

1. 本质问题概述

讨论“698协议**对象**”，这里**对象**主要是指：终端自己需要管理、使用的信息。

讨论“698协议对象的**实现**”，这里**实现**主要是指：信息在终端APP中如何表示和组织；如何与主站交换信息。

2. 问题回顾

分两块简述需要解决的问题：

- 1. 信息在终端APP中的表示和组织；
- 2. 与主站交换信息的模型和方法；

2.1 终端内部信息的表示和组织

在本周几次讨论中，我们主要提及了2种方法：

- 1. 将698协议文档中类和对象的标记方式，映射到C++中的class（包括采取继承等方式），然后再实例出object。
- 2. 脱离698协议文档，从业务实现出发，需要什么信息，就设计什么样的结构。

终端中的约束

结合终端这款具体的产品，技术要求文档：APP之间往往不直接交互数据，数据的交互通过 **数据中心** 完成。

技术文档中，对数据中心数据表示和组织已经进行了规范。

数据中心信息按如下方式组织与表示：

- 1. OAD（结合ROAD）索引到唯一一个元素；
- 2. 对信息内容格式有规定：使用了698协议中的基础数据类型、重新定义了编码方式；

2.2 与主站交换信息的模型和方法

终端中只有特定的APP与主站交互，在营销专业中是低压集抄APP。图1中将它称为接口APP，因为它主要起到**接口**作用：

- 信息存储在数据中心或某个APP中；
- 甚至可以由某个APP直接提供响应内容，由接口app转发至主站；
- 接口app完成了数据格式的转换，如A-XDR与数据中心数据内容格式的转换。

结合2.1小节，终端信息访问、组织和表示的总体框图如下：

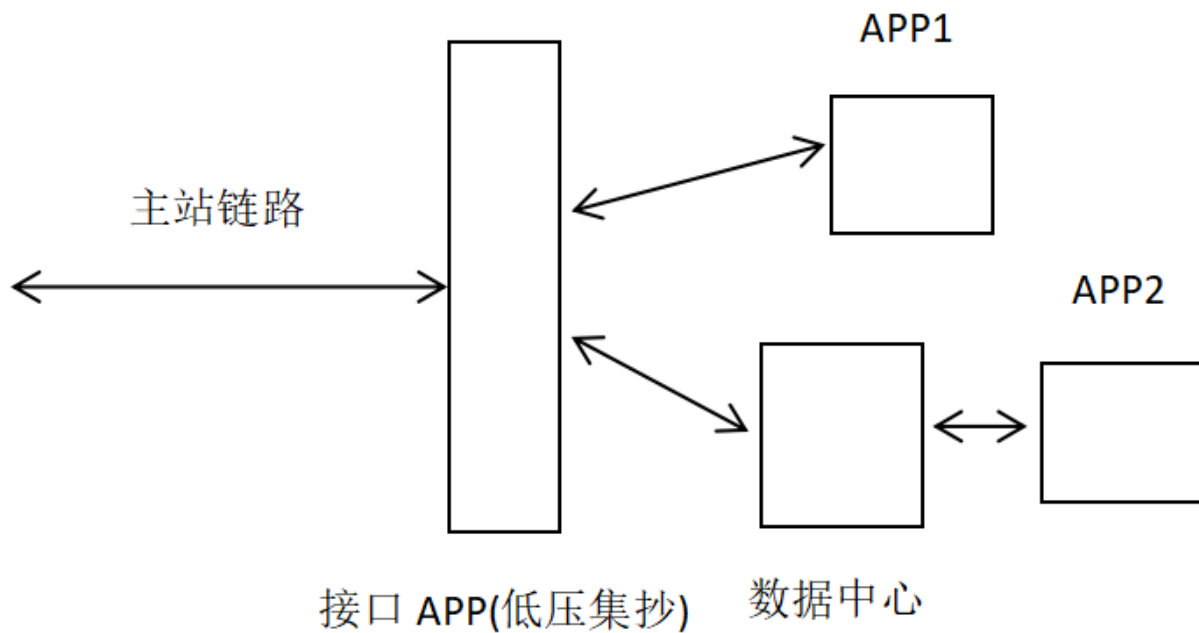


图1 信息交互模型

在终端APP中，信息的组织和分布，大致有两种情况：

1. 类似APP1，信息仅在APP中，没有被同步至数据中心。这类信息通过APP之间的消息接口通道获取；
2. 类似APP2，信息被存入数据中心。产生数据的APP，没有提供APP之间的消息接口访问途径。这类信息从数据中心获取。

3. 实现思路

将APP中信息的表示/组织与访问接口分开。

3.1 APP中信息的表示/组织

APP内部，即产生、使用信息的地方，从业务需求出发设计数据结构。

1. 可以完全脱离698协议，以方便业务操作的原则组织数据；
2. 可以按照698协议中的属性，描述数据；依据业务需求提供合适的接口函数；

3.2 访问接口

总体思路：

1. 由单个信息体（或称呼为对象）响应访问；
2. 对象的实现按698协议要求的属性和方法要求设计，重点在访问接口的实现。

在接口角度，信息按对象组织；与主站交互时每个对象处理各自信息交互。

这样做是为了降低问题规模：将访问分散到单个信息体中。

如协议文档中描述的：

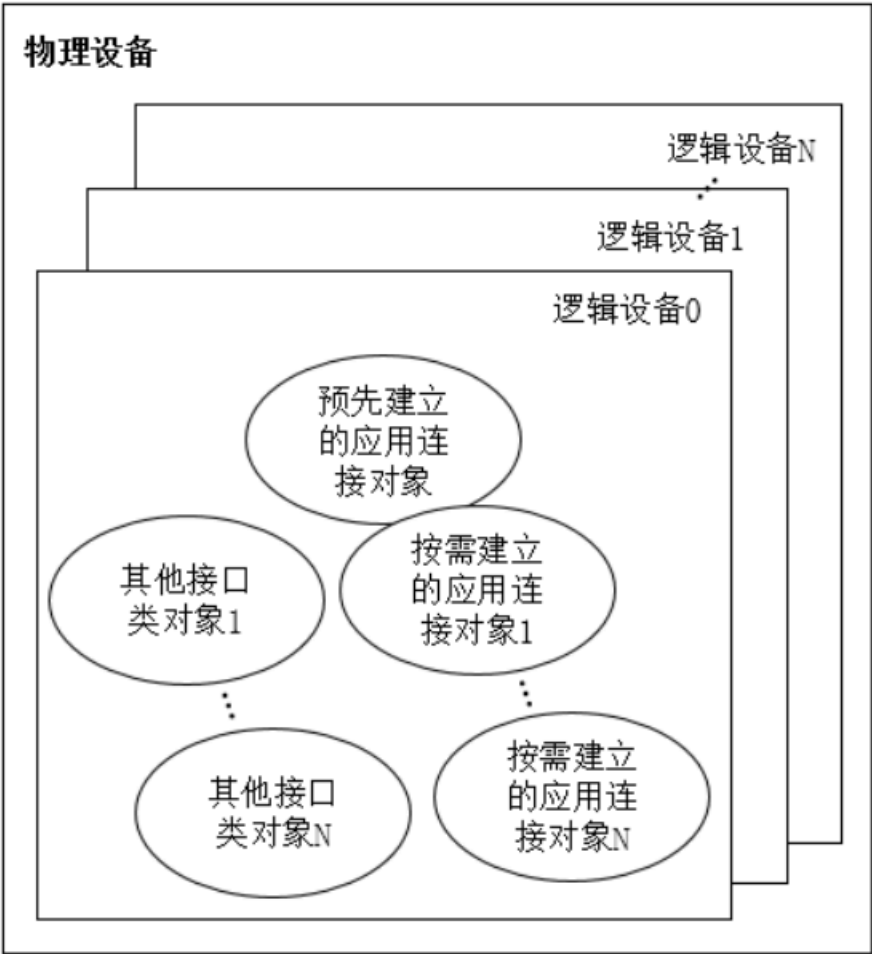


图2 由对象响应访问

访问接口上，对象侧重**接口函数**的实现；

接口函数重点在信息的查找与格式转换：

- 从数据中心或者某个APP中获取信息；
- 格式转换：序列化和反序列化

数据中心已经有良好的信息组织、表示方式。如果信息在某个APP中，则应该约束APP对外接口：

- 按OAD定位信息；
- 有数据内容有明确的格式定义；