block houses, etc.). Still, the data collection purpose exists, and data collection will likely continue, even though the resolution and blurring can be discussed.

These are examples of new interpretations on how personal data should be handled in vehicle tests. The situation is living, as ADAS products with more and more video collection are being introduced and vehicles start to collect data from crashes.

L3Pilot sees **a need to clarify data sharing regulation** regarding vehicle research. Especially worked-out legal examples on sharing videos in research projects would be welcome. What can make harmonization difficult, however, is that countries may have related national research legislation on top of the GDPR.

L3Pilot used a pseudonymisation process to ensure **anonymity of test subjects**, when sharing data between the pilot leaders, pilot data processing partners and evaluation partners. However, it should still be possible to differentiate between different trips in the consolidated database using IDs. For this to work, the following properties had to be fulfilled:

- Each ID must be anonymous. It should not be possible to figure out who the individual is or which pilot site they belong to.
- Each ID must be unique. No two participants shall share an ID across L3Pilot.

One way of satisfying both properties is to use a hashing algorithm. A hashing algorithm is a one-way process, from which the input cannot be calculated (or rather it would be computationally extremely expensive to do so). Therefore, test subject IDs were hashed using the secure hash algorithm (SHA) with a 256-bit key length. SHA256 is a cryptographic hash function, i.e. it maps data of arbitrary size to a fixed-length string. SHA256 is common (or even state-of-the-art) in many technological applications that require cryptographic protection. It is also widely used for the signing of certificates in the internet. SHA256 offers a certain amount of protection of sensitive information, however, there are some problems with just hashing the IDs:

- Two or more pilot leaders could use a simple numbering system for tracking the IDs, like 1, 2, 3, 4. Since hashing is deterministic, two different participants can end up with the same hashed ID.
- Using trial-and-error, one could go ahead and simply test possible inputs on the data. It is not impossible to find out the hash since some information is likely (e.g. a trip or driver id is likely to be a number, then try out combinations until you find a number). To avoid these problems, the typical approach is to use a “salt”, or a secret word, in conjunction with the ID, which is only known by the pilot leader. The ID and the salt are concatenated to generate the hash.

SHA256 returns a 64-digit hexadecimal ID string. The sheer length of the string can be annoying in practice, especially if it is used in questionnaires and administrative work. Therefore, only the first 8 digits are used for this project. This will not give quite as many combinations, but still enough for the purpose of L3Pilot activities (15^8 ~= 2.5 billion). It is therefore extremely unlikely, that two IDs will be the same, considering that we have a pool of ~1000 people.

Another advantage of SHA256 is, that it is well-known and widespread, i.e. basically all programming languages and even Microsoft Excel supports it. The complete process in L3Pilot is shown in Figure 3.4, using generic data.

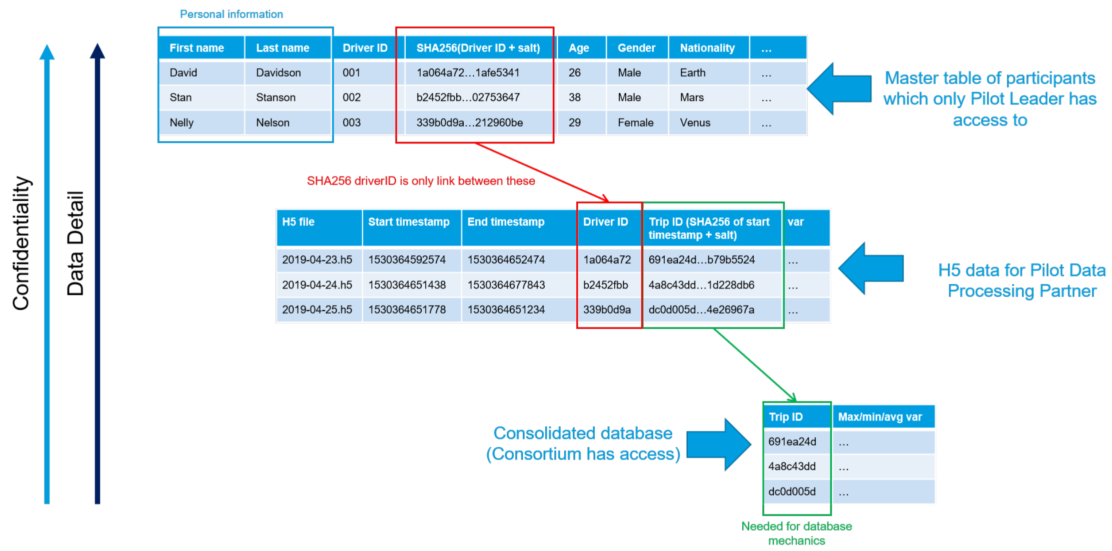

Figure 3.4: Example of the pseudonymisation process for IDs in L3Pilot.

Pseudonymisation does not provide anonymity in the legal sense, as that can only be achieved if all links to the driver can be cut and nobody can link the data to a person anymore.

Pseudonymisation in L3Pilot aims to hide the personal details from data analysts, but the test organisers still have in their safekeeping e.g. consent forms and original driver IDs.

Using the proposed process, data sharing between various partners is possible, but no identification of private information is possible. However, each pilot site can track their data through all databases within the project. This also allows for an easy maintenance of the data during the project.

3.6 Data processing

One tongue-in-cheek definition for big data is that it means a problematic amount of data. Something that is difficult to transport, seemingly takes forever to process etc. To tackle these hurdles, research projects must often focus on reducing data to a set of key variables and indicators.

Indicators can be calculated as computer background processes over several terabytes of data, and if analysts only review the indicators, they might need to handle a few megabytes, only.

In production systems, e.g. in fleet management, data loggers often include processing and filtering features built-in. Not all raw data would be saved or transmitted – but only event and summary data. Sampling frequencies can be set low, to meet only the main requirements for the data collection from a product.

In research projects, however, even minimum data requirements frequently include 5–10 Hz sampling of various signals. Since the systems are not tailored for product use, data collection may lack compression and automated event detection/cutting. Therefore, in many research efforts, the collected data is easily on the verge of “big data”. Later, after indicators and summaries have been calculated, and videos have been cut and compressed, this may no longer be the case.

Research and development therefore **benefit from modern big data tools to store and sort data**. Metadata must be included in files to enable efficient searches and tracking of e.g. software versions and test types.

Figure 3.5 shows an example, which benefits from modern data tools such as Apache Hadoop, and uses NAS (network-attached storage) hard drive facilities. The figure depicts how files from SSD drives are moved to NAS storage. Then, e.g. dropped video frames are repeated in a synchronisation step, after which the data is converted to L3Pilot’s common data format. Additional map-matching is used before reaching the final files. Analysts could therefore use a virtual desktop access to their data lake, to avoid copying data unnecessarily and risking privacy.

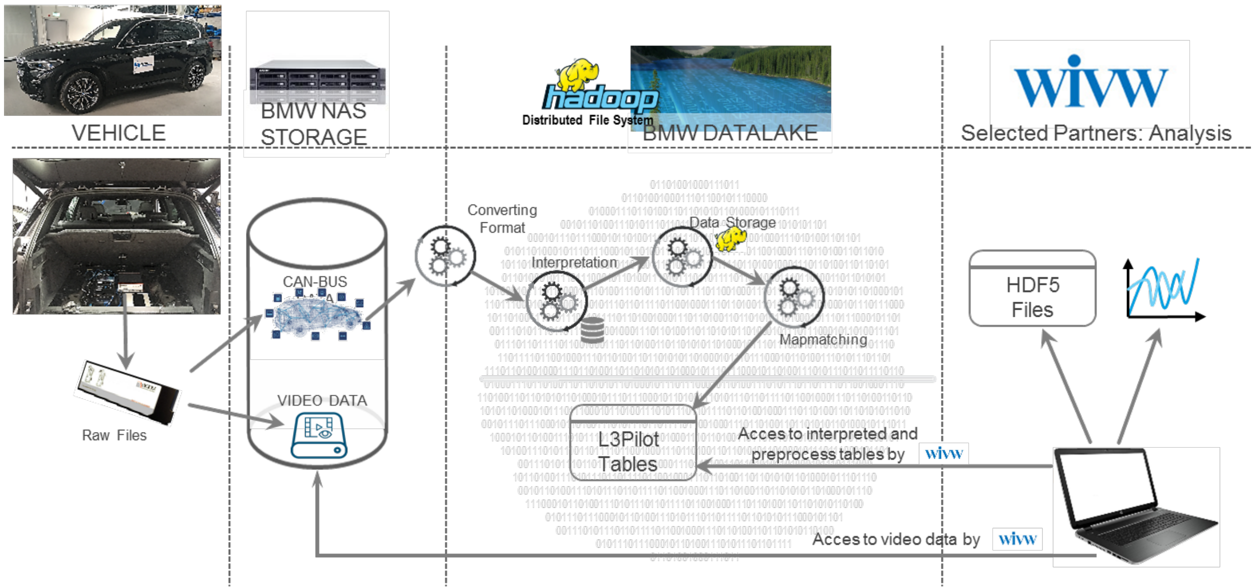

Figure 3.5: Example of Data Management in L3Pilot.

Pilot leaders had to all set up similar systems and data processing, to generate required common data format files. In a different pilot, processing steps after data logging included

- data cleaning
- data fusion
- re-sampling
- verification
- object processing
- image processing using neural networks
- scenario extraction
- conversion to common data format
- storing the data for analysis.

After the processing, data was transferred to an internal cloud storage, where the analysis partners also had access to.

The main efforts in data processing were performed at the pilot sites, converting raw data streams into common data format. This step included data fusion (e.g. combining different data sources from environmental sensing), synchronization (external objects, accelerometers, GPS, and ego vehicle measures) and re-sampling data to 10 Hz. Further, the DMPI framework and aggregation scripts were executed at the pilot data processing partners. Finally, analysis scripts were applied on the consolidated database at the analysis partners.

As mentioned previously, the approach was manually executed by each partner in the chain. There are clear benefits in having a common format, executing scrips having common definitions. Automating this chain would take this approach one step further. In a joint development effort like the DMPI framework in L3Pilot, you can expect many iterations and software versions. Datasets will have to be reprocessed several times.

There are promising approaches in federated computing, where a master node can request data to be re-processed using a new set of scripts. There have been initiatives in e.g. sharing a machine learning training model between partners, rather than the raw data. As a first step a request for processing to a partner could be initiated and then accepted or rejected by the partner. This would imply having a framework for controlling all instances, refined version control. But this would also make it easier for an analyst to request a new derived measure or performance indicator with much shorter lead times. In fact, this could give the researchers more flexibility both in correcting errors, but also taking knowledge of the data into new findings.

Considering upcoming even larger naturalistic tests where test drivers use the cars in their daily lives, it could be that you do not want to transfer all data to back-office. The federated approach could be viable also for the vehicles, forming an edge computing network, where the majority of data is processed within the vehicles. This would solve many of the issues of today (e.g. data transfer and storage, and GDPR). But the risks of doing so should not be underestimated: a small bug in the data collection and processing scripts could jeopardize a part or even the entire data collection. A combination is the most likely approach where parts of the data is excluded, some parts are processed and sent to back office servers even during the trip and some parts are solved in the measurement equipment (e.g. blurring faces within the camera module), and some parts stay the same. It is also likely that high-speed WLAN or 5G could make wireless transfer possible in the coming years.

4 Data Analysis

This report considers data analysis mainly from vehicle data and closely related toolchain programming perspectives. Raw sensor and vehicle data processing tools as well as evaluation tools for scaling up results for societal impact assessment fall outside the scope of this chapter. The tools discussed in this deliverable, however, provide numeric input to societal calculations and simulations.

The main benefits of **data sharing** for analysis purpose are: 1) getting more research results for the resources allocated for collecting data 2) multiple persons being able to scientifically validate results and build belief into the findings, e.g. to acknowledge feasibility of new driver support systems. Certainly, there are other benefits, such as new collaboration options and time/cost savings for not having to gather similar data again and again.

The main barriers that hinder data sharing are [4]:

- privacy and product confidentiality
- quality issues
- poor or missing agreements
- lack of resources or trust.

The latter topics can be tackled in large research projects that can, to some degree, ensure data quality, have assigned resources for data management, and use lawyers to draft agreements. **Product confidentiality** remains as the main hurdle. It can be tackled by considering what type and level of data will be shared.

Technical evaluation deals with detailed performance and at least parts of it are often confidential.

In L3Pilot, pilot data processing partners worked through the raw data with pilot leaders. What was shared within the large consortium (37 partners) were the results from this work: indicator-level data and numeric summaries of detected events/scenarios. The original raw data remains behind bilateral NDA agreements and availability would be agreed case by case. Templates for such third-party agreements about access to the data in some cases now exist as a result from negotiations between the partners. However, L3Pilot has not enforced such legal templates to be ready for further re-use.

Regarding the data that was shared for impact assessment, i.e. indicators and statistical distributions, there was a long discussion on how to prevent from identifying a company given the shared data. Distributions should e.g. tell behaviour on a motorway but not include GPS coordinates, directly identifying the test site.

A level of pseudonymisation was carried out and a list of distributions and indicators selected, but in the end, the intention was not to fully anonymize data (cf. Section 3.4). Via statistical analyses, it might remain possible to identify from which test site certain distributions originate from. The details on what data is shared in what capacity and in which form has to be clearly communicated and noted early on in order to make discussions with partners easier.

The shared distributions and indicators were selected in a way that they should not contain key company IP. They are distributions for analysing project-wide results. Even the distributions themselves won't be published in reports but summaries over all data from e.g. parking manoeuvres. Certainly, test site specific results can be discussed in reports as well. There was no intention to hide but instead promote national results.

This indicator-level sharing approach enables high-level impact assessment, where partners work together under a consortium agreement which includes non-disclosure clauses. The same partners also wish for further collaboration in the same research topics, building trust and common work practises.

As prototypes start to reach a product phase and function-specific regulations become available (e.g. highway automation laws and standards), it could be foreseen that even more company IP-related navigation data would become possible to share. After all, collaboration on legal safety concepts and e.g. in digital maps seems necessary to launch widely used products. The project coordinator also commented noticing partners becoming more positive towards data sharing over the project period, based on revisions of the consortium agreement.

As a future best practise, data sharing details should be tackled even further in EC and consortium agreement templates. However, today, with almost no standard clauses, the task for lawyers and data experts of different companies to define common terms and suitable clauses is a year-long effort. L3Pilot worked this out to a certain degree. It is a matter of opinion, whether the NDA clauses in the consortium agreement suffice for data sharing between pilot leaders and pilot data processing partners. General consortium agreement clauses cannot describe all details nor specific liabilities or data deletion processes.

4.1 Collaborative evaluation toolbox development

From the beginning of L3Pilot, data and evaluation groups discussed how **sharing data and calculation tools** makes up for a better project. Sharing data, at least at the level of indicators and well-defined summaries, enables combining test results from several test sites. Further, sharing calculation tools ensures that exactly the same definitions are used when calculating indicators – otherwise, even "average speed during a trip" might get calculated in many ways, using different scripts at each test site.

The scope of data processing scripts in a project such as L3Pilot has become very large. The number of derived measures, key performance indicators and events to be calculated out of test data amounts to hundreds. If test sites would each face such evaluation and calculation requirements alone, they would just end up carrying out a limited evaluation due to lack of resources.

Centralized data management, where data is collected to a single server or cloud, is suggested by FESTA as a best practice. Within L3Pilot, this suggestion was taken up and the process was modified to fit the need of partners to protect their intellectual property while still enabling analysis partners to evaluate all the recorded data and produce the evaluation results of the project.

Therefore, it was agreed that pilot leaders will team up with pilot data processing partners (Figure 4.1) (cf. also D3.4). The pilot data processing partners were responsible for detailed analyses and producing the higher-level indicators as an input for impact assessment. This approach was chosen in order to have a tight control over who views the detailed raw data, annotates videos containing personal data etc.

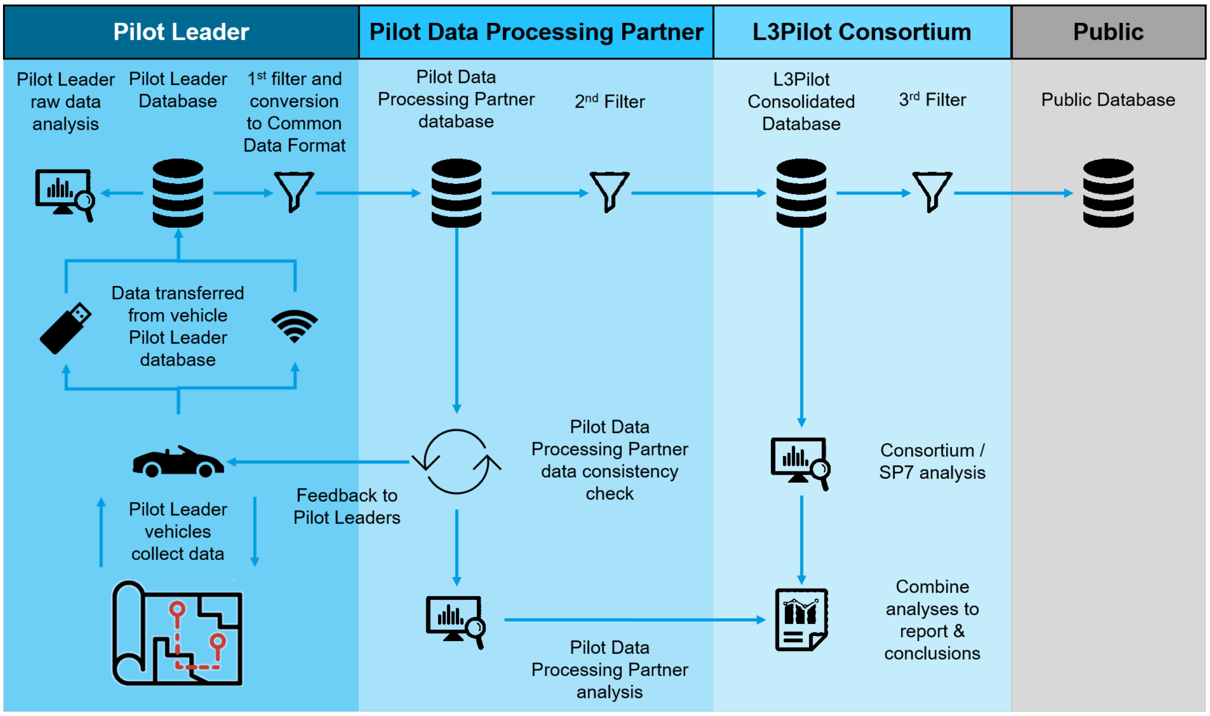

Figure 4.1: L3Pilot data flow.

From evaluation perspective, however, it was mandatory that practically all the calculations would be done with common scripts and code. After all, the log data was to be in the CDF. Minimally, the role of a pilot data processing partner could become to annotate video data, check data quality and run intermediate results, i.e. the indicators and events using common scripts, for higher-level analyses.

Practically the pilot data processing partners had wider responsibilities, e.g. to consult on test procedures, to analyse the quality of each signal, and to verify scenario detection. The sensor setup varied from pilot site to pilot site and the data processing partners had to compensate for variations. Local feedback affected the finalisation of common scripts.

This process has proven to be viable even in a large consortium such as that of L3Pilot with 14 pilot sites. It allowed sharing of data to some extent and the detailed analysis of vehicle data. However, it also means that during the development of the tools and the process, the teams of tool developers, data analysts and pilot leaders have to work together closely in order to put together a toolchain that works with the different set-ups at the various pilot sites. Here, the CDF proved helpful and was a good starting point to developing a common toolchain.

In order to develop such a distributed toolchain with so many partners involved, L3Pilot quickly adopted the concepts from big software development projects, mainly the usage of version controls tools. Among those, *git* is the most widespread tool. It allows an easy versioning of software code and working on code in parallel (temporal and spatial). For the flexible management of all the versions of the code and the other processes involved within the development, L3Pilot relied on the implementation of a git server by Gitlab, that was hosted by an academic partner. This approach allowed the management of most of the aspects of git from a convenient website (cf. Figure 4.2).

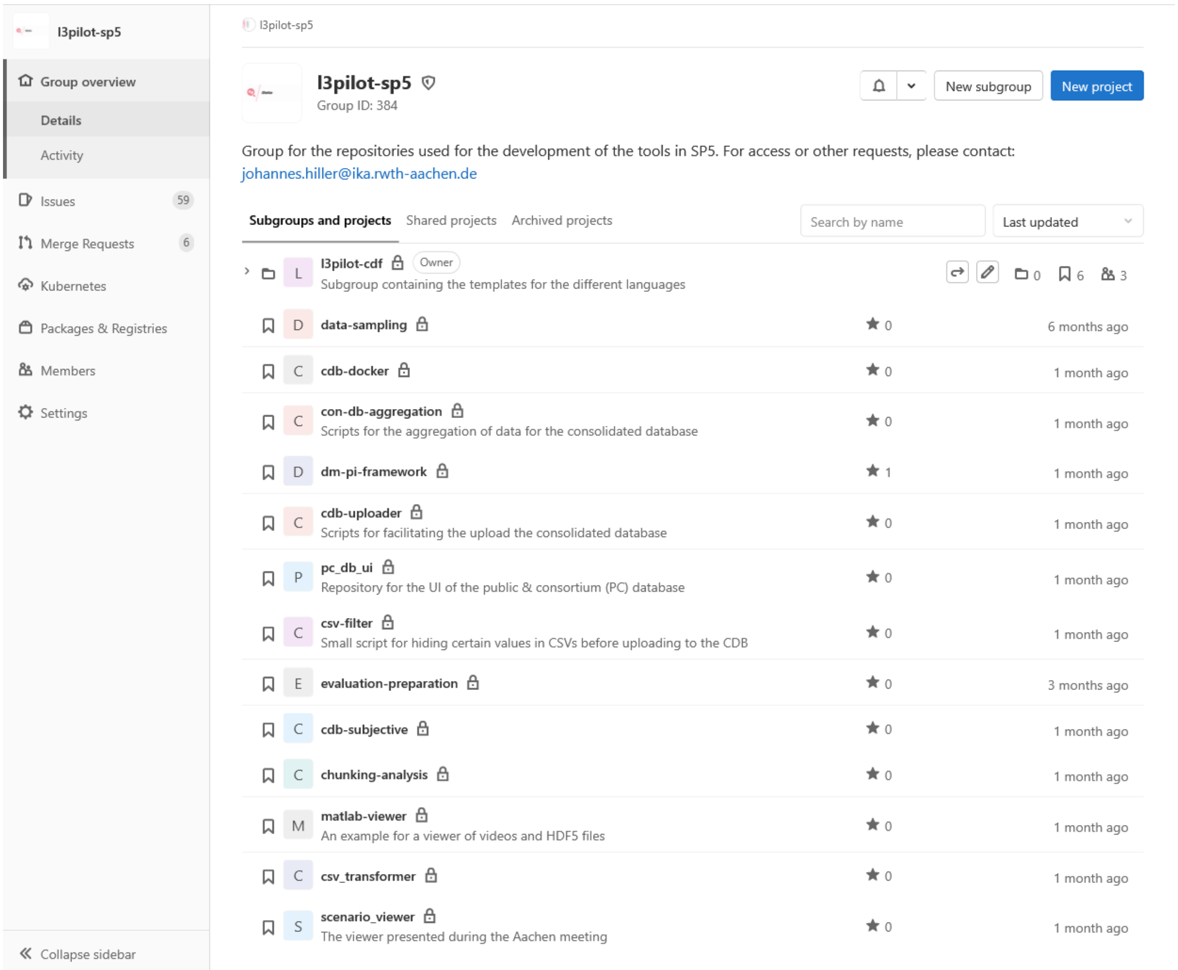

Figure 4.2: The web view of the Gitlab version control system.

Within the group for L3Pilot, multiple software projects could be created, and access could be easily granted to all of them by the administrator. Organizing the various developments into projects also allowed a granular assignment of rights to different developers. This made it clear to all other developers which developer was responsible overall for the development of different parts of the code. Those *maintainers* were also responsible for simple tests of new pieces of code and merging new developments into the main branch of work which was used by the analysts.

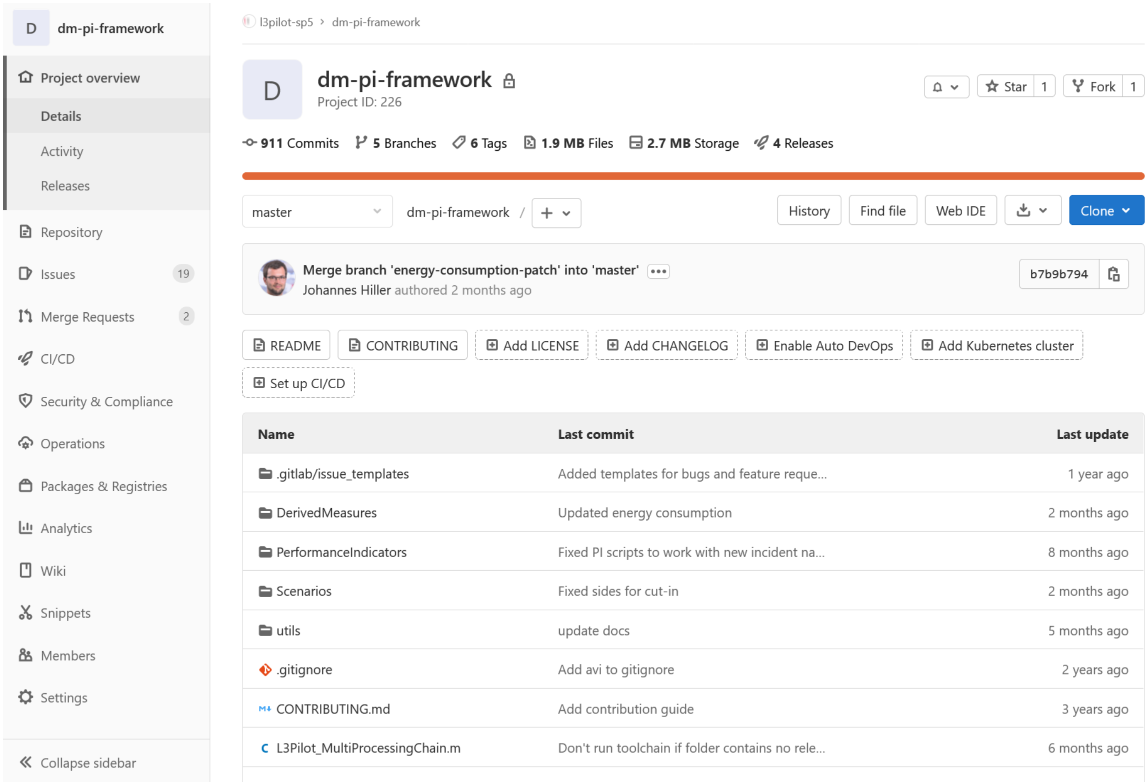

Figure 4.3: Web view of a single development project within the L3Pilot Gitlab group.

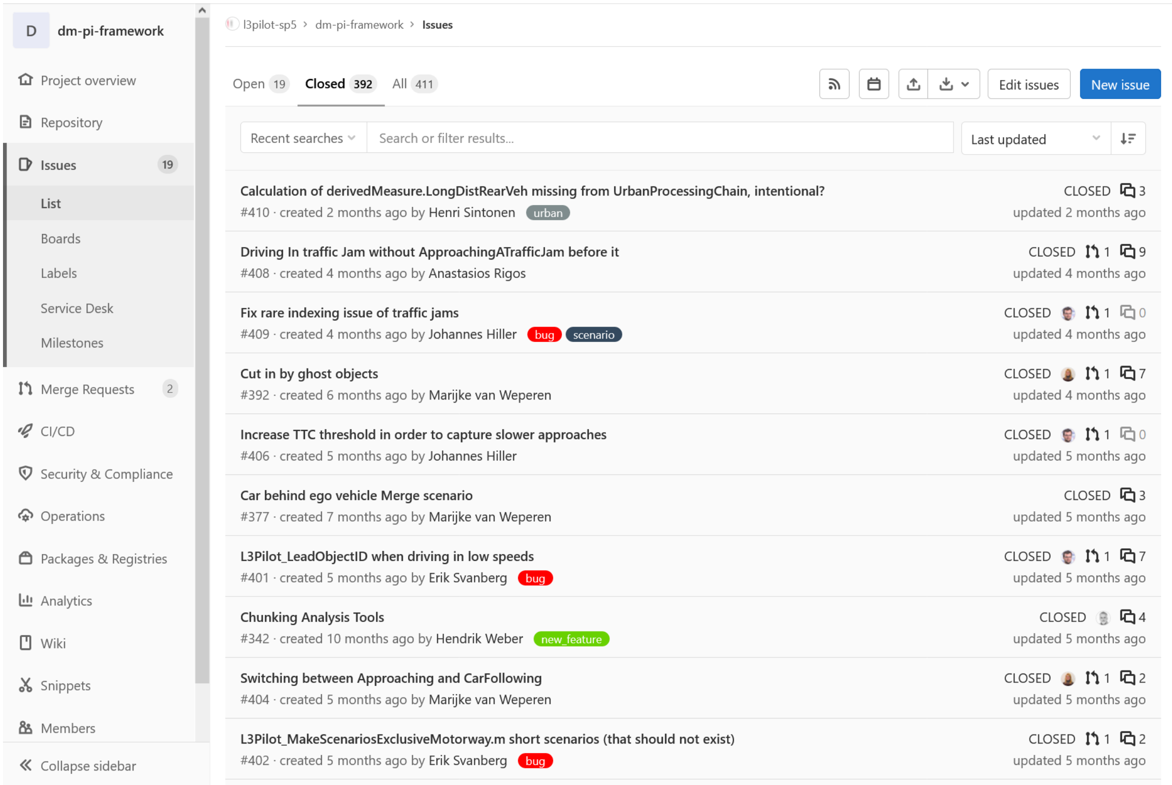

Figure 4.4: Example of some closed issues in one of the developed tools.

These responsibilities were also mirrored in the weekly calls held by the development group in order to organize the work that still needed to be done for the development of the tools. Especially in the early phase this included lively discussions on how to implement certain aspects or features brought up by the team responsible for the evaluation methodology. In a later phase of the project, analysis partners also took part in these calls and used them to report issues they encountered when using the toolchain on their data (cf. Figure 4.4).

Through issues, it was possible to document the discussions and work done on the code in between calls. It also allowed the easy traceability of the rationale behind some implementations. Developers, testers and analysts were encouraged to describe the issues as completely as possible in order to allow developers to easily find the underlying problem. This is of utmost importance for easy working with this kind of a system and with slightly varying data at the different pilot sites. For future projects, a clear and expressive description of issues within the development tools should be enforced (cf. Figure 4.5). Not doing this leads to much time spend on gathering information on the underlying problem and less time with the actual solving of the issue.

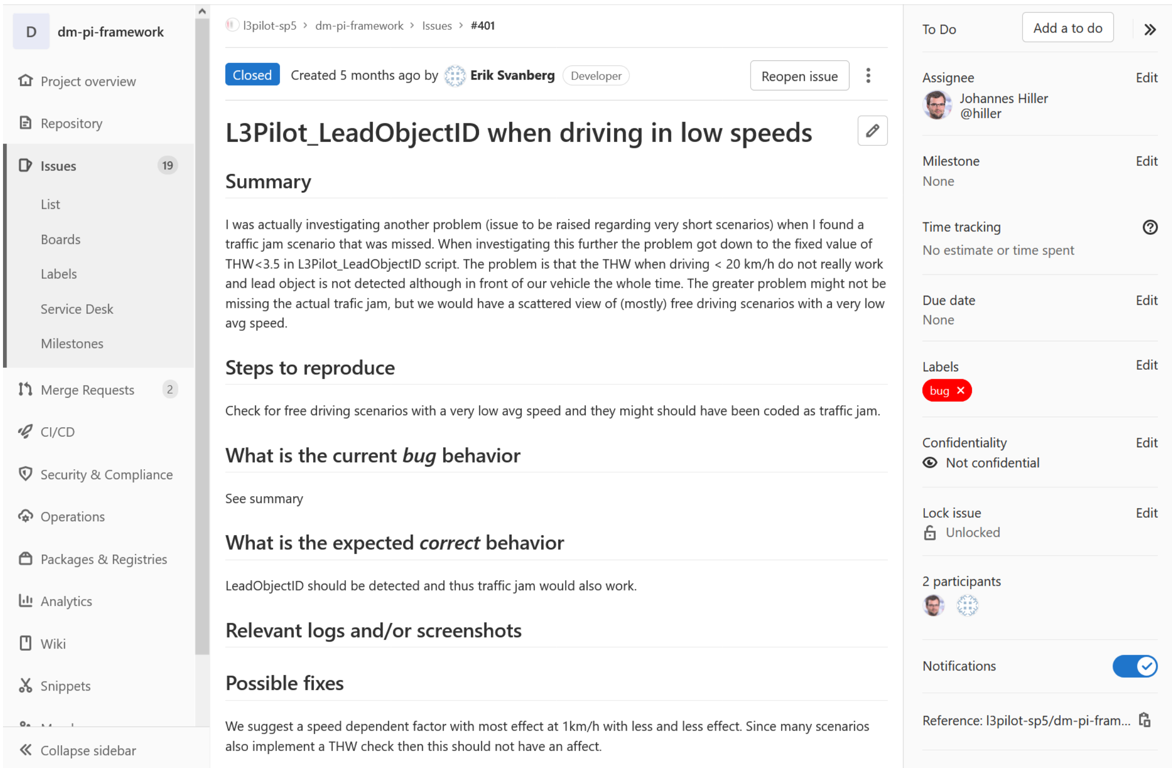

Figure 4.5: Detail on how issues were described.

Although this process was well conceived among the developers and established in numerous software projects around the world, it was not yet fully validated among the users of the toolchain, in this case the evaluation experts. For future projects it is therefore of utmost importance to improve the literacy of analysts in the basic software development tools such as git (or any other versioning tool). Similarly, experience should be gained in the management of the website, in order to enhance the development of the tools even further.

4.2 Data viewers and video annotation

For evaluation, it is a very common requirement to **be able to view video and vehicle data easily**. The purposes vary depending on the partner (e.g. pilot leader or a data processing partner), but for example they include:

- Verification of sensing algorithms
- Classification of situations from human perspective; video annotation
- Analysis of behaviour of either the ego vehicle or other road users.

In L3Pilot, a **video annotation tool** was used (from now on, MATLAB-viewer) for partners to annotate and confirm detected scenarios from video. The tool enabled to move quickly within the data to a detected scenario and confirm & annotate it. The tool required DMPI-Framework calculations to be executed before viewing a file.

Additionally, a few pilots used their own data viewers. One experiment first processed recorded data and stored detected driving situations in a database, so that different cases could be searched and viewed easily. They considered minimum searchable parameters to include vehicle number, AD revision, date, location, weather, AD system status, speed and speed limit. It showed that a combined use of the tools developed for the analysts and those available at the pilot site proved helpful for a first assessment of the recorded data. The converted data was viewed using the MATLAB-viewer, while possible issues were investigated in-depth using proprietary tools.

One additionally developed viewer (Figure 4.6) offered both a bird's eye view into situations and a 360° panoramic view. The panoramic view was useful especially when pausing to analyse a situation – but in easy cases, a front view and the bird's eye view over surrounding objects were enough to assess the driving situation. For scenario inspection, the birds' eye view was proved extremely useful since all objects are depicted with their assigned ids (lead vehicle is depicted in different colour) and within their corresponding road lanes (see Figure 4.7). On the other hand, manual video annotation using the video annotation tool was very difficult to be used for big datasets due to time implications and hence the automatic scenario detection offered by the project data processing scripts proved valuable.

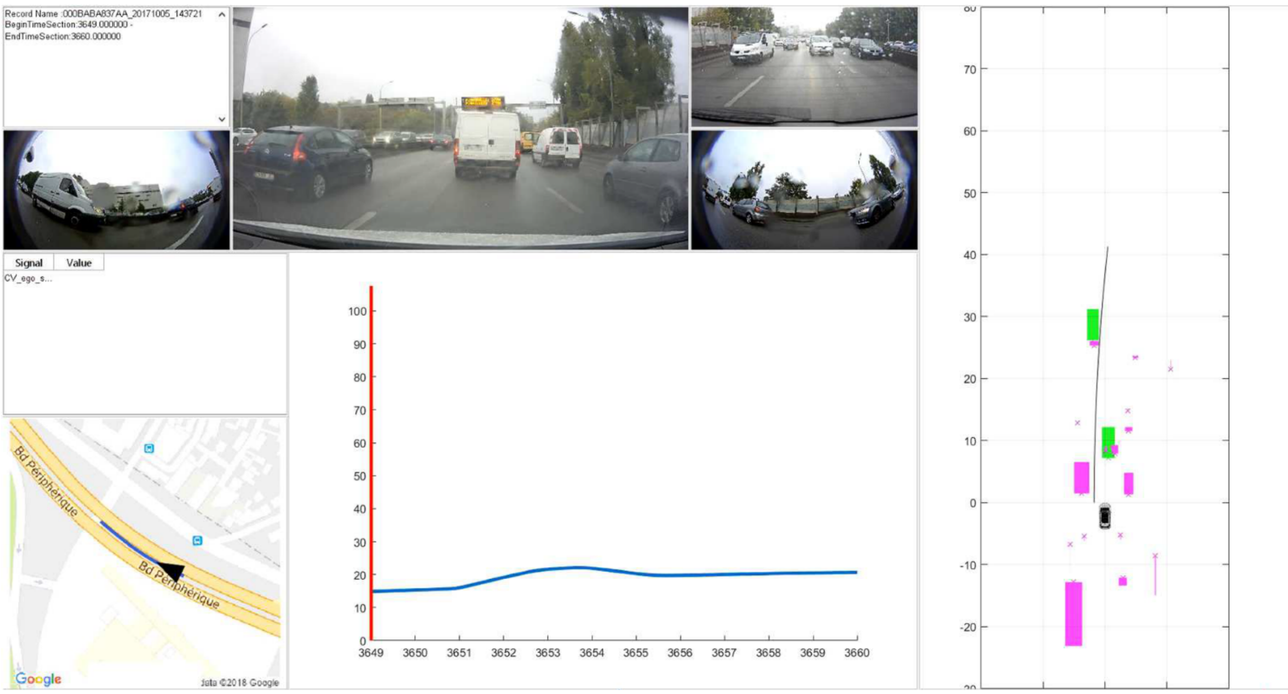

Figure 4.6: Data viewer example.

Another partner implemented a viewer based on the project's common viewer (cf. Figure 4.7) that allowed the review of the scenario labels. This proved helpful in assessing the overall picture in comparison to just single instances.

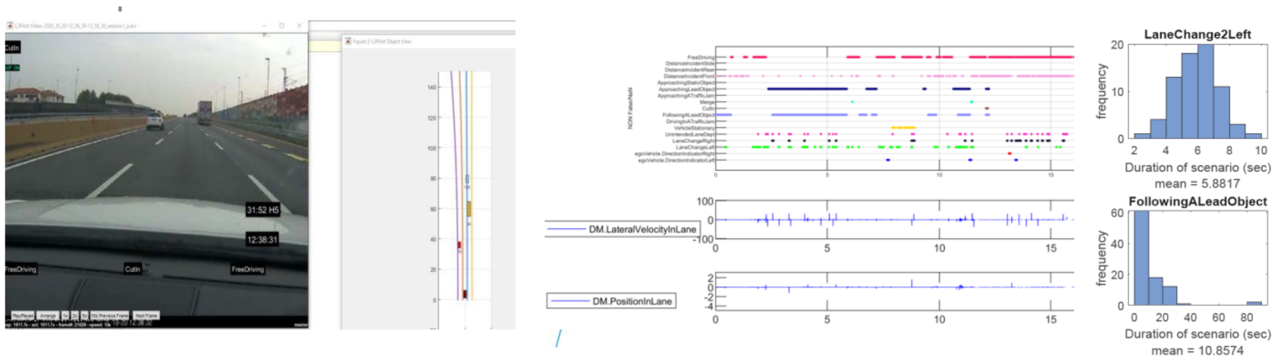

Figure 4.7: Data viewer example incorporating the MATLAB-viewer.

4.3 Data Analysis Toolchain

In L3Pilot, a toolchain was developed using the methods and the code versioning described in the previous sections. As programming language and execution environment, MATLAB was chosen, as most relevant partners were familiar with this language and even had related previous code available. MATLAB is not always the first choice in some environments and the issue of licencing exists. Other languages such as Python offer similar features minus the issues with licences. Therefore, the choice of MATLAB for tool development should be critically revisited in future projects. In the end, it is always a balance between usability and pre-exiting knowledge and the easy transfer to other environments such as computing clusters.

The **toolchain** developed in the project is shown in Figure 4.8.

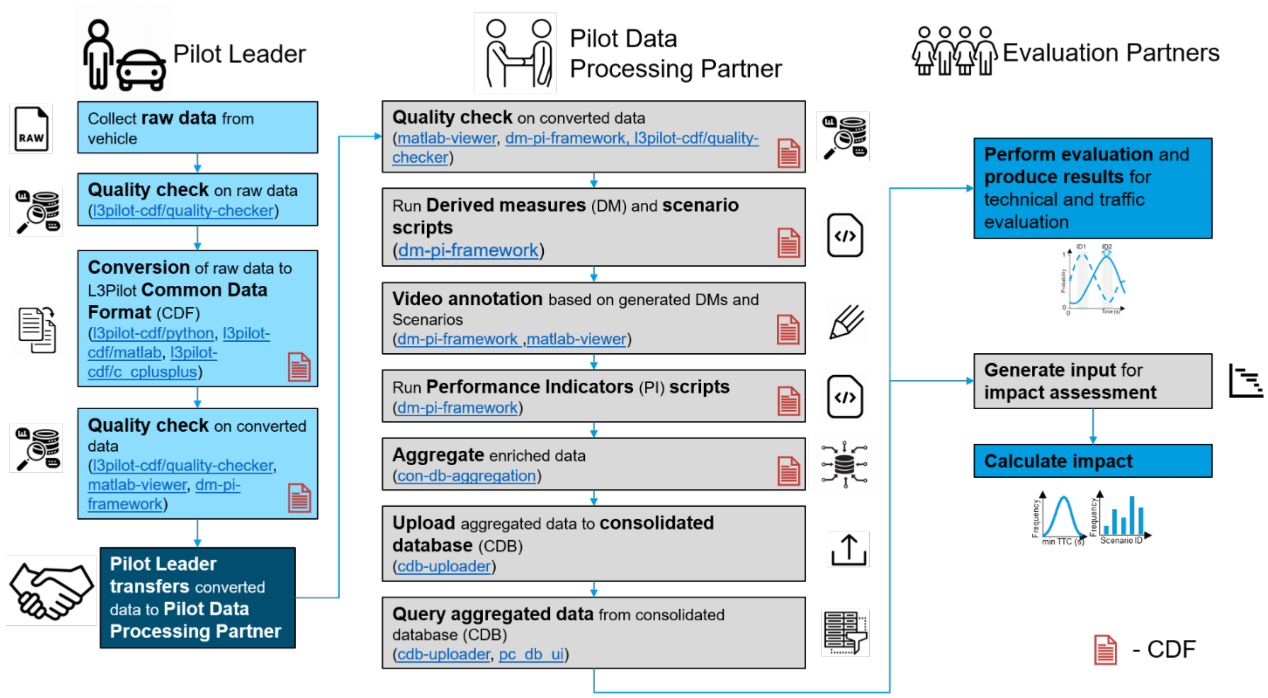

Figure 4.8: Data analysis toolchain developed within L3Pilot. Scripts are noted in brackets behind the individual steps.

During the project, it became clear that the process as described in Figure 4.8 was overly idealistic and did not account for the iterations needed during the process of converting the data and assuring data quality (cf. Section 3.4 for details on quality assurance) as well as the changes that were needed in the toolchain to better fit the data (cf. Section 4.1 for details on the process).

During the application of the complete toolchain from log files to a consolidated database, it became clear, that sharing the toolchain has its advantages in providing every partner the same calculations. However, it also became clear, that small differences in the vehicle data can lead to big differences in processing and with the results.

As an example, a small bug in one of the scenario detection scripts was detected late in the process. While the issue in the scenario detection script was easy to fix, rolling out the fix to all data was not as easy. As a first step, all pilot data processing partners had to ensure, that the fix worked correctly on their data. They then had to re-process all the data and go through the process of re-uploading the data to the consolidated database from where data analysts had to export it again. This example shows all the advantages and disadvantages of the chosen approach. While it was easy to fix the issue in the toolchain and make it available to all partners, the small differences in data made the application of the fixed toolchain to the data a long process.

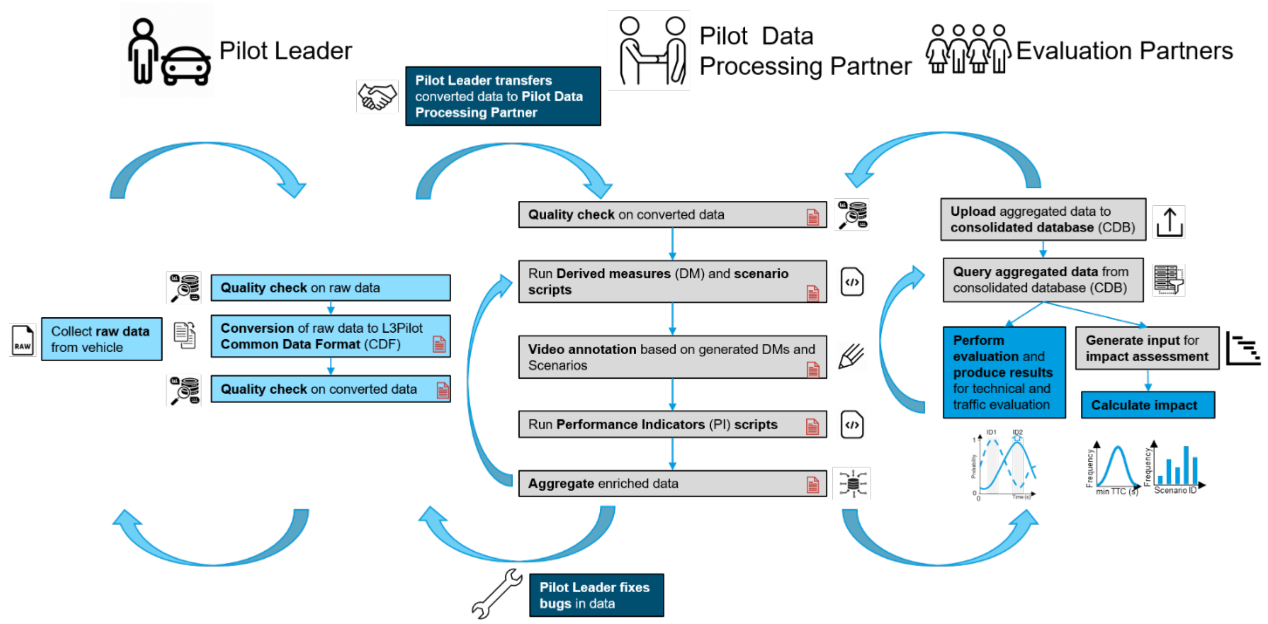

Figure 4.9: Realistic representation of the data process, including iterations.

While fixing of bugs in tools and reprocessing of data have always been present in similar projects, often on short notice at late stages, they have not been so obvious at the beginning of our project, as each analysis partner worked separately to deliver the results. Thus, in order to better utilize the advantages of the common toolchain, a more rigorous control of data quality and algorithms has to take place, including a more complete automation of different re-processing steps.

One suggestion here is to have a small amount of data available to all developers that is fully annotated manually with all necessary information and scenarios, so that they can work with this case for the development of the tools. In this way, they don't have to blindly implement tools while the datasets are still being collected on the road. This would also allow the application of automated (unit) tests on data samples to check for the conformity of the toolchain after every change. This method has already proven itself to be effective within many other fields.

Another suggestion is to have data quality guidelines ready as early as possible within the project for partners to work with. This way, both the pilot leaders and data processing partners can check the conformity of data and tools. As a basic approach, this could be a simple list with defined ranges of signals that need to be fulfilled. In a more advanced version, this would be a tool that highlights outliers of the ranges provided. A first version of such a tool was provided and well received among the pilot sites. Alternatively, this could also be implemented in the scripts which are responsible for the upload to the Consolidated Database (CDB). Unrealistic values would be excluded by default and would need manual confirmation by a pilot data processing partner before being added to the CDB.

Finally, aligning the timeline between tool developers, data providers and analysts could also positively contribute. Since most projects are already quite restricted in terms of time, having the scripts ready before collecting data seems utopian. But a more focused communication of requirements to data providers and analysts could go a long way in solving key elements of these issues.

Further issues arise, when data is collected on roads or in situations that are not covered by the toolchain. For example, this may happen when a script expects motorway driving but the recording is from a rural road, and there is no related check. Such cases can lead to strange issues in the toolchain and in the detection scripts.

One example is the existence of a tunnel at one pilot site. It had to be manually excluded, since this situation was unique, and would lead to a possible identification of the site in the results. One solution would be to address this aspect from the start of the development of the toolchain and include it in the processing steps.

Another example concerns changes in the routes that were selected for the experiments. Such changes can be due for example to new traffic rules or temporary construction sites. Modifying the itinerary can lead to quite different results. In this case, the toolchain might need to differentiate between data containing the construction site and data not containing the construction site. This would also include adding a version indicator as well as some kind of versioning system to the common data format and the data in general. A future toolchain should include these improvements.

5 Conclusions

This report brings together the lessons learned from L3Pilot with respect to data logging, data management and data analysis. The key findings are summarized here reflecting the various steps of the work in sequence.

First, the project established a process to **finalise data collection requirements.** An initial set of research questions by evaluation partners was translated into data needs and a list of vehicle signals to be acquired. Pilot sites then commented on their possibilities and difficulties to eventually collect these signals, and also on the testing priorities from their perspectives. Alternatives to collecting required information were discussed: a typical example regards the case when a certain vehicle signal was not easily available, but another source of information or possibly another evaluation method was applicable to approach the same research topic.

After an iterative discussion among evaluation experts and system developers, L3Pilot fixed its data collection requirements. The collected data was expected to enable evaluation of automated driving from several viewpoints, for example, safety, efficiency and environmental impacts.

As the evaluation goals involved the integration of data coming from several pilot sites to boost the analysis about impacts across Europe, the project had to develop **harmonized steps for data acquisition and processing**. A number of procedures involved difficulties due to the data collection requirements, in particular listing numerous vehicle signals and exact content, as well as adapting indicator calculation and other post-processing to various raw datasets coming from pilot sites. Most sites had existing data logging implementations with proprietary formats. Raw datasets were also considered confidential, at least for parts.

To enable an effective collaboration, easier sharing of data among partners, and development of common tools for data processing and analysis, L3Pilot decided to go for **a common log file format**. The format was named Common Data Format (CDF) and it was published as open source. While the pilot sites still logged their proprietary formats for development purposes, the CDF targets evaluation and provides a more condensed format. The CDF stores e.g. object classifications and locations instead of raw environmental sensor data.

The next issue to be tackled was **the implementation of tools able to operate on large development-related datasets** besides the evaluation data. These ranged from using SSD hard drives in RAID configurations in the vehicles enabling high-enough write speeds, to managing datasets in server farms and data lakes, where data was converted and viewed.

As the data collection included numerous sensors, partners had to focus on **time synchronisation,** so that each signal could be accurately time-stamped. Generally, the lesson was to timestamp data close to the source – and possibly again after processing/transmission steps. For handling multiple cameras at the same time, partners ended up using separate video processing equipment to combine camera feeds.

Partners noted that they would benefit from further development of data logging and test monitoring tools, when progressing from these pilots towards longer-lasting naturalistic drive tests.

Some partners already had traffic light type of displays for monitoring system status, to quickly see if every system is on green. Others sent a few key values from the vehicle to an internet server, to monitor test progress and success online. However, it seems appropriate to extend the list of indicators and events computed already in the vehicle, and sent to a test monitoring station.

Regarding data management and analysis, two main lessons learned can be highlighted. The first one is to target **as much harmonization as possible among the different test sites**, in term of formats, procedures and tools. The second aspect regards taking a great **care for data quality during all the steps** in the data flow. In this context, the project built up a toolchain for harmonized processing of the logged data to be used by all partners. Moreover, the team involved in data management provided a series of supporting means. These include (among others) examples for data conversion, a data viewer and a video annotation tool, scripts for computing derived measures, numerous driving scenario detection scripts, computation of key performance indicators, and analysis of data distributions.

With many partners involved, harmonization was not only about data: **a software development process** had to be set up including the use of collaboration tools such as version control. In L3Pilot, git was used in conjunction with Gitlab to enable versioning of software code and programming in parallel (temporal and spatial). The development group held weekly calls in order to organize the work that still needed to be done for the development of the tools especially in the early phase of the development. This also included discussions on how to implement certain aspects or features brought up by the team responsible for the methodology. In a later phase of the project and the development, analysis partners also took part in these calls and used them to report issues they encountered when using the toolchain.

An advice for future projects is versioning the data too (besides the tools themselves): the date of post-processing and the software version used should be tracked. Because of updates on the data logging setup in the vehicles or updates in the scripts for data analysis, there was the occasional need to re-run data through the toolchain. Versioning the datasets would have led to getting a better overview on the data that has been analysed – partners already had some tracking systems. A further way of harmonization could be to implement a process that allows to trigger the execution of the post-processing toolchain from a central place, even if the data is stored decentralised at different locations. By enabling this kind of central triggering, re-processing would not be the responsibility of each individual pilot leader anymore, thus guaranteeing that all datasets have been calculated with the same version of the evaluation scripts.

While previous FOTs have often commented about long times to reprocess data, L3Pilot did not experience this as an issue because of the capabilities of scalable cloud services and the rather short logging duration.

Regarding **data quality**, pilot sites have been advised on how to achieve high quality by providing a comprehensive checklist and even a tool to automatically check the quality of the data. A first approach for guaranteeing a good quality is to frequently monitor the data collection process so that as little as possible testing would have to be repeated due to bad or missing data. A second approach is to ensure that the data is usable for evaluation purposes and selected key performance indicators can be calculated.

In L3Pilot, at first, quality was the responsibility of the pilot leaders, ensuring correct logging in the sensors installed in the vehicles. One lesson learned was to have a “reference trip”, testing the vehicle functions and manually coding the time when doing it. The next step was ensuring correct format and full content of sample log files: check that they were according to the CDF specifications and fulfilled the minimum requirements for logged signals. For this purpose, L3Pilot offered a Java-based tool, L3Q, to check the basic contents of log files and output web reports per log file. Further, the pilot leaders (with the support of pilot data processing partners) analysed the collected data in terms of semantic checks e.g. analysing if hard braking could be detected.

As a whole, the data quality process is quite challenging. The data quality checklist gave the partners a lead on what to do to ensure high quality data. The project could though have defined more tools than the L3Q. A further set of semantic checks could have been automated, giving warnings or even preventing elements of data suspected to be incorrect. This could include short or long scenarios (based on the scenario definition), extreme values, but also combination of scenarios and measures.

Besides **protection of company IP**, the project saw that also **GDPR caused concerns on what data could be collected and shared.** Most initial concerns were not warranted, as GDPR is not to hinder research and has related research exemptions. Raw vehicle data was shared only under confidentiality agreements between a pilot leader and pilot data analysis partner. Those who had access to the raw data, used the toolchain to process a set of common indicators and other statistics. To ensure the anonymity of test subjects and making even test sites difficult to separate, test subject IDs and vehicle IDs were encrypted using a hashing algorithm. Videos revealing test subjects’ identity had been already annotated and processed into numeric state values. The resulting statistics for evaluation were shared between consortium partners.

Nevertheless, the legal situation regarding sharing video even between consortium partners seemed to be a moving target with various views. Therefore, research projects would welcome EU-level practices and legal templates for sharing e.g. the front view video from a car within a research consortium, to speed up work and to avoid lengthy legal preparations. The current best practice is to add video data sharing clauses to the consortium agreement, to reduce the need for bilateral agreements. Video sharing also needs to be agreed with test subjects in their consent forms.

Overall, executing pilots proved to be a crucial undertaking towards the deployment of automated driving in series cars, and data handling was among the major aspects to be considered. Besides evaluating SAE Level 3 and 4 functions, L3Pilot made considerable efforts in this direction, with a view on making available the gained experiences, focusing on user needs, and creating a Europe-wide piloting environment.

References

[1] Version 7 of the FESTA Handbook on FOT Methodology. December 2018. http://wiki.fot-net.eu/index.php/FESTA_Handbook (26.6.2019)

[2] Data Sharing Framework. 2019. FOT-Net and CARTRE support actions. https://knowledge-base.connectedautomateddriving.eu/wp-content/uploads/2020/04/Data-Sharing-Framework-v1.1-final.pdf (27.5.2021)

[3] The L3Pilot Common Data Format – Enabling Efficient Automated Driving Data Analysis. 2019. J. Hiller, et al. The 26th International Technical Conference and exhibition on the Enhanced Safety of Vehicles (ESV), the Netherlands. http://indexsmart.mirasmart.com/26esv/PDFfiles/26ESV-000043.pdf (26.6.2019)

[4] D1.2 Final FOT-Net Data Report. 2017. S. Koskinen, et al. http://fot-net.eu/wp-content/uploads/sites/7/2017/05/D1.2-Final-FOT-Net-Data-Report-v1.pdf (26.6.2019)

[5] L3Pilot Deliverable D4.1 Description and Taxonomy of Automated Driving Functions. April 2019. T. Griffon et al. https://l3pilot.eu/fileadmin/user_upload/Downloads/Deliverables/Update_15042021/L3Pilot-SP4-D4.1-Description_and_taxonomy_of_AD_functions-v2.0_for_website.pdf (28.05.2021)

[6] L3Pilot Deliverable D3.3. Evaluation methods. September 2019. B. Metz et al. https://l3pilot.eu/fileadmin/user_upload/Downloads/Deliverables/Update_28042021/L3Pilot-SP3-D3.3_Evaluation_Methods-v1.0_for_website.pdf (28.05.2021)

[7] L3Pilot Deliverable D6.5. Pilot Reporting Outcomes. L. Andreone et al.
Deliverable in progress. Downloadable later at https://l3pilot.eu/downloads

[8] L3Pilot Deliverable D7.3. Pilot Evaluation Results. H. Weber et al.
Deliverable in progress. Downloadable later at https://l3pilot.eu/downloads

List of abbreviations and acronyms

Abbreviation	Meaning
AD	Automated Driving
ADF	Automated Driving Function
AV	Automated Vehicle
CAD	Connected and Automated Driving
CDB	Consolidated Database
CDF	Common Data Format
DM	Derived Measure
DMPI	Derived Measures and Performance Indicators
FOT	Field Operational Test
GDPR	General Data Protection Regulation
GPS	The Global Positioning System
HDF5	Hierarchical Data Format version 5
ICT	Information and Communications Technology
IP	Intellectual Property
KPI	Key Performance Indicators
NAS	Network-Attached Storage
NDA	Non-Disclosure Agreement
NDS	Naturalistic Driving Study
PI	Performance Indicator
RAID	Redundant Array of Independent Disks
SDV	Software Defined Vehicles

Annex 1 Lessons learned from former projects

The current document presents a collection of lessons learned which have been published in other projects. The following projects have been checked for lessons learned: euroFOT, FOT-Net 1, FOT-Net 2, and FOT-Net Data, DRIVE C2X, TeleFOT, interactive, AdaptIVe, eCoMove, FOTsis, and UDRIVE.

Lessons learned from euroFOT

The following lessons learned have been extracted from the euroFOT project (Selpi et al., 2011).

Positive working methods

- Using common templates for gathering requirements.
- Having specialized/focused groups to enable joint efforts in performing specific tasks.
- Separating the handling of incoming disks/data from pre-processing stage.
- Using modular approach for developing the base data analysis tool.
- Using one data format throughout the pre-processing, uploading and analysis (based on MATLAB structures), once the raw data had been read.

Common DAS-hardware issues

- Broken compact flash and SD cards.
- Failing connectors and loose cameras.

Recommendations

- Use industrial-grade compact flash cards and SD cards.
- Give extra attention when installing connectors and cameras.
- Install inbuilt diagnostics with remote status reports to follow up the status of the DAS (data acquisition system).
- Be open to the potential benefits of using database features like partitioning and compression to enhance the standard relational database, using proper data warehousing, or even to other technologies (e.g. column-oriented databases, Hadoop database (HBase) or data warehouse for Hadoop (Hive)) for the management and analysis of FOT data of the scale of this project or larger projects.
- Perform data quality checks on sample and signal basis directly after data conversion (e.g. check for missing data, values out of range, wrong dynamic behaviour of data, incoherent relationships among data).
- Test the whole chain of data pre-processing, upload and analysis (if possible) until PI-calculation before the FOT.

- Try to fully use the power of state-of-the-art servers utilizing many cores. By using parallel techniques in the database server or splitting the queries into smaller pieces can boost the performance.
- Storing large amounts of data in a traditional relational database should be questioned for future projects of this scale or bigger.
- The data pre-processing and upload procedures should be flexible and allow changes after the FOT has started. This is necessary because new problems might occur that did not exist in the pilot.
- A “sandbox” database including a subset of the data should be created. When developing and refining scripts it is much more time efficient to test on a smaller dataset. When satisfied the analysis can be executed on the whole dataset.
- Consider the cost for making backups of the database. If raw or pre-processed data is stored and has backup there might not be need for taking backups of the continuous data tables.
- It is very important to develop and improve a web-based survey tool for data collection in order to save time and prevent data transcription mistakes.
- In order to reach more drivers and improve response rate, it is very good to duplicate or offer multiple options for filling questionnaires in (i.e. hard copy and electronic copy)

Lessons learned from FOT-Net

The FOT-Net initiatives (FOT-Net , FOT-Net2, and FOT-Net Data) published several documents how a European platform for enabling data sharing and data reuse could be developed. Even if the documents do not all contain explicitly a section “lessons learned” the contents can be seen as best practice regarding data sharing. Barnard et al. (2016) and Barnard, Y., Koskinen, S. and Gellerman, H. (2014) highlight the following information:

Collection and Managing Data in FOTs

FOTs produce large amounts of complex data and the main challenge is to make the data manageable and easily available for analysis. As each FOT has its own type of data and data management demands, a generic model that is suitable for all FOTs does not exist. Often the data recorders lay the foundations of the FOT dataset by grouping data into segments, e.g. by trips or events. By establishing these segments it is possible to attach rich metadata that can help flter the dataset.

Documentation is a key requirement. The data need to be described in detail, but also the tools and processes implemented during the FOT. To understand the dataset the study design (the objectives of data collection) is important information, as well as the detailed test protocols with relevant test scenarios. It is also important to describe the structure of the dataset, how it is organized and stored, to facilitate use of the data.

Data quality assurance is another key issue. Procedures ensuring the quality have to be applied from the very beginning of the planning and piloting of the data collection, and the subsequent storage and processing.

Data Sharing

In order to re-use data in a new project, the original project must be willing to share the dataset, and make it available in a way ensuring that it can be re-used. Sharing poses challenges, which may be of different nature:

- Technical challenges concern issues such as the quality of metadata, descriptions on implementations, how field tests were run, how the data was collected, the tools used to collect and store the data, the standards and formats used.
- Legal and organisational issues concern ownership, data protection and privacy issues. For example, a permission from the FOT participants is needed to allow third parties access the data. Re-engineering tested services or used sensor systems is also seen as a worry.
- Practical and financial issues concern questions about who is paying for the access, the training of new data analysts so they can understand the data, its limitations, tools, and physical access to the data.

The proposed Data Sharing Framework consists of seven different areas, which should all be addressed in order to provide or re-use data:

1. Agreements within the project collecting data, including consortium agreements, participant agreements and agreements with third party data providers
2. Availability of valid data and metadata, including a "standard" description of the documentation of the data
3. Data protection requirements both for the data provider and the analysis site
4. Security and personal integrity education for all personnel involved
5. Support and research services to facilitate the start-up of projects and offer research capabilities
6. Financial models to provide funding for the data to be maintained and available and for access provision personnel to be available
7. Application procedures and data sharing agreements

Data Sharing Platform and Strategy

A data sharing platform consists of sharing principles and data access procedures, data descriptions (metadata) and standards, and descriptions of technical tools and methods for data sharing. Such a platform is to be used in two major phases. The first concerns the steps to be taken and issues to be resolved from the very beginning of a FOT, such as defining participant agreements for allowing data re-use and resolving data ownership issues. The second is to address the actual data sharing and the procedures, templates and services needed for successful research on data gathered in earlier projects

Principles and procedures

The main principles and procedures in data sharing concern procedures for application and approval, and templates for application, data re-use in consent forms, data sharing agreements and data sharing texts for consortium agreements. In addition, support and research services, financial models, and personnel education on legal and ethical issues need to be developed.

A data-sharing platform needs to consider the views of different kinds of stakeholder, serving research, commercial or policy purposes. Data protection, privacy and security are crucial prerequisites to data sharing. Different types of levels of protection are required depending on the sensitivity of the data. Guidelines are needed for each level for how data could be practically accessed and shared.

Data descriptions and tools

The descriptions include details about the test execution and collected log data. The data descriptions cover the data collected in the tests, enhancing data such as map data and the equally important metadata, that helps understand the background to the collected data. Special attention should be paid to information about video annotations, the subjects and vehicles collecting the data, and the experimental protocol used, and study design applied.

Another issue is the harmonization of data of the same type across datasets. Ideally a minimum list of data types with minimum requirements for a harmonized dataset should be set-up, which all future FOTs are recommended to collect. In this way the research community would be able use a common set of data for a variety of purposes.

Tools and methods

The technical issues concern tools and methods for data access, processing, enrichment, and analysis. Tools that can be used to support data sharing as well as data post-processing methods and data aggregation types are of interest. Data processing tools and scripts are generally very helpful for an analyst when made available together with FOT data, allowing reading the original raw data, interpret it, enhance it and calculate indicators.

In the FOT-Net project a Tool Catalogue was developed and made available at the FOT-Net wiki (http://wiki.fot-net.eu/index.php/Tool_Catalogue).

Lessons learned from DRIVE C2X

The following lessons learned (regarding “data”) are extracted from the DRIVE C2X project (Malone, K. et al., 2014, section 5.3 Lessons learned and recommendations for future FOTs).

Measures and sensors

Use one type of logger, harmonize measures

DRIVE C2X developed a logger to record driver behaviour parameters. However, there were also other types of data loggers depending on test site. This led to time and resource demanding work of data conversion and additional data quality checking later in the project which caused time pressure for impact assessment. Therefore, it is recommended that only one type of data logger would be used in one project. Furthermore, the development of data loggers and measurements e.g. which performance indicators the measurements will cover needs to be decided commonly with good understanding of consequences of the decisions to what can be analysed and concluded at the end of project. All measures need to be harmonized as much as possible. More emphasis should be put on harmonization of questionnaire translation.

Data Acquisition

Pilot functions, pilot data processing, pilot data quality checks, pilot data analyses

Many activities in data acquisition are taken care of by test sites. Piloting of the tests is critical for the following phases of the project. To pilot system and data collection and processing takes more time than expected. Piloting should cover not only the technical functioning of the systems but also provide data to test the whole data processing chain and data analyses.

Data comes from several sources. These sources need to be checked for synchronisation, quality, consistency etc., before they are combined. Some work can be automated, while others must be done manually. If this is not conducted well, the following phases are suffering both in terms of quality and time.

Data quality addresses quality and completeness of the datasets.

The piloting of data analyses identifies mismatches in data inputs and outputs, units, structures and the like in an earlier and easier to modify stage.

Data base and data analyses

Engage competent data processors

Data processing in a FOT is time demanding and the time and expertise needed for it should not be underestimated. Competent people and resources are needed to handle the task. Preparation of the data for the final analyses requires an iterative process with requests from data processing to the tests sites, from impact analyses to and from data processing etc. This process urgently needs people who can efficiently communicate data issues, and this kind of expertise is also needed in the test sites.

Lessons learned from TeleFOT

The following lessons learned (regarding “data”) are extracted from the TeleFOT project (Malone et al., 2014, section 8.3. SP4 Analysis):

- Establish a data working group from the outset – no matter how much the project feels it does not need it.
- Make sure the analysts are fully involved from the beginning – not joining at a later stage.
- Limit the numbers of research questions and hypotheses to a manageable number.
- Study first the impacts on travel and driver behaviour together and only after that divide the analysis work to impact areas to avoid overlapping work.
- Specify the data ahead of the FOT beginning.
- Engage fully with stakeholders – ensure that their needs are met.
- Decide how logged data will be managed.
- Pilot the FOTs – as much as possible.
- Never under-estimate the amount of time needed for analysis.
- Never under-estimate the amount of time needed to summarise the analyses as a readable deliverable.
- Never try to be over-ambitious with the data analysis.
- Decide how to handle your data before data collection starts.
- Plan for the data analysis as far as is reasonably practical – ad hoc creativity just creates unharmonized approaches.
- Prepare for the unexpected.
- Have a coping strategy for the unexpected.
- Accept the fact that despite best intentions, the final analysis will not give all of the answers required and more analysis will be required and/or will be possible.

Lessons learned from UDRIVE

The following lessons learned (regarding “data”) are extracted from the UDRIVE project (Martin et al., 2017, section 5 Conclusions).

During the pilot and the installation phases there were some mistakes with incomplete and/or wrong system delivery by the DAS provider. The lesson learnt was to check all the materials before installing them in the vehicles. However, the fact that there were no status reports on the OMT (Online Monitoring Tool) during the piloting phase, made more difficult to check the functioning of the equipment.

Regarding the equipment, the most important problem was the broken DASs (Data Acquisition Systems). Some DASs had to be sent back to the provider or manufacturer and the repair time was very long. This problem could be mitigated by having spare DASs that could be shared around operation sites as needed.

The DAS should be adapted to each type of vehicle. For the scooters, the fact that the DAS was the same than for cars created a problem. The DAS weight (15 kg) exceed the maximum weight permitted for a normal PTW support bracket (5–8 kg) which made change the vehicle model, the top case and the whole equipment installation. For the trucks, the technical system was too sensitive for heavy use trucks which caused that part of the time for some of the trucks data are not complete. In naturalistic driving studies with trucks, the systems must be extra robust.

Around 90% of the participants were aware that they had been recorded when driving. The lesson that can be learnt, is that somehow the equipment should be installed even more hidden if possible, not to make participants change their behaviour.

Finally, the ethical and legal issues should not be underestimated either and approval from the competent national authorities for data protection should be sought from the very start, as soon as the pilot site locations are determined, in order to avoid delays in the workplan when operations have already started.

Other Projects

The lessons learned mentioned in the final reports of interactIVe (Alessandretti et al., 2014) and AdaptIVe (AdatpIVe, 2017) address the research topics covered in the project (e.g. human–vehicle integration, applications, automation, evaluation, and legal issues). There are no specific lessons learned addressing data logging, data management, or data analysis in related documents.

Furthermore, the examined documents of the public funded projects eCoMove (Vreeswijk, 2013), FOTsis (Pou, 2015), and IVBSS (Sayer, 2011) did not contain data-related lessons learned.

Annex References

AdaptIVe (2017). *Final project results*. Public deliverable D1.0 (version 1.0) of the AdaptIVe consortium

Alessandretti, G., Amditis, A., Metzner, S., Johannson, E. and Fahrenkrog, F. (2014). *Final Report*. Public deliverable D1.9 (version 1.2) of the interactIVe consortium

Barnard, Y. and Fernandez, L. (2013). *Lessons learned from Pilots on Cooperative Systems*. FOT-Net 8th Stakeholders Meeting. Brussels

Barnard, Y. et al. (2016), Data Management and Data Sharing in Field Operational Tests. In P Pagano (ed.), *Intelligent Transportation Systems: From Good Practices to Standards*. CRC Press, pp. 59-70. DOI: 10.1201/9781315370866-4

Barnard, Y. , Koskinen, S. and Gellerman, H. (2014). A Platform for sharing data from field operational tests. In: *21st Intelligent Transport Systems World Congress*. Detroit, USA.

FOT-Net2 (2014). *Project Final Report*. Brussels.

Selpi et al. (2011). *Data management in euroFOT*. Public deliverable D3.3 of the euroFOT consortium

Malone, K. et al. (2014). *Impact Assessment and User Perception of Cooperative Systems*. Public deliverable D11.4 of the DRIVE C2X consortium

Martin, O. et al. (2017). *Lessons learnt from OS preparations*. Public deliverable D35.1 of the UDRIVE consortium. https://doi.org/10.26323/UDRIVE_D35.1

Mononen, P. et al. (2012). *Final Report*. Public deliverable D1.15 of the TeleFOT consortium

Pou, A. et al. (2015). *Project Final Report*. Public deliverable D1.4 of the FOTsis consortium

Sayer, J. (2011). *Integrated Vehicle-Based Safety Systems Field Operational Test Final Program Report*. Ann Arbor

Vreeswijk, J. et al. (2013). *Report on test site operation – Verification and validation results*. Public deliverable D5.9 of the eCoMove consortium