Development Platform for Safe and Efficient Drive

Warning Functions Solution Design

Deliverable n.	D41.1 – Warning Functions Solution Design		
Sub Project	SP 4	Functions	
Workpackage	WP 4.1	Warning Functions	
Task n.	T 4.1.1	Requirement analysis and solution design	
Author(s)	R. Brimble, G. Reade et al.	File name	D41.1 Warning-Functions_WIP_1.0
Status	1.0		
Distribution	Public (PU)		
Issue date	2014/04/06	Creation date	2014/02/04
Project start and duration	1st of September, 2012 – 36 months		

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATION	DESCRIPTION
AUTOSAR	AUTomotive Open System ARchitecture
DAS	Driver Assistant Systems
HMI	Human Machine Interface

OEM	Original Equipment Manufacturer
SP	Sub Project
WP	Work Package

REVISION CHART AND HISTORY LOG

REV	DATE	AUTHOR	REASON
0.1	15.05.2013	R. Brimble	Outline Draft
0.2	14.06.2013	G. Reade	Added content.
0.3	08.08.2013	G.Reade	Detailed functional requirements for ROW, REW, reviewed with TR.
0.4	29.11.2013	R. Brimble	Incorporated CRF contribution.
0.5	04.02.2014	R. Brimble	Amended in response to INFINEON review.
1.0		R. Brimble	Raised to 1.0 for issuing
1.1	07.04.2014	M. Kutila (VTT)	Layout improvements

EXECUTIVE SUMMARY

This deliverable is an interim output of WP4.1 of the DESERVE project and presents a set of warning functions to be used as test cases.

The set of warning functions has been derived within the context of the DESERVE demonstrators with the aim of exercising and validating the DESERVE platform (SP2) and for providing modules for integration and test (SP5). The warning functions could also be used as demonstrations of DESERVE results for dissemination (SP7).

Work within WP4.1 is being conducted in two phases. During the first phase (project months 7 to 12) the warning function test cases have been proposed, functional and process requirements have been considered, and results have been communicated to SP2. During the second phase, the warning functions will be developed into prototypes using the DESERVE platform and results from these implementations will be input into SP5. Deliverable D41.2 will be created during the second phase to capture results from these implementations.

Delays in this workpackage have occurred due to delays in the availability of inputs plus a tendency for partners to view end demonstrators as the focus and associated uncertainty as to how the warning functions are infact a step towards those end demonstrators. These needs to be resolved.

In this report we describe:

- The methodology used.
- The set of warning functions selected.
- Inputs into other parts of the project.
- What is planned for the second phase.

INTRODUCTION

DESERVE WP4.1 (Warning Functions) is responsible for developing a set of warning function test cases for the DESERVE project.

The set of warning functions will be used to exercise and validate the DESERVE platform (SP2) and to provide modules for integration and test (SP5). The warning functions may also be used as demonstrations of DESERVE results for dissemination (SP7).

This deliverable is an interim output of WP4.1 and a second deliverable (D41.2 *Warning Functions Prototypes*) will be produced to capture implementation and test results from the second phase.

In this report we describe:

- The methodology used so far.
- The set of warning functions selected.
- Inputs into other parts of the project.
- What is planned for the second phase.

Appreciation of Requirements

DESERVE has far reaching aims including:

- Cross-domain software re-use and standardised interfaces.
- Easy and safety-compliant integration of software and hardware modules.
- Pre-validation and pre-certification of ADAS modules.
- Integrated, trusted, interoperable tools and tool-chains.

The set of warning functions presented here aims to help translate these aspirations into concrete requirements and real-world examples. The main objective is to support the development and validation of the DESERVE platform and the novelty and innovative content of specific warning functions is not an explicit aim. The implementation of these warning functons must be completed using the (emerging) DESERVE platform, processes and tools.

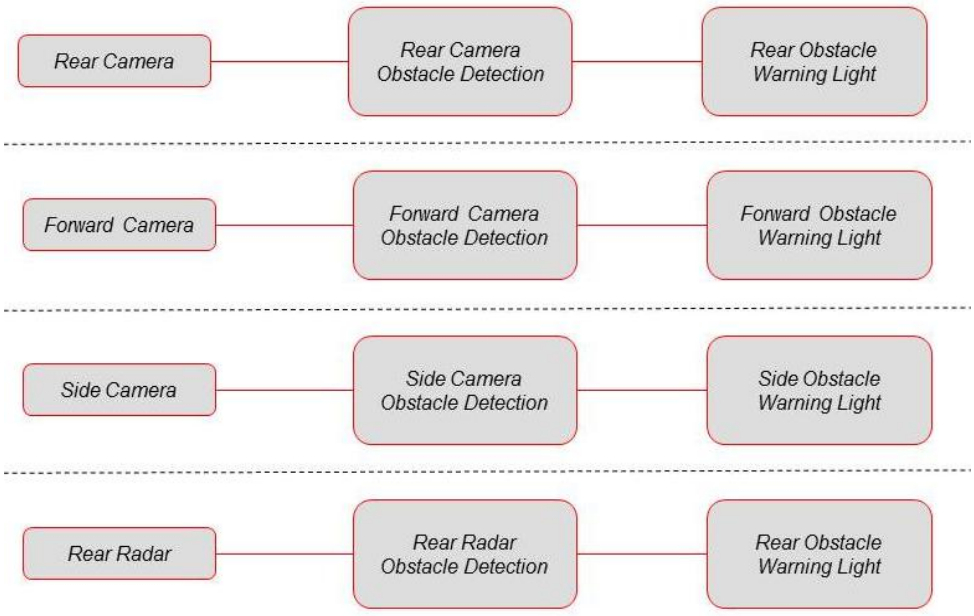
The implementations will provide building blocks for integration into the DESERVE application test cases (commonly referred to as the DESERVE demonstrators). This

means this workpackage should analyse the definitions of the demonstrators to identify relevant functions.

Since DESERVE is developing a generic platform this workpackage may also consider other functions to ensure coverage of the problem space balanced against partners skills and knowledge.

One of the key aspects DESERVE aims to improve is the practice of introducing a new "stove pipe" sub-system whenever a new function is added to a vehicle. A caricature of this is illustrated in Figure 1, where we have a different sub-system for each obstacle camera. Architectures developed with DESERVE results will provide greater integration between functions, including sharing resources and enabling different connection points to be created easily. WP4.1 aims to identify and develop a set of warning functions that illustrate this cultural change.

Figure 1 : A Stovepipe Design



Summary of Progress

WP4.1 kicked-off in May 2013 as planned in the DESERVE project schedule. A meeting was held between WP4.1 partners and objectives, deliverables and timescales were

discussed and responsibilities agreed. Subsequent tele-conferences and direct contact with individual partners has been used to progress work.

During this first phase (project months 7 to 12) a set of warning function test cases have been proposed, functional and process requirements considered and results have been communicated to SP2. The set of warning functions have been partially documented and this needs expanding during the second phase of this workpackage.

At the time of the kick-off there was a lot of uncertainty about the content of the application test cases as these were being defined in parallel to this workpackage. This resulted in useful iterative dialogue with partners defining the application test cases.

It has proven challenging to progress WP4.1 according to the initial schedule. The main impact of this for WP4.1 being preliminary design work has moved from phase 1 into phase 2. This has been due to a delay in the availability of inputs but also a partner focus on end demonstrators with uncertainty on how these functions provide components towards the end demonstrators. As the end demonstrators are evolved more completely the uncertainty associated with these functions will be reduced.

METHODOLOGY

WP4.1 is conducted in two phases conducted by seven partners (ASL, CRF, FICOSA, ICOOR, TTS, RAMB and VISLAB). The first phase runs until project month 12 and the second phase runs until project month 24. The work overall has been decomposed into three tasks:

- T4.1.1. Requirement analysis and solution design 1st Phase
- T4.1.2. Development of prototype solutions 2nd Phase
- T4.1.3. Laboratory and virtual testing 2nd Phase

Of these tasks only T4.1.1 is in scope for phase 1, which itself consists of four sub-tasks:

- T4.1.1.1 Establish functional requirements
- T4.1.1.2 Establish process requirements
- T4.1.1.3 Propose input requirements to SP2 and SP5
- T4.1.1.4 Design of Warning Functions

To balance project needs against resources the warning functions identified are to be developed to different levels of maturity. Some will be developed into full working prototypes, suitable for integration into the demonstrators, whilst others will remain laboratory or paper exercises.

The deliverables from SP1 that define the application core functions for future DAS have been used to set the context for WP4.1. Selection of the warning functions has been driven in the first place by the demonstrators and consequently the partners leading the demonstrators. These demonstrators cover passenger cars, commercial vehicles and motorcycles and will demonstrate Advanced Driver Assistance Systems (ADAS).

Partners were asked to provide suggestions for warning functions in the context of these demonstrators. Constraints for suggestions included requiring these functions to be relevant (to ensure selection was consistent with the context of the demonstrators), existing (to reduce the risk associated with creating novel functions) and sharable (to ensure data can be passed between DESERVE partners and sub-projects). Proposals were then made to partners leading the demonstrators to include specific warning functions as part of their architecture.

ANALYSIS OF INPUTS

DESERVE has identified 5 DAS Application Test Cases (commonly called demonstrators) to be used as a focus for the project.

Table 1 : DESERVE Application Test Cases

From SP1 the demonstrators to be developed are:
1. Automatic Emergency Braking for Pedestrians
2. Inter Urban Assist System
3. Driver Monitoring System
4. Adaptive Cruise Control with Autobrake
5. Driver Distraction and Intention Detection System

Deliverable D11.1 identifies 10 DAS groups containing 33 DAS core applications of the current and the near future car market. These core applications represent generic building blocks and all future DAS platforms should support them. There are 6 DAS core applications demonstrated within DESERVE which are contained within 5 DESERVE application test cases.

Table 2 : The 10 DAS Groups for DESERVE

1. Lane change assistance system
2. Pedestrian safety systems
3. Forward/Rearward (distant range)
4. Adaptive light control
5. Park assistant
6. Night vision system
7. Cruise Control System
8. Traffic sign and traffic light recognition
9. Map supported systems.
10. Vehicle interior observation

Table 3 : The 33 DAS Core Applications for DESERVE

Lane Change Assistance System 1. Lane Departure Warning System 2. Blind Spot Detection 3. Lane Change Assistance System 4. Overtaking Assistance System Pedestrian Safety Systems 5. Pedestrian Detection System Forward/Rearward looking system (distant range) 6. Collision Warning System 7. Low Speed Collision Avoidance System 8. Pre Safe System 9. Collision Avoidance System 10. Emergency Braking ahead 11. Electronic Emergency Brake Light 12. Intelligent Intersection	Park Assistant 19. Ultrasonic Park Assist System 20. Intelligent Park Assist 21. Rear View Camera System 22. Surround View Night vision system 23. Night Vision System 24. Night Vision System with pedestrian detection Cruise Control System 25. Adaptive Cruise Control 26. Adaptive Cruise Control -Stop & go Traffic sign and traffic light recognition 27. Traffic Sign Recognition 28. Traffic Light Recognition System 29. Curve Warning System 30. Fuel Economy System Vehicle interior observation

(Emergency Vehicle Detection) 13. Rear Approaching Vehicle 14. End-Of-Tail-Congestion Warning Adaptive Light Control 15. Adaptive High Beam Assist 16. Partial High Beam Assist 17. Inter Urban Light Assist 18. Map supported Frontal Lighting	31. Driver impairment warning System (drowsiness, fatigue, ...) 32. Driver/Rider visual Distraction Warning System (focus on the driving task, eye gaze evaluation) 33. Occupant Detection and Classification System

Table 4 : The 6 DAS demonstrated within DESERVE

1. Inter Urban Light Assist for passenger car	(Daimler)
2. Driver impairment warning System for cars	(CRF)
3. End-Of-Tail-Congestion warning for motorcycles	(Ramboll)
4. Blind-Spot-Detection for motorcycles	(Ramboll)
5. Motorcycle occupant detection and classification	(Ramboll)
6. Adaptive cruise control for heavy trucks	(Volvo)

IDENTIFIED WARNING FUNCTIONS

This section identifies the warning functions to be developed within DESERVE and maps these functions to the DESERVE demonstrators. Definitions of these functions can be found in subsequent sections. The set of warning functions to be developed within DESERVE are listed in Table 5.

Table 5 : The 7 Warning Functions for DESERVE

1. Reversing Obstacle Warning	(ASL)
2. Road Edge Warning	(ASL)
3. Lane Change Warning	(ICOOR)
4. Driver Visual Distraction Warning	(CONTI)
5. Driver Drowsiness Warning	(FICOSA, CONTI)
6. Emergency Braking Warning	(CRF)
7. Vunerable Road User Detection Warning	(VISLAB, CTAG)

Table 6 : Warning Functions mapped to Demonstrators

Warning function	Demonstrator	Application		
		Passenger cars	*Commercial vehicle*	*Motorcycle*
Reversing Obstacle Warning		X	X	
Road Edge Warning		X		
Driver visual distraction detection	Driver Monitoring System (FICOSA/CRF)	X	X	
Driver cognitive distraction detection	Driver Monitoring System (FICOSA/CRF)	X	X	
Driver drowsiness detection	Driver Monitoring System (FICOSA)	X	X	
Emergency braking warning	Automatic Emergency Braking for pedestrian (CRF)	X	X	
Erroneous trajectory warning	Inter Urban Assist (Daimler)	X	X	
Autobrake warning	ACC with Autobrake (Volvo)	X	X	
Visual and cognitive distraction detection	Rider monitoring platform (VTT)			X
Vunerable Road User Detection	Automatic Emergency Braking for pedestrian (CRF)	X		

DEFINITION OF WARNING FUNCTIONS

This section defines the warning functions to be developed within DESERVE. The design, implementation, and test, of these functions will be completed during phase 2 and documented in D412 Warning Function Prototypes.

Much of WP4.1 is about defining, developing, and testing interfaces between modules. The content of each module is arguably arbitrary and within some of the implementations surrogate modules may be used. This means that, for example, the decision-making aspect of a specific peripheral module may be driven by user-input and not an automated algorithm.

1. Reversing Obstacle Warning

This warning function uses rear facing sensors to detect and warn the driver about obstacles whilst a vehicle is reversing.

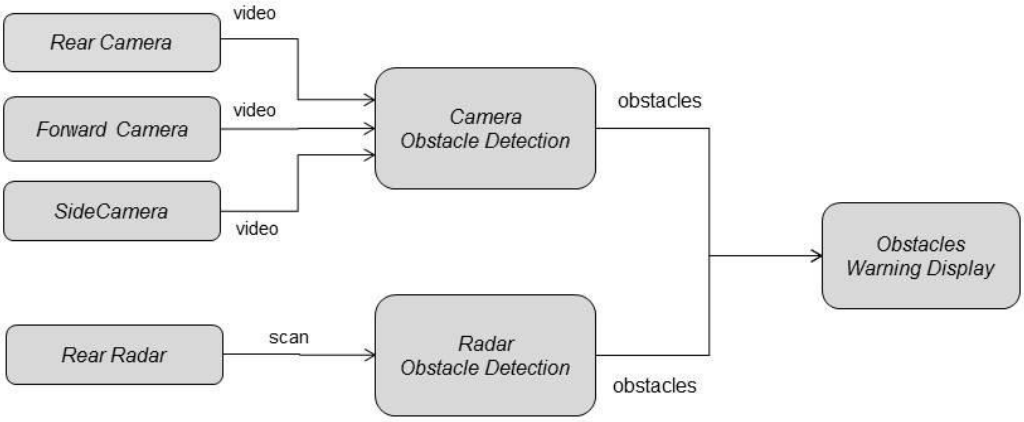
To exercise and validate the DESERVE platform this test case considers the use of additional cameras (e.g. front facing, side facing) and other sensors (e.g. radar).

To illustrate the DESERVE approach this function will be implemented as a modular design, with abstract interfaces that enable different modules to be integrated. For example, the nominal design is use of a single rear facing camera that provides input to a generic camera obstacle detection module which then sends obstacles data to an obstacles display module.

Using the DESERVE results this function will show how the reversing obstacle warning function could be modified to accept different sensor inputs (e.g. cameras mounted elsewhere on the vehicle, such as mirrors) and provide alert information to different actuation modules (e.g. dashboard lights, audio device). This may, for example, provide extra capability by using surround-view mirror cameras to detect obstacles that may be struck by the front sides while turning in reverse.

The warning function will initially be targeted for use on the passenger car application test case but will also be offered to the truck application test case to demonstrate migration of component systems across vehicle types. Note this would involve use of wide-angle both forward and reverse sensors.

Figure 2 : Reversing Obstacle Warning

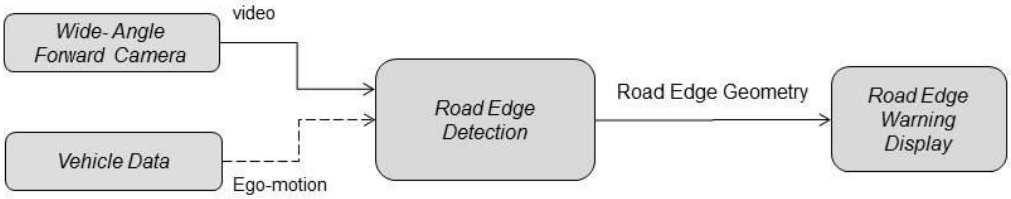


2. Road Edge Warning

Knowledge of where the road is enables a DAS to predict what a vehicle will do, know when a vehicle is about to enter an off-road condition and help classify the threat of detected objects.

This warning function test case will show how the road edge can be determined from a camera sensor for use within the CRF Passenger Car application test case. The function will accept video data from a wide-angle front-facing camera and maintain an estimate of where the road edges are. The geometry associated with the road edge will be generated as output and provide data which could feed into a road edge display (the display is not planned to be completed within this workpackage). This test case will also consider the use of vehicle ego-motion data as a means for improving edge detection.

Figure 3 : Road Edge Warning



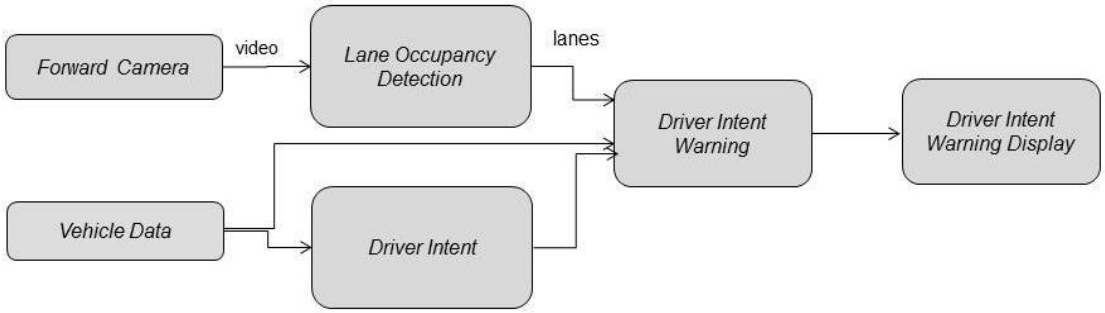
3. Lane Change Warning

The Driver Intent module being developed within SP3 will indicate when a driver is anticipated to turn left, turn right or carry straight ahead. This can be combined with a module that perceives whether the space (left, right or ahead) is free. Should the space not be free then a warning function could be used to alert the driver to the hazard.

This warning function uses a lane occupancy detection module to provide the external perception. This means a driver will be alerted if their anticipated steering intention would transition the vehicle into a lane that is occupied.

We additionally introduce vehicle data to only trigger the alert when the vehicle is moving at speed; i.e. slow transitions from road to off-road is considered not worthy of issuing a warning when the driver is manoeuvring slowly.

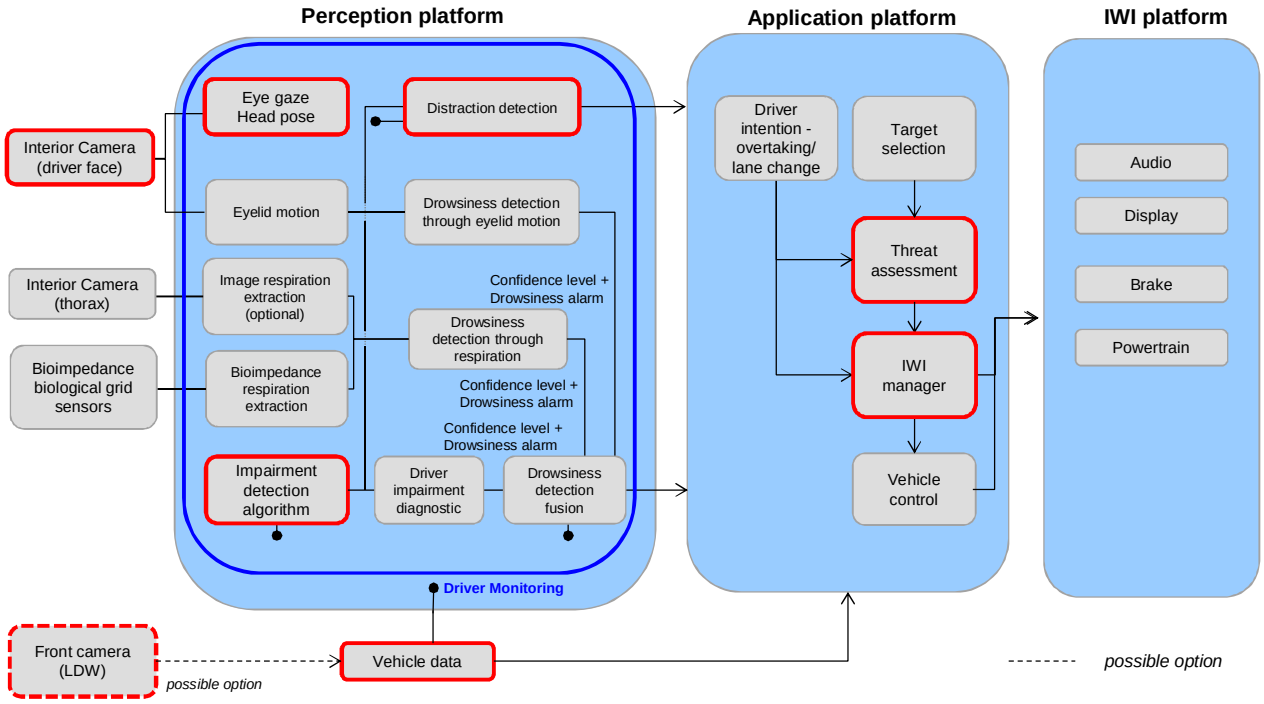
Figure 4 : Lane Change Warning



4. Driver Visual Distraction Warning

The functional blocks diagram, shown in Figure 5, describes the main software modules (highlighted in red) of the Driver Visual Distraction Warning function, included in both Perception and Application Platforms.

Figure 5 : Driver Visual Distraction Warning



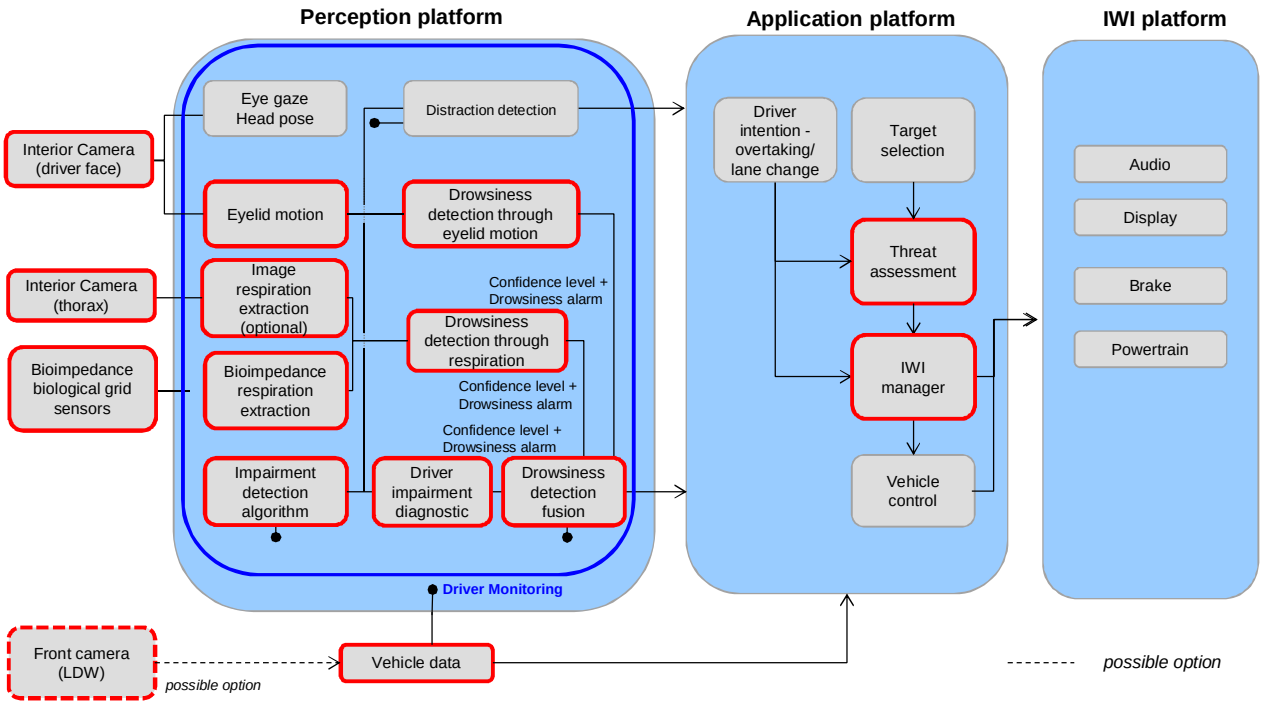
The Eye Gaze – Head pose module uses the video input from the camera. An impairment detection algorithm using several parameters (like Steering wheel angle, LDW data, pedal actions and so on…) from the vehicle CAN is also implemented in the platform. Impairment level diagnostics module then elaborate the alarm signal with a confidence level

The eye gaze and head position data are sent to a distraction detection module, which elaborates them with other parameters from the vehicle CAN, like actions on radio and navigation system in order to infer a driver level of distraction sent to the application platform. Finally the processed data are sent to the Application Platform that decides the intervention strategy.

5. Driver Drowsiness Warning

The functional blocks diagram, shown in Figure 6, describes the main software modules (highlighted in red) of the Driver Drowsiness Warning function, included in both Perception and Application Platforms.

Figure 6 : Driver Drowsiness Warning



The Eyelid Motion module uses the video input from the camera. A specific module is dedicated to the Bioimpedance biological sensors.

An impairment detection algorithm using several parameters (like Steering wheel angle, LDW data, pedal actions and so on…) from the vehicle CAN is also implemented in the platform. Impairment level diagnostics module then elaborate the alarm signal with a confidence level

The data of eyelid movement are used by a specific drowsiness detection module; similarly another module elaborates the data from the biological sensor (or/and from the camera). Both supply their warning combined with a confidence level. These warnings

and the ones form the impairment level diagnostics are used, combined with their confidence levels in the drowsiness detection fusion module to elaborate the data sent to the Application Platform.

6. Emergency Braking Warning

The functional blocks diagrams, shown in Figure 7 and Figure 8, describe the main software modules (highlighted in red) of the Emergency Braking Warning function, included in both Perception and Application Platforms.

Figure 7 : AEB Interurban Warning

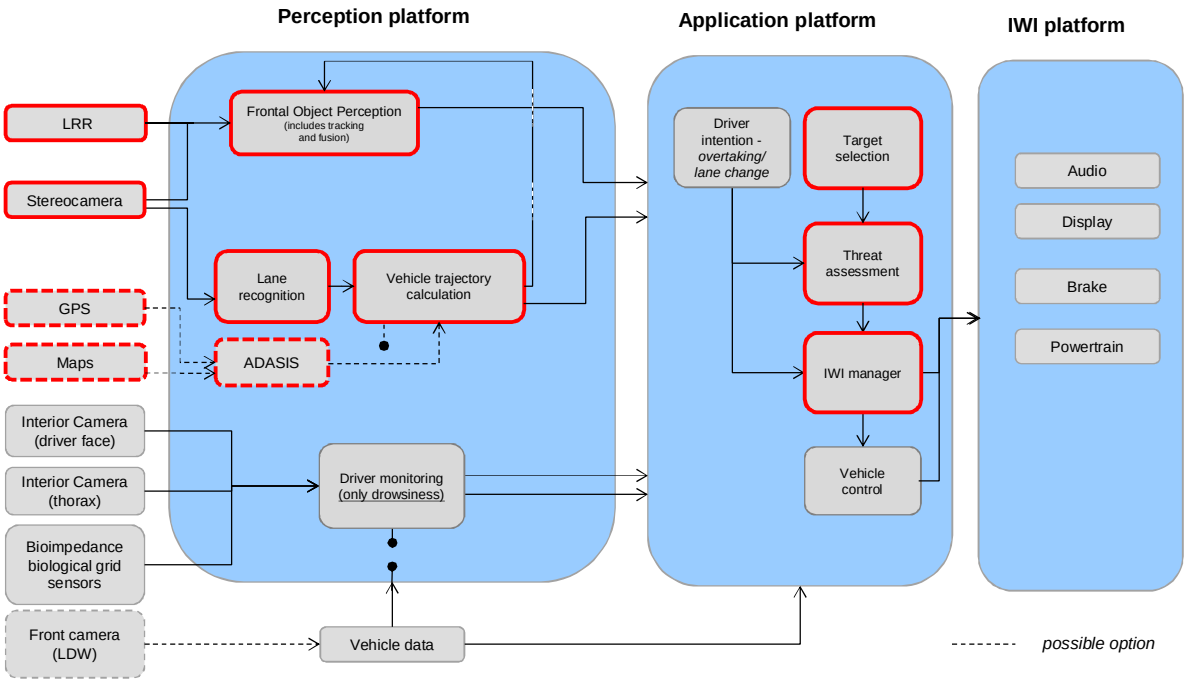
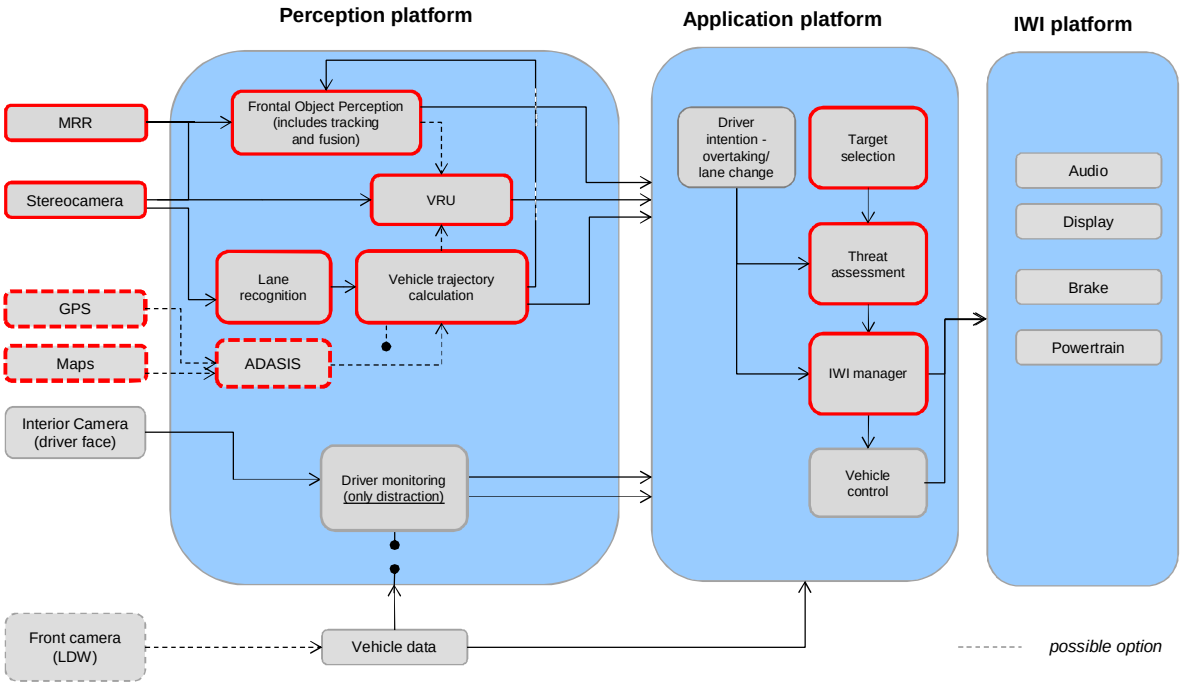


Figure 8 : AEB Pedestrian Warning



The Frontal Object Perception module aims at detecting every relevant stationary and moving obstacle in front of the host vehicle. Advanced filtering techniques and data fusion are executed in order to extract additional information from the sensor data. The module receives as input the sensor signals (mid-range radar, stereo camera) and the data from the Vehicle Trajectory Calculation module. The output is based on a list of detected objects (stationary and moving) with the required attributes.

The Vehicle Trajectory Calculation module provides the trajectory of the host vehicle as a list of points. The aim is to predict the driver's intention few seconds in advance estimating the path of the host vehicle and its dynamics with respect to a given fused road geometry.

The Lane Recognition module evaluates the position and geometry of high-contrast lane markers of the host vehicle lane on the road. This module uses a stereo camera as sensor input. The ADASIS Horizon module allows to extract the host vehicle position data as well as the road segment attributes from the digital map.

The outputs of the former modules are sent to the Application platform. In particular, the Target Selection module determines which vehicle is the most relevant target obstacle

related to the current and predicted driving situation. The priority of the target depends on its trajectory in comparison to that of the host vehicle.

The Threat Assessment module classifies and assesses the longitudinal and lateral risks associated to the current situation based on specific measures like TTC (time-to-collision) and TLC (time-to-lane-crossing). It also calculates the threat for a new alternative trajectory.

The IWI module is based on Information, Warning and Intervention manager. The Information Manager deals with the information to be provided to the driver. The Warning Manager analyses the results of the Threat Assessment and the Driving Strategy modules and makes the final decision about when to issue a warning and when to suppress it. The Intervention Manager analyses the results of the Threat Assessment and the Driving Strategy modules and makes the final decision about when to issue an intervention and when to suppress it. The Vehicle Control module determines in case of an intervention the desired longitudinal deceleration request based on the results of the components trajectory planning and control. If a braking action is needed it controls the correct amount of brake force to ensure that the vehicle is braked safely.

The software algorithms related to the IWI Manager module are different for the AEB Pedestrian and the AEB Inter Urban functions.

7. Vunerable Road User Detection Warning

The Frontal Object Perception module in Figure 8 is used by the Vulnerable Road Users module that detects, classifies and tracks pedestrians in front of the host vehicle. The VRU module will fuse information coming from a forward looking radar sensor, capable of accurately measuring the range to the object, and a vision sensor providing classification capability and accurate lateral positioning in order to get a sufficiently reliable pedestrian detection.

The VRU detection module could be implemented using input from the Vehicle Trajectory Calculation module to identify only those users at risk within the current trajectory. Alternatively the VRU detection module could be implemented without such trajectory information and result in output for all VRUs detected irrespective of their location with respect to the vehicle trajectory.

WP4.1 aims to show such flexibility in designs though defers the decision about which aspects will be addressed within this function until the next phase.

REQUIREMENTS

7.1. Functional Requirements

WP4.1 was tasked with establishing the required performance of the warning functions and the range of vehicles on which the warning functions must be deployed. The range of vehicles was expected to span commercial vehicles to passenger cars and other types of vehicles. Analysis was also required to consider the data acquisition, processing units and warning devices.

WP4.1 took the Application Database from SP1 as input to identify warning functions within scope. Functional requirements for these warning functions were then compiled. WP4.1 follows the convention adopted elsewhere in the project for capturing functional requirements. This helps consistency across the project. For each function requirement we capture the ID, Name, Definition, Importance, Type, Rationale and Responsibility in tabular form.

The set of functional requirements are maintained in a separate excel spreadsheet. For convenience this set is duplicated in this report (see Appendix A).

7.2. Process Requirements

WP4.1 was required to analyse the working requirements of the DESERVE platform, and the working requirements of the integration and test process. This included establishing requirements to produce generic solutions leveraging the DESERVE approach. Where DESERVE does not mandate any particular approach WP4.1 was tasked with providing an agreement between partners.

Partners working on WP4.1 were asked to provide general details on their process requirements and current toolchains. The information gained was not rich enough to provide a definitive picture for various reasons including confidentiality. It is clear that – as expected - there is a mixture of approaches with some partners using dedicated tools for particular steps and others using familiar tools which are made to fit the task in hand.

Management of Requirements ranged from using DOORS, Microsoft Excel Spreadsheets and Tagged Microsoft Word documents linked to in-house tools. Specifications were written using Microsoft Word and In-house standard proforma based on programming languages when precision was needed. General in-house dissemination of issues and documents included Wikis and Sharepoint.

Design methodologies were based around the V Model and ISO26262 and tools included Enterprise Architect, Rhapsody, SystemDesk and Matlab/Simulink/Stateflow. Software methods and tools included Matlab, QT-based ECU simulation, RTMaps, Simulink, C, C++, with tools running under Windows and Linux on desktops and dedicated servers.

There was typically a need to develop using tools based around the OS from chipset supplier and any of the AutoSAR toolsets. MISRA coding standards featured where relevant and builds and tests were managed by scripts and sometimes cross-compiled across platforms.

Configuration management is achieved using well-known open source tools such as SVN and GIT plus in-house release management tools. Review tools included Gerrit and Testing was achieved using Matlab, Symulink, Polyspace, using recorded data via CANalyzer and CANape.

Deliverable D132 provides a discussion of different methods and tools relevant to the development of DAS. It recognises that the DESERVE platform is an abstract concept that will be realised in practice using different tool-chains. This presents a challenge when defining mandatory requirements for those building concrete implementations.

Partners working on WP4.1 are only just getting to a stage where they understand which tool chain they will use to develop their prototype(s) and hence concrete process requirements that are not addressed by D132. Once these evolve then they will be documented in the update to this deliverable.

7.3. Inputs To SP2 and SP5

Prototyping a set of warning functions provides a key opportunity to give feedback to SP2 (ADAS Development Platform) and SP5 (Integration and Test). Specifically, WP4.1 has provided requirements as inputs to SP2 to help ensure the platform defined by SP2 supports the warning functions being developed. Implementation results from WP4.1 will

be fed into SP5 when available to act as functional building blocks suitable for integration.

The DESERVE demonstrators (which provide integrated system test cases) will be realised in association between partners as a collaborative effort. The warning functions to be designed and developed during WP4 are identified in Table 5 and mapped to the application type (passenger car, commercial vehicle and motorcycle).

7.4. Further Work

Test cases will be developed up to the stage of working prototypes - using the emerging DESERVE platform, process, and tools - during the second phase of the workpackage (months 12 to months 24 of the project).

CONCLUSIONS

Within this deliverable a set of warning function test cases has been identified for the DESERVE project. The document has described how these test cases have been derived from the needs of the project and how they map to the DESERVE demonstrators.

This deliverable concludes the first phase of WP4.1. The second phase of WP4.1 will develop the identified test cases into laboratory prototypes to exercise and validate the emerging DESERVE platform.

REFERENCES:

[1] www.deserve-project.eu

[2] DESERVE consortium, *DEvelopment platform for Safe and Efficient dRiVE*, Technical Annex Part B, 31.01.2013

[3] *D111 Application Needs*, DESERVE; Issue Date 2013/03/06

[4] *D112 Platform needs*, DESERVE; Issue Date 2013/04/08

[5] *D132 Method and Tools Specifications*, DESERVE; Issue Date 02.08.2013

Appendix A

Functional Requirements

ID	Req_DPR_LIC_1_v0.1
Name	Development Tool Licensing (within DESERVE project)
Definition	Any processes or tools added or propagated into partners from others, need to be freely available to the recipients. That is, no licensing issues or onerous implementation difficulties. Applies in scope of DESERVE project, not to work by partners outwith DESERVE.
Importance	C
Type	N
Rationale	This may already be part of DESERVE agreement.
Responsibility	All partners.

ID	Req_DPR_RQM_1_v0.1
Name	Management of Requirements Documentation
Definition	Requirements will be written and maintained in widely available desktop tool formats, or a similarly available online format. File formats are defined by Microsoft Office WORD and Excel, but any compatible tool may be used. Online format might be e.g. a wiki on the DESERVE website, for

	which all partners have write access. Requirements written in the same presentation as this. Outline requirements for all WP 4.1 Warning Functions collected into one document (this document). General requirements for all WP 4.1 Warning Functions collected into one document (this document). Detailed specific requirements for WP 4.1 Warning Functions in one or more* document(s)** per function or layer***, these documents referenced from the outline requirement. Requirements uniquely identified by e.g. as ID field of this, and with individual version numbers incremented at each requirement change****. Requirements identified as reviewed and agreed by partners… how?
Importance	C
Type	N
Rationale	*If detailed requirements were collected across partners, it would become very difficult to manage updates to the single document. **The term “document” here is used loosely as an individually managed block of information. ***See next requirement. **** The requirements will be inserted into a spreadsheet(s) which will automatically generate the required tags, which can then be copied back into this document (process from CRF/N.Pallero).
Responsibility	All partners.

ID	Req_DPR_RQM_2_v0.1

Name	Management of DESERVE Layer Requirements
Definition	Outline requirements should (shall?) split each Warning Function into layers of Perception, Application and Controller, and it is likely to be advantageous to write a separate document of requirements for each layer. Conversely, it may be possible to write requirements at one layer that are shared between multiple WFs.
Importance	C
Type	N
Rationale	DESERVE is about interoperability and platforms, and it should be possible to use one supplier's perception layer with another supplier's application: Therefore, the requirements for each should be separated.
Responsibility	All partners.

ID	Req_DPR_RQM_3_v0.1
Name	DESERVE Requirements for Maturity
Definition	Requirements for each function and/or layer shall identify the maturity of the item within scope of DESERVE. Also, useful to identify the maturity change that the work in DESERVE may achieve.
Importance	C
Type	N
Rationale	For DESERVE WP4.1 we are lab testing ADAS functions. It would be important to identify, in each case, and to review at critical points, whether the ADAS function maturity is at the start and intended to be at

	the end: • Laboratory prototype/experiment • Live testing prototype/experiment • Production intent but not safety related. • Production intent and safety related. • And, aspects of qualification for use. And this may define the development method (see requirement below). The WP plan (m12 report) can outline the intention per Warning Function.
Responsibility	All partners.

ID	Req_DPR_SAF_1_v0.1
Name	Management of Safety Needs
Definition	The requirements for each function, or each DESERVE layer of each function, shall identify the safety requirement for that function/layer. For layered ADAS functions, it will be necessary to identify the safety requirements both of each function/layer, but also its safety requirements imposed on lower or higher layers. For DESERVE, it would be necessary to identify the safety requirements for each function/layer as deployed in DESERVE*. The safety requirements written for each function/layer may be written in two separate requirements: “For DESERVE deployment” and “For Production deployment”; it is suggested the latter should be present, even

	if content is “TBD”, so as to flag up that more may be needed than for research.
Importance	C
Type	N
Rationale	*In the sense of a risk assessment specifically of the DESERVE deloyment. If it is only for laboratory testing in DESERVE and no hardware transducer output , then there may be no risk of harm and no safety requirement at all. If it is for demo car testing in DESERVE then there may be a need for expressed safety resrictions which may feed back into actual functional requirements. For example, the hedgehog** detection operation of automatic braking may adopt a functional requirement of automatic suppression above 5kph (so as to avoid injury to test engineers from false detections), and a non-functional requirement never to test on public highways (because of lack of completed safety qualified development). **Translation:  sweetclipart.com
Responsibility	All partners, especially systems integrators.

ID	Req_DPR_DEV_1_v0.1
Name	Development Method
Definition	For each function and/or layer an appropriate development process shall be identified in requirements. Corresponding to the maturity issue discussed previously, it’s particularly important to identify what is to be achieved at the end of the development.

Importance	C
Type	N
Rationale	Appropriate processes will be quite different according to function maturity (requirement above). For example, for laboratory prototyping, it may be most productive to adopt a form of agile process in which we aim to complete a new experiment every week. For scheduled delivery of a working mass-production product, a very controlled product delivery process with an emphasis on specification, detail & regression testing is more appropriate. Applicable in WP4.1 to lab testing only, so most likely a scheduled output of multiple experiments is appropriate.
Responsibility	All partners.

ID	Req_DPR_DEV_2_v0.1
Name	Development Process Maturity
Definition	None for DESERVE WP 4.1.
Importance	O
Type	N
Rationale	As WP4.1 requires partners to test their own functions in their own laboratories, and only up to laboratory test stage, there is no need for known process maturity in development. More relevant to DESERVE overall and may be defined from SP1, but not yet (Sept 2013). Much more relevant when targeting tests on vehicles or operating

	controls.
Responsibility	All partners.

ID	Req_DPR_DEV_3_v0.1
Name	Development Planning
Definition	For each function and/or layer a project plan or schedule shall be created. This would be expected to be concurrent with or immediately after, identification of requirements. For developments encompassing deployment onto vehicles, especially with control outputs, one would expect the plan to include appropriate “gateways” where the development qualifies (or not) to operate potentially dangerous controls.
Importance	C
Type	N
Rationale	Following previous requirement “Development Method”, this requirement is effectively to plan the development according to the chosen method.
Responsibility	All partners.

ID	Req_DPR_DEV_4_v0.1
Name	Design Method
Definition	None for DESERVE WP 4.1.

Importance	O
Type	N
Rationale	It's in the nature of DESERVE's modular approach, that the interfaces on modules will need to be clearly agreed, and maintained as stably as possible. DESERVE should surely seek a clearly expressed architecture, and a method for defining interfaces (at all levels), and we suggest, some interfaces (or elements of) can be standardised. But this is not WP4.1, and it is a source of concern (to ASL) that earlier subprojects have yet to provide such things. There is no need in DESERVE, to adopt common design methods inside modules; indeed this will remain an area of competitive advantage.
Responsibility	All partners.

ID	Req_DPR_DEV_5_v0.1
Name	Software Development Tools
Definition	For scope of DESERVE WP4.1 the partners shall use tools identified as part of DESERVE framework. This effectively mandates certain testing tools, but not implementation tools.
Importance	O
Type	N
Rationale	See also requirement above regarding licenses. This requirement helps towards but would not alone achieve interoperability.

Responsibility	All partners.

ID	Req_DPR_DEV_6_v0.1
Name	Software Interfaces
Definition	Not required for WP4.1
Importance	O
Type	N
Rationale	May be defined from SP1, but not yet (Oct 2013).
Responsibility	All partners.

ID	Req_DPR_DEV_7_v0.1
Name	Coding Standards
Definition	Not relevant for WP4.1
Importance	O
Type	N
Rationale	As WP4.1 requires partners to test their own functions in their own laboratories, there is no need for shared coding standards. More relevant to DESERVE overall and may be defined from SP1, but not yet (Oct 2013). Adherance to some coding standards (e.g. MISRA) may be required for acceptance onto test vehicles. Interfaces may need to be AutoSAR compilant for use on test or

	production vehicles, may be defined in SP1, but not yet (Oct 13).
Responsibility	All partners.

ID	Req_DPR_DEV_8_v0.1
Name	Software Build Process
Definition	Not relevant for WP4.1
Importance	O
Type	N
Rationale	As WP4.1 requires partners to test their own functions in their own laboratories, there is no need for build process disclosure. More relevant to DESERVE overall and may be defined from SP1, but not yet (Oct 2013).
Responsibility	All partners.

ID	Req_DPR_DEV_9_v0.1
Name	Configuration Management
Definition	Not relevant for WP4.1
Importance	O
Type	N
Rationale	As WP4.1 requires partners to test their own functions in their own laboratories, there is no need for shared configuration management.

	More relevant to DESERVE overall and may be defined from SP1, but not yet (Sept 2013). *Can a repos be added to the workspace.vtt.fi website? How would that work as for interfaces & libraries?*
Responsibility	All partners.

ID	Req_DPR_TES_1_v0.1
Name	Test Data Management
Definition	Not relevant for WP4.1
Importance	O
Type	N
Rationale	As WP4.1 requires partners to test their own functions in their own laboratories, there is no need for shared test data management. More relevant to DESERVE overall and may be defined from SP1, but not yet (Oct 2013). Test data for video systems is extremely large, and not always feasible to transfer via WAN. So sharing data is likely to require duplication, and then sharing identification data.
Responsibility	All partners.

ID	Req_DPR_TES_2_v0.1
Name	Test Plans

Definition	Testing required for each function/layer shall be defined in a systematic way, taking into account the maturity required at the end of the process. It is suggested this is best done with a test plan for each function and/or layer. The test plans should be clear about the objectives of each test, e.g. proof of concept, reliability extension, acceptance, regression; indeed it may be preferable to split the planning of these tests.
Importance	O
Type	N
Rationale	
Responsibility	All partners.