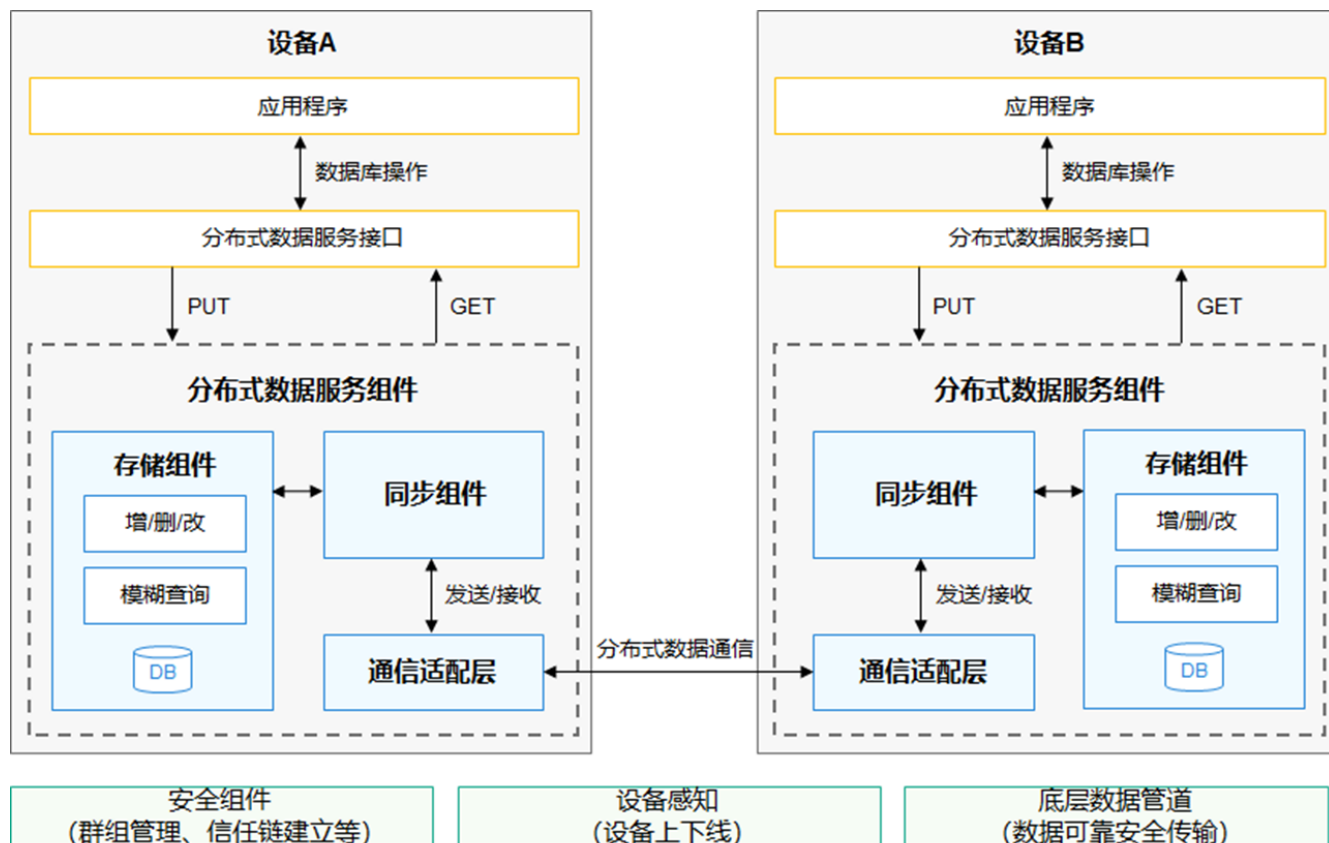


## 鸿蒙系统分布式数据管理

鸿蒙系统分布式数据管理的核心组件设计旨在实现跨设备的数据共享与一致性，其架构围绕以下几个关键部分展开

### 分布式数据库各个组件



### 数据访问组件（数据服务接口）

- 提供统一的接口给上层应用，使得应用能够如同操作本地数据一样轻松访问和操作分布式环境下的数据。这包括对关系型数据（如联系人、日历）和键值型数据（如缩略图、设备信息）的访问。
- 特性：**支持数据订阅、变更通知、远程查询、冲突监听等功能，便于开发者在多端协同和跨端迁移场景下处理数据同步问题。

### 分布式数据库

- 关系型：日历、联系人、备忘录、图库元数据等
- 键值型：缩略图、设备器件信息、包信息

### 分布式数据对象

- 分布式内存数据库之上，JS对象型封装
- 提供一个超级终端范围内的“全局变量”，为开发者屏蔽“多端协同”和“跨端迁移”场景下数据同步处理
- JS对象封装

### 用户首选项

- 一般用于记住用户的一些选择
- 开机从xml/json文件中读取所有的选项

- 用户首选项仅支持根据key存取

## | 存储组件

### | 功能

提供结构化数据的读写能力，并记录多端数据操作日志，以支撑分布式数据的一致性。支持SQLite、LevelDB等底层数据库引擎，适应不同数据存储需求。

### | 数据模型

采用KV (Key-Value) 数据模型，适合存储结构简单、业务关系较弱的的数据，具有良好的读写性能和较低的同步复杂度。

Key最大支持896Byte，Value最大支持4MB

存储组件维护数据三元组 (UserID, AppID, DB) ，确保数据隔离性。

### | 网络模型

不依赖云中心节点，是一个无中心、对等的模型。

- 任何一个同步过程总是发生在两个设备之间
- 不同设备上的同一个数据库之间发生数据同步
- 同步时都按key-value 条目，以时间先后顺序逐步同步

### | 结构化存储

存储组件提供结构化数据的读写能力。记录多端数据操作日志，用于支撑分布式数据一致性。结构化地记录数据和索引，支撑高效率的数据访问。



## | 同步组件

### | 同步模式

1. 手动同步：由应用程序调用sync接口来触发，需要指定同步的设备列表和同步模式。

同步模式分为：

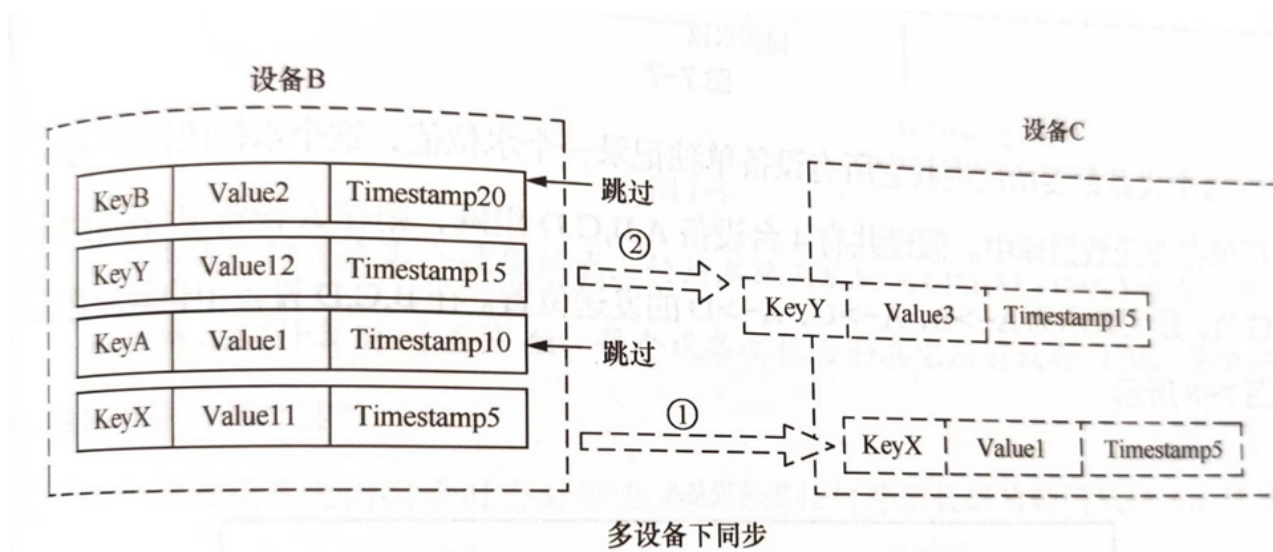
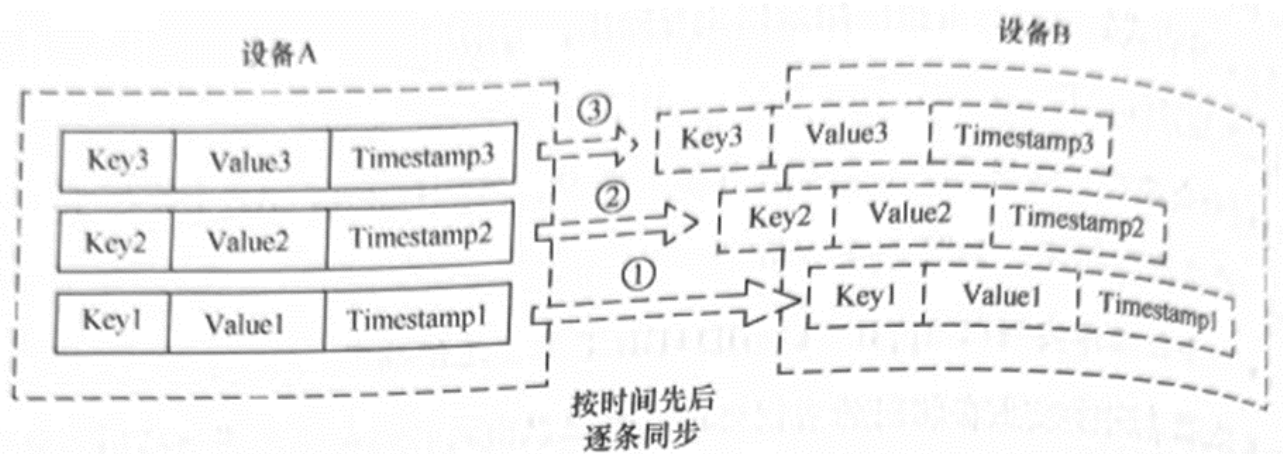
- PULL\_ONLY (将远端数据拉到本端)
- PUSH\_ONLY (将本端数据推送到远端)
- PUSH\_PULL (同时)

2. 自动同步：根据同步时机自动同步

同步时间：包括设备上线、应用程序更新数据等

### | 数据同步过程

数据同步是按条目逐步同步的，同步的顺序是由旧到新



同步水位：记录上次已同步到了哪个位置，是一个时间戳值。

## 时间同步

分布式数据库需要确定操作发生或条目产生的先后顺序，不仅仅是针对单设备，对多设备也尤为重要，它是分布式数据库能正常运行的基础。把在多设备上产生的动作先后顺序识别出来，时钟机制。

**本地虚拟时钟**：本地虚拟时钟是鸿蒙系统在分布式数据管理中采用的一种机制，旨在解决分布式系统中时间同步和时钟漂移问题，以确保数据同步的正确性和一致性。在分布式环境中，各个设备可能使用不同的时钟源，且时钟之间可能存在偏差，这种偏差随着时间累积可能影响数据同步的顺序和准确性。

特点：

- 单调递增且不回退**：本地虚拟时钟是一个在每个设备上独立维护的计数器，它的值只会单调递增，不会因为系统时间被用户调整或NTP校时等因素出现回退现象。这有助于保持时间序列的逻辑一致性。
- 步进间隔均匀**：虚拟时钟的增加步长保持均匀，确保时间的流逝在各个设备上表现得一致，有助于同步逻辑的稳定执行。
- 物理与逻辑部分结合**：虚拟时钟的值通常由两部分组成，一部分反映实际物理时间，另一部分则可能用于调整和补偿因时钟漂移、回绕等问题带来的误差，从而达到更精确的时间同步。

## 数据冲突解决

- 在执行数据更新或者数据同步写入时，不可避免遇到操作同一key值，当不同的操作在处理同一key值时就相当于数据产生了冲突。
- 本地操作的数据写入与本地已有的数据会产生本地数据冲突，本地写入会采用覆盖式处理这种冲突

- 同步操作，其他设备操作的数据也可能会操作该key值，比如本地已经有一条K1数据的情况下，远端设备又同步过来一条K1的数据，此时会产生同步数据冲突

### 解决机制：

1. 自动解决冲突机制：通过时间戳解决数据的冲突，采用Last Write Win的机制（基于提交时间戳，选择最新数据提交）
2. 用户解冲突：基于自动解冲突机制来处理的，先对数据按照时间戳机制自动解冲突，并且告诉用户冲突双方的数据信息；用户拿到冲突数据后，可再次安排数据的选取及重新写入

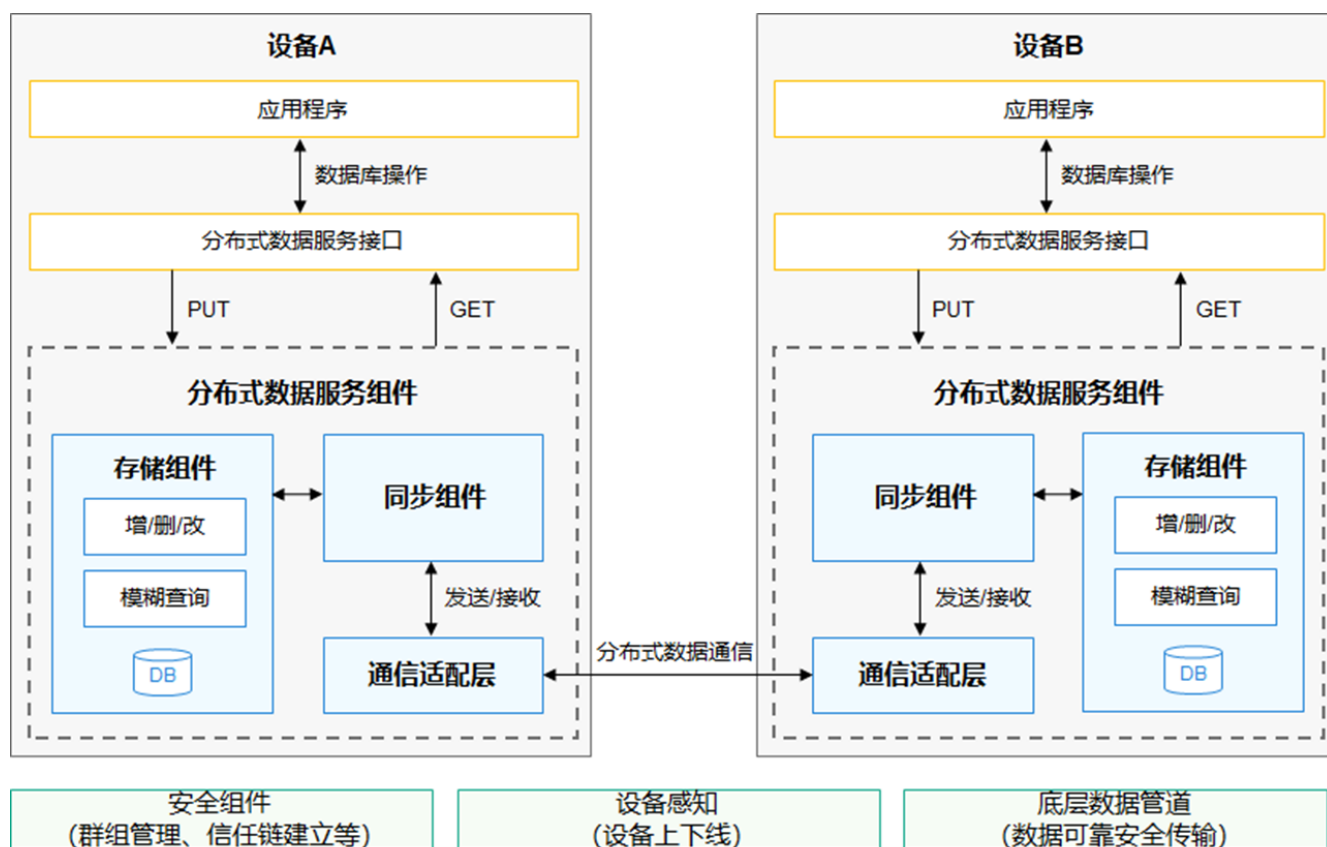
## 通信组件

维护已连接和断开设备列表的元数据，上下线信息发送给上层同步组件，通信适配层的接口将数据封装并发送给连接的设备。通信适配层的灵活设计使数据在不同网络条件和设备配置下均能高效传输，支持设备的上下线信息管理，确保同步过程的可靠性和效率。

## 鸿蒙分布式数据服务实现/使用框架

上述提到的各个组件共同组成为应用程序提供分布式数据库的服务。

如图



1. 应用程序通过调用分布式数据服务接口实现分布式数据库创建、访问、订阅功能
2. 服务接口通过操作服务组件提供的能力，将数据存储至存储设备上
3. 存储组件调用同步组件实现将数据同步
4. 同步组件使用通信适配层将数据同步至远端设备
5. 远端设备通过同步组件接收数据，更新至本端存储组件，通过服务接口提供给应用