UNIVERSITÀ DI PISA

Industrial Applications

The User Mobility Profile Project Specification

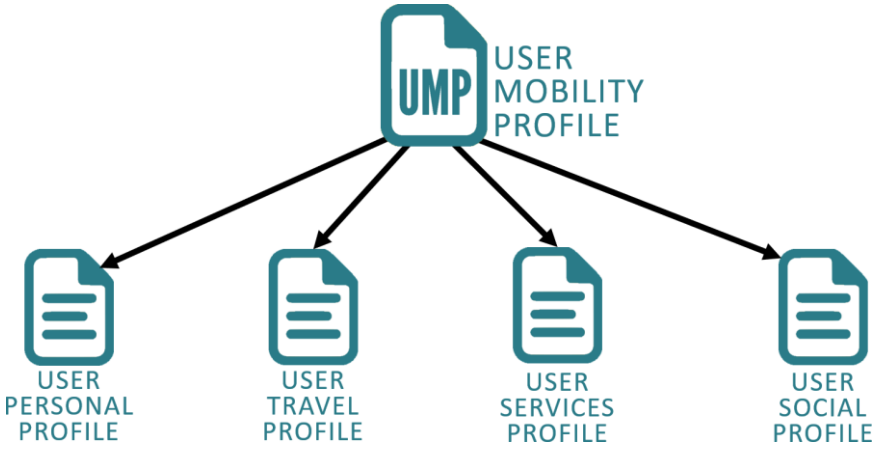
Table of contents

Introduction

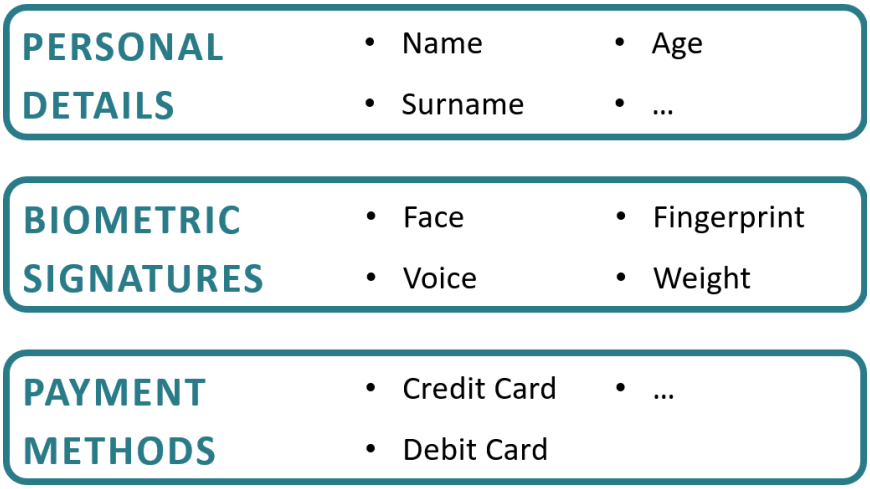
The following represents the preliminary specification document of a distributed platform- and operating system-independent profiling system for passengers of autonomous vehicles termed *"The User Mobility Profile (UMP)"*, a name that is also used to refer to the collection of information and preferences associated with a passenger of an autonomous vehicles and the corresponding data structure.

UMP Composition

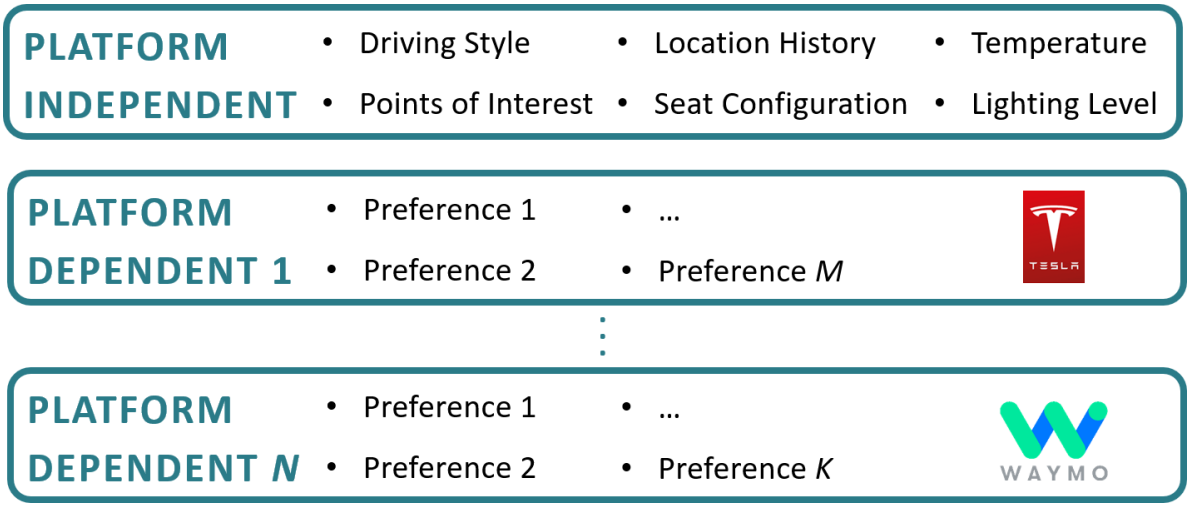
The User Mobility Profile was envisioned as a modular and extensible data structure composed of the following modules:



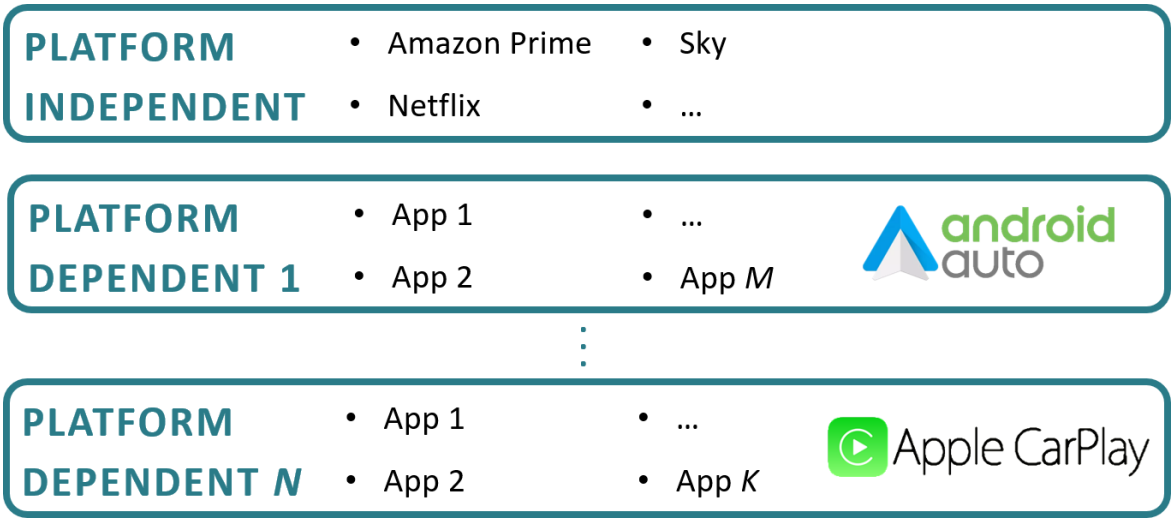
- The *User Personal Profile (UPP)* module contains all user personal information that does not belong to another category, such as his personal details, biometric features, and will be used for user identification purposes as thoroughly discussed later as well as payment methods.



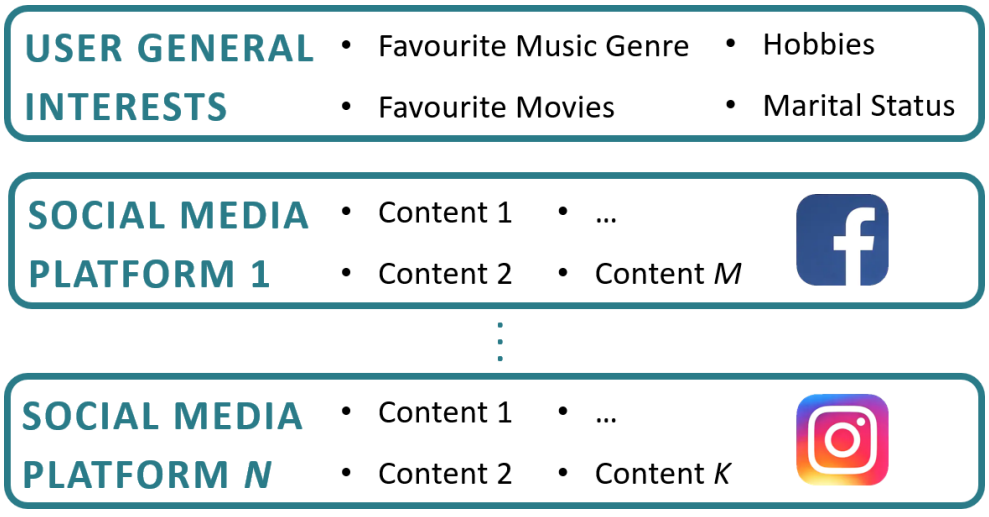
- The *User Travel Profile (UTP)* module holds all the user travelling and ambience preferences, and is divided into a platform-independent set, whose information is applicable regardless of the vehicle the user is currently travelling in, and in multiple platform-dependent sets, whose information is applicable instead only to specific platforms, which may consist in a particular vehicle, operating system or hardware architecture.



- The *User Services Profile (USeP)* module contains the list of all the applications and services the user is subscribed to, and again is divided into a platform-independent and platform-dependent sets, where each of the platform-dependent applications also specifies a download URL from which the corresponding app can be retrieved.

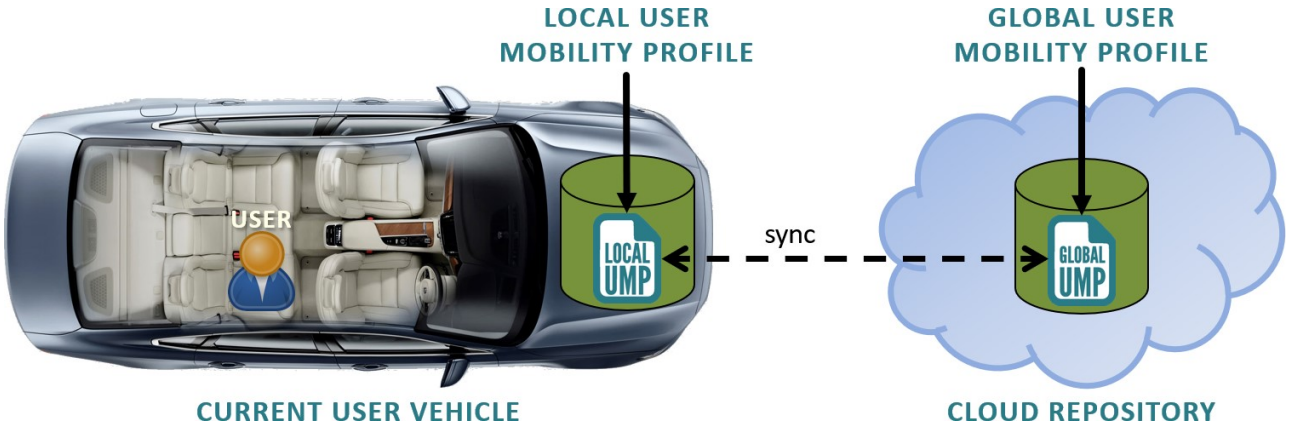


- The *User Social Profile (USoP)* module holds a collection of the user general interests and preferences as well as a selection of his or her social media contents, allowing them to be shared in the car environment with the purpose of enhancing sociality with the other passengers.



Global and Local UMP

In terms of its use within the system, the UMP data structure is differentiated into a *Global UMP*, representing the entire collection of user information and preferences, which as will be discussed later is stored in a dedicated cloud service, and a *Local UMP*, holding the subset of the information contained in its respective Global UMP that is applicable within the specific car environment the user currently occupies, a collection that is stored in a module deployed within the car called *User Mobility Profile Manager (UMPM)*, whose details again will be discussed later.



Functional Description

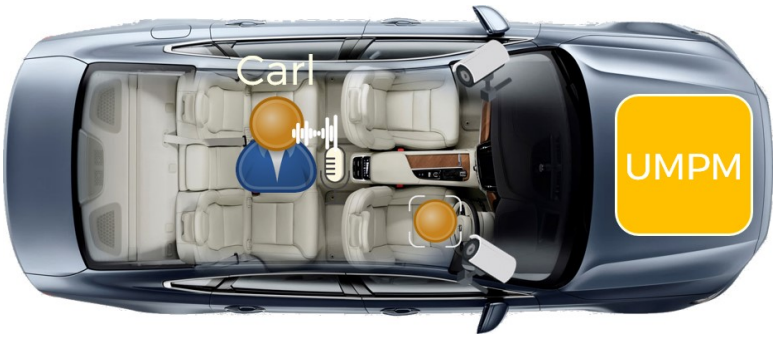
A high-level description of the functionalities offered the system is presented below:

Premises

- The pre-existing user global UMP is stored in a dedicated cloud repository, where the user can initialize its contents and permissions through a web interface.
- The car is equipped with biometric sensors, such as face-recognition cameras, microphones and more, that continuously scan the passenger compartment for user biometric input.

1) User Biometric Features Sampling

Whether as part of an explicit authentication process when entering the vehicle or shortly thereafter, due to the continuous scanning of the biometric sensors, one or more of the user biometric features is collected and sent to a local *User Mobility Profile Manager (UMPM)* module.

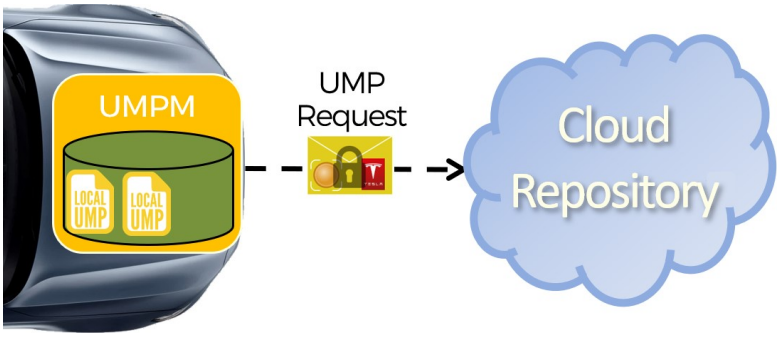


2) User Biometric Features Sampling

Within the UMPM the sampled biometric features are matched against the corresponding features found in each of the UMPs currently stored in the module, with the purpose of identifying the user and matching such input, and the possibly associated command, with his or her set of information and preferences.

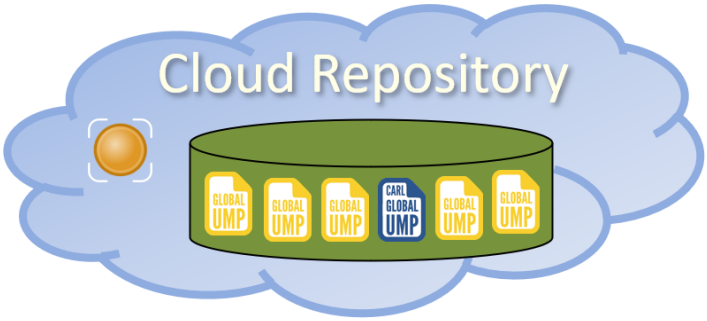
3) Forward UMP Request to the Cloud

If a match for the biometric input was not found in the local UMPs, a temporary profile for such user is created and a request for retrieving the user UMP containing the biometric features and information on the car platform is forwarded to the cloud repository.



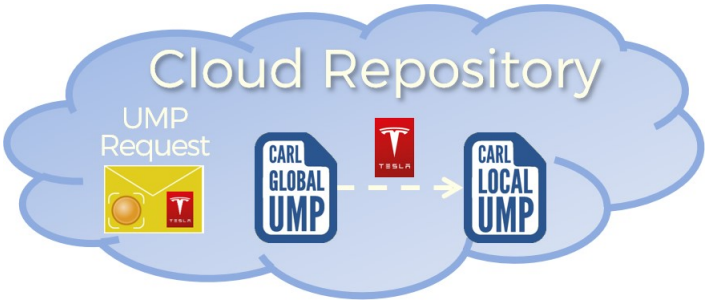
4) User Cloud Biometric Identification

Within the cloud repository, similarly to what was performed in the car environment, the user biometric features are checked against the corresponding features stored in the entire collection of UMPs until a match is found.



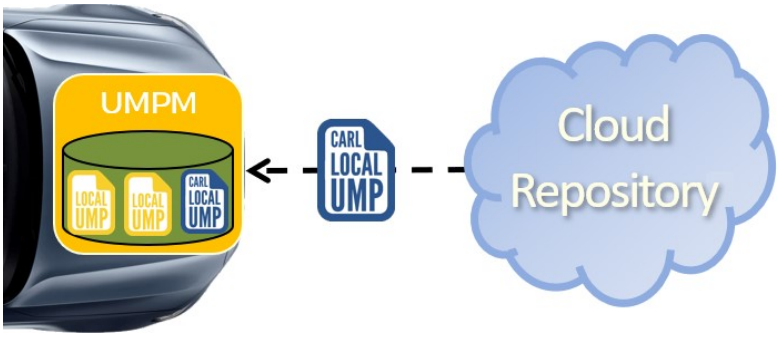
5) Creation of the Local UMP

Once the user has been identified within the cloud, their local user mobility profile is crafted from its global profile by discarding all platform-dependent information that is not applicable in the vehicle currently occupied by the user.



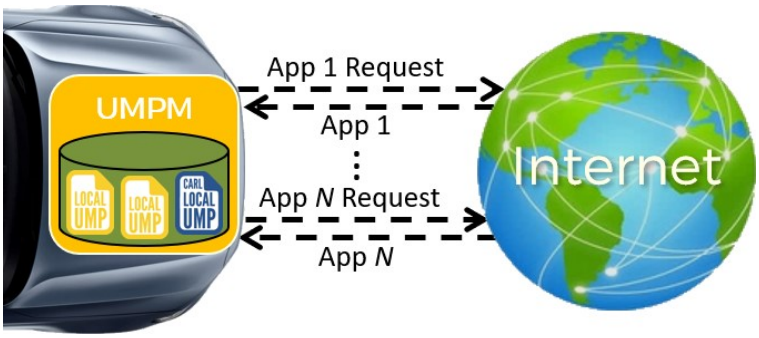
6) Returning the Local UMP

After its assembly, the user local mobility profile is returned to the requesting UMPM for it to be used within the car local environment.



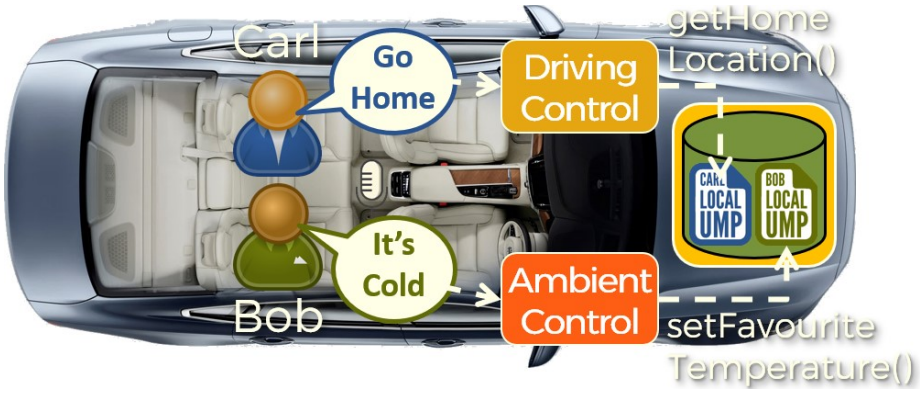
7) Retrieval of the User Applications

As a new local profile is loaded within the UMPM, depending on the user and/or the car policies, each or in general a subset of the applications specified within their user services profile, along with their configuration, is retrieved from the internet and installed in the local environment.



8) Using the User Mobility Profile

In addition to biometric identification purposes the UMPM module enables the applications running in the local environment, given the appropriate permissions, to read and update the contents of the passengers' local profiles, allowing for their self-configuration according to the users' preferences.



9) Synching the User Mobility Profile

Once the user leaves the vehicle, either explicitly or implicitly if their biometric features are not sampled within a given time interval, the information in their local UMP that was modified or added is sent to the cloud repository, where it will be merged with the user's global mobility profile.

Requirements Definition

Functional Requirements

- The system shall define a modular data structure, to be named “User Mobility Profile (UMP)”, for holding the set of information associated with passengers of autonomous vehicles, which must be organized into the following modules:
    - A "User Personal Profile (UPP)" module, holding the set of user personal information such as name, surname, age, address, gender, payment methods, as well as a set of biometric features including but not limited to their iris, face and speech patterns.
    - A “User Travel Profile (UTP)” module, containing all user travelling and ambience preferences, which shall be organized on the one hand into a platform-independent set, including information such as seat configuration, preferred travelling style, ambient temperature, lighting level, location travelling history and points of interests, and on the other in a set of platform-dependent collections, containing information specific to a certain platform, which may consist in a vehicle, operating system and/or furniture models.
    - A “User Services Profile (USeP)” module, containing the list of applications and services the user is subscribed to, again to be organized into platform-independent and platform-dependent sets, where each application must include a download link for its retrieval.
    - A “User Social Profile (USoP)” module, holding a collection of the user general interests and preferences as well as a selection of social media contents.

- An Application Programming Interface (API) for accessing and managing the information contained in the User Mobility Profile must be defined that shall include the following functionalities:
    - The matching probability of the comparison between an input and the corresponding user biometric feature.
    - The possibility of creating, reading, updating and deleting information according to a hierarchy of permissions.

- Depending on the context in which it is employed, the User Mobility Profile shall be differentiated into a *“Global UMP”* and a *“Local UMP”*, with the former representing the entire collection of user information and preferences, to be stored in a dedicated cloud service and the latter comprising the subset of information contained in the related Global UMP that is applicable within the car environment the user currently occupies, to be stored in a local car module named *User Mobility Profile Manager (UMPM)*.

- A module for managing the set of user mobility profiles of the current passengers of an autonomous vehicle, to be named *“User Mobility Profile Manager (UMPM)”*, shall be defined, and must include the following features:
    - A database for storing the user mobility profiles of the current passengers.
    - The association between the biometric inputs being passed from the car operating system to the corresponding source user, and thus their set of information and preferences.
    - If a biometric input could not be associated with any of the local UMPs, the creation of a temporary profile associated with such biometric feature and the forwarding towards the dedicated cloud service of a “UMP Request Message”, containing such biometric input and a set of information identifying the current car platform.

    - The merging of the local UMPs received from the cloud service with their associated temporary profiles, along with requesting from the operating system the retrieval and installation of the app specified in the newly retrieved User Services Profile (USP).
    - Allowing third-party applications to create, read, update and delete information from the stored UMPs according to a hierarchy of permissions.
    - The upload towards the cloud service of the UMP information that was added or modified since its retrieval once a passenger leaves the car.

- A Cloud Infrastructure hosting a service supporting the distribution and synchronization of the user mobility profiles must be defined, which shall present the following features and functionalities:
    - A database for storing the users' global mobility profiles.
    - For every "UMP Request Message" received from a User Mobility Profile Manager (UMPM) deployed within a vehicle, the biometric feature included in the message must be matched against the entire set of global UMPs to find its corresponding user, after which their local UMP must be created using the car platform information included in the message and returned to the requesting UMPM.
    - Once an updated local UMP is received from a UMPM, the cloud service must merge updated information with the corresponding user's global mobility profile.

- A Web Service allowing users to initialize and manage their user mobility profiles must be deployed, which must include a user interface offering the following functionalities:
    - Initialize, browse, update and delete the information contained in their user mobility profile.
    - Define a hierarchy of permissions for third-party services in accessing and modifying information in their UMP.

Non-Functional Requirements

- ### Timeliness
  Since the passengers' interactions with the services offered by the car ecosystems are strictly tied to the functionalities provided on the one hand by the local UMPM module and on the other by the supporting cloud service, the latency of their operations must be minimized to prevent a degradation of the Quality of Service (QoS) experienced by the end-users, with the overall distributed service attuning to the definition of soft real-time system.
  In particular, the matching between the input biometric features with the ones contained in the users' UMPs should be performed through hardware-accelerated machine-learning algorithms.

- ### Security
  Since the system manages sensitive user information, both internally in the car environment and within the cloud service, the best security practices must be in place to enforce the privacy and protection of the user data, with particular attention to the information held within the User Personal Profiles (UPP) and in the communications over the internet between the UMPMs and the cloud service.

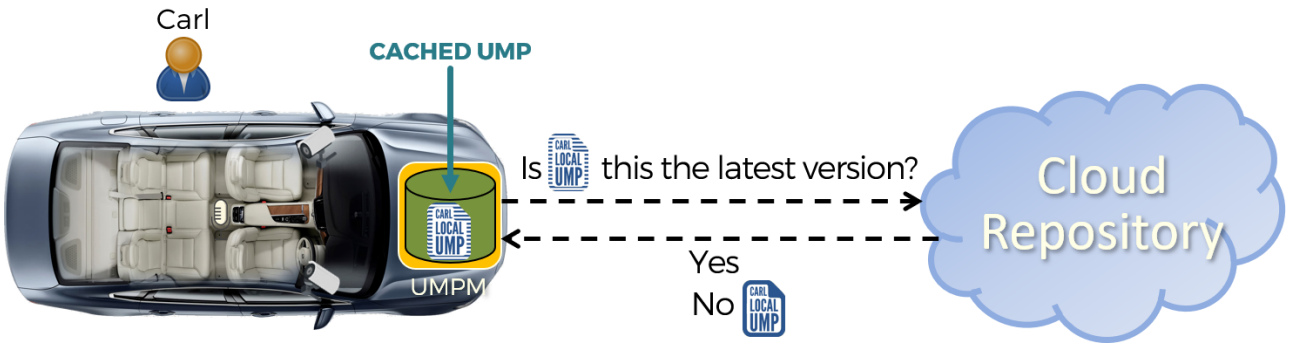
- ### Portability
  The User Mobility Profile data structure and the User Mobility Profile Manager (UMPM) module must be designed to be both operating system and hardware independent, with implementations requiring minimal effort to be ported to different car platforms and architectures.

- Scalability
  The cloud infrastructure servicing the UMPMs request must be capable of horizontally scaling, in terms of both throughput and latency, according to the projected user adoption, which must be estimated in the later stages of the development process.

- *BASE* Requirements
  The system must be developed attuning to the requirements expressed by the *BASE* paradigm, as outlined below:
  - *Basically Available* — The system must present a high degree of availability, with particular reference to the services offered by the cloud infrastructure (>99%), which can be obtained by exploiting data redundancy and horizontal scalability as previously discussed.
  - *Soft State* — The data handled by the system is subject to continual updates to more recent versions.
  - *Eventual Consistency* — The data handled by the system will eventually converge to its latest version.

Meeting the *BASE* Requirements

In order to meet the *BASE* requirements, when passengers leave a car their UMPs, and depending on the car policies their applications and configurations, are temporarily cached locally for a possible later reuse should users take another ride in the same car.
Hence, when a user enters a car, if his or her UMP is found in the local cache it is made immediately available for use, and a request checking for an updated version is forwarded towards the cloud service.



Project Development Risks Assessment

Technological Risks

1) Coding language

The language of choice is Python. It adapts well to every need thanks to the presence of numerous libraries and being one of the most used languages now it is constantly updated. New libraries and therefore new features are added almost every day making it a multi-purpose, effective and well-structured language. The possibility that a better and more 'powerful' language emerges has been evaluated and this would require a rewrite of the prototype code if the performance were better. Furthermore, python currently remains the best language for machine learning and all that follows, thanks to the presence of well-stocked libraries dedicated to this purpose. The code rewrite problem could be mitigated by splitting them into independent modules that communicate with each other via sockets. So, if a better language is found for one of the modules, it would be enough to rewrite only that module in the new language, thus limiting any necessary updates.

Risk Level: Medium to Low Risk Costs: Medium to Low

2) Database Management System (DBMS)

We have chosen to use a NO-SQL type DB due to the need to meet the BASE requirements, in particular MongoDB adapts well to the needs in terms of user information store. Our choice therefore falls on document database technology. Mongo Db is currently the most used and most efficient document database, if a new document DB technology were to come out, the information management bases in the DB would remain the same, as in any case of the document DB type and this feature abstracts from the referring technology, therefore it would be enough only to update the code, from the syntactic point of view, to satisfy the new technology.

Risk Level: Low Risk Costs: Medium to Low

3) Machine Learning Technologies

Machine learning: we have chosen to use deep neural networks, as they are currently the most effective technology, for our purposes. They are reliable, widely used and in constant development and updating. This allows them to remain valid over time. If a new, better technology were to come out, we would have to replace the machine learning module and change the recognition parameters, in any case by dividing it into modules, the possible need for change is limited. The need for change remains minimal as neural networks still have scores very close to 100% accuracy in many cases.

Risk Level: Low Risk Costs: Medium

Non-Technological Risks

1) Legal fragmentation and roadblocks for employing biometric recognition systems

Since at the present day the use of biometric recognition systems for user identification purposes lacks a well-defined regulatory framework from governing authorities, the biometric matching mechanism employed in the project as it has been envisioned could be subject in the future to a set of new constraints and regulations, possibly on a per-country basis, in order to adhere to the local laws.

Risk Level: High Risk Costs: Medium to Low

2) A competing profiling system becomes the de-facto standard in autonomous vehicles

Should a competing service become the de-facto standard for user profiling in autonomous vehicles to the point that employing the proposed system would no longer be convenient for the interested parties, the entire project along with its development costs would be wasted.

Risk Level: Medium Risk Costs: Very High

Project Work Breakdown

Required Professionals

The following is a list of the professionals required for developing the project, consisting on the one hand of the development team members and on the other of a number of external supporting consultants:

Development Team Members	
Professional	**Role**
Software Engineer	High-level software design and implementation
Machine Learning Engineer	Design and implementation of the biometric recognition systems employed in the UMPM and in the cloud service
Data Engineer	Data modelling and communication mechanisms between the different system modules
Embedded Systems Engineer	Implementation of the User Mobility Profile Manager (UMPM) module in the selected car platform(s)
Database Engineer	Design and implementation of the databases used for storing the UMPs, both in the UMPM and in the cloud service
Cloud Systems Engineer	Deployment of the cloud infrastructure hosting the service supporting the distribution and synchronization of the UMPs
Front-end Web Developer	Front-end development of the User Mobility Profile Web Service
Back-end Web Developer	Back-end development of the User Mobility Profile Web Service

External Consultants	
Professional	**Role**
Legal expert on biometric recognition systems	Legal advisor on the employment of biometric recognition systems
Expert in mobile applications development	Counsel in designing and organizing the User Mobility Services Profile (USeP)
Expert in AV operating systems	Counsel for implementing the UMPM in the selected car platform(s)

Technical skills requirement

Development Team Members	
Professional	**Desired Skills**
Software Engineer	• Proficient in C++ • Excellent grasp of fundamental computer science concepts • Experience using common design patterns in the software industry • High standards for code quality, maintainability, and performance
Machine Learning Engineer	• Must have 3-5 years of professional experience with deep learning • Experience productionizing machine learning models • Strong and effective communication with peers and business users • Expert knowledge of at least one programming language, ideally Python
Data Engineer	• Experience coding in Python. • Implement high-quality test-driven code. • Experience working with structured and NOSQL databases. • Identify code quality issues and implement tests to improve future processes. • Translate business requirements into actionable data tasks.
Embedded Systems Engineer	• Proficient in programming/Scripting languages such as C and Python. • Experience developing SPI interface and working with SPI devices that drive/diagnose I/O and flash memory • Test and troubleshoot new and existing systems, including in-car validation • Design appropriate embedded control systems solutions, including integration of control systems hardware and real-time data acquisition networks
Database Engineer	• Manages and tunes the logical and physical database design • Excellent query tuning skills • Excellent MongoDB skills, with the ability to generate complex queries • An extremely detail-oriented work ethic
Cloud Systems Engineer	• Excellent analytical, decision-making and problem-solving skills • Experience in containerization and management tools such as Docker Swarm • An understanding of basic network stack (IPv4, Routing, DNS) • Skills for developing, deploying & debugging cloud applications
Front-end Web Developer	• Mobile development and .net experience • Create amazing UX client-side apps with React • Solid understanding of basic software engineering principles such as data structures and algorithms • Experience developing the front end of engaging, visually appealing web applications
Back-end Web Developer	• Solid understanding of basic software engineering principles such as data structures and algorithms • Experience developing the back end of web applications • Familiarity with RESTful APIs • Have an understanding of code versioning tools.

Work Packages Definition

The projected activities required for developing the system were organized into the following work packages:

Work Package	Description	Development Team Members	External Consultants
Design of the UMP and its API	Detailed design of the UMP data structure and its base API	• Software Engineer • Data Engineer • Machine Learning Engineer	• Legal Expert on Biometric Recognition Systems • Expert in Mobile Applications Development
Cloud Service Development	Design and deployment of the cloud service and infrastructure	• Data Engineer • Database Engineer • Cloud Systems Engineer • Machine Learning Engineer	-
Web Service Development	Design and development of the UMP web service	• Front-end Web Developer • Back-end Web Developer • Data Engineer • Database Engineer	-
UMPM Development	Development of the UMPM on the selected car platform(s)	• Software Engineer • Database Engineer • Machine Learning Engineer • Embedded Systems Engineer	• Expert in AV operating systems

Work Packages Tasks Breakdown

The individual tasks constituting each work package, along with their assigned team members and estimated man*hour cost, are outlined below:

Design of the UMP and its API		
Task	**Assigned Team Members**	**Man*Hour Cost**
Design of the User Personal Profile (UPP)	Software Engineer Machine Learning Engineer	450
Design of the User Travel Profile (UTP)	Software Engineer	400
Design of the User Services Profile (USeP)	Software Engineer	650
Design of the User Social Profile (USoP)	Software Engineer	350
Design of the UMP Base API	Software Engineer Data Engineer	850
	TOTAL	2700

Cloud Service Development		
Task	**Assigned Team Members**	**Man*Hour Cost**
Development of the Database for storing the Global UMPs	Database Engineer	850
Development of the Biometric Features Matching System	Machine Learning Engineer	550
Development of the UMPM Communication API	Data Engineer	700

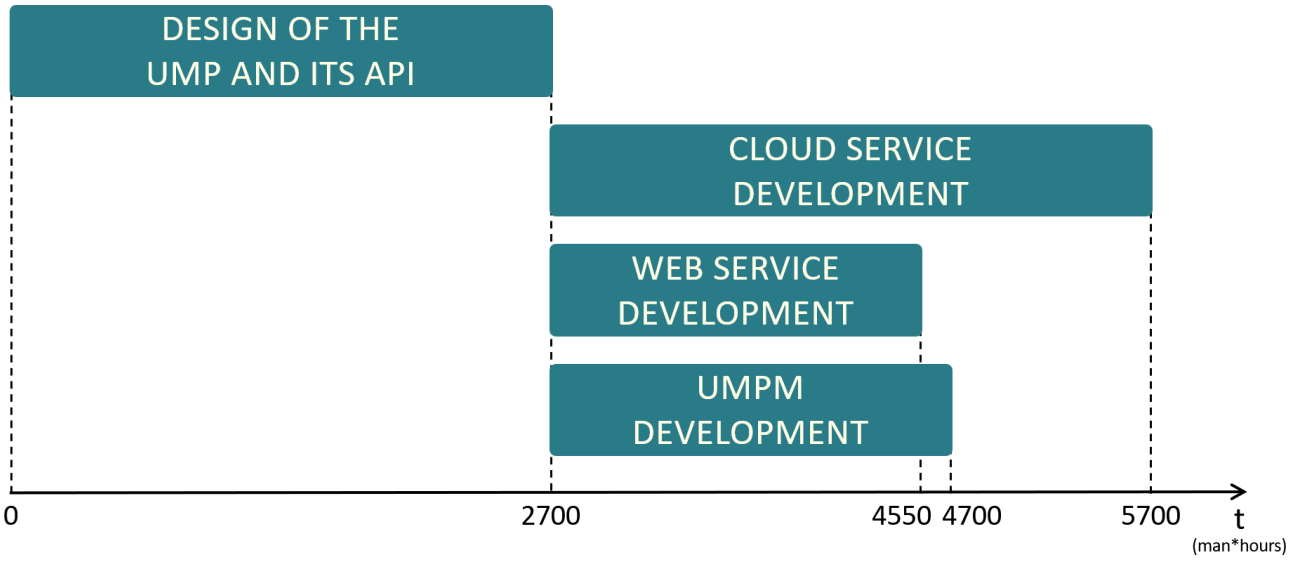
Set up a Content Distribution Network for servicing requests	Cloud Systems Engineer	900
	TOTAL	3000

Web Service Development		
Task	**Assigned Team Members**	**Man*Hour Cost**
Front-end Web Development	Front-end Web Developer	750
Back-end Web Development	Back-end Web Developer Database Engineer	800
Integration with the cloud infrastructure	Data Engineer	300
	TOTAL	1850

UMPM Development		
Task	**Assigned Team Members**	**Man*Hour Cost**
Porting of the UMP to the selected car platform(s)	Embedded Systems Engineer, Machine Learning Engineer, Database Engineer	1200
Integration with the car operating system	Embedded Systems Engineer	500
Integration with the cloud infrastructure	Data Engineer	300
	TOTAL	2000

Project Development Schedule

An estimation of the project development schedule expressed in a man*hours scale is outlined below, where once the initial "Design of the UMP and its base API" work package has been completed the others can be carried out in parallel:

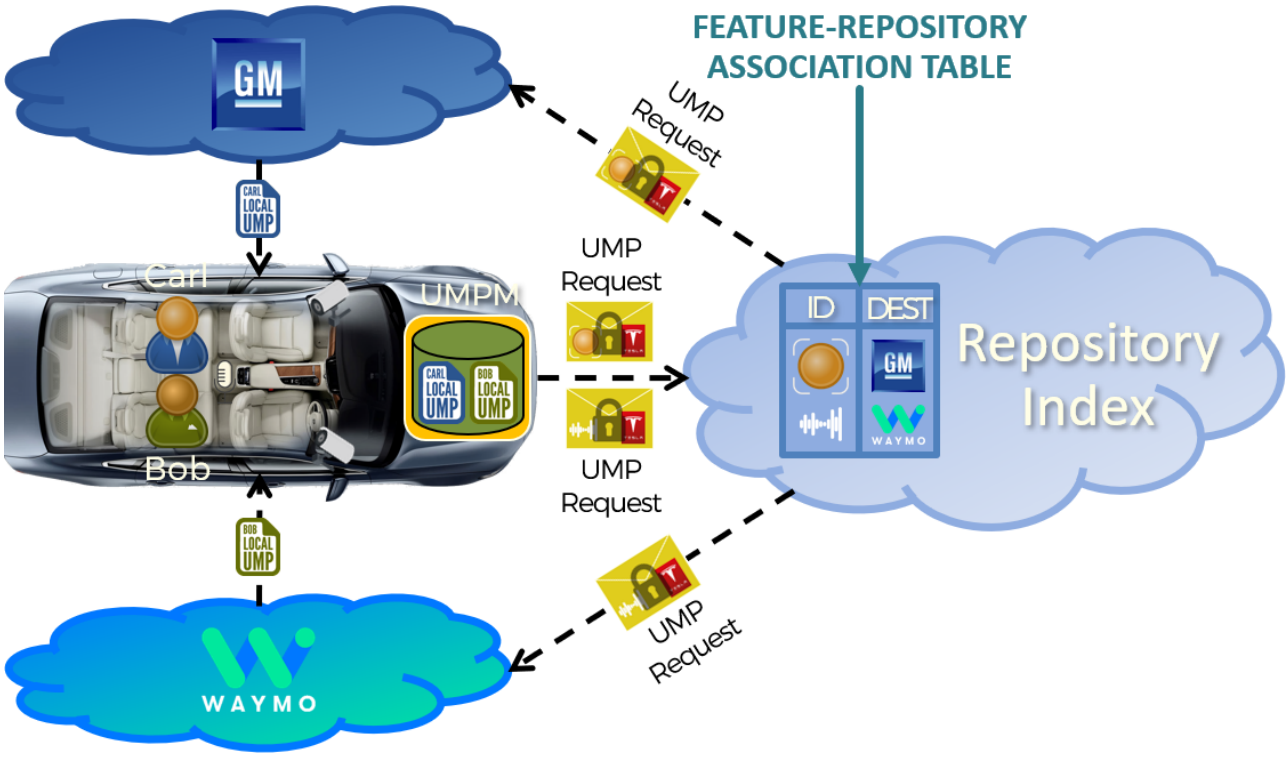


Possible System Extensions

The following features represent possible project extensions whose application is to be evaluated in future system revisions.

Multi-Repository Cloud Service

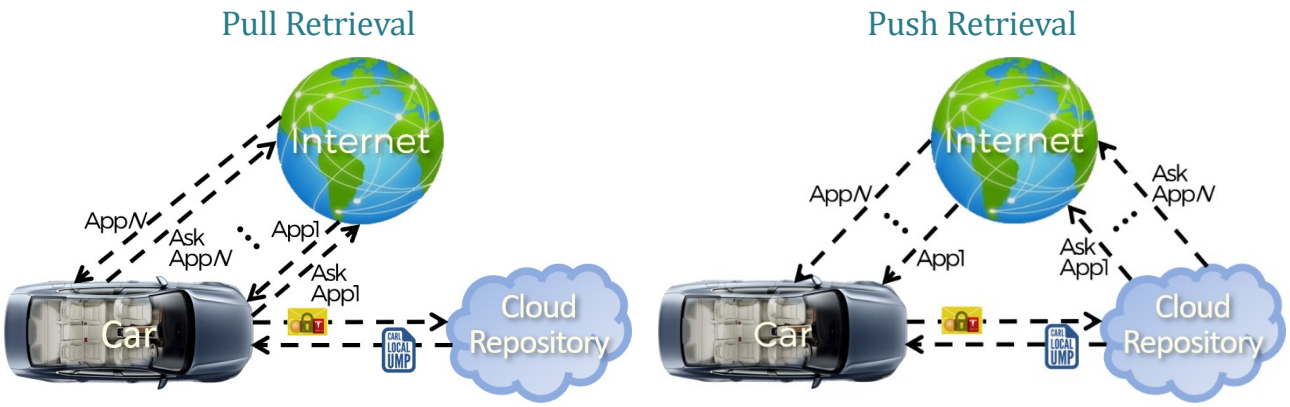
While in the system as it has been envisioned the entire collection of user global mobility profiles is stored within a single repository, as an extension it is also possible to distribute such dataset across different repositories, in a scenario where each car manufacturer has its own cloud service hosting the global UMPs of their customers.
Employing this architecture would require an intermediate step in the Local UMP retrieval consisting in a repository index holding a feature-repository association table, allowing the routing of UMP requests coming from the UMPMs to the repository associated with the biometric features contained in the message, repositories that would in turn return the user local UMP back to the requesting UMPM.



This architecture for retrieving the user UMPs, while presenting the advantage that no single entity or corporation would own the entire dataset of UMPs, would also carry a set of disadvantages, including a higher complexity, possible services interoperability issues as well as a higher latency in retrieving the local UMPs.

Push vs Pull Applications Retrieval

In the system as it has been envisioned, every time a new local UMP is retrieved from the cloud, the car system selects, according to its policies, the set or subset of the user applications to be retrieved and installed in the local environment, thus attuning to a "Pull" paradigm.
An alternative approach for retrieving such applications would instead adopt a "Push" strategy consisting in initiating the application retrieval directly from the cloud service as soon as the user biometric features are matched to their global UMP, as depicted below:



While on the one hand the adoption of a push strategy would reduce the latency in retrieving the applications, on the other it would increase the resource requirements of the car platform in terms of computational capabilities, storage space and transmission bandwidth, as well as the general power consumption, also incurring the risk of useless data overprovisioning should passengers use only a subset of their applications during travel.

Inter-car Communications

One last extension aimed at easing the bandwidth and computational requirements of the cloud service consists in enabling autonomous vehicles to interchange the users' UMPs and possibly their applications and configurations, which can be obtained through the creation and management of ad-hoc inter-car networks: