

团 体 标 准

T/TIAA XX—202X

车载小程序技术要求

Technical requirements for vehicle mini-program

(征求意见稿)

2022- XX - XX 发布

2022 - XX -XX 实施

中关村车载信息服务产业应用联盟 发布

目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 车载小程序概述与框架	2
5.1 概述	2
5.2 车载小程序框架	3
6 车载小程序框架技术要求	5
6.1 车载小程序接口技术要求	5
6.2 车载小程序接口协议技术要求	5
6.3 车载小程序基础 API 接口	6
6.3.1 登录接口	6
6.4 获取车辆信息 API 接口	10
6.5 导航接口技术要求	12
6.6 多媒体控制接口技术要求	12
6.7 车载小程序屏幕适配技术要求	15
6.8 语音交互要求	15
6.9 车辆信息系统要求	16
7 监控技术要求	16
8 安全技术要求	16
附录 A	17

前 言

本文件按照 GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中关村车载信息服务产业应用联盟标准化委员会提出并归口。

本文件起草单位：腾讯科技（深圳）有限公司、重庆长安汽车股份有限公司、合肥工业大学、博泰车联网科技（上海）股份有限公司、博世（中国）投资有限公司、北京梧桐车联科技有限责任公司、北京车百智能网联科技有限公司、湖南湘江智芯云途科技有限公司、重庆桴之科科技发展有限公司、惠州华阳通用电子有限公司

本文件主要起草人：张亚军、李法镜、XX



车载小程序框架和技术要求

1 范围

本文件规定了车载小程序的概念，车载小程序通用的框架特征和技术要求以及车载小程序能力框架。

本文件适用于车载多媒体服务和车载小程序应用。
其他智能汽车相关程序也可参考使用本文件内容。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 40855-2021 《电动汽车远程服务与管理系统信息安全技术要求及试验方法》
GB/T 40856-2021 《车载信息交互系统信息安全技术要求及试验方法》
GB/T 40857-2021 《汽车网关信息安全技术要求及试验方法》
GB/T 40861-2021 《汽车信息安全通用技术要求》
GB/T 36464.5-2018 《信息技术 智能语音交互系统 第5部分：车载终端》

3 术语和定义

下列术语和定义适用于本文件。

3.1

车载信息交互系统 on-board information interaction system

安装在车辆上的通信系统，具备下列至少一项功能：

- a) 对外可通过蜂窝网络、短距离通信等通信技术建立连接并进行数据交换等功能，对内通过汽车总线与电子电气系统进行信息采集、数据传递与指令下发等功能；
- b) 实现通话、录音、导航和娱乐等相关服务功能。

[来源：GB/T 40856-2021]

3.2

车载小程序 vehicle mini-program

车载小程序是区别于车载APP的轻应用，类似微信小程序的理念，轻量化、无需下载安装、即用即走。

3.3

应用软件 application software

在车载信息交换系统上，为实现支付、娱乐等功能的一类软件。

注：包括在车载信息交互系统中已预装的应用软件和后期可安装的应用软件。

[来源：GB/T 40856-2021]

3.4

用户数据 user data

有用户产生或为用户服务的数据。

[来源：GB/T 40856-2021]

3.5

GCJ-02

GCJ-02 是由中国国家测绘局（G表示Guojia国家，C表示Cehui测绘，J表示Ju局）制订的地理信息系统的坐标系统，是在WGS84经纬度的基础上执行加密算法而成。

4 缩略语

下列缩略语适用于本文件。

HTML	超文本标记语言	HyperText Markup Language
OEM	代工生产	Original Equipment Manufacturer
URL	统一资源定位符	Uniform Resource Locator
SDK	软件开发工具包	Software Development Kit
LBS	基于位置服务	Location Based Services
POI	兴趣点	Point of Interesting
ASR	自动语音识别	Automatic Speech Recognition
TTS	语音合成	Text To Speech
NLP	自然语言处理	Natural Language Processing
HMI	人机接口	Human Machine Interface
APP	软件应用	Application
UTF-8	可变长度编码	Unicode Transformation Format
VIN	车辆识别码	Vehicle Identification Number

5 车载小程序概述与框架

5.1 概述

车载小程序是一款基于HTML5和JavaScript框架的车载场景新应用。结合智能语音系统，车载小程序将内容/服务搜索与语音互操作无缝融合，支持全程语音交互，有效提升车载环境下的应用交互体验。

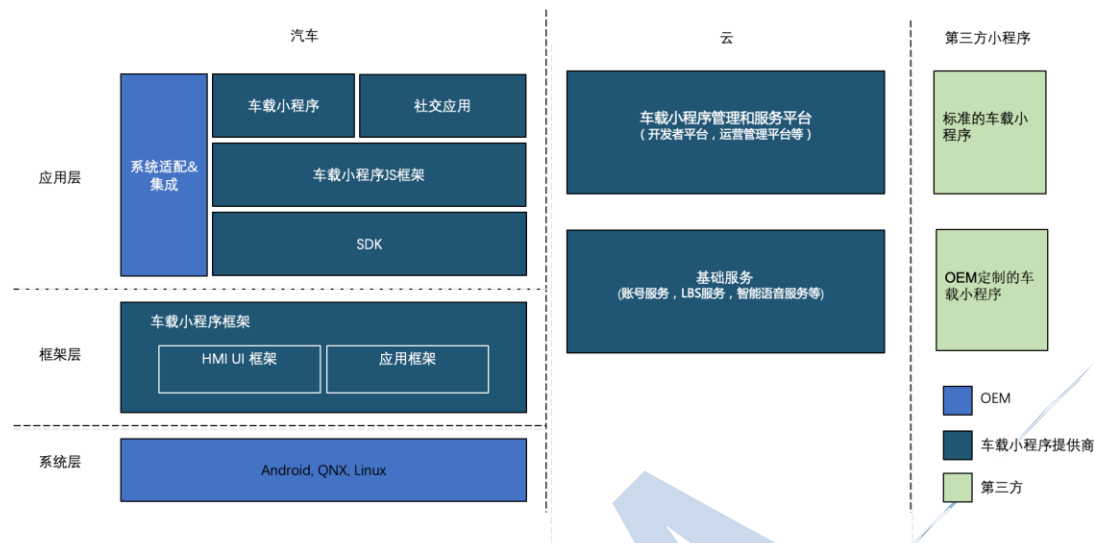


图1 车载小程序系统框架

应用层：基于JavaScript框架和兼容应用的车载小程序。

框架层：包括系统UI框架、应用框架。

车载操作系统：宜支持包括但不限于Android、QNX、Linux等系统。

云服务：车载小程序管理平台及服务。宜支持基础服务，比如LBS服务，智能语音服务。

第三方车载小程序：提供标准车载小程序接口和OEM车载小程序接口。

5.2 车载小程序框架

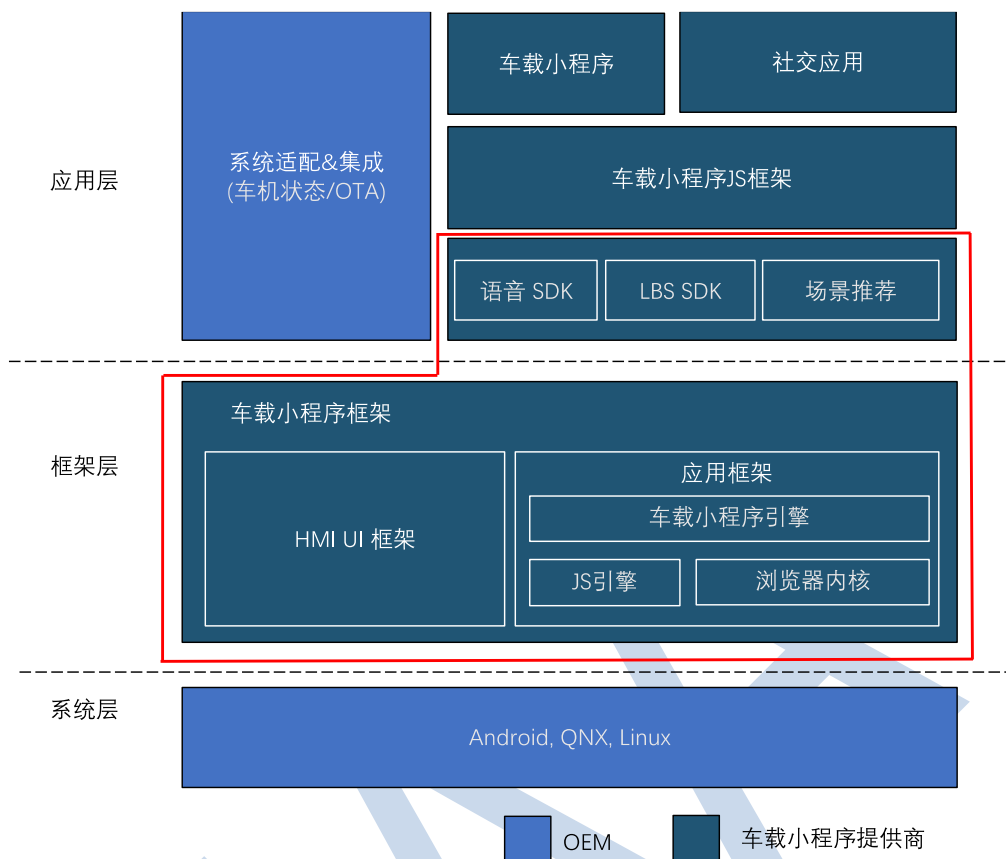


图2 车载小程序框架

车载小程序无需安装，动态加载和更新的机制可以极大的方便用户的使用和内容消费。车载小程序结合智能语音系统，将内容/服务搜索与语音系统无缝集成，全程支持语音交互，有效提升车载环境下的应用交互体验。

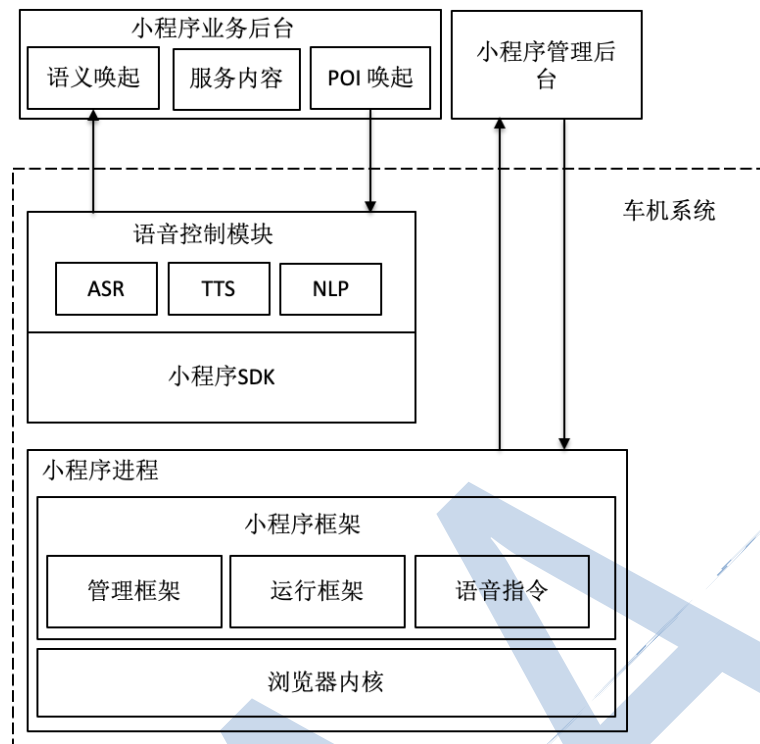


图3 车载小程序模块架构

- 车载小程序功能模块：支持语音唤起，基于 POI 位置唤起。
- 车载小程序管理模块：车载小程序升级和配置。
- 语音控制模块：支持 ASR、TTS、NLP。
- 车载小程序 SDK：提供语音推荐接口。
- 车载小程序线程：包括车载小程序管理模块、操作模块和语音指令模块。
- 所有这些模块都可以在 Web 浏览器服务中运行。

6 车载小程序框架技术要求

6.1 车载小程序接口技术要求

车载小程序接口：

- 车载小程序初始化接口：初始化车载小程序；
- 车载小程序启动界面：通过传递 URL 启动车载小程序；
- 车载小程序语音接口：可以通过语音执行相应的语音命令；
- 车载小程序多媒体控制接口：通过该接口对多媒体内容进行播放控制操作和状态更新；
- 车载小程序状态监控接口：注册侦听和接收有关车载小程序状态的通知。

6.2 车载小程序接口协议技术要求

车载小程序接口协议应为 HTTPS POST 类型接口，请求使用 application/json，UTF-8 编码，请求返回为 Json 对象。

输入参数：

属性	必填	说明
----	----	----

appid	是	小程序 id, 唯一标识
sign	是	签名
timestamp	是	请求时间戳, 单位到秒
nonce	是	请求随机串, 客户端生成的 9 位字母组成的字符串
code	是	消费 code 码
version	是	协议版本
grant_type	否	authorization_code

示例代码:

```
POST 请求 body:
{
  "appid":"60258",
  "nonce":"abcde6789",
  "timestamp":"1606286021",
  "version":"1.0",
  "code":"29225ba02ee811eba252b55ce8cf45c8",
  "grant_type":"authorization_code",
  "sign":"626802ae19bd45a33855594b3d2450df"
}
```

6.3 车载小程序基础 API 接口

6.3.1 登录接口

用户授权登录。

参数:

属性	类型	必填	说明
timeout	number	否	超时时间, 单位 ms
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

回调参数:

属性	类型	说明
code	string	用户登录凭证

示例代码:

```
login({
  success (res) {
    if (res.code) {
      //发起网络请求
      wx.request({
        url: 'https://example.com/onLogin',
        data: {
          code: res.code
        }
      })
    } else {
      console.log('登录失败!' + res.errMsg)
    }
  }
})
```

6.3.2 获取用户信息接口

获取用户信息。

返回结果：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数
withCredentials	boolean	否	是否带上登录态信息。 当 withCredentials 为 true 时，要求此前有调用过 login 且登录态尚未过期； 当 withCredentials 为 false 时，不要求有登录态。

回调参数：

属性	类型	说明
userInfo	number	用户信息对象，不包含 openid 等敏感信息
rawData	number	不包括敏感信息的原始数据字符串，用于计算签名
signature	number	用于校验用户信息
encryptedData	number	包括敏感数据在内的完整用户信息的加密数据
iv	number	加密算法的初始向量

示例代码：

```

getUserInfo({
  success: function(res) {
    var userInfo = res.userInfo
    var nickName = userInfo.nickName
    var avatarUrl = userInfo.avatarUrl
    var gender = userInfo.gender //性别 0: 未知、1: 男、2: 女
    var province = userInfo.province
    var city = userInfo.city
    var country = userInfo.country
  }
})
encryptedData 解密后为以下 json 结构
{
  "openId": "OPENID",
  "nickName": "NICKNAME",
  "gender": GENDER,
  "city": "CITY",
  "province": "PROVINCE",
  "country": "COUNTRY",
  "avatarUrl": "AVATARURL",
  "watermark": {
    "appid": "APPID",
    "timestamp": TIMESTAMP
  }
}

```

6.3.3 获取系统信息接口

获取系统信息，判断车联环境。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

回调函数参数：

属性	类型	说明
brand	string	不同汽车品牌对应的技术平台
model	string	手机型号
pixelRatio	number	设备像素比
screenWidth	number	屏幕宽度
screenHeight	number	屏幕高度
windowWidth	number	可使用窗口宽度
windowHeight	number	可使用窗口高度
language	string	系统设置的语言
version	string	版本号
system	string	操作系统版本
platform	string	客户端平台
SDKVersion	string	客户端基础库版本

Orientation	string	当前屏幕展示状态。 横屏全屏：landscape-full; 横屏半屏：landscape-half; 竖屏全屏：portrait-full; 竖屏半屏：protrait-half;
device	string	设备类型

示例代码：

```

getSystemInfo({
  success (res) {
    console.log(res.device)
  }
})
try {
  const res = wx.getSystemInfoSync()
  console.log(res.device)
} catch (e) {
}

```

6.3.4 检查登录状态接口

检查登录态是否过期。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

示例代码：

```

checkSession({
  success () {
    //session_key 未过期，并且在本生命周期一直有效
  },
  fail () {
    // session_key 已经失效，需要重新执行登录流程
    login() //重新登录
  }
})

```

6.3.5 获取用户设置接口

获取用户的当前设置。返回值是用户授权结果。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

回调函数参数：

属性	类型	说明
authSetting	AuthSetting	用户授权结果

示例代码：

```

getSetting({
  success (res) {
    console.log(res.authSetting)
    res.authSetting = {
      "scope.userInfo": true,
      "scope.userLocation": true
    }
  }
})

```

6.4 获取车辆信息 API 接口

6.4.1 获取速度接口

获取车辆行驶速度。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

示例代码：

```

getSpeed({
  success: function(res) {
    console.log('获取的车辆速度', res);
  }
})

```

6.4.2 获取剩余油量接口

获取车辆剩余油量。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

示例代码：

```

getRemainPower({
  success: function(res) {
    console.log('获取的车辆油量', res);
  }
})

```

6.4.3 获取车辆 vin 接口

获取车辆vin。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

示例代码：

```
getVehicleId({
  success: function(res) {
    console.log('获取的车辆 vin', res);
  }
})
```

6.4.4 获取方向盘转向角接口

获取方向盘转角。

参数：

属性	类型	必填	说明
success	function	否	接口调用成功的回调函数
fail	function	否	接口调用失败的回调函数
complete	function	否	接口调用结束的回调函数

示例代码：

```
getWheelRotationAngle({
  success: function(res) {
    console.log('获取的方向盘转角', res);
  }
})
```

6.4.5 监听车辆信息接口

监听车辆信息变化。

参数：

属性	类型	说明
speed	Array	监听速度变化
keyCode	Array	监听 keyCode 变化

示例代码：

```
addVehicleEventListener({keyCode: ['keycode'],speed: ['speed']}, function(data) {
  console.log('车辆信息返回--', JSON.stringify(data) );
  data.forEach(function(item){
    if (item.hasOwnProperty('speed')) {
      console.log('车速信息',item.speed)
    }
    if (item.hasOwnProperty('keycode')) {
      console.log('keycode',item.keycode)
    }
  })
})
```

6.5 导航接口技术要求

6.5.1 地图接口

参数：

属性	类型	默认值	说明
destination	Object		目的地

Destination参数：

属性	类型	必填	说明
latitude	number	是	GCJ-02 纬度，范围为 $-90\sim 90$ ，负数表示南纬
longitude	number	是	GCJ-02 经度，范围为 $-180\sim 180$ ，负数表示西经
address	string	是	地址的详细说明
name	string	否	位置名

示例代码：

```
const tamLatitude = 39.908692;
const tamLongitude = 116.397433;
const tamAddress = "天安门";

navigateMap({
  destination: {
    latitude: tamLatitude, // GCJ-02 坐标
    longitude: tamLongitude, // GCJ-02 坐标
    address: "" // 地址
  }
})
```

6.6 多媒体控制接口技术要求

6.6.1 音乐播放数据格式

表 1 音乐播放数据字段

数据字段	说明
operationType	"play", "pause", "stop", "resume", "seek"
src	音频的数据源
title	音频标题
coverImgUrl	封面图 URL，用于做原生音频播放器背景图
singer	歌手名
epname	专辑名
currentTime	当前音频的播放位置（单位：s）
state	"pause", "stop", "play", "timeupdate", "waiting", "error", "occupied", "pretrack", "nexttrack"
errCode	错误码，详见附录 A。

表 1 音乐播放数据字段（续）

数据字段	说明
errMsg	错误信息
paused	暂停
duration	音频的时长
buffered	缓存

6.6.2 播放列表

播放列表，需要保存到统一播放卡片的曲目列表信息，列表中每个 item，有如下属性：

属性	类型	必填	说明
title	string	是	音频标题
singer	string	是	歌手名
coverImageUrl	string	否	封面图 URL
dataUrl	string	是	音频的数据源
options	string	否	自定义字段，用于开发者存放描述当前曲目的定制信息

示例代码：

```

var curPlayList = [{
  coverImgUrl:'http://image.biaobaiju.com/uploads/20180803/23/1533309822-GCcDphRmqw.jpg',
  title: '音频名称 1',
  singer: '歌手 1',
  dataUrl: 'http://music.163.com/song/media/outer/url?id=476592630.mp3',
  options: JSON.stringify({
    epname: 'xx 专辑'
  })
},
{
  coverImgUrl:
'https://qpic.y.qq.com/music_cover/ezXpob9biaedyoUWFJDttEQc9HsqSUPliaPVnKrwk194QrXna9xZ5
wbg/300?n=1',
  title: '音频名称 2',
  singer: '歌手 2',
  dataUrl: 'https://cdn.kaishuhezi.com/audio/story/M-2-01-20150408.mp3'
},
{
  coverImgUrl:
'https://qpic.y.qq.com/music_cover/HvzXRphXYwDIM7tpX0XM4dCF2K6tu1k38ts4H8sD5icicTtoeZPs
M09g/300?n=1',
  title: '音频名称 3',
  singer: '歌手 3',
  dataUrl: 'https://cdn.kaishuhezi.com/audio/story/M-1-01-20150408.mp3'
}
];
const audioPlayer = wx.getBackgroundAudioManager()
audioPlayer.playInfo = {
  playList:curPlayList,
  context: JSON.stringify({key:1000,timeStamp:536760})
}
audioPlayer.src = "http://music.163.com/song/media/outer/url?id=476592630.mp3";

```

6.6.3 音乐播放流程

车载小程序音乐播放流程图见图4，具体如下：

- a) 车载小程序向车载小程序应用框架发送音频播放控制信息；
- b) 车载小程序应用框架解析音频播放控制信息，生成音频播放控制请求；
- c) 车载小程序应用框架向车载操作系统发送音频播放控制请求；
- d) 车载操作系统根据音频控制请求，对音频进行播放控制；
- e) 车载小程序应用向车载小程序应用框架发送音频状态获取请求；
- f) 车载小程序应用框架根据音频状态获取请求，获取音频的实时状态信息；
- g) 车载小程序应用框架向车载小程序应用发送音频的实时状态信息；
- h) 车载小程序应用根据音频的实时状态信息，更新音频的实时状态。

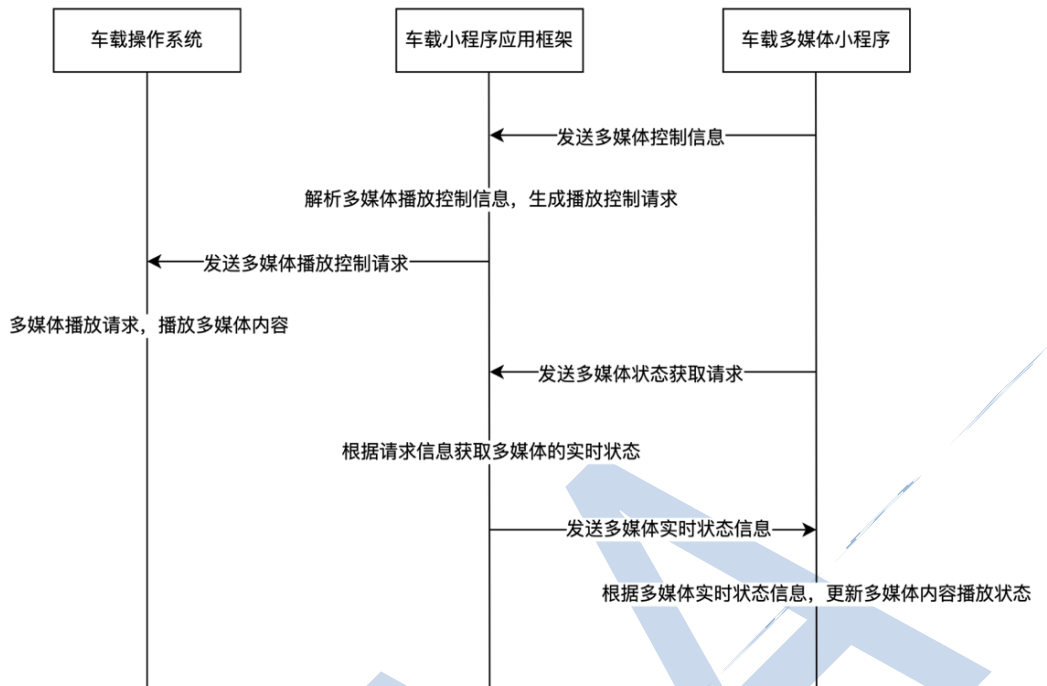


图4 车载小程序音乐播放流程

6.7 车载小程序屏幕适配技术要求

车机屏幕有多种尺寸，且屏幕方向分横屏和竖屏两种。

车载小程序框架应保证车载小程序可视区域的宽高比例。车载小程序只存在横屏和竖屏的区别（此处横屏和竖屏指小程序可视区域，而不是车机屏幕）。本文件应考虑两种适配方案（横、竖）。可以使用前端通用接口获取可视窗口的宽度和高度，判定可视区域的横竖模式。也可以使用前端通用接口获取可视窗口的宽度和高度，并判定可视区域的横竖模式。

示例代码：

```

<br/>
    if (window.width > window.height) {
        return 'horizontal'
    } else {
        return 'portrait'
    }
  }
<br/>

```

6.8 语音交互要求

- 支持智能语音调起、切后台、关闭相关小程序；
- 支持小程序内的语音交互，包括打开链接、播放音视频、上下翻页等；
- 支持语音的纠错能力，例如语音“播方”可以纠错成“播放”；
- 语音交互能力支持动态更新；

- 车载终端智能语音交互系统的平均响应时间请参考表 3。

表 3 语音识别率

系统	子系统	基础目标
语音静态识别率	-	≥95%
语音动态识别率	-	≥90%
离线响应速度	导航场景语音控制响应速度	≤4s (T7 平台) ≤3s (8666 平台)
	音乐场景语音控制响应速度	≤4s (T7 平台) ≤3s (8666 平台)
	电话场景语音控制响应速度	≤3s (T7 平台) ≤2s (8666 平台)
在线响应速度	导航场景语音控制响应速度	≤4s (T7 平台) ≤3s (8666 平台)
	音乐场景语音控制响应速度	≤4s (T7 平台) ≤3s (8666 平台)
	电话场景语音控制响应速度	≤3s (T7 平台) ≤2s (8666 平台)

6.9 车辆信息系统要求

- 支持通过车载中控界面点击拉起小程序；
- 支持通过语音拉起小程序，语音包含固定术语，语义解析；
- 支持 Push 通知调起小程序；
- 支持基于 LBS 地理围栏分发小程序；
- 支持基于内容服务，搜索直达小程序具体页面。

7 监控技术要求

支持监控启动时长，内存占用，打开失败率等性能指标监测。

8 安全技术要求

- 支持车载中控根据车辆当前状态控制小程序能力；
- 支持车载中控对小程序内音视频语音与车载语音等冲突控制能力；
- 小程序框架存在校验机制，防止框架代码被篡改；
- 支持小程序内核漏洞动态更新的能力；
- 小程序只能使用公开的 JavaScript 接口，不能使用未公开的 JavaScript 接口或者其他途径调用本地能力；
- 每个小程序独立运行，小程序之间不能互相获取数据。

附录 A

(资料性)

错误码

errCode(String 类型)	错误码描述
-404	系统不存在车联账号服务
-403	需要先登陆
-402	权限 scope 非法
-401	用户拒绝授权
-1	服务错误
3	参数错误
5	认证错误
101	userid 不存在
104	包名无效
153	sessionKey 无效
250	carid 不存在
317	车机已登录其他车联用户
321	车机未登录车联用户