1. **数据库**

* 系统使用的数据库包含：MariaDB、MongoDB、Redis
* MariaDB：用于关系型数据存储，其中包括用户信息、设备配置、点位信息、服务信息等。
* MongoDB：用于文档型数据存储，其中包括设备数据信息。
* Redis：用于缓存、实时点位信息存储，DB0用于网关（dc3-gateway）使用，其中包含用户鉴权相关的信息；DB1用于数据服务（dc3-dbs）使用，其中包含缓存和实时点位信息。
* 目前第一版本数据服务设计为云端统一存储和查询，第二版将调整为边缘测数据库，建立数据库集群服务，在云端集中存储和查询，保证数据高可用性。
* **约定**：进入service层的数据对象不能为空（传入对象不为空，子对象和属性可以为空）。
* **约定**：系统只存在Post和Get请求，其他类型请求一律禁止，修改，删除，查询列表使用Post请求，查询单条数据使用Get请求。

1. **容器**

* 全容器部署，MongoDB需要搭建集群，确保数据高可用性，第一版暂不使用volumes，第二版需要实现volumes，进行数据管理

1. **用户登录**

* 用户登录可以使用用户名、手机号、邮箱进行登录，登录请求之后后端会返回token，除了登录以外的所有请求操作需要在header中传入token，每个token的有效时间为60分钟，当token失效时，客户端需要进行再次登录进行授权。
* 用户密码需要通过获取对称加密密钥privateKey进行加密传输（第一版：密码明文传输；第二版：密码使用对称密钥加密传输）。
* 数据库中的用户信息需要进行加密存储，需要加密的字段有用户名、手机号、邮箱、密码等关键用户信息（第一版：密码加密存储，其他信息明文存储；第二版：全部使用对称加密存储）。

1. **统一数据封装**

* 统一返回数据格式Json，数据包含3个字段（ok,message,data），其中ok表示true和false，分别表示请求成功和失败；message主要用于ok=false状态下的错误描述；data用于返回数据体，泛型。
* 微服务之间不采用数据封装，对外提供接口的返回数据需要有统一的拦截进行数据封装。
* dbs服务不对外提供接口服务，仅用于微服务间交互。

1. **分页逻辑**

* 采用Mybatis-Plus的mapper文件的select注解分页功能进行分页，数据返回格式为IPage<T>。
* 分页默认从第一页开发，每页10个元素。
* 分页信息包含：页码、行数、排序（升序和降序，对应true和false，默认为升序）。
* **约定**：由于ID是自增的，涉及查询关系型数据库数据排序时，时间排序约定使用ID；支持排序的字段应该是数字或者比较小的字符串；查询列表类均需要进行分页查询，不设置分页参数，采取默认分页机制返回数据。

1. **视频流转码服务**

* 提供新建、删除、修改、查询、停止、启动、查询运行状态功能，第二版需要提供日志输出功能。
* 新建：通过接口新建一个转码任务。
* 删除：通过接口删除一个转码服务，如果任务正在运行中，不能删除。
* 修改：通过接口修改一个转码服务，如果任务正在运行中，不能修改。
* 停止：停止一个正在运行的任务，不能操作已停止的任务。
* 启动：启动一个停止运行的任务，不能操作已启动的任务。
* 查询：提供任务列表查询，单个任务查询接口。
* 创建任务个数不设置上限。