White paper:
Cooperative Mobility Device

In car platform for ITS & Mobility

Authors: H.Verbeek, AutomotiveNL
R. van Venrooy, TomTom

Version: 1.0

Date: 17-12-2012

Purpose

The purpose of this white paper is to share the ideas and considerations underlying a new concept to accelerate the adoption of advanced ITS and mobility services by the consumer: the Cooperative Mobility Device (CMD). The contents from this paper can form the basis of project proposals and industrial development activities.

Table of contents

Glossary

CMD	Cooperative Mobility Device
CMD-Pod	device within the CMD concept (without display functionality)
ITS	Intelligent Transport Solutions/Systems
WiFi-p	short range mobile communication technology
CAN	Controller Area Network, in-vehicle data network
OEM	Original Equipment Manufacturer (car manufacturer)
UI	User Interaction
OS	Operating System
ADAS	Advanced Driver Assist Systems
GNSS	Global Navigation Satellite System
DAB	Digital Audio Broadcasting
TPEG	Transport Protocol Experts Group (traffic/travel data standard)
USB	Universal Serial Bus
LTE	Long Term Evolution (advanced cellular network)
4G	4th generation cellular network
NDS	Navigation Data Standard (physical storage format for maps)
SDK	Software Development Kit
HMI	Human Machine Interaction
Bluetooth	standard for short range wireless communication
GSM	Global System for Mobile communications
CE	Consumer Electronics
IP	Intellectual Property
FOT	Field Operational Trial
DITCM	Dutch Integrated Test site for Cooperative Mobility

Abstract

In this white paper the concept of the Cooperative Mobility Device (CMD) is presented as a response to the challenge of introducing smart mobility applications and services to the in-car environment on a large scale in such a way that drivers are enabled to complete their trip in a safe, convenient and efficient way while contributing to the societal objectives of emissions reduction, congestion reduction and overall traffic safety. There is a need for new concepts because the structure of the current automotive supply chain is not adequately suited to implement all new services as fast as desired. However, looking to past in-car revolutions some lessons can be learned and applied to the current challenge.

This leads towards the concept of the CMD that allows multiple applications and services to interact on a single platform and that integrates user interaction mechanisms from these applications seamlessly. It is argued why the separation of the display and the processing unit(s) is a key characteristic of the CMD concept. Within this concept a wide variety of functionalities can be realized, depending on the mix of technology enablers that are implemented such as short range communication (WiFi-p), vision technology and vehicle data network connectivity (CAN). The various functionalities and services will be implemented through separate apps from different parties, allowing a vast community of developers. To ensure a seamless user experience it is mandatory that an integrated UI mechanism is adopted that is capable of interacting with all the individual apps and that can produce enticing and supportive advice to the driver.

The concept should be affordable, adaptable, flexible, open, scalable and acceptable. Cost levels should be within the range of current high end consumer devices and not in the range of accessory list prices from car manufacturers, allowing both aftermarket and OEM implementations.

The CMD-Pod is presented as a key device within the framework of the overall CMD concept: it is the device that allows decoupling of UI and processing functionality, decoupling of vehicle related hardware and nomadic devices and integrating enabling technologies as options. Some basic solutions for the mechanical integration of the CMD-Pod -primarily linked to the location in the vehicle -are discussed. Finally an outline is shown about the phased deployment of the CMD concept, based on opportunities in near future projects.

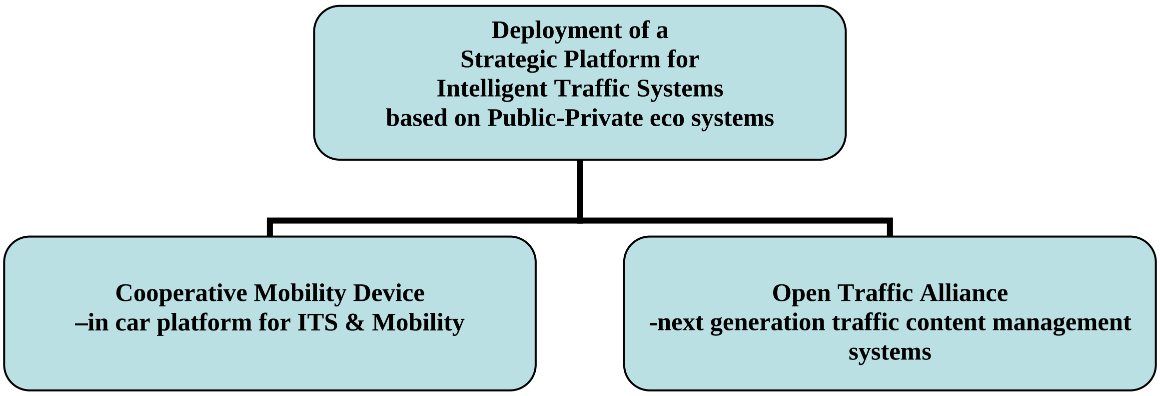
Introduction

Goal

The goal of the proposed CMD concept is to enable the high volume deployment of services in the area of ITS and smart mobility by offering an open, sustainable and affordable in car platform that is able to communicate with back-offices and road side infrastructure. It builds upon the results from the former SPITS project (2009-2011) in which the technical viability of such a platform has been successfully demonstrated.

Scope

The content of this white paper describes a part of the wider scope of new ways to deploy smart mobility services as presented in the white paper 'Deployment of a Strategic Platform for Intelligent Traffic Systems based on Public-Private eco systems'*. For this platform the in car part is discussed in the current white paper 'Cooperative Mobility Device – in car platform for ITS & mobility'. The part related to infrastructure and back-office is presented in the white paper 'Open Traffic Alliance –next generation traffic content management systems'. The diagram below summarizes the relationships between the various papers.



* this paper is still in the editing phase at the release of the current document

Need for new concepts

Mobility innovations restrained

In the area of Intelligent Transport Systems a host of new applications and services have recently been demonstrated in worldwide research programs, proof of concept projects and field operational tests. In most cases the technology level has proven to be sufficient and benefits for travel comfort, safety and environment are appealing. However, a widespread and large scale adoption of these new applications can only be realized when consumer electronics innovations are deployed within vehicles, but this has proved difficult for the following reasons:

- Integration of state-of-the-art consumer electronics for in-car usage is a complex task
- Automotive sales channels are fragmented over car brands and types
- Life-cycles of cars and consumer electronics are vastly different
- Car OEMs generally will not allow another, independent service provider to use their proprietary in-car platform
- Car interiors are not suited to accommodate (replaceable) electronics devices
- Consumer electronics don't comply with automotive quality standards and as a rule are not designed to function in such an environment
- The high level of customization required for OEM adoption conflicts with the high volume requirement for affordable consumer devices
- Retrofit systems are mostly restrained to a single application or service provider, prohibiting re-use by other parties

From the above it is clear that an innovative approach is necessary to achieve a significant take-up of valuable mobility services.

Learning from past revolutions

In the past such an approach has been successfully adopted in the area of in vehicle audio features. Fast-paced developments in consumer audio functionality resulted in ever new quality standards (like Dolby) and content carriers (like cassettes, CD's), raising the demands of the consumer-driver. Eventually the introduction of the standard DIN slot in the dashboard enabled the innovations in home audio systems to reach the in car environment. Next to the definition of the slot size this required standardization of electrical power and signal interfaces within the vehicle accessory system.

Another breakthrough has been the large scale introduction of navigation functionality within cars. Originally developed for in-dash systems fitting the above mentioned DIN-slot, massive take up only took place after the introduction of car-kit systems that could be easily put on the windscreen and powered from the cigarette lighter. The fact that the standard interfaces from the vehicle perspective had already been established (the cigarette lighter interface and the generic properties of the windscreen) enabled an even faster introduction than in the audio/DIN slot case.

Looking into these two examples the following characteristics appear:
Killer application
A new car integration concept for consumer electronics is triggered by an emerging killer application that is 'pulled' into the car by customer needs; first this need was superior audio experience, later it was route guidance convenience.
Integration by isolation
The paradox is that the initial attempt to integrate the new functionality in the existing vehicle environment is not successful enough for mass adoption; only after isolating the new functionality to a certain extent from its hosting environment –the car dashboard- an effective integration occurs.
Simplicity in hindsight
Looking back from the current state of the art the new concepts don't look like rocket science: a hole in the dashboard, a cigarette lighter… however, long periods of time elapsed before it came 'obvious' to adopt the new concept. After introduction this simplicity of the solution supported a rapid market acceptance.

Challenge and opportunity

Also for the new and valuable (cooperative) mobility applications the challenge is their integration into the vehicle is such a way that it appeals to users and enables high volume take up. Although some car manufacturers have product offerings for a subset of these applications the proposed systems show limitations as to openness and adaptability. Although this makes sense from the perspective of securing the proprietary supply chain and the customer contact point, it hampers a fast and wide proliferation of new services from a variety of sources.

From the customer point of view a low acceptance threshold is achieved when the new applications can be boxed into a device with integrated user-interface (screen) combined with an easy installation process. This was the case at the introduction of stand-alone, retrofit personal navigation devices, a concept that by now has entered the second half of the life-cycle S-curve. As more navigation functionality becomes available as software components for standard in-dash hardware and smart phones the volume of separate device solutions is declining. This indicates that the 'functionality-in-a-box' solution fills the gap between 'no - functionality' and 'seamless-integrated -functionality'.

The challenge at hand is how this would look like in the case of (cooperative) mobility applications. The multitude of mobility services (as opposed to a single application) that will grow exponentially in the years ahead will require a very high system flexibility; to accommodate this requirement a separation between processing unit and screen seems a logical step. The multitude of services mentioned above is also the basis for the large business opportunity: when finding a suitable concept a host of new functionality can be unleashed to the market in an economical feasible way, affordable for the consumer and with little integration hassle.

CMD concept

The basic idea

The solution to the 'gap-filling' challenge for mobility services in the car described above is the Cooperative Mobility Device (CMD). It is a concept that enables mobility services from various sources and suppliers to be presented to the in car user in an intuitive way, sharing necessary system functionalities and allowing controlled extensions and upgrades. It is a called 'device' because it involves a combination of hardware and software, but this hardware can be distributed over various hardware units as will be explained later.
Fundamental to the CMD concept is the combination of two characteristics that seem contradictory to each other at first sight:

- Allowing multiple applications and services to interact on a single platform;
- Integrating user interaction mechanisms from these applications seamlessly.

The first characteristic is realized by building upon an existing open source based software platform that can be combined with a large variety of hardware platforms: Android OS for tablets, smart phones and computers. The second characteristic is the result of a dedicated 'shell-ware' that integrates the user interactions from the various mobility services into a single user experience. It is in fact an application connected to a routing engine with a certain control over the execution and presentation of other, related applications.

> *Example:*
> The CMD features a 'speed-to-green' advice from provider X to advise on the optimum, real time driving speed to catch the next green traffic light. At the same time provider Y transmits the actual dynamic speed limits as retrieved from the local road authorities and provider Z continuously offers the optimum multimodal trip planning depending on real time traffic conditions. All three applications can be developed, deployed and sustained independently using the open platform and standardized protocols; Yet the user only gets a single, easy to interpret advice on speed and direction.

Functionalities and enablers

The CMD concept can –potentially- offer a wide variety of functionalities that can be used in various mobility services and these functionalities are in turn linked to technological enablers, components and building blocks that can be part of a specific execution of a CMD. The most important technological enablers are listed below, followed by a cross reference list with the functionalities that are supported.

- State-of-the-art smart phone platform electronics
  For a fast adoption of the feature innovations from the consumer electronics market into the car environment it makes sense to use generic available smart phone platform technology because this technology is backed by huge development resources and boasts highest performance/price ratios.

- Camera electronics & computer vision

Through smart phone usage consumers are becoming used to affordable camera functionality whereas most car brands offer advanced vision based functions . Combining these technologies will allow after market offerings to a wide user group at affordable price levels.

- Advanced positioning technology
  Various technologies enable a more accurate and/or more robust positioning; these include 3D acceleration sensors, gyro sensors and ADAS maps. In the near future higher accuracy GNSS positioning will be added to this list.

- DAB electronics and TPEG technology
  The digital broadcast technology DAB is expected to replace the current analogue broadcast technology. Its market penetration is still low due to the higher system costs and –up to now- insufficient benefits to balance these costs. Integration within the CMD concept could change this, using DAB's potential for unidirectional large data package transmission. TPEG technology is an established means for transferring traffic related, location referenced data and hence will be reused in the CMD environment.

- USB and Wifi connectivity
  These are standard solutions to connect consumer devices, wired and wireless respectively. It is expected that over time ever more vehicle integrated devices will feature at least one of these connectivity technologies.

- Wifi-p connectivity
  Using the physical connectivity layers of the widely adopted Wifi standard, Wifi-p technology is specifically designed for high speed mobile usage; it enables vehicles to connect real time to each other and to the road side infrastructure. This enables those smart mobility use cases that depend on safe, secure and robust connectivity with a very low latency. Car manufacturers have already announced standard integration of Wifi-p in their vehicles as from 2015.

- Cellular connectivity
  Just as current cellular connectivity allows speech and data transfer and hence internet access, future developments like LTE/4G will only broaden its application field, offering ever higher bandwidth and lower latency.

- Standard and modular run-time map formats
  Current navigation systems rely on proprietary, static maps that only can be updated via bulk upload mechanisms. The next generation maps will conform to the open standard NDS and will allow frequent partial updates from the 'map-content-cloud'.

The following listing indicates features that are enabled by the technologies above; note that the list is by no means exhaustive.

Enabling technologies								
Smartphone platform electronics								
Camera electronics & computer vision								
Advanced positioning technology								
DAB electronics and TPEG technology								
USB and Wifi connectivity								
Wifi-p connectivity								
Cellular connectivity								
Standard/modular run-time map formats								
Features								
Software innovations from smart phones								V
3rd party apps, validated for in-car usage								V
Object/pedestrian detection							V	
Collision avoidance			V				V	
Virtual/Augmented reality							V	
Map content enhancement							V	
Traffic sign recognition							V	
Lane guidance							V	
Advanced driver assistance	V		V			V	V	
Pay as you drive						V		
Advanced map content services						V		
High quality audio					V			
Free-to-air traffic information					V			
Map content provisioning					V			
Interoperability of devices in the vehicle				V				
Shockwave damping			V					
Free to air car to car content sharing			V					
Cooperate driving between vehicles			V					
Platooning			V					
Traffic light interaction			V					
Headway control			V					
Tethering			V					
Online internet connection		V						
Smart mobility app market access		V						
Hybrid navigation	V	V						
Over the air software & content updates		V						
Map update services		V						
Phone		V						
Fresh maps (up to date content)	V				V			

Table: Enabling technologies and features for the CMD concept

Decoupling the UI

From the previous paragraph it becomes obvious that different versions of the CMD concept will exist, each with its own combination of technologies, depending on feature demand, system cost and technological maturity. This implies that hardware encapsulating these components should be highly adaptable while preserving high volume cost advantages of the components. Furthermore, the positioning of this hardware in the car lay out will depend on the enabling technologies at hand: when camera technology is included a windscreen mounting is essential, for connection to the vehicle CAN databus an in-dash mounting makes more sense. This will prohibit a cost effective combination of the display and the processing unit(s) of the CMD, and hence the separation of these two –at least at the conceptual level- is key to the concept. Note that there can be actual implementations of the CMD concept in which this separation is invisible, for example when an existing smart phone or a next generation PND would be used as the single integrated hardware platform.

Apps

Another key concept of the CMD is its ability to run applications from various service providers. These applications resemble apps as currently deployed on tablets and smart phones although in general there will be more content exchange between the app running on the client -the CMD- and the back office of the service provider; an example would be the 'speed-to-green' advice mentioned earlier, requiring frequent and real time status updates.
Whether the majority of the data processing in an application will reside within the client or in the back office remains a matter of choice: probably it is best to set up architectures is such a way that both extremes can be covered.

To allow independent parties to effectively develop apps for the CMD platform an extensive and freely available SDK should be available, preferably including simulation tools to predict real life situations.
What differentiates these smart mobility apps from currently known apps is that there will be some specific requirements and restrictions related to the in vehicle usage; furthermore they will have to adhere to strict interface definitions to allow a proper interaction with other applications like route and trip planners. This means that a validation phase should be part of the introduction process of an application and also that these apps will be most probably be developed by professional organizations with knowledge of the automotive / traffic domain.

Integrated UI concept

Further adding to the challenge for the app developer is the requirement for the integrated, seamless user experience that is crucial to the CMD concept: This will have to be covered by the 'shell-ware' on the CMD: the software environment that controls the activity, visibility and accessibility of the various mobility apps that are –or could be- running at the same time. This 'shell-ware' should also control the way in which advice is forwarded to the driver: advice on optimum or maximum speed, lane guidance, headway control, safety alerts and so on.
This forms a major human factors and HMI design challenge with the added complexity that it should be able to interact with various applications, each providing its own inputs and advices to the driver. Some form of standardization and best practices definition is surely needed in this area.
It is expected that in the future driver state estimation software will become an important component to assist in the task of prioritizing and filtering the various inputs to the driver.

High level requirements

After having established the key concept, the technological enablers and the features of the CMD the high level requirements are listed in this paragraph:

Affordable –	for the ‘average’ consumer, hence the product pricing range should reflect typical CE products, not high end car option products
Adaptable -	the product should allow for iterative adaptations that improve quality, stability and robustness
Flexible -	it should be possible to apply various hardware, software and functional configurations based on a single platform architecture
Open -	multiple vendors and service providers should be able to use the platform standards
Scalable -	the concept should be relatively easily scalable to millions of users
Acceptable –	the advice and the way it is presented should be acceptable for the end user, not enforcing or prohibitive

CMD implementation variants

The CMD concept can be implemented in various ways, each offering a subset of the enablers and hence functionalities as mentioned before. These implementation variants are characterized mainly by the adopted hardware topology and this topology depends largely on two dimensions:

- Device type: Nomadic device combined with in-car hardware (allowing retrofit solutions) or exclusively in-car hardware (mostly OEM solutions)
- Enabling technologies: basic (GPS, GSM, USB, Wifi, Bluetooth), WiFi-p (short range connectivity), CAN (connectivity to the vehicle CAN databus), Camera.

The diagram below shows several implementation variants (not exhaustive) along these two dimensions and it is obvious that many more combinations can be considered. When combining this hardware variety with a diversity of application software packages the result will be a multitude of actual implementations. This again stresses the importance of the six high level requirements mentioned earlier to ensure a steady growing deployment of the CMD concept.

This multitude of potential implementation variants requires that the CMD concept should be defined primarily in a pre-competitive effort to ensure interoperability at the later stages when commercial offerings from competing industrial players will reach the market. A well communicated and transparent boundary between the competitive and the pre-competitive part of the platform development will remain an important point of attention in the near future.

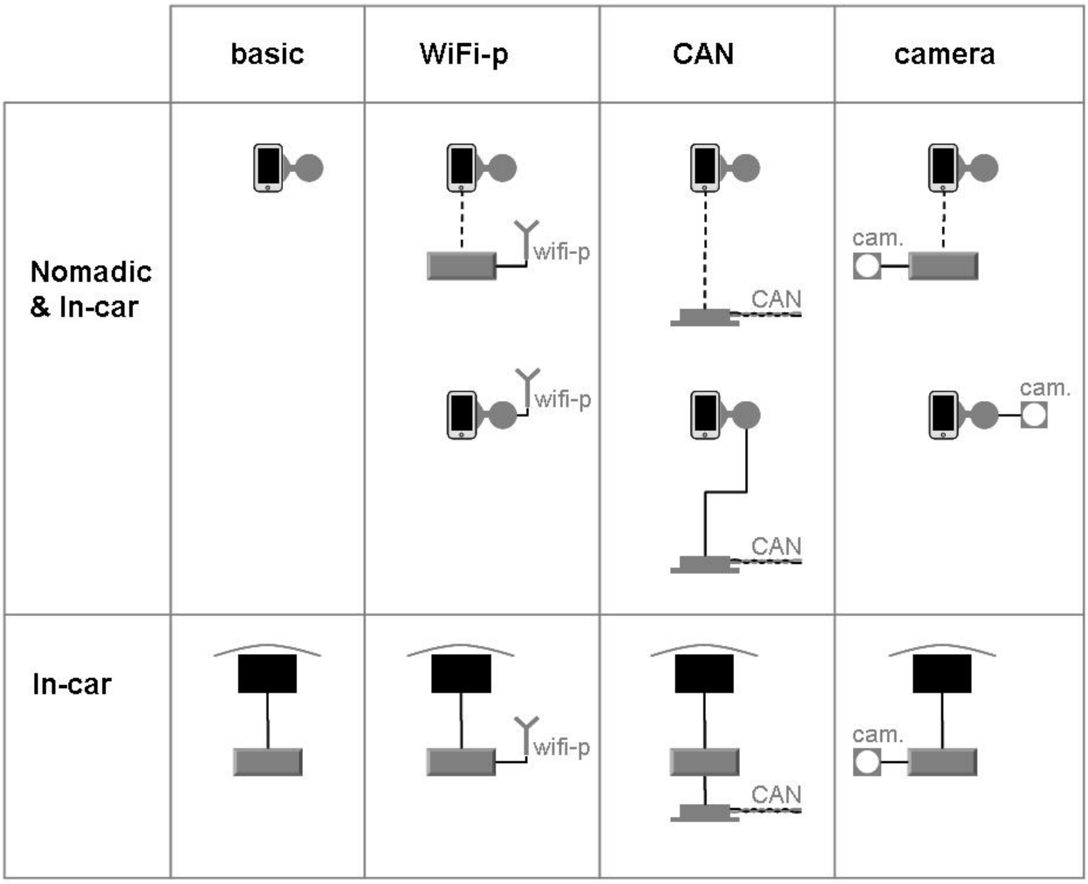


Diagram: Some CMD implementation variants (not exhaustive)

Note : smartphone can also be tablet or PND

Cost levels

An important requirement for the CMD is affordability as this is one of the main obstacles to large scale deployment of advanced mobility services. The interpretation of affordability is of course depending on the scope and functionality of the system (and of the services it supports). Considering the fact that the basic CMD functionality can be achieved using software on a high end smart phone or high end satnav device it seems realistic that the lower consumer cost level for the CMD concept will be in the EUR 250-400 bracket. For the enhanced functionalities, requiring dedicated hardware components and potentially additional vehicle installation efforts a doubling of these amounts can be expected; considering the fact that multiple applications and services can be supported with a single platform this cost level contrasts in a positive way against the current list prices of single application systems on OEM pricelists. The cost levels for the integrated in-car systems will be highly dependent on the technical and commercial choices made by the car manufacturer.
Note that the adopted business models can handle these costs in different ways: for example either by allocating these costs completely to the end user or by adopting a subscription model with a service provider absorbing the initial investments.

CMD pod

Car integration challenge

As indicated earlier in this paper a major challenge is to find a way to actually provide all the promising mobility services to the in-car user; how to get to the 'functionality-in-a-box' solution while a part of this functionality is depending on vehicle residing elements such as WiFi-p antenna's, camera or CAN-data?
This excludes the straightforward adoption of a consumer-device-only solution: The chance that smart phones for example will show interoperability with the vehicle CAN infrastructure seems remote in de midterm future, and integration of advanced vision technology in nomadic PND's will be difficult. As indicated earlier full integration in the original dashboard layout is not an option either, as well from a hardware as from a software point of view.

A solution for this apparent deadlock is to combine in an innovative way the elements 'functionality-in-a-box', the UI/processing division and the consumer device performance/price ratio with its high user acceptance and take-up rate: The CMD-Pod is born.

CMD Pod concept

The CMD-Pod is an actual device within the framework of the overall CMD concept: it is the device that allows

- Decoupling of UI and processing functionality
- Decoupling of vehicle related hardware and nomadic devices
- Encapsulating/integrating enabling technologies as options

Hence the CMD-Pod may consist of the following building blocks:

- Processing unit & memory
- Software for app installation, handling and usage
- Software for external UI control
- Microphone and speaker(s)
- Antenna's for GSM, WiFi, Bluetooth, WiFi-p
- SIM card
- DAB receiver
- Integrated or separately connected camera
- Interface towards vehicle CAN bus
- USB interface(s) for extension hardware
- Mechanical mount for consumer electronics device
- Mechanical fixture to the vehicle

Depending on the anticipated combination of the CMD-Pod with other nomadic or in-car devices and the desired functionality some of the building blocks in the list above can be omitted.

Mechanical integration

As for the mechanical integration of the CMD-Pod in the vehicle the following four basic options exist, primarily linked to its location in the vehicle:

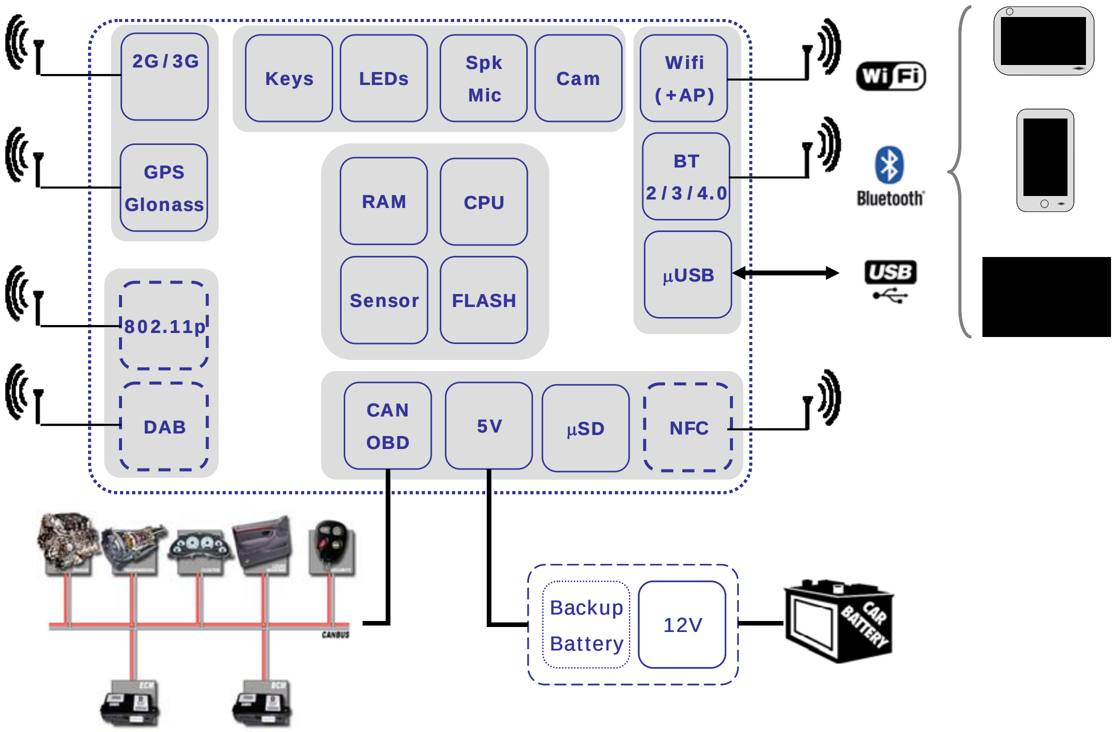
Option	Location	Key driver(s)	Issues
A	Dashboard top	Integrated Camera CE device mount	Universal dash fixture problematic
B	Lower windscreen	Integrated Camera CE device mount	Camera calibration problematic
C	Rearview mirror	Integrated Camera	Retrofit wiring
D	Concealed	CAN connectivity Clean dash	No or external camera External antenna's

The CMD-Pod can retrieve its power from the standard accessory wiring or from the cigarette lighter interface. The addition of a simple separate display (attached to the CE device mount) upgrades the CMD-Pod to a complete CMD system in case there is no nomadic device or in-car display available. To be able to combine a wide variety of CE nomadic devices there probably will be a well defined mechanical interface in the CMD-Pod design to separate the basic elements of the CMD-Pod with the part that takes care of the CE device mounting. As an alternative there could be two separate mechanical products (POD and mounting) with a wired USB link or a wireless WiFi/Bluetooth link between the POD and the device containing the display, using a remote display app.

Architecture

As for the architectural issues of the CMD-Pod there is no need to go into much detail in the scope of this white paper; only a high level electrical architecture is shown in the figure below to indicate the standard and optional building blocks within the CMD-Pod and to illustrate the separation between the processing and the UI part of the CMD concept.

The software architecture must support a distributed system approach, allowing every client-server application to adopt the optimum division of tasks between client and server. This optimum may change in the future when requirements, bandwidth and processing power evolve. There should be middleware software to handle over-the-air updates of software, data and maps in a robust and efficient way with minimum user involvement. On the UI layer a software framework is needed to integrate the inputs from the various apps into a seamless user experience, with a separate module to manage driver attention load. On the lower level of the software preferably a widespread operating system with a large developer community should be used (e.g. Android).
The detailed architecture will be defined in a next phase of development, taking into account the most recent developments in the various international CE, software and ITS markets.



Electrical architecture CMD-Pod

Towards deployment

Deployment challenges

The key challenges for deploying the CMD concept (including the CMD-Pod) apart from the generic risks of new product development and introduction are the following:

- Market acceptance of a system that requires some form of installation effort into the vehicle for its premium features; this can be circumvented by first introducing a basic version without CAN integration (and possibly camera)
- Choosing the appropriate standards and platform technologies for worldwide adoption in a market that is still developing
- Overcoming the 'chicken and egg' dilemma of available services depending on system penetration and vice versa, with the need to simultaneously develop or adapt business models for all partners and stakeholders involved.

From the above it is obvious that there is no 'silver bullet' solution that can be designed in a single step but that an iterative development and deployment path should be followed in which the features, the services and the penetration levels will grow over time. This is reflected in the proposed phased introduction described later on.

Competencies & partners

To be able to respond to the challenges above at least the following competencies must be combined in a pre-competitive partnership:

- Hardware development
- Industrial design
- Module & knowledge suppliers for the enabling technologies
- Vehicle networks know how
- Software development
- App development
- System Integration
- UI design and development
- Testing and piloting
- Mid-High volume industrialization, marketing and distribution
- Service deployment
- Billing processes

Note that the focus in this list is on the client side, whereas the server side competencies are listed in the other white paper (ref. 'scope'). In a next phase appropriate partners can be invited to join the initiative.

Open innovation & IP

One of the issues requiring special attention in a pre-competitive setting as described above is Intellectual Property (IP). For the information exchange within the consortium activities some proven approaches exist that distinguish between background and foreground knowledge although these approaches are mostly used in research project environments and less in development projects. Probably a more difficult issue is the use of IP-fenced technological building blocks from various partners within or outside the consortium. It is obvious that such building blocks will be (re-)used to the advantage of the overall system, but this should be done without barring entry from other parties. The stakeholders representing the final customers and users (including governmental bodies) can play a critical and neutral role to prevent this from happening.

Business considerations

The CMD is not a standalone product but a part of a larger system comprised of public and private back offices, road side systems, data streams and services. Hence the business case for the CMD is not as straightforward as for other consumer targeted products. As a result there is also a necessity for prototyping and development of the business models in parallel with the technical developments, especially within the living lab phase (ref. next section). As an example a 'zero-investment-annual subscription fee model' could be used when an insurance company is driving the implementation whereas a 'lump-sum payment at purchase with 3 years of free services model' could be offered when a service provider is leading. Although there is quite some research documentation and some limited market experience it can not yet be predicted what will become the most attractive business models for mobility services based on the CMD concept. This is an additional complexity during the development phases of the CMD.
However, an additional opportunity exists for companies that are well equipped to develop and market devices in small to medium volumes as there will be a growing market for affordable platforms that can be used (with some project specific adaptations) in the various Field Operational Trials (FOT's) that run all

over the world in research and pilot projects. Currently these projects often use their own specific platforms with higher costs, higher lead times and lower quality.

Phased introduction

Three phases can be distinguished from the preliminary developments up to the final high volume roll out of products and services: Try-out, Living Lab and Roll out, each with an increased level of validation and volume (and penetration rate) as indicated in the table below.
However, this should not be regarded as a linear waterfall timing schedule as there will be parallel activities in all three phases depending on the level of features or technologies; for example more advanced cooperative functions will be developed in EU collaborative research projects while at the same time basic versions of the system are already in the roll out phase.
The transition from try-out towards the roll out phase will be fully embedded within the living lab facilities and knowledge of DITCM (Dutch Integrated Test site for Cooperative Mobility) in which most relevant Dutch partners are already participating. The CMD platform will become a core facility within the DITCM innovation line P2, development environment.

Phase	Deliverable	Volume	Carrier projects	Timing
Try-out	Proof of Concept	<100	BB SRE in car3 (basic functions)	2013 Q1-Q3
			RWS_'innovatief inkopen' (extended functions)	2013 Q2→
			TKI/STW projects EU projects (advanced functions	2013- 2016
Living Lab	Large scale user experience	<100.000	BB informed travelers (within NL)	2013 Q4→
			EU cars2020 projects (within EU)	2014→
Roll out	High volume adoption of products and services	>100.000	Public-Private Partnership driven, commercially viable projects	2014

Deployment phases (BB=Beter Benutten, Dutch governmental initiative)

Note that again here the focus is on the client side (in car) development; in parallel the server side developments must be secured including services development and open data exchange. For a part these activities can be part of other projects not mentioned in the table above.

Outlook

It is anticipated that with the combined competencies from various Dutch partners a CMD could be developed, validated and marketed that will deliver a breakthrough in unleashing the potential of in-car smart mobility services.
The necessity of mobility innovations in the dense Dutch road network, the positive experiences in triple helix collaboration and the absence of dominating passenger car manufacturers put the Dutch partners in pole position for the first lap in the race for international market leadership in the area of mobility solutions. The time to move forward is now, and we must do it together.