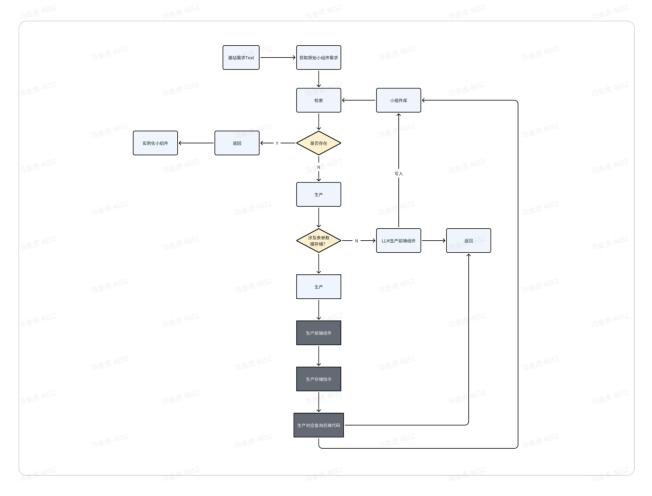
小组件灵活存储的后端系统设计

背景

为丰富生成网站和局部修改的丰富度和能力,设计了小组件功能,这里要设计一套系统,来存储各种各样即时生产的小组件产生的各式异构数据。同时,要考虑这些存储的查询,包括写接口和读接口设计。

业务流程



从上述一般的业务流程来看,不涉及后端存储的小组件这里不用care,剩下的只关注涉及后端存储的小组件,也即上述黑色的3个流程框。

无论是预先生成还是实时生成的组件,都会被缓存,然后后续在小组件库中被检索。

这里生产的小组件还分3种情况。

- 1. 预先设计生成
- 2. 实时LLM生成
- 3. 提供了较为通用的后端接口,供开发者使用来生产具有数据存储能力的小组件

这3种情况对应的解决方案,最好的做法当然是一种通用手段能覆盖3种需求,但是实际上我们可能很难找出这种方案,甚至这种方案不一定是综合来看最划算的,尤其是考虑到他们的优先级以及实施的工期问题。下面分别讨论3种情况的解决方案,综合共性以及优劣,给出最终的选型建议参考。

总体思路:

第一步: 各种方案是怎么回事讲清楚

第二步: 总结每个方案的特点(从角色和可做的操作(生成、更改)两方面,组件类型和组件实例是

不同的)安全性、可操作性、生成效率、数据的操作(存、查)、指出生成阶段、更新阶段等。

第三步: 列表格直观比较

第四步: 选型决策

选型及设计论述

主要涉及两方面,一是后端存储,一是后端业务逻辑

这种模式的特点是,可以预先设计异构的数据如何处理,以及接口如何处理,写好代码,被选中的小组件直接调用即可。现在我们基于这个最简单的方式来设计方案。

这里主要面临两个任务,一是如何设计存储Schema,来存储各种异构的数据,同时保障一定的查询效率;二是提供对这些异构数据的操作接口——路径、参数及返回值设计。

传统的页面数据存储一般是提出功能->抽象数据->设计表结构->按功能写查询接口->优化索引->考虑数据规模分库分表等等。基本是一个功能或者数据结构对应一套从数据设计到查询设计的完整链路,由研发人工保障整体的功能、效率、安全性。

后来出现一种低代码的平台设计场景,即预设一些基本组件,组件可自由可视化拖拽组合,灵活实现一定的业务场景。这里的组合,对应于前端的组件JSON组合,也对应于后端的接口参数更改、数据结构灵活变更、查询内容的变更。

以上描述来看,我们的场景和上述有一定的相似性,(但是实际上形似而神不似,差别还是挺大,低代码平台很多固化的场景和流程, 使用的协议和设计可能特别复杂,我们没有这个时间和团队来支持)除了不太有组件组合场景外,其他的诸如业务场景自由度比较高,异构数据较多,而且要实现从查询接口到数据最终落地的一整个链路等特性比较类似。

这里列出几种常见的解决思路。

传统开发法

即每个相同结构类型对应一个表和处理接口。行模型是数据库设计的标准数据建模技术。

因为组件也是一个个开发的,每个小组件对应一个不同的业务逻辑。所以每开发一个组件,即开发一套对应的后端组件逻辑元组(接口路径-接口参数-流程-数据结构(数据表)-数据库)。**即同一类型的小组件使用一个后端逻辑元组**

这种方法就和低代码完全不同,因为低代码的定义就限制了它是可视化编辑,高度复用,而这个方法就是传统的代码编辑与开发。如果有复用,也只是相似的业务组件可以小范围人工抽象,其性价比似

乎还比不上一个组件一个开发元组。

直接小组件ID加页面ID查询,会完全泄露隐私(需要有相应的鉴权手段)。

优点:

开发逻辑简单,流程明晰,高度支持小组件的几乎所有功能

缺点:

用户端几乎无法修改(除非做预留设计),复用度很低,开发量大,不灵活,每个小组件都需要对应的后端提供开发。

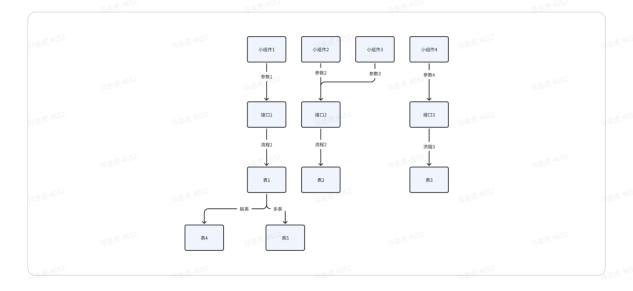
特点总结:

前端:按类型生产小组件

新建: 研发手动设计

修改: 研发手动设计和修改

其他人: 无法修改



使用单文件

客户的某个页面的小组件(唯一ID)对应一个文件,类似CSV的文件,当然也可以做成大Json. 每次读取都是读整个Json, 传给前端,所有查询过滤等逻辑均做到前端。后端只需要接整个数据整个更新或者整个返回即可。更新时需加分布式锁,所有针对单个文件的写操作都需串行。一个组件实例使用一个文件,而不管类型是否相同

如果要加更复杂筛选功能,比如权限控制部分数据行或者列的返回,后端的工作则大增,资源消耗尤其是内存消耗会飙升。

优点:

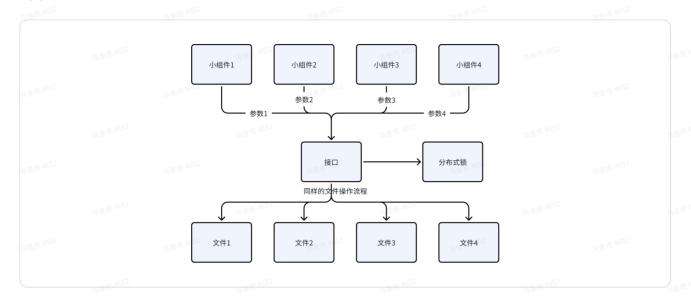
接口完全复用,实现简单,贴近前端,所有数据操作逻辑均由前端实现;用户端可以直接修改结构(需前端处理历史数据);数据天然隔离。

缺点:

无法做权限校验等行级别逻辑操作,性能低下,存在一定安全风险

无法做多表联查等对应的数据库操作,这样一来只能支持平铺的单实体逻辑

难点:客户端存在并发的情况下,不能使用数据快照来做更新数据操作,而是使用后端语义的 append/delete等做单条处理,后端加分布式读写锁来处理数据语义



横向设计法

程序生成动态实体,行模型是数据库设计的标准数据建模技术

即每新建一个组件实例,则新建一个表(或者多个表,即表组)(把单文件直接用表实现),不用管复用的事。

那么组件实例对应的结构更新对应着表的schema更新,新建组件需要提交表单数据的Schema,更新表单结构需要给出Modify,对于数据库总之是DDL操作。

这里理论上是可以做联表、多表查询等复杂操作的,毕竟是数据库实现的,但是无法做自动化或者自动化的代价太大。

EAV(行代替列)

E即实体,也即我们需要存储的对象,这里要支持的对象是各种各样的;A即属性,各种不同对象的不同属性,同一对象的属性也会变更;V即某个对象的某个属性的真实值。

实体表E

| 实体ID | 实体类型 | 实体名称 |
|-----------------------------|---------|------|
| 马 宜^{克 A652} | Product | 商品1号 |
| 2 冯金虎人 | Post | 帖子2号 |

属性表A

| 属性ID | 属性名称 | 值类型 |
|----------|------|-----------------|
| 1 冯金虎 4 | 评论内容 | string |
| 2 | 更新时间 | date |
| 3 | 价格 | int 海雏虎 4652 |

值表V(优化方式: v-{type})

| 实体ID | 属性 | ID & 8. 4652 | 值 | |
|--------------------|-------------------------|--------------|------------|-------|
| 2 | ⁵⁵² 1 | | hello | |
| 2 465 ² | 2 | | 2014-08-29 |) 193 |
| 1 | ⁵ 3 | | 15 at 4652 | |

优点:

• 灵活性高:能够轻松添加或删除属性,无需修改表结构。

• 扩展性强: 支持无限多的属性,满足复杂数据管理的需求。

• 节省空间:避免了传统数据库中的宽表问题,减少了存储空间浪费。

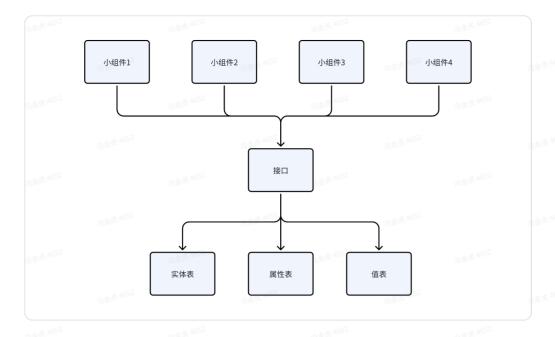
缺点:

• 查询复杂:由于数据分布在多个表中,查询时需要进行多次表连接,可能影响性能。

• 数据完整性维护困难:在更新或删除数据时,需要确保数据的一致性和完整性。

• 性能问题: 在数据量较大或者列较多时,查询性能低下。

在我们这个场景下的使用方式:



新建小组件:

表都是建好的,后端不用做任何动作

更新小组件:

表还是建好的,后端不用做任何动作

某个小组件提交了新建数据(一个Json,包含数据本体和一些固定meta),后端需要固定流程处理, 先新建实体,然后查找到属性或者新建属性,然后组成值表的插入行数组,执行值表写入

所有属性都是跟随具体的实体记录变动的,所以也无所谓结构的变更了,每个实体本身在值表里就对应了一组具体的属性,已经是完全的Schema-less了。

查询需要3表联动查询,这里就不展开了。

这里还需要做的是实体表里要加上Type作为不同类型(以前的Table,比如是一个具体的小组件)的载体。

安全和隔离层面看,每个客户或者每个网站可以使用一套EAV表3元组。

元信息+宽表

元信息 (Metadata):元信息存储与实体、属性相关的配置、定义等信息。它描述了表结构的动态变化,使系统可以理解和处理不同的字段和数据类型,而不必直接修改数据库结构。

宽表 (Wide Table):宽表是在数据库中提前定义了足够多的列,以适应可能出现的各种字段需求,它用来存储不同元信息的实际数据记录(实体)

借鉴salesforce的metadata-driven model

https://architect.salesforce.com/fundamentals/platform-multitenant-architecture

metadata表

| tenant_id | entity_type | field_id | filed_name | data_type | column_name |
|-----------|-------------|----------|------------|-----------|-------------|
| | | | | | |

| 3. 1 R 4652 | Post | 1 % | title | str mark 4652 | column_1 |
|--------------------|----------|------------|-----------|-------------------------|----------|
| 1 冯金 | Post | 2 9金虎 4652 | content | text | column_2 |
| 2.4652 | Commnent | 1 | rating | int 冯金虎 4652 | column_2 |
| 2 | Commnent | 2号金麂 4652 | text | text | column_4 |
| 2 4652 | Product | 2 | price | int ^{冯金虎 4652} | column_2 |
| 2 | Product | 1 冯金虎 4652 | name (38) | str | column_1 |

data表

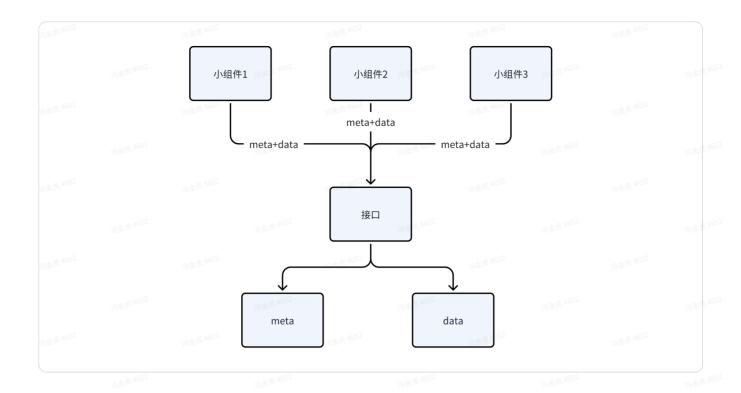
| tenant_id | entity_id | entity_typ | column_1 | column_2 | column_3 | column_4 | column |
|------------------------------|-----------------------|------------|---------------|------------------------|-----------|--------------------|-----------|
| 2 1 ^{8,4652} | 1 冯金聚46 | Post | "Title 1" | "Content xx" | NÜLL | NULL 3 4 4 4 5 5 2 | NULL 4652 |
| 1 3金素 4652 | 2 冯金庆46 | Product | "P Name 1" | 1500 | NULL | NULL 海傘原 4652 | NULL |
| 2 周金素 4652 | R 4652 3 海錠便 46 | Comment | NULL 332 A532 | 5 ^{ac/R} 4652 | NULL NULL | "Commen t" | NULL |

优点:

- 1. **灵活性高**:通过元数据驱动的设计,可以动态添加和修改实体类型和字段,而无需更改底层数据库结构。
- 2. **性能较好**:共享表结构避免了 EAV 模型中频繁的 JOIN 操作,提高了查询效率。

缺点:

- 1. 复杂性增加:维护元数据表和数据表的映射关系较为复杂,增加了系统管理难度。
- 2. 数据冗余和存储浪费: 宽表可能包含大量未使用的列,导致存储空间浪费。
- 3. 扩展受限: 虽然灵活,但某些情况下不如 EAV 模型在处理大量可变字段时那么高效。



文档型数据库

这个说白了就是支持大Json的存储。Mysql/MongoDB/ES都可以一定程度上支持。 类似以下存储结构

| ID | Name | Туре | Data | | | | |
|--------|----------------------|----------|----------|----------|----------------|-----------|----------|
| 1 | Product1 | wid-1 | | 冯金虎 46 | 52 | 冯金虎 4652 | |
| | 冯金虎 4652 | | 冯金虎 4652 | 1 { | | | |
| | | | | 2 | "Name":"电脑' | ١, | |
| | 冯金虎4552 | 冯金虎 4652 | | 3 四金虎46 | "CPU":"Intel | 11,金成4652 | |
| | 冯金虎 465 ² | | 冯金虎 4652 | 4 | "GPU":"4090" | | |
| | Ban | | Par. | 5 } | | | |
| | 冯金虎 4552 | 冯金虎 4652 | | | | | |
| 2 4652 | Post1 | wid-2 | 冯金虎 4652 | | | | |
| | | | | 1 { | | | |
| | 冯金虎 4552 | 冯金虎 4652 | | 2 冯金虎 46 | "ID": 1, | | |
| | 冯金虎 4652 | | 冯金虎 4652 | 3 | "Content": " | | |
| | | | | 4 | "IP": "187.1 | 63.9.1" | |
| | 冯金虎 4652 | 冯金虎 4652 | | 5 | | | |
| 金虎4652 | 冯金虎 4652 | | 冯金虎 4552 | | <u>冯金庆4652</u> | | <u> </u> |
| 3 | Post2 | wid-2 | | | | | |
| | 冯金虎4552 | 冯金虎 4652 | | 1 { | | | |
| | 冯金虎·4652 | | 马金虎 4652 | 2 | "ID": 2, | | |
| | | | | 3 | "Content": " | world", | |

```
4 "IP": "xxx.xxx.xxx",
5 "Sex": "Male"
6 }
```

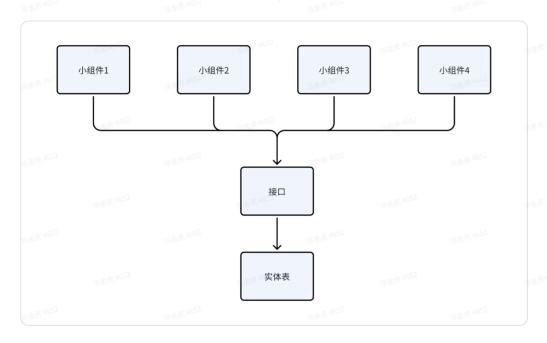
新建小组件:

这里也是新建时没有Schema限制,所以后端不用做任何设计;

更新小组件:

如果还没有记录数据,则无需做任何操作;如果已经有记录,则需要根据具体的变更把相关的数据遍历一遍,按要求修改对应的JSON结构;

某个小组件提交新数据,则直接插入;更新新数据,也是直接替换相应记录。



安全和隔离层面看,每个客户或者每个网站可以单独建一个表。

优点:

使用简单,操作简单

缺点:

性能是灾难,基本无法支持自适应的联表查询

不同研发方式的小组件流程、支持度、特点

这里的研发专指跟小组件存储的网络请求和数据存、取操作相关。

操作:

- a. 小组件新建
- b. 小组件更新

- c. 数据存
 - d. 数据查取

角色:

- a. 内部研发
- b. 建站客户
- c. 建站站点用户
- d. 外部研发

组件维度:

- a. 组件类型
- b. 组件实例

阶段:

- a. 新建页面
 - b. 更新页面
 - c. 托管使用页面(使用时)

官方预先生成的组件

特点:人工参与,相对可控

| | 传统开发法 | 使用单文件 | 横向设计法 | EAV | Meta driven | 文档数据库 |
|----|------------|--------|----------------------|--------|------------------|---------|
| 前端 | 内部研发:按一 | 内部研发:按 | 内部研发:按 | 内部研发:按 | 内部研发:按 | 内部研发:按 |
| 如何 | 个小组件类型建 | 一个小组件类 | 一个小组件类 | 一个小组件类 | 一个小组件类 | 一个小组件类 |
| 新建 | 一个数据结构、 | 型建一个数据 | 型建一个数据 | 型建一个数据 | 型建一个数据 | 型建一个数据 |
| 小组 | 请求接口、参数 | 结构、请求接 | 结构、请求接 | 结构、请求接 | 结构、请求接 | 结构、请求接 |
| 件 | 构造与响应解 | 口、参数构造 | 口、参数构造 | 口、参数构造 | 口、参数构造 | 口、参数构造 |
| | 析;小组件代码 | 与响应解析的 | 与响应解析的 | 与响应解析的 | 与响应解析的 | 与响应解析的 |
| | 按类型存储; | 代码;小组件 | 代码;小组件 | 代码;小组件 | 代码;小组件 | 代码; 小组件 |
| | 其他角色无法参 | 代码按类型存 | 代码按类型存 | 代码按类型存 | 代码按类型存 | 代码按类型存 |
| | 与 | 储供检索引。 | 储; | 。储; | 储;4652 | 储; |
| | — J | 用; | PS 32 | | 13 m | |
| | | 其中与后端交 | 冯金虎 4652 | | 冯金虎 4 | |
| | | 互的接口是同 | | | | |
| | | 一个,除了小 | 冯金虎4 | | 冯金虎 4652 | |
| | | 组件独有的 | 冯金虎 465 ² | | 冯金虎 ^A | 652 |

| | | data结构,其 他公共参数与 | 冯金虎 46 ⁵² | | 冯金虎名 | :52 |
|----------|-----------------|--------------------------|---------------------------------------|---|---|--------------------------|
| | | 响应都一致。 | 冯金麂 4 | | 冯金虎 465 ² | 冯金虎 4652 |
| | | | 冯金虎 4652 | | 冯金庆 ⁴ | 552 |
| | 内部研发:一个 | 内部研发:后 | 内部研发:后 | 内部研发:后 | 内部研发:后 | 内部研发:后 |
| | 小组件类型对应 | 端提供通用接 | 端提供通用接 | 端提供通用接 | 端提供通用接 | 端提供通用接 |
| | 一个表或者多个 | 口,所有小组 | 口,所有小组 | 口,设计好3 | 口,设计好2 | 口,所有小组 |
| | 表, 手动设计建 | 件类型共用一 | 件类型共用一 | 张通用表建 | 张通用表建 | 件类型共用一 |
| | 表; | 个接口组;接 | 个接口组;接 | 模,所有小组 | 模,所有小组 | 个接口组;接 |
| | 一个小组件类型 | 口参数、返回 | 口参数、返回 | 件类型共用一 | 件类型共用一 | 口参数、返回 |
| | 对应 一组增删改 | 值都是通用 | 值都是通用 | 个接口组;接 | 个接口组;接 | 值都是通用 |
| | 查接口 和对应参 | 的;也就是写 | 的;也就是写 | 口参数、返回 | 口参数、返回 | 的;也就是写 |
| 后端 | 数与响应; | 好通用接口和 | 好通用接口和 | 値都是通用 | 値都是通用 | 好通用接口和 |
| 加姍 如何 | 其他角色无法参 | 流程后后端就可以不用参与 | 流程后后端就可以不用参与 | 的;也就是写 | 的;也就是写 好通用接口和 | 流程后后端就可以不用参与 |
| 新建 | 与 | 可以不用参与 新建了; 没有 | 可以不用参与 新建了。 | 好通用接口和 流程后后端就 | 対 通用接口和 流程后后端就 | 可以不用参与 新建了; 没有 |
| 小组 | | 数据存储建模 | 冯金虎 4652 | 可以不用参与 | 可以不用参与 | 数据存储建模 |
| 件 | | 这一阶段,在 | 其他角色无法 | 新建了。 | 新建了。 | 这一阶段,在 |
| | | 流程运行写入 | 参与; | | 冯金虎 400 | 流程运行写入 |
| | | 时才会有数据 | 冯金虎 465 ² | 没有业务数据 | 没有业务数据 | 时才会有数据 |
| | | 建模控制。 | | 存储建模这一 | 存储建模这一 | 建模控制。 |
| | | | 冯金虎 ⁴ | 阶段,在流程 | 阶段,在流程 2-4-5-2-3-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1-1 | 冯金虎 4652 |
| | | 其他角色无法 | ~ # 465 ² | 运行 <u>查询</u> 时才 | 运行 <u>写入</u> 时才 | 其他角色无法 |
| | | 参与; | 冯金匹。 | 会有隐式的建 模控制。 | 会有隐式的建 模控制。 | 参与; |
| | | | · · · · · · · · · · · · · · · · · · · | | 冯金虎 4652 | 冯金虎 4652 |
| | | | | 其他角色无法 | 其他角色无法 | |
| | 冯金虎 | 1654 | 冯金虎 4654 | 参与;************************************ | 参与; | 654 |
| 前端 | 内部研发更新小 | 内部研发直接 | 内部研发更新 | 内部研发更新 | 内部研发更新 | 内部研发更新 |
| 如何 | 组件库对应类型 | 更新组件原型 | 小组件库对应 | 小组件库对应 | 小组件库对应 | 小组件库对应 |
| 更新 | 代码,其他人无 | 代码 | 类型代码,其 | 类型代码,其 | 类型代码,其 | 类型代码,其 |
| 小组 | 法参与 | | 他人无法参与 | 他人无法参与 | 他人无法参与 | 他人无法参与 |
| 件类 | | | A652 | | ∞ A | 552 |
| 型 | 冯金郎 | | 周金郎 4 | 冯金风 | 冯金灰" | |
| 后端 | 内部研发更新小 | 不用更新 | 不用更新 | 不用更新 | 不用更新 | 不用更新 |
| 如何 | 组件库对应后端 | | 口令度 4652 | | de A | 552 |
| 更新 | 接口和存储部 | | 治 並11 | | 冯兹庇 | |
| 小组 | 分,其他人无法 | | 坦金虎 A | | 冯金虎 4652 | 四金虎 4652 |
| 件类 | 参与 | | - 1652 | | | 152 |
| 型 | 冯金虎 | Ann. | 周金虎 402 | 冯金虎 4002 | 冯金虎 ^A | |
| wegi | 从建站站点客户 | 匹配到小组件 | 匹配到小组件 | 匹配到小组件 | 匹配到小组件 | 匹配到小组件 |
| c建/ | 的建站chats里 | 原型,实例化 | 原型,实例化 | 原型,实例化 | 原型,实例化 | 原型,实例化 |
| 改页 | 提取意图,去小 | 小组件实例, | 小组件实例, | 小组件实例, | 小组件实例, | 小组件实例, |

| 面时 如何 | 组件库向量匹配(如RAG)到合 | 与网页代码— | 与网页代码一起保存; | 与网页代码一起保存 | 与网页代码一起保存 | 与网页代码一 起保存 |
|----------|-----------------|----------|----------------------|-----------|------------------|---------------|
| 新建 | 适的小组件类 | 海金茂 4652 | 冯金虎 | NE IN IT | 冯金虎 4652 | 冯金虎 4652 |
| 小组 | 型,嵌入网页区 | | 同时实例化时 | | 确定结构后需 | 确定结构后需 |
| 件实 | 块,新建一个小 | | 前端需要向后 | 冯金虎 4652 | 要申请后端 | 要申请后端 |
| | | | 端发起数据表 | AK52 | DDL建表操 | DDL建表操 |
| 例 | 组件实例; 本质 | | 结构创建申 | | 作,后端接口 | 作,后端新建 |
| | 上是使用或实例 | | 请,后端接口 | ·□金虎 4652 | 组中的DDL功 | □ |
| | 化,而不是新 | | 组中的DDL功 | | 能执行元信息 | collection |
| | 建。 | | 能执行数据表 | 4652 | 添加记录(实 | (文档型数据 |
| | | | 创建。 | | 际上是把DDL | 库比较灵活, |
| | | | 冯金虎 4652 | 冯金虎 4652 | 转换成元信息 | 这里实际上也 |
| | | | | A652 | 表的DML) | 可以一个组件 |
| | | | 冯金郎。 | | 冯金既 | 型以 1组 1 |
| | | | 冯金虎 46 ⁵² | 冯金虎 4652 | 冯金虎 ^A | 652 |
| | | | | | | collection, |
| | | | 冯金虎 | 452 | 冯金虎 4652 | 转换到运行时 |
| | | | (5) | :52 | | 就是实例化时 |
| | | | 冯金虎4652 | 冯金虎 4602 | 冯金虎 ^A | 可以先匹配信 |
| | | | | nk52 | = 4652 | 息表,看看是 |
| | | | 冯金虎* | | 冯金既 | 否已存在 |
| | | | 冯金虎 4652 | 冯金虎 4652 | 冯金麂 ^A | collection, |
| | | | | | | 没有则新建, |
| | | | 冯金虎 4 | 4652 | 冯金虎 4652 | 否则使用旧 |
| | | | _ 1657 | - 4657 | | 的) |
| | 7큐 하나 후 라 디 와 # | 1.42件京何三 | | | | |
| | 建站客户只能替 | 小组件实例可 | 小组件实例可 | 小组件实例可 | 小组件实例可 | 小组件实例可 |
| | 换小组件类型, | 以开放修改数 | 以开放修改数 | 以开放修改数 | 以开放修改数 | 以开放修改数 |
| wegi | 新建一个实例, | 据结构的前端 | 据结构的前端 | 据结构的前端 | 据结构的前端 | 据结构的前端 |
| c建 | 删除原先的实 | 接口给建站客 | 接口给建站客 | 接口给建站客 | 接口给建站客 | 接口给建站客 |
| 页 | 例,无法更新小 | 户,只需要前 | 户,客户可以 | 户,客户可以 | 户,客户可以 | 户,只需要前 |
| 面- | 组件结构。 | 端更新数据相 | 用来修改结 | 用来修改结构 | 用来修改结构 | 端更新数据相 |
| 发布 | | 关结构并保存 | 构; | 后保存实例代 | 后保存实例代 | 关结构并保存 |
| 后无 | | 即可。 | 前端需要把结 | 码即可; | 码即可; | 即可。 |
| 历史 | | | 构的diff转化为 | | 确定结构后需 | 1950年 |
| 数据 | | | DDL需求向后 | 冯金虎 4652 | 要申请DDL建 | 652 |
| 时如 | | | 端发起数据表 | | 表操作,后端 | |
| 何更 | | | 结构修改申 | 4652 | 接口组中的 | 冯金虎 4652 |
| 新小 | | | 请,后端接口 | ·□全虎 4652 | DDL功能执行 | 552 |
| 组件 | | | 组中的DDL功 | 1-22 Jan | 元信息修改记 | |
| 实例 | | | 能执行数据表 | 4652 | 录(实际上是 | 冯金虎 4652 |
| 结构 | | | | | | 污巫" |
| -H 13 | | | 创建。 | 冯金虎 4652 | 把DDL转换成 | 652 |
| | | | | | 元信息表的 | |
| | | | 冯金虎。 | 1,52 | DML) | 冯金虎 4652 |
| | | | | | | |
| wegi | 建站站点客户只 | 小组件实例可 | 小组件实例前 | 小组件实例前 | 小组件实例前 | 小组件实例可 |

| = # | 피 호계 소호 | | | 76 BD = 1 - A652 | | |
|------------|-----------|------------|--------------------|------------------|------------------|---------------------------|
| 面发 | 型,新建一个实 | 据结构的前端 | 改,然后申请 后端DDL修改, | 改即可; | 改,然后申请 后端DDL修 | 据结构的前端 |
| 布后 | 例,删除原先的 | 接口给建站客 | Lan A | 至于后端的数 | ~ # 4652 | 接口给建站客 |
| (产 | 实例,无法更新 | 户,但是需要 | 后端执行后 | 据结构要不要 |) 改; | 户,但是需要 |
| 生历 | 小组件结构。 | 同时提供处理 | (不一定能成 | 修正,这里我 | 后端修改DDL | 同时提供处理 |
| 史数 | | 历史数据的办 | 功,部分有数 | 倾向于不修 | 仅修改元信息 | 历史数据的办 |
| 据) | | 法,依然在前 | 据的DDL修改可 | 正。如果是修 | 表记录即可, | 法,依然在前 |
| 如何 | | 端处理,后端 | 能无法成功) | 正结构的话要 | 实际上是DML | 端处理,后端 |
| 更新 | | 就是一个文件 | 将结果给前 | 执行批量的 | 操作。 | 就是一个 |
| 小组 | | proxyo | 端;后续前端 | DML操作。这 | 历史数据不用 | proxy。 |
| 件实 | | 历史数据按照 | 会按照新数据 | 里的DML操作 | 修正,因为每 | 历史数据按照 |
| 例结 | | 前端的历史数 | 结构渲染。 | 既改了结构, | 次查询都会先 | 前端的历史数 |
| 构 | | 据处理办法处 | 历史数据按照 | 也修正了历史 | 从meta表中 | 据处理办法处 |
| | | 理 | 新的数据结构 | 数据。 | 新建数据 | 理。 |
| | | | 格式处理 | 不过即使后端 | 別達致語 Schema. | |
| | | | | 不修,查询时 | Schema. | |
| | | | | 组装返回前端 | | |
| | | | | 也会按照前端 | | |
| | | | | 新的结构使用 | | |
| | | | | 数据的。 | | |
| | 170 | 700 | | 交入1/日口20 | 100 | 75.00 |
| 数据 | 1. 数据按小组 | 1. 数据按小 | 1. 数据按小组 | 1. 数据全部 | 1. 数据物理 | 1. 数据按照 |
| 存储 | 件类型存储 | 组件实例 | 件实例存储 | 打平存 | 层面全部 | 小组件实 |
| 特点 | (表组与类 | 存储(文 | (表组与实 | 储,无任 | 打平存 | 例存储 |
| | 型映射) | 件与实例 | 例映射) | 何映射 | 储,无任 | 2 (也可以 |
| | 2. 多租户共享 | 映射) | 2. 多租户天然 | 2. 多租户共 | 何映射 | 按照类型 |
| | 数据库,共 | 2. 多租户天 | 隔离数据 | 享数据 | 2. 实际元信 | 存储) |
| | 享数据表, | 然隔离数 | 3. 有数据库 | 库,共享 | 息(隐式 | 2. 多租户共 |
| | 仅通过租户 | 据 | DDL操作 | 数据表, | Schema | 享数据库 |
| | ID加以区分 | 3. 无数据库 | u金虎 ^A | 2 仅通过租 |)是与组 | 3. Schema- |
| | 3. 只有DML, | 操作 | 4. 可以实现较 | 户ID加以 | 件实例映 | less, 所有 |
| | 无DDL | | 复杂的数据 | 区分 | 射的 | Schema- |
| | | 4. 数据是孤 | 联查或SQL | 。 3. 无任何数 | 3. 多租户共 | less的优 |
| | | 国 海金属 4652 | (需提前定 | 据库DDL | 享数据 | 缺点它都 |
| | | 5. 查询性能 | 义场景协 | 操作,只 | 库,共享 | 有 |
| | | 低下 | 议) | 有DML操 | 数据表, | |
| | | | 5. 查询性能高 | 作 | 仅通过租 | 4. 性能很差 |
| | | | | 4. 存查操作 | 户ID加以 | (无索引 支持), |
| | | | | 4. 任宣撰TF 引擎需仔 | 区分 | 又 _持), 不支持大 |
| | | | | 细设计, | 4. 无任何数 | 数据量 |
| | | | | 存储高度 | 据库DDL | |
| | | | | 灵活 | 操作,只 | 5. 无高级查 泡末持 |
| | | | | 57. | 有DML操 | 询支持 |
| | | | | | 作 | |
| | | | | | | |
| | | | | | | |

| | 冯金原 冯金虎 ⁴⁵⁵² 冯金贵 | A652 海金虎 A652 A652 | | 5. 查询性能 中低,写 入性能中 低 | 存查操作 引擎需仔 细设计, 存储高度 灵活 | 52 海金原 A ⁶⁵² |
|---------------------|--|---|--|--------------------------------------|--|-----------------------------------|
| | | 冯金虎 4652 冯金虎 4652 | | 6. 无法支持 高级SQL 查询 | 6. 查询性能中高等,写入性能 | 冯金虎 ^{A652} |
| | | 海金虎 4652 A652 | | | 中高等7. 能执行一 | 冯金虎 A652 |
| | | 冯金虎 4652 | | | 定的高级 SQL,但 是特别高 | 冯金虎 A652 |
| | | A652 冯金虎A652 | | | 级的也支 持不了 8. 唯一索引 | 552 冯金虎 A652 |
| | | A652 海金原 A652 | | | 等东西需要从逻辑 层面或者 加表支持 | 52 冯金凤 ⁴⁵⁵² |
| | 冯金宗 | 1652 | 冯金庚 465 ² | 冯金虎 4652 | 海金熊 | 152 |
| 流程 控制 特点 | 一个类型一个流程开发,前后端都有高度控制的流程逻辑,支持很复杂的业务逻辑流程控制 | 后端通用流程,只支持通用的数据增删改查;只能依赖前端做特性数据逻辑处理; | 后端通用流程,一是DDL流程,二是通用的数据增删改查;三是通过约定协议可做一定的联查 | 后端通用流程,主要就是EAV映射转换流程,即列转行这个引擎流程 | 后端通用流程,主要就是把表结构设计到元信息里, 找到宽表(数据表)的列映 射,实时重建 | 后端通用流程,支持通用的数据增删改查,也能支持简单聚合查询 |
| | | 也可以提前做 好简单流程 库,供组合选 | | | Shema和 ORM. | 52 |
| | | 择(后面通 用,不再赘 述) | | | 冯金虎 4652 冯金虎 ⁴ | 另金虎 A652 |
| 写入 数据 (使 用 | 提交到小组件类型固定的后端接口,写到类型对应的数据表 | 后端控制好读 写锁,采用 append/delet e/update模式 | 直接执行关系 型数据库的 ORM写入即可 | 前端提交正常 的数据结构, 后端需要通过 列转行引擎, | 前端提交正常 的数据结构, 后端需要通过 组件实例ID从 | 前端提交正常 的数据结构, 后端直接写入 对应的 |
| 时) | | 单条或批量更 新,不是全量 | | 插入3张固定 表。 | 元信息记录找 到宽表映射 | collection |

| | 为金虎和52 冯金虎和52 冯金虎 | 性问题),可 以类比即时设 计的协同编 辑,只是不用 实时广播。 | 冯金虎 4652 冯金虎 ^A | | 按照ORM写入 数据 | |
|---------------------------|--|---|--|--|---|--|
| 查询 数据 (使 用 时) | 提交到小组件类型固定的后端接口,从类型对应的数据表读取 | 后端控制好读 写锁,采用全 量或者分页返 回数据快照 | 直接执行关系 型数据库的 ORM查询即可 | 需要通过EAV 映射引擎查询 出行,转列, 组装成正确的 数据结构后返 回前端 | 通过组件实例 ID从元信息里 找出Schema 映射列,然后 按照ORM从宽 表里取数据 | 后端直接查询 对应的 collection |
| 多租 户据 离方 | 共享数据库,共享数据表,仅通过租户ID加以区分 | 数据库级别隔 离 | 数据表级别隔离 | 共享数据库, 共享数据表, 仅通过租户ID 加以区分 | 共享数据库, 共享数据表, 仅通过租户ID 加以区分 | 数据表级别隔离 |
| 数据 一致 性 | 数据库正确使用 保证了数据的一 致性 | 需要前后端使 用非快照+读 写锁控制一致 性 | 关系型数据库 正确使用保证 了数据的一致 性 | 关系型数据库 一定程度保证 了一致性,但 是历史数据的 一致性很难保 证 | 数据一致性要 后端保障,需 要设计特殊的 规则来支撑 | 数据一致性要 后端保障,通 过文档型数据 库操作可以有 基本保障 |
| 安全 性 | 多租户仅通过租户ID加以区分数据,有一定安全隐患; 现实必须要解决的一个前提问 | 1. 多租户的 数据间是 安全隔离 的 2. 需要对组 件实例鉴 | 1. 多租户的数据间是安全隔离的 2. 需要对组件实例鉴权,以防越页面 | 1. 多租户仅 通过租户 ID加以区 分数据, 有一定安 全隐患; | 1. 多租户仅 通过租户 ID加以区 分数据, 有一定安 全隐患; 2. 需要对组 | 1. 多租户的 数据间是 表级别离 全 需要对例 2. 需要实例 |
| | 题:多租户的用户查询一个租户页面的小组件数据的时候,必须要有鉴权措施, | 权,以防 越页面取 数据 | 取数据 3. 支持一些行 和列级别的 数据筛选, 不过需要提 | 需要对组 件实例鉴 权,以防 越页面取 数据 | 件实例鉴 权,以防 越页面取 数据 | 权,以防 越页面取 数据 3. 有建表操 作,无更 |
| | 否则可以随意拉取到别的租户的页面数据(只需暴力尝试不同页 | 冯金虎 4652 Jacob | 前定好查询 协议 4. 数据库DDL 操作,需要 | 3. 无数据库 DDL操 作,仅有 DML操作 | 3. 无数据库 DDL操 作,仅有 DML操作 | 新表操作 4. 支持一些 行和列级 别的数据 |
| | 面ID或者小组件 ID作为参数) | 海金虎 ^{A652} 1652 | 做好严格的 验证监控 | | 海金虎 A652 海金虎 A | 筛选,不 过需要提 前定好查 询协议 |

| 冯金虎 4652 冯金虎 4652 冯金虎 4652 | | 3. 很难做查 询和行列 级别数, 筛选, 为后端用 有通用取 数逻辑, 所以会把 | 海金虎 4652 海金虎 4652 海金虎 4652 | 4. 支持 行和的数, 分别的选需 时间 的 分别的 大型 的 的 大型 的 大型 的 大型 的 大型 的 大型 的 大型 的 | 4. 支持一些行和列级别的数据筛选,不过需要好前定好查询协议 | 2 冯全虎 46 ⁵² 2 |
|----------------------------------|-----------------------------------|--|----------------------------------|---|--|---|
| | | 整个小组件实例的数据全部下发到前端,前端选择性展 | 冯金克·4552 冯金克·4652 冯金克·4552 | | 冯金克 4652 冯金克 4552 | 为金克 4652 2 为金克 4652 |
| | | 元字任展 示,这方 面可能存 在数据隐 患 | 海金虎 4652 海金虎 4552 海金虎 4652 | | 汽金泵 455 汽金泵 4552 汽金泵 455 | 2 汽金泉A652 2 |
| | 1. 只有内部研 发可以生成 | 1. 后端通 用,意味 | 1. 后端相对通用 | 1. 后端相对 通用 | 1. 后端相对通用 | 1. 后端相对通用 |
| | 和更新,其 他人只能使 用 | 着投入很小就可以 支持绝大 | 一些协议需 | 2. EAV引擎复 杂度高, 需要仔细 | 2. 运行时元 信息重建 和查询复 | 有DDL建 表操作 表数和组 |
| | 2. 复杂的且有 必要价值的 小组件可以 | 多数的简 单存取需 求 | 度的复杂查 询 | 设计和开 发 3. 无DDL操 | 杂度高, 需要仔细 设计和研 | 件实例数 正相关, 会有无限 |
| 结论 | 用这种方式 做到组件 库,供生成 使用 | 对复杂存储和查询功能基本不支持, | 4. 有DDL操 作,既有新 | 作 作 4. 表数固 定,仅数 | 发 3. 无DDL操 作 | 膨胀的风险 |
| | 3. 只从小组件 使用上来看 应该是体验 最好的 | 遇到这种 场景需要 pass掉此 方案 | 实例数正相 | 据膨胀 5. 运行效率 较低 | 4. 表数固定,仅数据膨胀 5. 运行效率尚可 | 低下 |
| | | | 冯金庚 AJS2 | | 冯金虎 4652 | 周金虎 4652 |

LLM辅助生成

特点:全/半自动化,需要仔细审核上述流程中LLM需要做到什么步骤,额外增加哪些工作和代价 此时角色打平(内部研发、建站客户),只有创建者这一角色。

LLM辅助生成可以认为有两种形式,一是全部是新的,二是基于组件库来修改,下面的几个论述流程 实际上已经覆盖了这两种情形,只是交互层面提供入口即可。

| | 传统开发法 | 使用单文件 | 横向设计法 | EAV | Meta driven | 文档数据库 |
|----------------|----------------------------|--------------------------------|---|---|--------------------------------|--------------------------------|
| | 创建者提出 Prompt, LLM返回小 | 创建者提出 Prompt, LLM 返回小组件前 | 创建者提出 Prompt, LLM返 回小组件前端代 | 创建者提出 Prompt, LLM 返回小组件前 | 创建者提出 Prompt, LLM 返回小组件前 | 创建者提出 Prompt, LLM 返回小组件前 |
| | 组件前端代 码,和对应 的后端接口 | 端代码,和对 应的后端接口 控制模块; | 码,和对应的后 端接口控制模 块; | 端代码,和对 应的后端接口 控制模块;其 | 端代码,和对 应的后端接口 控制模块;其 | 端代码,和对应的后端接口控制模块;其 |
| | 控制模块 | Prompt需要包 含以下方面: | Prompt需要包含以下内容: | 他的不需要了 | 他的不需要了 | 他的不需要了 |
| | | 1. 与后端交互 的接口是同 | 1. 生成数据表 DDL设计的 | 海金R ⁻⁴⁻⁵ 主聚 4652 | 周金原 周金原 4652 | |
| 前端 如何 | | 一个,除了 小组件独有 的data结 | 指令 2. Prompt返回 | | 吳金虎 | |
| 新建 小组 供業 | | 构,其他公 共参数与响 | 值结构里包 含数据表 DDL | z虎 4652 冯金虎 4652 | 汽金虎 4652 汽金虎 4652 | |
| 件类 型 | | 应都一致, 需要嵌入这 些参数 | 3. 这个值需要和小组件类 | 主 虎 4652 | 冯金虎 4652 | |
| | | 2. 全量的 data数据 | 型代码保存 在一块,以 供实例化时 | | 吳金原 四金虎 4652 | |
| | | 处理逻辑。 然后LLM生成 | 使用 当然也可以不用 | | 冯金虎 | |
| | | 的小组件前端 代码保存到库 | 生成DDL, 而是和人工生成一 | 表 4652 设备 18 4652 | 冯金虎 4652 四金虎 4 | |
| | | 里即可 | 样,被动做结构 转换,运行时建 表 | 2便 4652 | 周金虎 A652 | |
| | | | 13金原 4652 | | 冯金贵 | |
| 后端如何 | 无法实现, 现在让LLM 控制后端代 | 后端通用接 口,无LLM参 与 | 后端通用接口, 无LLM参与; | 后端通用接 口,无LLM参 与 | 后端通用接 口,无LLM参 与; | 后端通用接 口,无LLM参 与 |
| 新建 小组 件类 | 码和数据 库,基本无 法实现 | | | i使465 ² 冯全使465 ² | 冯金虎 4652 冯金虎 | |
| 型 | | | | 表表 4652 | 冯金茂 4652 | |
| 前端如何更新小组 | 冯金宪 4652 | Prompt修改组 件前端代码, 然后保存 | Prompt修改组 件前端代码,然 后保存,包含 DDL部分内容 | Prompt修改组 件前端代码, 然后保存,已 经包含DDL部 分内容 | Prompt修改组 件前端代码, 然后保存 | Prompt修改 组件前端代 码,然后保存 |

| 件类 | | | | 冯金麂 4652 | | |
|----------|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|---|--|
| 型 | | | | e:戊 4652 | | |
| 后端 如何 | | 不用更新 | 不用更新 | 不用更新 | 不用更新 | 不用更新 |
| 更新 小组 | | | | e庚46 ⁵² | | |
| 件类 型 | | | | 冯金虎 465 ² | | |
| | 冯金虎 4652 | <u>冯</u> 金虎 465 |) | ₹度 4652 | <u>冯</u> 金虎 4652 | 冯金虎 4652 |
| | 冯金虎 4652 | 1. 匹配到小组 件原型,实 例化小组件 实例,与网 | 1. 匹配到小组 件原型,实 例化小组件 实例,与网 | 1. 匹配到小组 件原型,实 例化小组件 实例,与网 | 匹配到小组 件原型,实 例化小组件 实例,与网 | 匹配到小 组件原 型,实例 化小组件 |
| | | 页代码一起 保存 2. 生成页面 | 页代码一起 保存; 同时实例化时前 | 页代码一起 保存 2. 生成页面 | 页代码一起 保存 确定结构后需 | 实例,与 网页代码 一起保存 |
| | | 时,直接新 | 端需要向后端发 | 时,直接新 | 要申请后端 | 确定结构后需 |
| | | 生成一个小 组件实例 | 起(LLM给的) DDL数据表结构 | 生成一个小组件实例 | DDL建表操 作,后端接口 | 要申请后端 DDL建表操 |
| | | 这种时候 | 创建申请,后端 接口组中的DDL | 冯金虎 4652 | 组中的DDL功 能执行元信息 | 作,后端新建 一个 |
| | | Prompt包含的 东西与上面生 | 功能执行数据表 | 全限 4652 | 添加记录(实 | collection |
| wegi | | 产组件类型时 | 创建。 | | 际上是把DDL | (文档型数据 |
| c建/ | | 要求一致。 | 2. 生成页面 | 冯金虎 460-2 | 转换成元信息 | 库比较灵活, |
| 改页 面时 | | | 时,直接新 | 金虎 4652 | 表的DML) | 这里实际上也 |
| 如何新建 | | | 生成一个小 组件实例, 同时执行 | 冯金虎 4652 | 2. 生成页面 时,直接新 生成一个小 | 可以一个组件 类型映射一个 collection,转 |
| 小组件实 | | | DDL数据表 创建申请, | 全虎 4652 冯金虎 4652 | 组件实例, 此时也需要 | 换到运行时就 是实例化时可 |
| 例 | | | 后端接口组 中的DDL功 能执行数据 | 表表 4652 | 确定结构后 需要申请后 端DDL建 | 以先匹配信息 表,看看是否 已存在 |
| | | | 表创建 | 海金虎40-2- _{全虎4652} | 表操作,后 端接口组中 的DDL功 | collection, 没有则新建, 否则使用旧 |
| | | | | 海金泉 4652 李泉 4652 | 能执行元信 息添加记录 (实际上是 | 的) 和无LLM类 |
| | | | | 冯金原A652 | 把DDL转 换成元信息 表的 | 似。 |
| | | | | 主虎 4652 冯金虎 4652 | DML) | |
| | | | | | | |

| wegi | | 小组件实例可 | 小组件实例可以 | 小组件实例可 | 小组件实例可 | 小组件实例可 |
|------|-------------|----------------------|-------------|----------------|---------------|---------------------|
| c建 | | 以开放修改数 | 开放修改数据结 | 以开放修改数 | 以开放修改数 | 以开放修改数 |
| 页面- | | 据结构的前端 | 构的前端接口给 | 据结构的前端 | 据结构的前端 | 据结构的前端 |
| 发布 | | 接口给建站客 | 建站客户,客户 | 接口给建站客 | 接口给建站客 | 接口给建站客 |
| 后无 | | 户,通过 | 可以通过LLM修 | 户,客户可以 | 户,客户可以 | 户,只需要前 |
| 历史 | | Prompt让LLM® | 改结构; | 通过LLM用来 | 通过LLM来修 | 端更新数据相 |
| 数据 | | 按规则修改 | | 修改结构后保 | 改结构后保存 | 新史·斯·威·斯伯 关结构并保存 |
| 时如 | | 1女人が火リットと人 | 前端需要把结构 | 存实例代码即 | 实例代码即 | 即可。 |
| 何更 | | | 的diff转化为 | | 可; | ᅜᅡᆸᅌ |
| 新小 | | 冯金虎 465 | DDL需求向后端 | 可; | 马 海金虎 4652 | 和无LLM类 |
| 组件 | | . # A65 ² | 发起数据表结构 | → A652 | 确定结构后前 | 似。 |
| | | 冯金伦。 | 修改申请,后端 | 冯金虎。 | 端需要申请 | |
| 实例 | | 以会院 465 | 接口组中的DDL | s·虎 4652 | DDL建表操 | |
| 结构 | | P3 | 功能执行数据表 | | 作,后端接口 | |
| | | 冯金虎 4652 | 修改。 | 冯金虎 4652 | 组中的DDL功 | |
| | | | | | 能执行元信息 | |
| | | 冯金虎 465 | | 虎 4654 | 修改记录(实 | |
| | | 山金虎 4652 | | ·13余虎 4652 | 际上是把DDL | |
| | | P | | 100 Mar. 11 | 转换成元信息 | |
| | | 冯金虎 465 | | 主虎 4652 | 表的DML) | |
| | | | | | 和无LLM参与 | |
| | | 冯金虎 4654 | | 冯金虎 4654 | 类似 | |
| | w A652 | ≈ A65 | I. | ≈ A654 | X12X | = 465 ² |
| wegi | <u>冯金</u> 尼 | Prompt里需包 | 小组件实例前端 | 小组件实例前 | 小组件实例前 | 小组件实例可 |
| c页 | | 含数据转换或 | 执行结构修改, | 端执行结构修 | 端执行结构修 | 以开放修改数 |
| 面发 | | 者历史数据处 | 然后申请后端 | 改即可; | 改,然后申请 | 据结构的前端 |
| 布后 | | 理逻辑 | DDL修改,后端 | 至于后端的数 | 后端DDL修 | 接口给建站客 |
| (产 | | 冯金虎 465 ² | 执行后(不一定 | 据结构要不要 | 改; | 户,但是需要 |
| 生历 | | | 能成功,部分有 | 修正,这里我 | 后端修改DDL | 同时提供处理 |
| 史数 | | 冯金虎 465 | 数据的DDL修改 | 倾向于不修 | 仅修改元信息 | 历史数据的办 |
| 据) | | .63 | 可能无法成功) | 正。如果是修 | 表记录即可, | 法,依然在前 |
| 如何 | | 冯金虎 4652 | 将结果给前端; | 正结构的话要 | 实际上是DML | 端处理,后端 |
| 更新 | | _ 105 | 后续前端会按照 | 执行批量的 | 操作。 | 就是一个 |
| 小组 | | 冯金虎 ⁴⁰³ | 新数据结构渲 | DML操作。这 | 冯亚严 | proxy。 |
| 件实 | | 冯金虎 4652 | 染。 | 里的DML操作 | 历史数据不用 | 历史数据按照 |
| 例结 | | | 历史数据按照新 | 既改了结构, | 修正,因为每 | 前端的历史数 |
| 构 | | 冯金虎 465 | 的数据结构格式 | 也修正了历史 | 次查询都会先 | 据处理办法处 |
| | | 冯金虎4652 | 处理。 | 数据。 | 从meta表中新 | 据处理办法处 理。 |
| | | 冯兹风 | 疋 垤。 | 夕 人打凸 0 | 建数据 | 注 0 |
| | | | | 不过即使后端 | Schema. | 和无LLM类 |
| | | 冯金灰。 | | 不修,查询时 | 和无LLM参与 | 似。 |
| | | 冯金虎 4652 | | 组装返回前端 | 类似 | |
| | | | | 也会按照前端 | | |
| | | 冯金虎 465 | | 新的结构使用 | 冯金虎 4652 | |
| | | | | 数据的。 | | |
| | | 冯金虎 465 ² | | | 四金虎 | |

| 数据 | | 1. 数据按小组 | 1. 数据按小组 | 1. 数据全部打 | 1. 数据物理层 | 1. 数据按照 |
|--------------|----------|---------------------|-----------------|---------------------|--|------------|
| 存储 | | 件实例存储 | 件实例存储 | 平存储,无 | 面全部打平 | 小组件实 |
| 持点 | | (文件与实 | (表组与实 | 任何映射 | 存储,无任 | 例存储 |
| | | 例映射) | 例映射) | 2. 多租户共享 | 何映射 | 也可以 |
| | | 2. 多租户天然 | 2. 多租户天然 | 数据库,共 | 2. 实际元信息 | 按照类型 |
| | | 隔离数据 | 隔离数据 | 享数据表, | 隐式 | 存储) |
| | | 3. 无数据库操 | 3. 有数据库 | 仅通过租户 | Schema) | 2. 多租户共 |
| | | 作 | DDL操作 | ID加以区分 | 是与组件实 | 享数据库 |
| | | 25 AGS | | 3. 无任何数据 | 例映射的 | 3. Schema- |
| | | 4. 数据是孤岛 | 4. 可以实现较 | 库DDL操 | 3. 多租户共享 | less, 所有 |
| | | 5. 查询性能低 | 复杂的数据 | 作,只有 | 数据库,共 | Schema- |
| | | 下 | 联查或SQL (需提前定 | DML操作 | 享数据表, | less的优 |
| | | 冯金虎 465 | (| 4. 存查操作引 | 仅通过租户 | 缺点它都 |
| | | 冯金虎4652 | 义吻京协议) | 等需仔细设 | ID加以区分 | 4652 有 |
| | | 12G 3300··· | | 新市 新 | 4. 无任何数据 | 4. 性能很差 |
| | | 冯金虎 465 | 5. 查询性能高 | り,仔個向 度灵活 | 年 DDL操 | 4. 注形很差 |
| | | | | | 作,只有 | 支持), |
| | | 冯金虎 4652 | | 5. 查询性能中 | DML操作 | 不支持大 |
| | | .05 | | 低,写入性 | | 数据量。 |
| | | 冯金虎 403 | | 能中低 | 5. 存查操作引 | |
| | | 冯金虎 4652 | | 6. 无法支持高 | 擎需仔细设 | 5. 无高级查 |
| | | | | 级SQL查询 | 计,存储高 | 询支持 |
| | | 冯金虎 465 | | 章虎 4652 | 度灵活 | |
| | | _ 4652 | | | 6. 查询性能中 | |
| | | 冯金虎 4652 | | 冯金虎 ⁴⁰²² | 高等,写入 | |
| | | ASS | | Le 4652 | 性能中高等 | |
| | | 冯金既 | | 銀174-*** | 冯金灰 | |
| | | 冯金虎 4652 | | 冯金虎 4652 | 7. 能执行一定 | |
| | | | | | 的高级 | |
| | | 冯金虎 465 | | 拿虎 4652 | SQL,但是 | |
| | | 冯金虎465 ² | | ~ ≈ 4652 | 特别高级的 | |
| | | 冯兹氏 | | 冯金庆 | 也支持不了 | |
| | | - no a 465 | | 4 g 4652 | - 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- | |
| | | 冯延匹 | | 35.00 | 8. 唯一索引等 | |
| | | 冯金虎 4652 | | 冯金虎 4652 | 东西需要从 | |
| | | | | | 逻辑层面或 | |
| | | 冯金虎 465 | | 全虎 4652 | 全 相层面或 者加表支持 | |
| | | 冯金虎 4652 | | 归金虎 4652 | 日 川北 又 〕 寸 | |
| | | | | ling same. | (2) | |
| | 冯金虎 4652 | 冯金虎 465 |). [3] | 拿虎 46 ⁵² | 冯金虎 4652 | 冯金虎 4652 |
| 流程 | | 后端通用流 | 后端通用流程, | 后端通用流 | 后端通用流 | 后端通用流 |
| 控制 | | 程,只支持通 | 一是DDL流程, | 程,主要就是 | 程,主要就是 | 程,支持通用 |
| ,工.p., 特点 | | 用的数据增删 | 二是通用的数据 | EAV映射转换流 | 把表结构设计 | 的数据增删改 |
| . 57111 | | 改查;只能依 | 增删改查;三是 | 程,即列转行 | 到元信息里, | 查,也能支持 |
| | | 赖前端做特性 | | 这个引擎流程 | 找到宽表(数 | 简单聚合查询 |
| | | 不以日1 4位 10人 1寸 1土 | | | 1んごり込入 (対 | |

| | | 数据逻辑处理; | 通过约定协议可 做一定的联查 | 冯金虎·4652 | 据表)的列映射,实时重建 | |
|----------------|----------------------|---|-----------------------------|--------------------------------|---|----------------------------|
| | | 也可以提前做 好简单流程 | | 主录 4652 冯金虎 ⁴⁶⁵² | Shema和ORM. | |
| | | 库,供组合选择(后面通用,不再赘述) | | 主死 4652 冯金宗 4652 | | |
| | 冯金虎4653 | 后端控制好读 | 直接执行关系型 | 前端提交正常 | 前端提交正常 | 前端提交正常 |
| | | 写锁,采用 append/delet e/update模式 | 数据库的ORM写 入即可 | 的数据结构, 后端需要通过 列转行引擎, | 的数据结构, 后端需要通过 组件实例ID从 | 的数据结构, 后端直接写入 对应的 |
| 写入 数据 (使 | | 单条或批量更 新,不是全量 快照(客户端 | | 插入3张固定表。 | 元信息记录找 到宽表映射 列,实时重建 | collection |
| (使 用 付) | | 全量快照的话 存在数据一致 性问题),可 | | | Schema, 然后 按照ORM写入 数据 | |
| | | 以类比即时设 计的协同编 辑,只是不用 | | | | |
| | 冯笙邝 | 实时广播。 | 海(4652 | 一个度 4652 | 冯兹匹 | 冯兹环。 |
| 查询 数据 (使 | 冯金茂 4652 | 后端控制好读 写锁,采用全 量或者分页返 回数据快照 | 直接执行关系型 数据库的ORM查 询即可 | 需要通过EAV映射引擎查询出行,转列,组 装成正确的数 | 通过组件实例 ID从元信息里 找出Schema 映射列,然后 | 后端直接查询对应的 collection |
| 4) 引 | | 汽金 克 465 | | 据结构后返回前端 | 按照ORM从宽 表里取数据 | |
| | | 冯兹匹 | | | | |
| 多租 中数 居隔 | 馬並飛 4652 | 数据库级别隔离 | 数据表级别隔离 | 共享数据库, 共享数据表, 仅通过租户ID | 共享数据库, 共享数据表, 仅通过租户ID | 数据表级别隔离 |
| 哥方 t | | 與金虎 465 3 | | 加以区分 | 加以区分 | |
| 女据 | 海金虎 465 ² | 需要前后端使 用非快照+读写 锁控制一致性 | 关系型数据库正 确使用保证了数 据的一致性 | 关系型数据库 一定程度保证 了一致性,但 | 数据一致性要 后端保障,需 要设计特殊的 | 数据一致性要 后端保障,通 过文档型数据 |
| −致 生 | | 另金原 ⁴⁰³² 月金虎 ⁴⁶⁵ | | 是历史数据的 一致性很难保 证 | 规则来支撑 | 库操作可以有 基本保障 |
| | _ | <u>海部</u> 是 4652 | 海金原 4652 |)另金原 A652 | 73. mil. | 4652 |

| た法实现 | 安性 A 全 | 1. 多据隔 需实以取 很和数因有逻会组数发前展面数租间离 要例防数 难行据为通辑把件据到端示可据户是的 对鉴越据 做列筛后用,整实全前选,能隐的安 组权页 查级选端取所个例部端择这存患数全 件,面 询别,只数以小的下,性方在 | 多据隔 需实以取 支和数不前协 数操做验租间离 要例防数 持列据过定议 据作好证户是的 对鉴越据 一级筛需好 库,严监约金 件,面 行的,提询 L要的 | 1. 多过以据安 需实以取 无DQ作 支和数不前协租租区,全 要例防数 数L有 持列据过定议户户分有隐 对鉴越据 据操ML 一级筛需好仅加数一患 组权页 库作从 些别选要查说,件,面 | 多过以据安需实以取无DD有隐对鉴越据据作,面对鉴越据据作,面对监过定议 3. A. 4. 4. | 数表全 需件权越数 不知知 经 不知 不知 不知 的 知 知 的 知 的 知 的 知 的 知 的 知 的 知 |
|----------|--------|---|---|---|---|--|
| 高金度 4652 | 结论 | 后端味小持的需 复查本,场域是简求 杂询不遇景掉 存功支到需此方案 | 用 2. 一要 支度询 有作建 表实关限 1. 一要 支度询 有作建 表实关限 2. 操有更 组正有的 2. 操有更 组正有的 4. 人。 数例,膨胀 | 用 2. EAV引擎复杂度高,需要仔细设计和开发 3. 无DDL操作 4. 表数固定,仅数据膨胀 5. 运行效率较低 | 用 2. 运行时元信 2. 息重建和查 询复杂度 高,需要仔 细设计和研 发 3. 无DDL操作 4. 表数固定, 仅数据膨胀 4. | 通 有表 表件正会膨险 运低用力操 数实相有胀 行下组数,限风率 |

开放的开发接口

选型中要更重视安全性,然后开放给出相应接口并给出严格限制即可支持功能。

选型总结

| | 传统开发法 | 使用单文件 | 横向设计法 | EAV | Meta driven | 文档数据库 |
|-----------|--|--|---|---|---|--|
| 选型 决策 成 估 | 1. 后端开发成本: 5 (非常高,所以得分低,预估中等复杂度1天一个) 2. 前端开发成 | 1. 后端开发成本: 3(预估3天,主要是与文件系统对接,分布式锁逻辑防并发问题) | 1. 后端开发成本: 3(预估3天,处理自动化DDL) 2. 前端开发成本: 2(需要自己处理 | 1. 后端开发 成本: 4 (预估5 天,要实现EAV查询 引擎与严 格自测) 2. 前端开发 | 1. 后端开发 成本: 4 (预估4 天,实现 字段的 meta表和 data表映 射转换引 | 1. 后端开发 成本: 3 (预估3 天) 2. 前端开发 成本: 2 (需要自 己处理数 |
| | 本: 2 (相 对来说较 高,但不一 定比LLM生 成慢) | 前端开发成本: 2 (需要自己处理数据逻辑) LLM调试成 | 数据逻辑) 3. LLM调试成本: 3(需要理解接口规则,还可 | 成本: 2 (需要自 己处理数 据逻辑) 3. LLM调试 | 擎)2. 前端开发成本: 2 (需要自己处理数 | 据逻辑) 3. LLM调试 成本: 3 (全在前 端,需要 |
| | 3. LLM调试成本: 暂时无法支持 4. 运行效率曲线: 5(满 | 本: 4(全 在前端,需 要理解接口 规则,处理 数据分页展 | 以做分页、 筛选等操 作,并前端 LLM Prompt 自 | 成本: 3 (全在前 端,需要 理解接口 规则,并 | 据逻辑) 3. LLM调试 成本: 3 (全在前端,需要 | 理解接口 规则,并 前端LLM Prompt 自动化操 |
| | 分,基本遇到效率问题都会被解决) | 示等逻辑, 并前端 LLM Prompt 自 动化操作) | 动化操作) 4. 运行效率曲 线: 4 (速 度尚可,支 | 前端LLM Prompt 自 动化操 作) | 理解接口 规则,并 前端LLM Prompt 自动化操 | 作) 4. 运行效率 曲线: 1 (速度比 |
| | 5. 效果: 每个小 组件单 独开发 | 4. 运行效率曲 线: 1 (速 度比较差, | 持一定程度 的复杂查 询) 5. 效果: | 4. 运行效率 曲线: 2 (速度比 较差,较 | 作) 4. 运行效率 曲线: 4 | 较差,不 支持复杂 查询) 5. 效果: |
| | (前后 端) 对需求 支持度 | 不支持复杂 查询) 5. 效果: | □ 后端只 有一次 性的工 作,后 续前端 | 多的联表 和多行操 作,不支 持复杂查 询) | (速度较 好,支持 一般复杂 查询) 5. 效果: | □ 后端 只有 一次 性的 工 |
| | 最高 可以用 于构建 复杂组 件库 | 性的工作,后续前端调用 | 调用 编集 44 | 5. 效果: | | 作, 后续 前端 调用 |
| | | | | | | 52 |

| 金虎 4652 | □ 复杂操 | 海金虎 🗀 | 支持基 | □ 后端只 | 后端 | □ 复杂 |
|---------|-------|-------|----------|--------|----------|------|
| | 作不支 | | 本的增 | 有一次 | 只有 | 操作 |
| | 持,只 | | 删改查 | 性的工 | 一次 | 不支 |
| s 4652 | 支持基 | | 和一定 | 作,后 | 性的 | 持, |
| 金民 | 本的增 | | 程度的 | 续前端 | 工程金属 | 只支 |
| | 删改查 | | 复杂查 | 调用 | 作, | 持基 |
| | | | 询 | | 后续 | 本的 |
| 金虎 4652 | □ 运行效 | | | | 前端 | 增删 |
| | 率偏 | | 运行效 | ○ 支持基 | 调用 | 改查 |
| | 低,不 | | 率中 | 本的增 | | |
| ~# 4652 | 能承担 | | 高,表 | 删改 | 支持 | □ 运行 |
| 並从 | 大数据 | | 可以加 | 查,也 | 基本 | 效率 |
| | 量(拍 | | 索引, | 支持列 | 的增 | 偏 |
| | 脑袋: | | 所以效 | 过滤和 | 删改 | 低, |
| 金虎 4652 | 5000 | | 率比较 | 分页 | 查和 | 不能 |
| | 行) | | 好 | | 一定 | 承担 |
| | | | 天然支 | □ 运行效 | 程度 | 大数 |
| 金虎 4652 | | | 持并发 | 率偏 低,能 | 的复 | 据量 |
| | | | 处理, | 承担一 | 杂查 | 伯 |
| | | | 单个小 | 定量的 | 询, | 脑 |
| | | | 组件存 | 数据查 | 如果 | 袋: |
| 金虎 4652 | | | 到几十 | 询,但 | 需要 | 全部 |
| | | | 万数据 | 是全部 | 支持 | 数据 |
| | | | 量都可 | 总量有 | 唯一 | 50万 |
| 金虎 4652 | | | 以存 | 限制, | 性等 | 行) |
| | | | 运维不 | 比如 | 特殊 | |
| | | _ | 友好, | 50万 | 操 | |
| 金虎 4652 | | | 要做 | 行记录 | 作, | |
| 金区 | | | DDL操 | 后,需 | 得独 | |
| | | | 作,需 | 要想办 | 立开 | |
| | | | 要要到 | 法分库 | 发辅 | |
| 金虎 4652 | | | 账号权 | 分表等 | 助模 | |
| | | | 限 | 优化操 | 块; | |
| | | | 冯金虎 4852 | 作 | 不支 | |
| 金虎 4652 | | | | □ 运维友 | 持多 | |
| | | | | 好,无 | 字段 | |
| | | | 冯金虎 4852 | 任何 | 排序 | |
| | | | | DDL操 | 功能 | |
| 金虎 4652 | | | | 作 | 能; | |
| | | | _ 1457 | | | |
| | | | 冯金虎 4852 | | | |
| 金虎 4652 | | | | | 冯金虎 4652 | |
| | | | | | | |
| | | | 冯金虎 4652 | | | |
| 金虎 4652 | | | | | 冯金虎 4652 | |
| | | | | | 1-32 | |

| 冯金虎 4652 | 冯金虎4652 | 冯金虎 4652 | | | □ 运行 | □ 一种 |
|-----------------|---------------------|----------|---------------|-----------------|----------------------|---------------------|
| | | | 冯金虎 4652 | | 效率 | 是 |
| | | | 治亚 [6] | | 17. | DDL |
| | 冯金虎 4652 | 冯金虎 4652 | | | 高, +n, | ,需 |
| | | | | | 加上 | 要建 |
| | | | 冯金虎 4852 | | 表的 | 表, |
| | ~ ≈ 4652 | 冯金虎 4652 | | | 路由 索引 ※ | 但不 用改 |
| | 冯金庆 ^{A632} | 冯兹尼 | | | (可 | 表, |
| | | | 冯金虎 4852 | | 手动 | 查询 查询 |
| | | | 197 | | 提前 | 效率 |
| | 冯金虎 4652 | 冯金虎 4652 | | | 建) | 一般 |
| | | | | | 是/ ,总 | (这 |
| | | | 冯金虎 4652 | | 量不 | 种就 |
| | 冯金虎 4652 | 冯金虎 4652 | | | 太受 | 不如 |
| | Tag 300 11 | 1334 | | | 限 | 横向 |
| | | | 冯金虎 4652 | | 制; | 法效 |
| | | | | | 单表 | 率 |
| | 冯金虎 4652 | 冯金虎 4652 | | | 总量 | 好, |
| | | | = AG52 | | 不能 | 缺点 |
| | | | 冯金虎 4852 | | 超过 | 却— |
| | 冯金虎 4652 | 冯金虎 4652 | | | 1000 | 样 |
| | | | | | 万行 | 多) |
| | | | 冯金虎 4852 | | 冯金虎 4652 | 冯金克 4 <u>652</u> |
| | 冯金虎4652 | 冯金虎 4652 | | | □ 运维。 | 种是 |
| | 冯金区 | 冯金风 | | | 友 | 无 |
| | | | 冯金虎 4652 | | 好, | DDL, |
| | | | 治亚100 | | 无任 | 查询 |
| | 冯金虎 4652 | 冯金虎 4652 | | | 何 海童鬼科 52 | 效率 |
| | | | | | DDL | 极低 |
| | | | 冯金虎 4852 | | 操作 | 海金虎 ⁴⁶⁵² |
| | 冯金虎 4652 | 冯金虎 4652 | | | 冯金虎4552 | 一不支 |
| | 1-3 | 1935 | | | j20) 300°° | 持改 |
| | | | 冯金虎 4652 | | 冯金虎 4652 | 动已 产生 |
| | | | 100 | | | 发据 数据 |
| | 冯金虎 4652 | 冯金虎 4652 | | | 冯金虎 4652 | 数据 的字 |
| | | | | | (5) | 段名 |
| | | | 冯金虎 4852 | | 冯金虎 465 ² | +X17 |
| 冯金虎 4652 | 冯金虎 4652 | 冯金虎 4652 | | 冯金虎 4652 | 冯金虎 4652 | |

总结:

支持LLM自动生成和开放性接口的方式,需要综合对LLM的支持,考虑到后端应该稳定而不变更(LLM做不到),同时支持一定的运行效率,运维友好性等,可以选用Meta driven的方式,除了不能实现较复杂的查询,从开发的难度、LLM通用性和查询效率以及支持的数据量上看,都可以支撑我们的初始阶段需要。后续如果满足不了需求,也可以再迁移;