为什么需要独立一个业务模块Dji-bus

# 背景

## 问题/需求描述

## Djilink角度

1. djilink承担了太多业务，后期维护变得困难，扩展业务影响底层的稳定性/难以和system\_server通信，逐渐会演变为3288的dji\_sys和dji\_server,维护的人都知道，蛋疼。举例子，tom+上熊+key调试的蜂鸣器业务。容易
2. 业务扩展性低，属于c++的一个程序，在init进程调起。和system\_server通信困难，比如需要AMS提供最近任务信息，app前后台切换
3. 稳定性要求高，既然作为一个link的核心进程，要保证稳定性，原则就是尽量不再改动
4. 专注于专用功能，当初定位就是一个通信功能和一些非常核心的升级服务，不应该承接app层琐碎的业务

## 系统APP角度

* 1. 链接浪费，目前要走V1，必须建立一个链接,，那么有10个app，就必须10条链路，并且要分配senderId
  2. 业务定制复杂，即使一个很简单的飞机链接判断，要走V1协议国语复杂，每一个app都必要写V1包的繁琐代码

## Dji-bus的功能需求

1. 是link和所有app的桥梁，一个适配器。
2. 对link屏蔽了所有业务细节，对app们屏蔽了数据链路，组包，解包等V1细节，
3. 和apps紧密配合（这里也需要解耦的），实现业务的定制
4. 开发周期不能太长（时间要求）

# 设计

## Dji-bus的软件形态架构

运行于system\_server，作为server.jar的一个子服务，理由如下

1. 业务扩展性强：system\_server的AMS/PMS/WMS等为APP层提供了强大的交互功能，Dji-bus作为system\_server的一个子服务，自然集成了这些交互
2. 贴近android原生服务架构，设计模式可以借鉴。

## Dji-bus开发实施

### Dji-bus与djilink

CS模型，非常简单，dji-bus的id为0204？，复用系统升级的？？

底层复用Dpad\_jni.so 的C++代码，jni实现调用

封装一下就好了

附加一个字段，标志APPID？？？

### Dji-bus与apps通信

dpad-core：里有V1的包格式的协议，要完成

Apps：依赖dpad-core，根据业务组合一个pack+包名，通过进程间通信发送给Dji-bus

服务名字：DJiBusManager

DusManager.DjiManager,V1Manger,DjiLinkManger，DjiProtocolManager，RcManager，

模仿

广播机制:还可以有堵塞机制，谁有限发送设计

回调设计?

AOA怎么样

首先需要在system\_server添加一个dpad-core.jar

# REF 原生设计

WifiManager

USBManager

SensorManager

LocationManager

# Todo

## 安全性设计

没有加固怎么整

## Binder太多堵塞怎么办