



General Technology Description

iviLINK

CONFIDENTIAL

20 pages

Table of Contents

1.	General Approach.....	3
1.1.	Scope of the Document.....	3
1.2.	Assumptions and Design Basics	3
1.3.	Task to Solve	4
1.4.	Product Overview	5
1.5.	Glossary.....	5
1.6.	Technology Overview	7
1.6.1.	Applications Level.....	8
1.6.2.	Service Level	8
1.6.3.	Profile Level	8
1.6.4.	Data Channels	8
2.	Integration and Relations to Other Technologies	9
2.1.	General Information.....	9
2.2.	Bluetooth.....	9
2.3.	DLNA™	9
2.4.	AppLink.....	9
2.5.	MirrorLink™, VNC and RTP	10
3.	Feature List of SDK v0.9 (Technology Preview).....	11
4.	Change History	11
5.	Approve History	11

1. General Approach

1.1. Scope of the Document

This document describes the overall goals, design and architecture of the iViLINK (formerly known as AXIS), the technology for seamless multimodal data exchange between mobile and automotive infotainment, navigation and multimedia devices (IVI). iViLINK is the technology, set of the protocols and SDK to enable mobile applications with an access to the automotive hardware and software infrastructure via gateway or IVI device. Being symmetrical, iViLINK supports in-vehicle applications with access to mobile portion of functions and features.

1.2. Assumptions and Design Basics

- iViLINK connectivity technology is prepared to provide unified connectivity approach for mobile and automotive devices. The technology is designed to cover peer-to-peer connectivity scenarios and mostly focused on mobile to head unit and mobile to headrest communication.
- iViLINK is designed to cover multiple usage scenarios for touch screen, simplified and headless automotive devices.
- iViLINK is symmetrical and doesn't limit connectivity functions for both directions (mobile to automotive and automotive to mobile) unless it is done by the SW or system integrator using system build approach or policies.
- iViLINK is provided in a form of Software Development Kit, i.e. a tool of the software applications and firmware development. It lets developers enable their products with additional features to control the remote side and do content sharing, but it is not designed to modify existing systems behaviors.
- iViLINK is created in assumption of provisioning multilevel controllability. OEM or SW manufacturer is capable to manage all the functions of the device even if applications and profile installation is allowed on it.
- iViLINK is oriented on incremental upgrade of the system. Profiles and applications can be added in the runtime. It helps introducing the technology and start promoting it to the market prior to the entire set of features is designed, validated and ready for deployment.
- iViLINK is designed as a secure technology. Encryption, authentication and DRM mechanisms are embedded to the SDK and can be used to create safe and secure environment acceptable for the automotive use.
- iViLINK is not a replacement of the various of other mobile to automotive connectivity technologies. It is designed to act as a shield over the existing content and applications sharing methods and shall unify and standardize the approach to access the remote side for applications.
- iViLINK doesn't define or limit the physical nature of the connection, encapsulating any serial channel provided by the system.
- iViLINK is designed as relatively system independent technology. Reference implementations are provided for a set of the most popular mobile and automotive operating systems.

1.3. Task to Solve

At the current level of the automotive infotainment and mobile market maturity, mobile phones, tablet PCs and other portable electronics have very important value, being used both as everyday application and content consumption devices and sources of the content to be displayed with using of preinstalled vehicle systems.

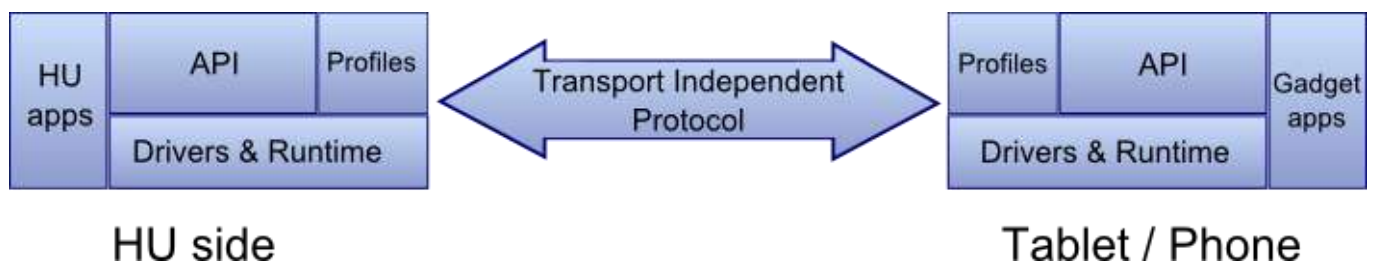
However, actually there are no commonly accepted and fully institutionalized standards of how to perform seamless transfer of multimedia, infotainment and navigation information across the mobile devices. It makes multiple IVI products incompatible to each other and limits its integration to mobile gadgets.

From the other side, multiple technologies reused from other IT areas like DLNA, Bluetooth, VNC etc. can be adopted for in-car usage and successfully applied for media and information transfer. Therefore joining these approaches under one managing framework providing transparent access and seamless usability may provide excellent synergy between the various types of preinstalled IVI and portable mobile devices.

iviLINK solves the problem of two-way symmetrical integration between preinstalled car IVI devices and consumer personal devices.

iviLINK is a portable application development framework that provides secure, OEM controlled two-way connectivity and system abstraction for:

- Screen/Controls access
- Audio / Video sharing
- Internet connectivity sharing
- Telephony services sharing
- Navigation functions sharing
- Remote API access



There are following typical use cases for applying of this technology in a regular conditions of the passenger car:

- Use smartphone as a source of applications for Head Unit
- Use tablet or smartphone as a display to head unit, including screen less and low cost head units such as 1-DIN modules
- Share smartphone screen to the Head Unit (use Head Unit as a screen for remote mobile gadget)
- Synchronize head unit and smartphone/tablet contents
- Share media content to the multiple vehicle and consumer devices including PND and RSE devices
- Integrate remote navigation service
- Integrate rear and side view cameras, including wireless ones

- Integrate aftermarket equipment with minimum or no adoption and wired connectivity
- Update older in-vehicle equipment with new technologies support without impacting of the major behaviors, UI and application level.

1.4. Product Overview

iviLINK is a Software Development Kit enabled with framework and development tooling, helping head unit developers to synchronize data, applications and multimedia with mobile devices.

iviLINK includes:

- Profile oriented protocol (see below), which performs control functions and data exchange
- In band and out of band multimedia data exchange functionality
- Sample applications
- Development toolchain

The iviLINK nativity for the developer depends on the role of the code the developer brings to the entire system.

For system integrator, iviLINK acts as a set of mandatory processes to be launched. These processes support the functioning of the iviLINK data channels management, profiles, services and applications control.

For application or profile developer, iviLINK is a set of libraries to be linked to the application or profile code and providing APIs to access services (for applications) or data channels (for profiles).

For OEMs iviLINK is a set of services and profiles providing these services to be simply selected and configured.

For the end user, iviLINK is an approach for seamless access of the remote resources after pairing and authentication.

1.5. Glossary

Authentication – process of permissions confirmation upon the previously paired devices connected to each other. It assumes exchange of the encrypted key phrases to make sure that the remote side has required permissions proven.

Application – application adopted to iviLINK use by linking with iviLINK libraries and using iviLINK API to register supported services

Carrier –physical transport, any type of serial link supported by iviLINK connection manager. iviLINK supposes to work with TCP/IP, Bluetooth, USB.

Carrier Adapter – entity that encapsulates the low level I/O for given Carrier and performing low level handshake to prepare the data channel and launch authentication or pairing process

Connectivity Agent – entity that provides establishes multiplexed connection over the carrier link

Channel Supervisor – entity providing logical data channels over the connection established by connectivity agent

Frame – atomic element of data transmission, data package

Handshake – process of negotiation that dynamically sets parameters of a carrier established data channel between two devices to help high level modules use reliable serial data link

Logical channel – duplex transmission channel, associated with profile instance

Maximum transmission unit - size (in bytes) of the largest protocol data unit that the layer can pass onwards

Multimedia stream – binary stream of raw data with no types or entities recognizable on iViLINK infrastructure level. Multimedia streams are supported primarily to transfer large amounts of media data in real time

Physical channel – system-provided serial link (RS232 cable, BT SPP, TCP/IP over wired or wireless link, etc.)

System channel – logical channel selected for iViLINK internal data exchange and never provided for use in profiles

Pairing – process started once upon the devices supporting iViLINK and not being paired yet connected to each other. Pairing assumes identification of the connected sides, confirmation of the permissions to access each other and sharing of the RSA public keys for further authentication.

Physical frame – atomic data unit transferred over the physical connection

Profile – named and versioned set of modules (dynamic libraries) with predefined API, responsible for remote method call and streaming content exchange. Profile converts application-level operations in a form of API calls to the process of formatted data exchange via RPC and PAD. Each profile has two role-specific implementations deployed on paired devices and mutually connectable for data exchange according for the data exchange protocol (contract) defined by the given profile specification.

Profile API – named contract, defining the methods how the set of profiles provide the define type of functionality to the requesting application. Profile API contains method specifications and the definition of how these methods to be used. Multiple profiles implementing the same API are polymorph and can implement the same part of services on selective basis.

Profile API repository – database, contain Profile APIs.

Profile implementation – representation of profile for selected role, version and hardware architecture in form of dynamic linked module.

Profile instance – a copy of the profile in the application address space, bind to the corresponded remote side for solving dedicated applied task.

Profile repository – contain profiles (map of profiles). Simple folder with XMLs and subfolders named by platforms.

Profiles update process – process built on top of Profile repository and responsible for profiles updating, uploading and downloading

Protocol data unit – minimal data chunk with atomic transmission property over physical carrier

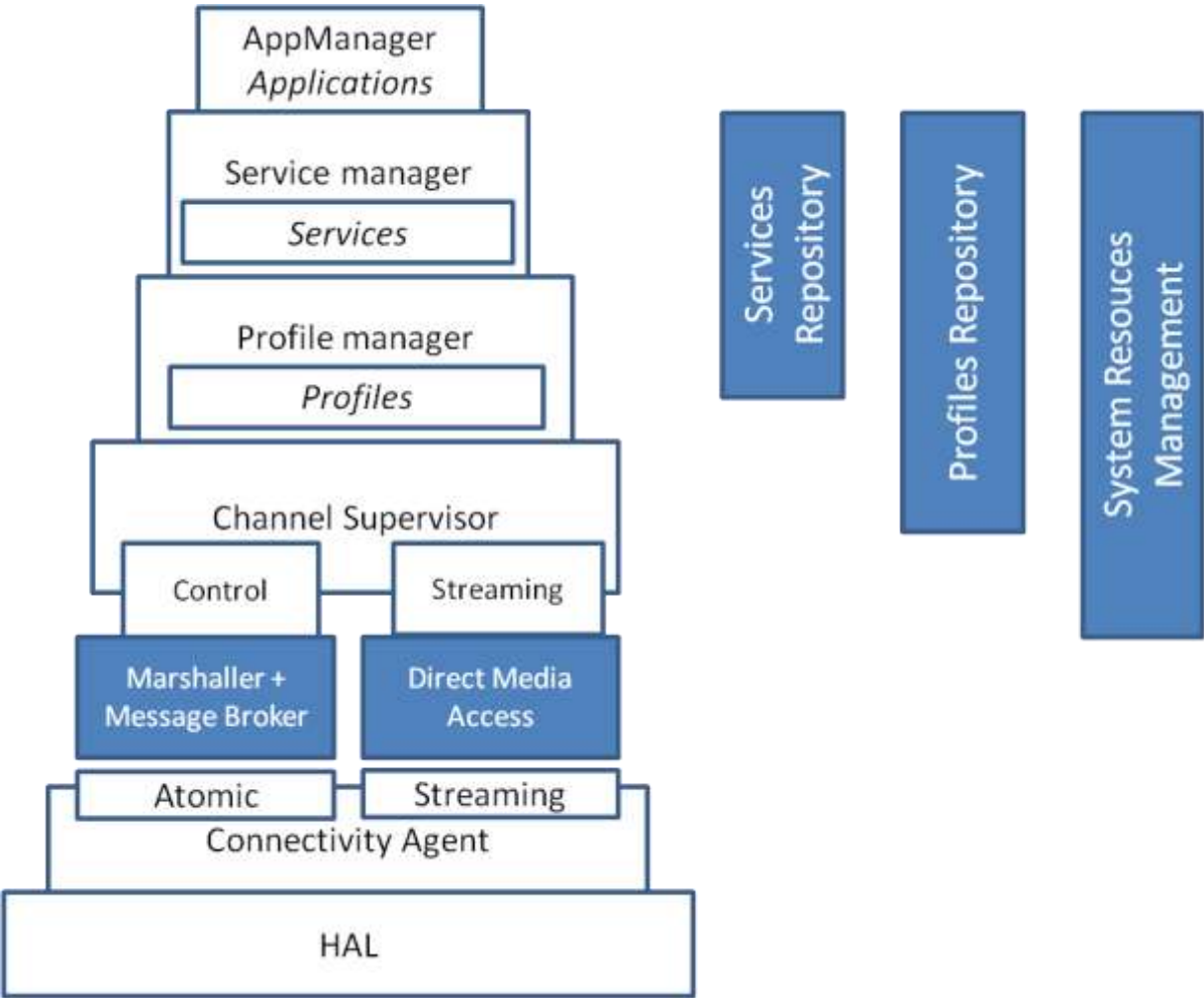
Remote procedure call – method used for restrict profile interaction only strong-typed messages. Also RPC used for avoid built distribute finite automata or protocol-based solutions.

Service – Named list of Profile API names (ID's), dedicated for direct requesting from applications. Service allow grouping of Profile API in a single entity in Service repository and allow atomic initializing.

Service repository – contain services (map of services). Actually is simple folder with XML files.

System Control Profile – holder of the Physical Channel#0 and processor of the administrative requests

1.6. Technology Overview



iviLINK design is based on the layered approach to architecture with strict rules of interaction between the layers. Lowest level is the Connectivity Agent responsible for media independent data transmission. Higher level data may be transferred in two ways: control messages or streaming. The first approach is oriented on control data transfer; the second one is mostly applicable for media data transportation.

Over the channel supervision layer, the level of profiles is applied. iviLINK approach treats profile as polymorphic chunk of code responsible for higher abstraction and hiding of network data exchange. Profiles are be grouped to services – named set of profiles suitable for atomic initialization and accessing and fully covering some complete use case.

Services do grouping, encapsulating and managing of profiles bind by the functional sets. Application operates with services, but not profiles directly. OEM or system integrator may select services supported on the particular device depending on the requirements and marketing needs.

1.6.1. Applications Level

iviLINK doesn't define how the applications to be developed and doesn't limit applications developers in style and approach how their applications to be made. iviLINK supplies Application Manager as a framework mechanism for remote launching of the applications to serve data coming from the connected side. To be capable to respond Application Manager requests each application supporting iviLINK shall do self-registering with using corresponded API. Once registered it may be called upon the request to support the service inquired by the connected side.

1.6.2. Service Level

Services are the software entities grouping fragments of functionality implementing sets of features valuable for the user. For example, it maybe "media player" service, "vehicle settings" service, "cameras view" service etc. Service is containing multiple data types for exchange and has no direct relation to the method and format of the data exchange, but only refers to the way how these data are used and applied in the application. I.e. service is to be designed to implement some particular use case.

iviLINK requires all the services on the device be predefined and preprogrammed (in contrast to profiles used by these services which can be replaceable and upgradable). This lets system vendor to lock the entire set of the functions and behaviors supported by the device even in case of installable applications are allowed.

Services have no user/programmer available runtime representation. Being a part of SDK, Service Manager simulates all necessary APIs using Service Repository, i.e. description of the services available on the device and the set of profiles required to make this service working

Service doesn't define the specific profiles (or profile implementations), but only profile APIs to be supported on the device to provide applications with the appropriate service. If no profiles with appropriate API are available or active, the corresponding service becomes disabled and blocked. If several profiles with the same API are available, Service Manager will perform parametric match of the best usable profile implementation.

1.6.3. Profile Level

Profiles make iviLINK maximum flexible technology in class. Profile is responsible for transferring of some data type or corresponding protocol e.g. file transfer, mpeg playback, remote diagnostics gathering protocol etc. Profile contains profile API and implementation. Same API can be supported by several of profiles, this makes profiles polymorph, i.e. same role can be handled on different manner by using different profile implementations.

If several profiles with the same API are available on the device, each profile is to be described with a set of attributes (e.g. best frame rate or approach to access diagnostics data). These attributes are used by Service Manager to select proper profile when binding to the service requested by application.

From the implementation standpoint, profile contains the record in Profile Repository, dynamic linked libraries for Profile Implementation and pre-assigned API UID. UID lets group profiles with the same API and do dynamic selection upon the Service Manager request.

Using signed dynamic libraries for profile implementation lets download profile from Internet or remotely connected device while iviLINK connectivity is in progress. As a profile is representing some data format or communication technology, it allows upgrading of the older device software without any impact to service or application levels.

1.6.4. Data Channels

Profile is using data channels to communicate with remote side. iviLINK SDK provides two types of the data channels: multimedia for real time data flows and command for message based data exchange. iviLINK contains

data scheduler, channel allocation and multiplexing mechanisms which let using iViLINK over any type of serial link like TCP socket, Bluetooth Serial Port or USB.

iViLINK also lets using Out of the Band approaches of data transfer on the profile level.

2. Integration and Relations to Other Technologies

2.1. General Information

iViLINK is designed to act as a shield over existing well established common and proprietary technologies of mobile to automotive communication. According to the iViLINK approach, all data transfer mechanisms including side technologies shall be implemented as profiles. Existing solutions may be aligned to profile API and architecture with using wrapping modules.

iViLINK allows supporting two types of communication: in band and out of the band. In band communication means that the profile uses channel supervisor and lower levels of the software stack to perform data exchange, out of the band method lets the integrated technology use its own native data transport and iViLINK just makes sure that both of connected side are properly identified and authenticated and controls the connection via profile API. Usually side technologies use the second communication schema if used under iViLINK framework control.

iViLINK supports parallel usage of various side technologies if there's no conflict on out of the band channels allocation between them.

2.2. Bluetooth

iViLINK uses Bluetooth Serial Port Profile as one of the carrier types. Natively it doesn't use higher level Bluetooth profiles for data transfer. However specific iViLINK profiles are capable to utilize Bluetooth functionality as out of the band channels for media and control data exchange.

Most typical usage of Bluetooth as a separate connectivity technology is telephony, high quality (advanced) audio transfer and OBEX data exchange.

On some operating systems like Android, usage of Bluetooth as out of the band high level carrier is only a method to capture and route some types of media content (audio) due to operating system limitations. It shall be applied even in case if iViLINK core uses another carrier to establish its main channel.

2.3. DLNA™

DLNA might be treated as a standalone mechanism to manage and transfer media data between mobile and automotive devices. From the other side, DLNA is system level technology not adopted for in-application usage. iViLINK lets wrap DLNA to its common media service APIs.

With DLNA incorporated to iViLINK the system becomes capable to provide service with using DLNA if both sides support it or simpler technologies like RTP, HTTPlite or RTSP in DLNA is not supported by one of the paired devices.

2.4. AppLink

iViLINK is capable to incorporate voice menu based functionalities of Ford AppLink and similar services with packaging it to profile envelopes. It lets using existing audio-based applications functionality with extensions of video and data sharing provided by alternate iViLINK profiles. It also helps in integration of AppLink functions with external applications using iViLINK framework capabilities. Parallel use of AppLink and visual iViLINK profiles enables vehicle applications with Graphics + Sound capabilities.

2.5. MirrorLink™, VNC and RTP

iviLINK is a complementary technology to MirrorLink™. VNC and RTP profiles are incorporated to the SDK and available in a package with iviLINK framework.

In contrast to MirrorLink or VNC, iviLINK is an application-oriented approach, not system-oriented approach. iviLINK package contains VNC sample apps providing basic VN functionality joint with iviLINK capabilities and APIs.

3. Feature List of SDK v0.9 (Technology Preview)

- Protocol stack
 - HAL
 - Connectivity Agent
 - Channel Supervisor
 - Profile Manager
 - Profile Repository
 - Service Manager
 - Application Manager
 - Application Repository
 - Authentication service
- Samples
 - Climate profile
 - Seat settings profile
 - Climate app
 - Seats app
- Quick start documents package

4. Change History

Version	Date	Status	Change description	Author/Editor
0.1	31 Mar 2012	draft	Initial	Konstantin A. Khait
1.0	31 May 2012	First version	Fixed and extended	Konstantin A. Khait

5. Approve History

Version	Approval Date	Issue ID	Approved By