



# 金融产品指导师——智能小投 设计说明书

**Product Owner:** 邱俊源

**Scrum Master:** 张凡

**Group Members:** 曾旭祥、邵典、范一迪、杨继超

# 智能小投产品设计书目录

一、内容	4
1.1 服务目的与描述	4
1.2 服务细节	6
二、需求	8
2.1 利益相关者分析	8
2.2 用户故事	8
2.3 关键功能需求	13
2.4 关键非功能性需求	14
2.5 环境分析	15
三、风险与环境影响	16
3.1 假设和环境依赖	16
3.2 开放性风险	16
3.3 瓶颈及问题	18
3.4 解决方法初探	19
四、业务架构	20
4.1 企业战略分析	20
4.2 价值链	21
4.3 业务架构图	26
五、应用架构	27
5.1 逻辑应用架构	27
5.2 应用技术观点	29
5.3 序列流	30
5.4 Interfaces-APIs（以微服务划分）	30
5.5 微服务	32
5.6 容器化	32
六、数据架构	34
6.1 数据模型	34
6.2 数据范围	34

6.3 数据流 .....	35
七、基础设施架构 .....	37
7.1 业务分析 .....	37
7.2 技术选择 .....	38
7.3 Hosting 网络托管 .....	39
7.4 性能分析 .....	40
7.5 敏感数据保护 .....	45
7.6 访问控制 .....	46
7.7 可扩展性 .....	48
7.8 可用性 .....	51
7.9 信息生命周期管理 .....	52
7.10 部署 .....	53
八、安全架构 .....	57
8.1 关键资产 .....	57
8.2 威胁模型 .....	57
8.3 安全解决方案设计 .....	60
九、云原生架构 .....	63
9.1 云原生架构四要点 .....	63
9.2 云原生架构部署 .....	63
附录 1 LINDDUN GO 威胁卡片 .....	66

# 一、内容

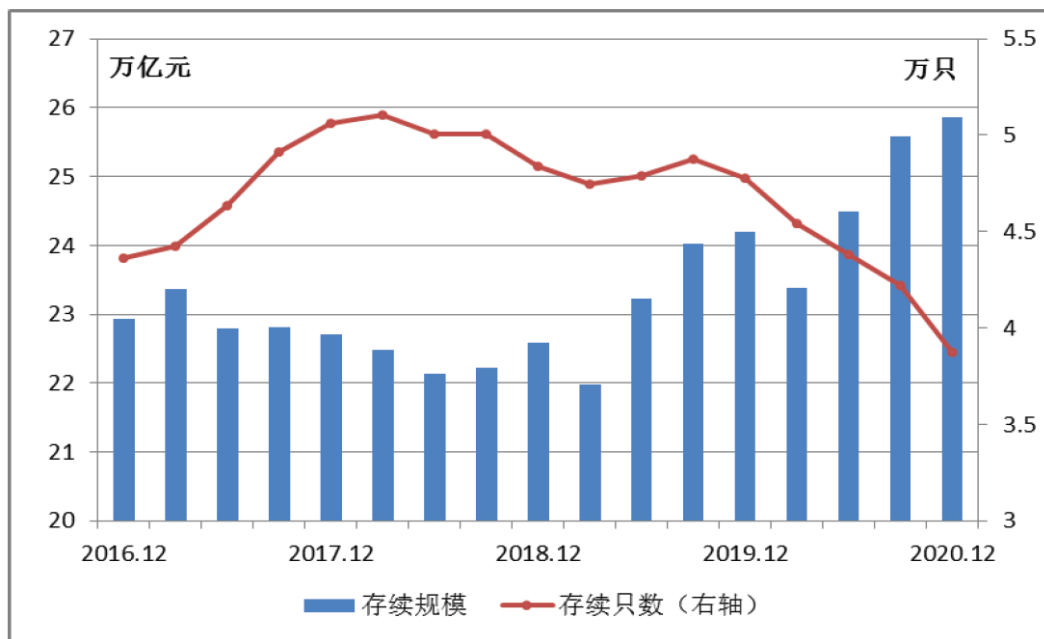
## 1.1 服务目的与描述

### 1.1.1 项目背景

2018 年 4 月《关于规范金融机构资产管理业务的指导意见》（以下简称“资管新规”）发布以来，为各大资管机构（主要代指银行）培育正规理财产品提供了宽松的政策条件。条款发布后的次年已然成为我国银行系理财子公司的设立元年。据统计，目前 6 家国有大行理财子公司已全部开业，12 家全国性股份制银行中已有 10 家完成理财子公司的获批筹建，从政策和市场供应者角度推进了各类理财产品的蓬勃发展；而从市场需求角度来看，随着国民人均可支配收入的提高，广大人民群众的投资需求也日益增加。

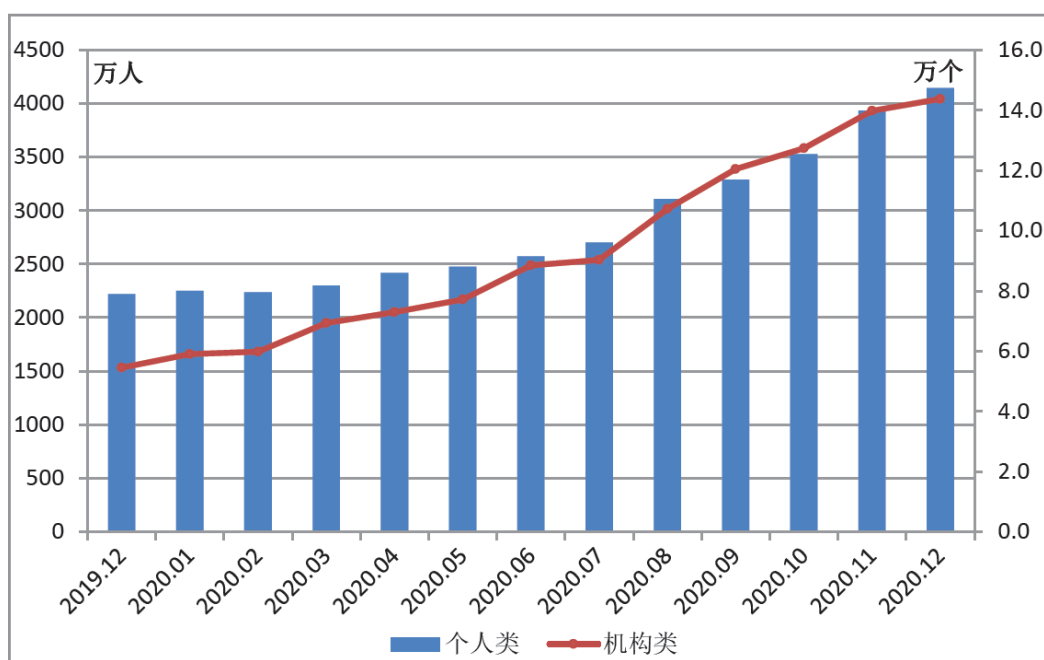
**金融理财产品种类各异，数量众多。**如各类互联网平台，包括（新浪微财富、网易现金宝等）、电商（阿里巴巴余额宝、腾讯理财宝、京东小金库等）、搜索网站（百度百赚、百发等）发布的互联网理财产品；基于各类货币基金产品发布的理财产品，例如华夏现金增利货币基金推行的活期通、汇添富基金余额宝等；各大银行发布的各银行自身的系列理财产品（如工商银行薪金宝、平安银行平安盈、中国银行活期宝等）；电信运营商主推出的财务余额自动理财类服务，如中国电信翼支付联合民生银行推出的“添益宝”等。

其中，银行因为其信用高、业务广、基础客户群体数量大等特点，是理财产品市场中的主力军。根据 2020 年中国银行业理财市场年度报告，银行理财产品市场平稳增长。由图 1-1 可观察到，2020 年 12 月统计得到的银行理财产品存续规模达 26 万亿元，存续只数约为 3.7 万只。在购买理财产品的群体中，个人投资者是理财市场绝对主力。截止 2020 年底，全市场持有理财产品的投资者数量为 4162.48 万个，其中个人投资者占比 99.65%，并且个人投资者数量近年来仍在不断上涨，如图 1-2 所示。投资者数量的增长一方面是由于新产品的不断发行，另一方面理财新规发布后产品起售金额的大幅下降，也让越来越多的普通投资者进入理财市场。因此，我们有理由相信这一增长的趋势将会在未来一段时间内持续。



资料来源：银行业理财登记托管中心

图 1-1 理财产品存续情况 (2016-2020)



资料来源：银行业理财登记托管中心

图 1-2 2020 年持有理财产品的投资者数量变化趋势(2019-2020)

但与此同时，伴随着金融理财产品种类与数量的日益增长，消费者对理财产品的认知度并未提高，尤其对银行理财的一些专业术语一知半解，对于一些理财新手来说，更是难上加难。繁杂的理财产品种类与金融产品大量且晦涩的专有名词

词是大多数普通消费者面临的一项信息壁垒。如何在繁复的资源中获取各类理财产品的精准信息，并准确理解各类理财产品晦涩的专有名词，获取各种理财产品的性质和属性，已经成为大众面临的主要问题之一。

因此，本项目提供服务的目的便是想通过我们的努力能够在某些程度上帮助到这部分消费者，让他们能够分清楚各种理财产品的属性以及各种专业术语的解释等等，从而让消费者真正买到适合他们的、符合他们个人需求的理财产品，避免消费者盲目购买高回报的理财产品。我们的理想是所有消费者都能有正确的理财观念，加强对理财产品的认知度，从而都能做到财产的良性增值。

### 1.1.2 服务内容

为了解决上述项目背景提到的消费者对于理财产品的认知问题，本项目提供了如下服务内容：

1. **金融名词查询**。用户可以在聊天页面输入任何不清楚的、想查询的金融名词，如直接输入“期权”，随后我们便会返回给用户查看此金融名词的具体含义。我们希望能够通过这样的方式让更多的用户能够加强对各种专业名词的了解，在以后查看理财产品信息时能够更加从容并能正确理解其中表达的含义。

2. **理财产品属性查询**。和金融名词的查询类似，用户可以在聊天页面输入任何想查询的理财产品的具体信息，比如用户可以输入某理财产品的收益率是多少，随后我们便会返回查询结果给用户查看。我们希望能够通过这样的方式使消费者直接查询自己对理财产品感兴趣的信息，从而节省消费者的时间，不用翻看整个理财产品说明书。

## 1.2 服务细节

为了帮助用户获得更加完整且安全的体验，并优化项目功能的完整性，我们设计了以下服务细节：

1. **反馈功能**。用户在使用过程中若是遇到了什么问题，或是对本项目有什么意见或建议，便可在聊天页面对本项目进行意见的反馈。在用户提出反馈意见后，我们会及时查看并积极解决，这样相信用户体验将会进一步提高。

2. **历史记录功能**。我们考虑到用户可能需要重复对某个内容进行查看，所

以提供了历史记录功能，用户可以在一定时间内查看自己曾经问过的问题，节省时间不用每次重新提问。

3. **闲聊功能**。由于所有数据都存在数据库中，所以难免会有金融名词或是某个理财产品的信息并没有收录在数据库中。所以我们提供了这个闲聊功能，在没有搜索到问题的答案时即调用闲聊模块减少此时的生硬感，同时用户也能直接提问闲聊的问题，这样也能丰富用户的体验。

4. **安全性**。为了考虑用户数据隐私安全，我们特意对项目的数据安全保护进行了设计。比如我们会将数据库分为云上数据库与本地数据库，用户的核心个人信息（身份证、手机号等等）会放在本地数据库中，并以此与云上的用户的其他信息进行链接保持信息的一致性。通过这样的操作就能更好地保证用户信息的保密性与安全性。

5. **信息更新**。正如第 2 点所提到的，用户在有些时候可能会找不到问题的答案，所以这就要求我们对数据库进行持续稳定的更新。我们将持续跟进最新的理财产品与金融名词，将它们的信息即使存入数据库中供用户查看。同时若用户没有查找到问题的答案，这个问题将会记录到数据库中，我们也会定期查看数据库中没有被回答的问题，并以此来更新对应的金融名词解释或是理财产品属性；若是因为问答机器人的逻辑问题导致无法找到正确答案，我们也能及时对问答机器人进行修改。相信随着时间的推移，不管是问答机器人的问答逻辑还是数据库中信息的完整度都能得到持续提升。

## 二、需求

### 2.1 利益相关者分析

内部利益相关者主要有最高管理层、职能经理(PO、SM)、项目团队成员。  
外部利益相关者主要有用户、竞争对手、推广合作商。

表 2-1 利益相关者权力-利益矩阵

利益相关者	权力		利益	
	高低	具体表现	高低	具体表现
高管人员	很高	项目的发起人、裁定项目所需资源并支持和保护项目经理及其团队免受其他组织压力	较高	股利、分红等, 长期发展受益的威信、荣誉、成就等
职能经理 (PO、SM)	较高	项目负责人, 直接参与项目的推进和审查	较高	晋升、奖金、荣誉等
项目团队成员	较高	项目的主要生产力	较高	晋升、奖金、荣誉等
用户	很高	体验产品, 并不断提出意见与建议	很高	需求尽可能被满足
竞争对手	较高	引导行业的创新发展方向, 相互竞争的威胁作用	低	技术、思想等方面的学习与借鉴, 促进竞争对手不断的技术创新
推广合作商	很高	提供产品推广资源	很高	丰厚的利益回报

### 2.2 用户故事

我们完善了之前的用户故事, 加入了一些新的功能性和非功能性需求, 并且对每个用户故事进行了风险评估。具体的用户故事如下表所示。



表 2-2 产品功能性和非功能性需求

编号	标题	工作项类型	状态	负责人	优先级	风险	需求类型	需求来源
INTEL-INV-2	智能小投产品的用户故事	史诗	进行中	邱俊源	最高	中	功能需求	产品规划
INTEL-INV-3	→登陆和注册	特性	打开	邱俊源	最高		功能需求	产品规划
INTEL-INV-6	→→作为用户，我希望能加入验证码以防止出现强行破解密码导致账户被盗的情况出现	用户故事	打开	邱俊源	最高	中	安全需求	产品规划
INTEL-INV-7	→→作为用户，我希望能加入找回密码功能以使我能够在忘记密码的时候重置密码	用户故事	打开	邱俊源	最高	中	功能需求	产品规划
INTEL-INV-8	→→作为用户，我希望能加入记住用户名的功能以方便我每次快速进行登录	用户故事	打开	邱俊源	最高	低	功能需求	产品规划
INTEL-INV-9	→→作为用户，我希望登陆时要填写的个人信息尽可能地少以让我快速	用户故事	打开	邱俊源	普通	低	体验优	产品规划

	注册体验产品	事					化	划
INTEL-INV-10	→→作为产品经理，我希望能够获取用户的理财偏好以实现更精准的查询结果返回与推荐	用户故事	打开	邱俊源	普通	中	体验优化	产品规划
INTEL-INV-11	→→作为产品经理，我希望客户的身份能够得到验证以防止非法用户注册	用户故事	打开	邱俊源	最高	高	安全需求	产品规划
INTEL-INV-69	→→作为设计者，我希望系统增加完善的访问控制权限管理，以方便不同身份和目的的用户获取不同的服务与信息	用户故事	打开	范一迪	最高	中	安全需求	产品规划
INTEL-INV-71	→→作为设计者，我希望有完整的 API 文档，以便开发	用户故事	打开	曾旭祥	较高	低	技术需求	产品规划
INTEL-INV-93	→→作为设计者，我希望能够将与用户的交互页面制作的美观大方	用户故事	打开	邵典	普通	低	功能需求	产品规划
INTEL-INV-94	→→作为设计者，我希望用户能够在本地客户端登录云服务器端	用户故事	打开	邵典	较高	低	功能需求	产品规划
INTEL-INV-98	→→作为用户，我不想自己的密码暴露给任何人，包括系统管理员，因此密	用户故	打开	张凡	较高	高	安全需	产品规

	码需要经过诸如 hash 之类的操作后才能保存在数据库中	事					求	划
INTEL-INV-102	→→作为设计者,我希望用户的核心个人信息能够被合理的 de-identified 或 de-centralized 以减缓用户信息泄露的问题	用户故事	打开	邱俊源	较高	高	安全需求	产品规划
INTEL-INV-5	→问答机器人	特性	打开	邱俊源	最高		功能需求	产品规划
INTEL-INV-12	→→作为用户,我希望问答机器人能加入闲聊机制以减少搜索不到答案时的生硬感	用户故事	打开	邱俊源	较高	低	体验优化	产品规划
INTEL-INV-13	→→作为产品经理,我希望能够有用户反馈的功能以收集用户使用的意见以进行后续产品的迭代更新	用户故事	打开	邱俊源	最高	低	功能需求	产品规划
INTEL-INV-14	→→作为用户,我希望加入历史记录功能以让我更好查阅已经查询过的问题	用户故事	打开	邱俊源	较高	低	功能需求	产品规划
INTEL-INV-70	→→作为设计者,我希望提高系统的可用性与容错性,以防止频繁宕机的产生	用户故事	打开	范一迪	最高	高	技术需求	产品规划

INTEL-INV-72	→→作为设计者，我希望能实现数据与服务的云端上传，以实现多态主机的远程操控	用户故事	打开	范一迪	较高	中	技术需求	产品规划
INTEL-INV-73	→→作为设计者，我希望将各个模块拆分成微服务，以提高可拓展性和可靠性	用户故事	打开	曾旭祥	最高	低	技术需求	产品规划
INTEL-INV-74	→→作为运维人员，我希望将微服务容器化，以便部署上线	用户故事	打开	曾旭祥	较高	低	技术需求	产品规划
INTEL-INV-88	→→作为用户，我希望开发者能够及时更新数据库，让我能够及时获得最新的信息	用户故事	打开	杨继超	较高	中	体验优化	产品规划
INTEL-INV-89	→→作为用户，我希望开发者能够优化问答机器人的逻辑算法，提高问题的检索速度	用户故事	打开	杨继超	较高	中	体验优化	产品规划
INTEL-INV-95	→→作为设计者，我希望用户向我们提供的他个人信息和对问答机器人提出的相关问题我们能够很好的保存并不泄露出去	用户故事	打开	邵典	较高		功能需求	产品规划
INTEL-INV-96	→→作为运维人员，我希望用户数据维护和更新应该根据不同用户采取	用户故事	打开	邵典	普通		功能需求	产品规划

	不同方式，频率和信息周期会存在不同。	事					求	划
INTEL-INV-97	→→作为设计者，我希望产品能够很好的扩展性，以便能够实现在多种环境下的适配和开发。	用户故事	打开	邵典	较高		技术需求	产品规划
INTEL-INV-99	→→作为设计者，我不想用户提出的所有的数据库中都没有的问题都调用问答机器人，因此，我们需要在调整一下调用聊天机器人的策略，在用户提出的数据库没有的问题但是是金融 产品相关的问题时，返回数据库中 没有相关金融产品之类的消息。	用户故事	打开	张凡	普通	低	功能需求	产品规划
INTEL-INV-103	→→作为用户，我希望能拥有方便的隐私设置以使我不需要阅读繁琐并且不熟悉的隐私协议后进行繁琐的设置	用户故事	打开	邱俊源	较高	中	体验优化	产品规划

## 2.3 关键功能需求

1. **登录、注册。**用户的登陆和注册功能
2. **反馈。**用户关于产品改进建议或体验感受的反馈功能。
3. **历史记录。**记录用户相关问答，以便其更好查阅已经询问过的问题。
4. **理财产品信息查询。**查询理财产品的相关信息，如产品代码、截止日期，

投资组合等等。

5. **金融名词解释查询**。查询产品内的金融名词解释，如风险投资的含义、产品存续期的解释等。

6. **闲聊**。当用户输入的问题既不是咨询理财产品相关的问题，也不是查询金融产品相关的名词解释问题后，则与用户进行闲聊，以提升用户体验。

## 2.4 关键非功能性需求

### 2.4.1 安全性

用户在问答机器人中的问答过程可能包含敏感的个人信息，用户登录注册时的用户名、密码、个人资料等都属于敏感数据，这就要求数据方面，考虑数据的敏感性，采用多重加密等手段。系统方面，防范各种攻击手段，以防用户数据的泄露。另外，还应建立完善的权限管理机制。

### 2.4.2 高并发

通过设计保证系统能够同时并行处理大量请求，相关指标如响应时间、吞吐量、并发用户数等需要达到对应的要求。

### 2.4.3 可拓展性

系统应具备可拓展性，如添加证券相关的问答功能、添加理财产品的推荐功能等。

### 2.4.4 问答响应时间

用户提问后，系统进行 NLP、查询到最后回复，其中的响应时间不应太长，否则用户体验不佳。

### 2.4.5 易操作性

前端界面应设置合理，观感简洁明了，操作便捷，用户使用产品的过程应当是流畅的，不需要有冗余的操作影响体验。

## 2.5 环境分析

### 2.5.1 市场背景研究

从市场需求角度来看，随着国民人均可支配收入的提高、金融理财产品数量和种类的丰富、理财新规发布后产品起售金额的大幅下降，越来越多的普通投资者进入理财市场，市场需求也愈加充分。

伴随着金融理财产品种类与数量的日益增长，消费者对理财产品的认知度并未提高，繁杂的理财产品种类与金融产品大量且晦涩的专有名词是大多数普通消费者面临的一项信息壁垒。

### 2.5.2 用户痛点

用户难以在繁复的资源中获取各类理财产品的性质和属性等具体信息。用户难以准确理解各类理财产品晦涩的专有名词。

### 2.5.3 外部竞争者调查

目前市面上关于理财产品的问答，都是静态的内容，只提供标准的、固定的少数问题，供用户参考。

如平安银行的理财常见问题。

#### 理财常见问题

##### 理财常见问题

##### 1、客户理财收益如何计算？

客户实际理财收益按下式计算：理财收益额 = 理财金额 × 持有到期年化收益率 × 实际理财天数 / 365

举例：假定理财本金为人民币20,000,000元，实际理财天数为30天，投资者持有到期年化收益率为3.70%，则理财收益额为： $20,000,000 \times 3.70\% \times 30 / 365 = 60,821.92$ 元

##### 2、客户通过什么渠道购买理财产品？

客户可通过网银（需开通网银购买理财产品功能）或柜台购买理财产品。

##### 3、滚动型产品在开放期如何操作？

顶尖计划/卓越系列/和盈系列滚动型产品，开放期内可通过网银或柜台进行新认购、全部赎回、部分追加、部分赎回等操作，但余额不能低于最低限额（顶尖计划100万，卓越系列/和盈系列500万）。

图 2-1 平安银行的理财常见问题

另外，关于理财产品的具体信息，也没能做到解答，用户痛点仍然有待解决。

## 三、风险与环境影响

### 3.1 假设和环境依赖

**假设 1:** 用户询问的问题都是他们迫切想要知道的，不是随便提出的垃圾话题。这样保证使用产品的用户都是正确的目标用户。

**假设 2:** 当前市场实现智能问答机器人的技术都不被恶意限制以至于无法使用。

我们的产品是基于强大的产品互通网络和互联网基础上的，同时使用并依赖于图数据库的存储功能和关系数据库的存储功能，并且使用 Vue 进行用户交互。任何一个节点都需要对应技术的支撑和对应安全技术的保护。与实现问答机器人的基本技术利益相关者需要建立互利共赢。

### 3.2 开放性风险

使用波特五力进行分析，可以看到市场上的开放性风险分为以下几个部分：

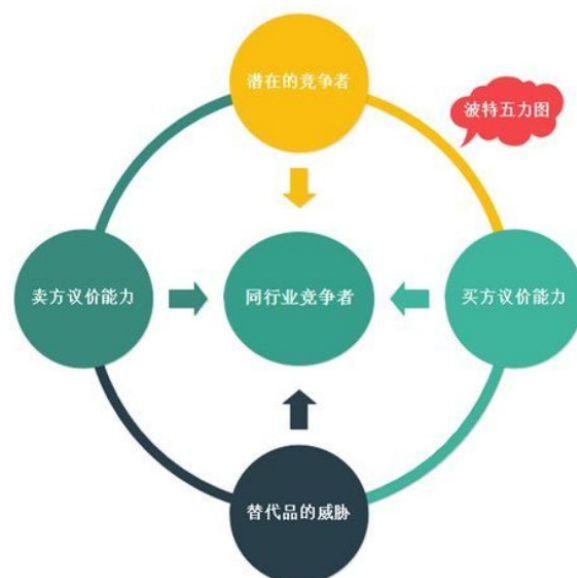


图 3-1 波特五力模型

#### (1) 潜在的竞争者

关于问答机器人的市场，当前已经存在一些十分具有市场地位的产品，如新华教育集团、美莱、金牌橱柜等大企业对此行业的投入，使得到现在为止问答机



器人的技术相对成熟。但大部分企业涉足的领域是服务行业，金融智能投资问答系统还暂时没有较多实力强劲的竞争者，但其实问答原理相同，如果现有问答机器人企业想要进入金融领域的小方向是十分有可能并且耗能低。

**(2) 买方议价能力**

我们作为提供智能问答机器人的一方，买方主要定位在在一段时间内无人工作下需要对用户进行金融投资顾问服务的企业群体。因此，买家市场非常广泛，在一定程度上减少了买方议价的能力。但另一方面，智能问答机器人现在并不是刚需，很多企业仍处于初步阶段，客户量不大，因此没有一定需要使用机器人来代替人工回答。因此我们在考虑买方议价时，更多的是要定位准确目标买家，并且根据其目标个性化的修改。

**(3) 卖方议价能力**

此产品卖方即为技术支持方。从数据库支持到前端界面支持，产品大量使用一些开源的代码来自主完成平台搭建，一次卖方方面几乎不用考虑价格制定是否处于优劣势。但是页面安全方面的协议，机构签名等基础支出，需要与相关监管部门进行沟通和监督，这作为确定的成本支出不考虑在议价能力内。

**(4) 替代品的威胁**

基于深度学习的问答机器人仍然处于发展阶段，当前市场尚未出现比智能问答机器人更加能满足企业无人工干预的回答客户问题的智能技术，因此暂时不用担心。

**(5) 同行业竞争者**

表 3-1 同行业竞争者

	快商通	网易七鱼	复星环球科创
文本智能	✓	✓	✓
智能检查	×没有	×没有	✓
人机耦合	×没有	×没有	✓
智能反馈	✓	✓	✓
学习测试	✓	×没有	×没有
网络回答稳定性	×没有	✓	×没有
数据分析	×没有	×没有	×没有

### 3.3 瓶颈及问题

很长一段时间以来，在智能化的应用中，问答机器人一直被业界重视，很多人工智能技术主要就体现在智能问答机器人上。所以，从长远来看，未来的机器人要有更好的发展才能走得更远。目前，随着时代的发展，智能问答机器人的也遇到了瓶颈，智能小投也不例外，需要着手解决的问题主要有以下几点：

- 1. 应答不够准确，意图识别不到
- 2. 维护 FAQ 工作量巨大
- 3. 知识管理不统一，重复工作
- 4. 模型比较复杂，调试困难
- 5. 只能相似匹配，不能给出答案
- 6. 还不够智能

为了实现足够的智能，问答机器人必须能够准确获取用户传达的文字信息、图片信息等其他各种信息。不同类型信息的特征也不同，因此一般实现功能需要识别以下不同问题的特点，如下：

表 3-2 用户提问信息内容概况

	语言信息	互动交流	其他信息（包括图片、表情形式）
用户可获得的信息	语言信息丰富	语言回复交流频率低	非语言信息丰富
	非语言信息丰富	没有非语言信息，纯文字交流	拥有视觉冲击带来的特殊效果
	情绪信息丰富	有关情绪方面的信息难以获取	信息传递有一定的不确定性
对时间/地域的要求	不受地区限制	不受地区限制	不受地区限制
	获取对应内容时间成本较高	获取对应内容时间成本较低	获取对应内容的时间成本高
获取信息特点	内容专业性高	难以建立可靠的信息交流模式	丰富性多样性

	信息内容可操作性 高	信息同步性较差	信息的多面性
	信息防盗安全性高	信息有效性较低	信息可转移和复制性高

### 3.4 解决方法初探

- (1) 统一多渠道知识库一次编辑多渠道应用，自动拆分 FAQ；
- (2) 采用深度学习准确率更高，知识图谱直接给出答案而不是找出相似问题。

# 四、业务架构

参考《企业级业务架构设计：方法论与实践》，我们为产品智能小投搭建起了一个初级的业务架构。首先，我们搭建了企业的战略分析模型。之后，以产品价值链为“横轴”为智能小投产品搭建起了一个业务架构。

## 4.1 企业战略分析

虽然目前我们的“企业”就仅有智能小投一个产品，但我们也需要有一个深刻、清晰的企业战略蓝图来指导我们完善目前已有的产品和开发即将面世的其他产品。我们使用由 BMGovernance 公司设计的企业战略分析模型<sup>1</sup>对我们的“企业”进行分析得到我们的企业战略分析模型。原模型和我们的企业战略分析模型如下图所示。

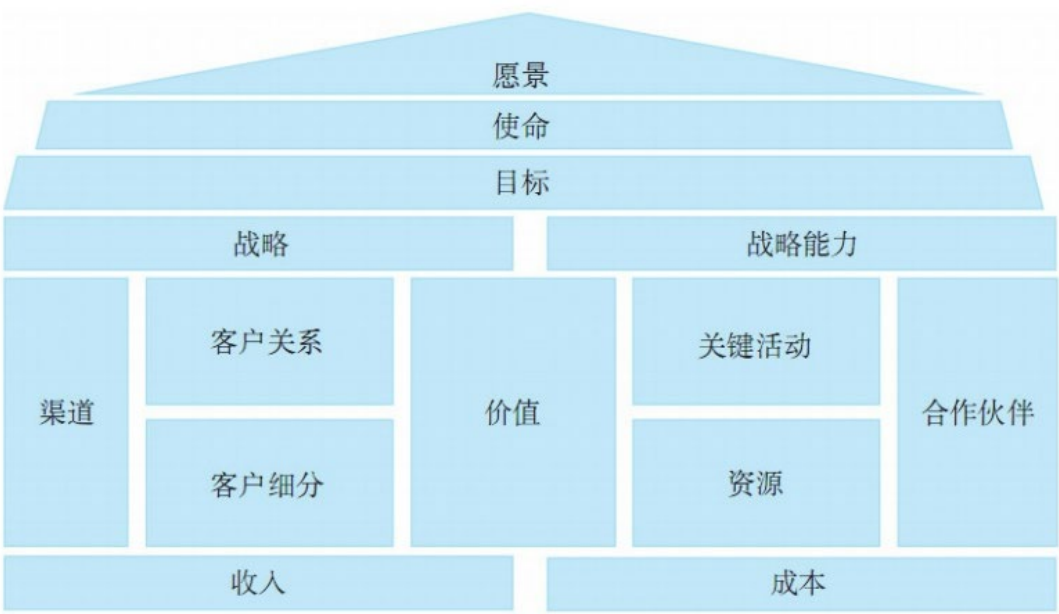


图 4-1 BMGovernance 公司设计的企业战略分析模型

<sup>1</sup> <http://www.bmgovernance.com/language/zh/>

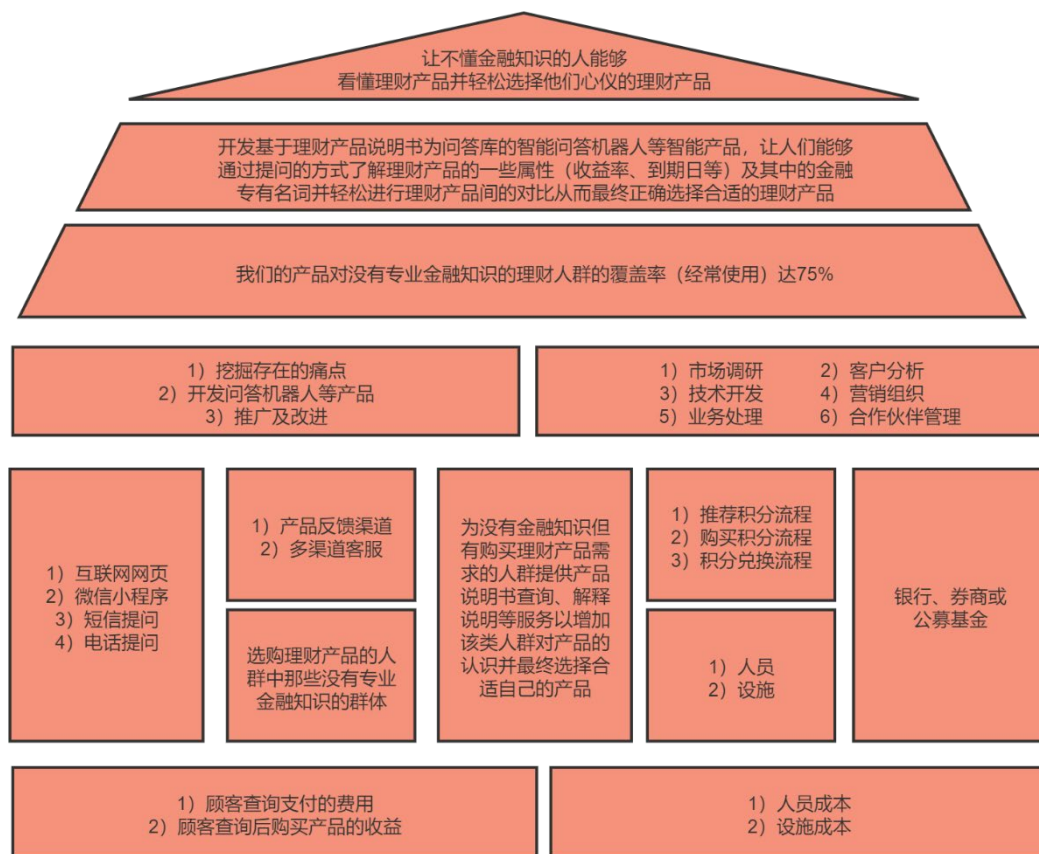


图 4-2 本企业的战略分析模型

## 4.2 价值链

管理学上分析企业竞争力通常多使用价值链模型，而这个久经考验的管理方法也很适合用来做横向的企业级分析。因此，我们首先构建智能小投产品的价值链作为整个业务架构的基础。该产品的价值链如下图所示。



图 4-3 价值链

### 4.2.1 产品设计环节的分析

智能小投的“产品设计”环节定义了 1 个活动，称为“设计上架产品”，其大致流程如下图所示。

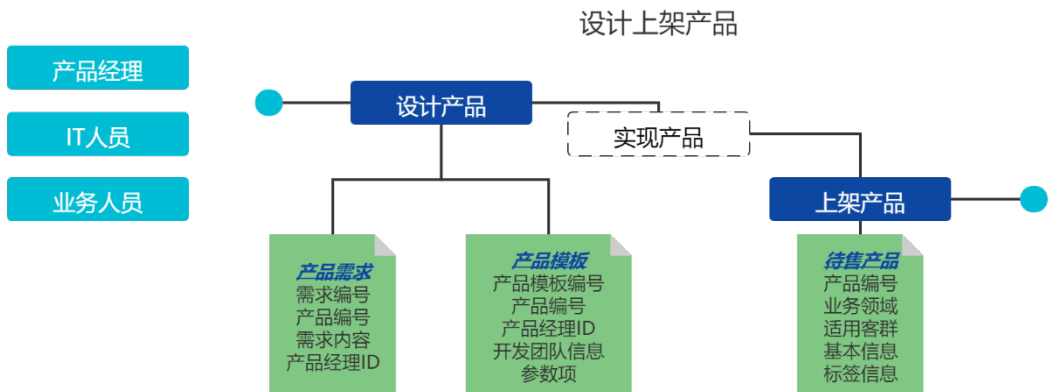


图 4-4 “设计上架产品”流程图

在这个活动中共包含 3 个角色：产品经理、IT 人员、业务人员。产品经理和 IT 人员共同完成设计产品这一任务，由 IT 人员根据设计好的产品进行下一个实现产品的任务。最后由业务人员负责上架产品。产品经理负责分析产品功能性需求，IT 人员负责补充产品的非功能性需求。之后产品经理设计并运用产品模板为业务部门整理业务需求，并提交给 IT 人员进行开发。由于实现产品的过程非常复杂，因此，在业务模型中可以用一个虚拟的任务来代表。开发完成后，业务人员添加关于产品的基本信息、标签信息等，做上架前的最后配置，配置完成后该产品就成为一个待售产品，可以随时出售。这个活动中，我们主要关注产品需求、产品模板、待售产品这 3 个实体，前 2 个由任务“设计产品”创建，最后 1 个由任务“上架产品”创建。

### 4.2.2 客户营销环节分析

在营销环节中，我们定义了“客户获取和维护”和“查询”这 2 个活动。第 1 个活动是面向智能小投产品的新用户和老用户，第 2 个活动就是营销的目的了——为他们提供产品信息查询服务。这 2 个活动的简要情况如下图所示。

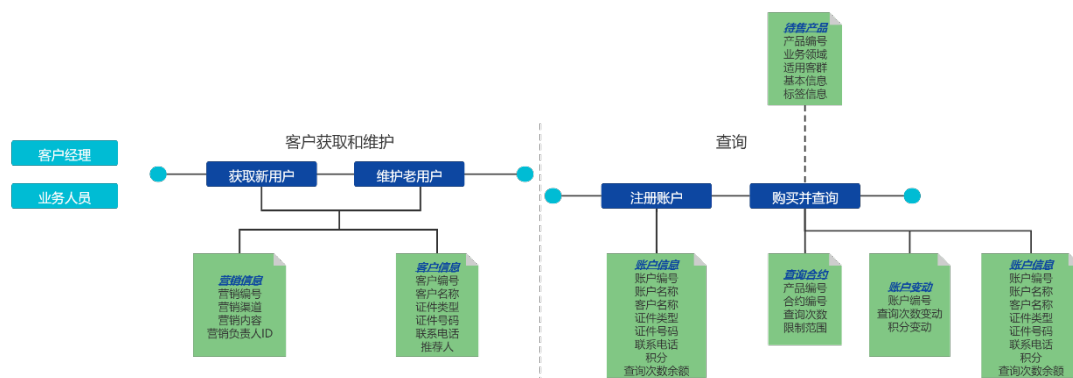


图 4-5 客户营销流程图

在客户获取和维护这个活动中共包含 2 个角色：客户经理和业务人员。客户经理和业务人员共同完成这一活动中的两个任务——“获取新用户”和“维护老客户”。业务人员主要负责营销活动的开展，产品经理则主要负责录入和维护客户信息。这个活动中，我们主要关注“营销信息”和“客户信息”这 2 个实体。

客户信息建好以后，即可进入业务办理过程——查询活动。在“查询”这个活动中仅包含业务人员 1 个角色。在前一个活动中获取到的新用户将在平台注册账户，并和老客户一样购买查询次数并进行查询。在“注册账户”任务中，要审核用户的证件。为了简化这里我们暂时略过这些内容，仅关注“注册账户”任务对“账户信息”实体的创建。完成账户开立之后，就是购买并进行查询了。无论客户是按次数付费还是按月付费，都是与我们的产品建立了一个“查询合约”，代表了一种服务接收方与服务提供方之间的供需关系。而合约主要记录的要素其实来自于我们在上一个环节中创建的“待售产品”。因此，“购买并查询”这个任务读取了“待售产品”实体，将其实例化建立了“查询合约”“账户变动”这两个实体。由于查询次数和积分的变化，该任务还变更了“账户信息”实体。在整个活动中，业务人员则扮演着客服的角色，指导、协助用户完成一系列注册、购买和查询活动。

### 4.2.3 运营管理环节分析

在运营管理环节中，我们定义了“业务处理”和“产品改进”两个活动。由于业务处理的过程非常复杂，因此，在业务模型中可以用一个虚拟的任务来代表。我们将重点关注“产品改进”这个活动。这 2 个活动的简要情况如下图所示。

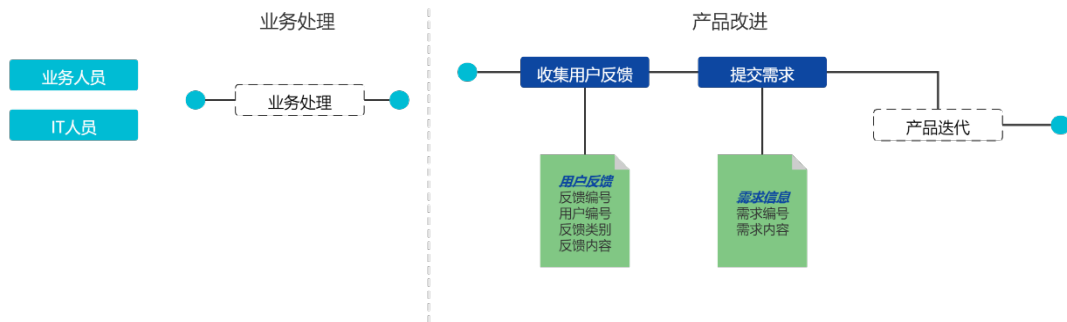


图 4-6 运营管理流程图

这 2 个活动均由 IT 人员和业务人员共同完成。由于使用 DevOps 模式进行开发，IT 人员不仅参与开发工作，也会负责产品上线后的运维工作。而业务人员则负责更新产品数据库、审核用户信息、将用户意见包装成需求给到 IT 团队等工作。

在“产品改进”活动中，业务人员收集用户的反馈（通过产品反馈按钮提交的反馈、微信客户群提出的反馈等）并将其转换为新的需求交给负责维护和升级的 IT 团队。IT 团队则根据新的需求更新产品，改善用户体验。由于“产品迭代”这个任务的过程十分复杂，因此，在业务模型中可以用一个虚拟的任务来代表。在整个活动中，我们主要关注用户反馈、需求信息这 2 个实体，由“收集用户反馈”和“提交需求”两个任务创建。

#### 4.2.4 风险控制

智能小投的“风险控制”环节定义了两个活动，分别为“产品风险控制”和“业务风险控制”。IT 人员负责产品的风险控制——网络安全、信息安全等内容。分析人员负责业务方面的风险控制——符合行业规范、法律规范等要求。由于风控的过程较为复杂，因此，在业务模型中我们使用两个虚拟的任务来分别代表每个活动。其大致流程如下图所示。

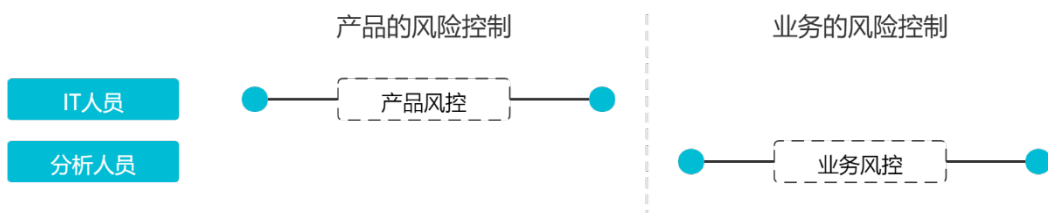


图 4-7 风险控制流程图



## 4.2.5 统计分析

在统计分析环节，我们定义了 1 个活动，称为“数据分析”。由于数据分析的内容过于复杂，因此，在业务模型中我们使用四个虚拟的任务来分别代表。其大致流程如下图所示。



图 4-8 统计分析流程图

## 4.2.6 组件设计

根据上述过程描述，我们可以设计如下组件和主题域。



图 4-9 组件设计

1. 将数据实体“产品需求”、“产品模板”和“待售产品”都放在“产品管理”主题域中，而将与之相关的“设计产品”、“实现产品”和“上架产品”三个任务聚合成“产品管理”组件。
2. 将数据实体“营销信息”和“客户信息”放在“营销管理”主题域中，将与之相关的“获取新用户”和“维护老用户”两个任务聚合成“营销管理”组件。
3. 将数据实体“查询合约”放在“合约”主题域中，将与之相关的“签订查询合约”任务放到“合约管理”组件。
4. 将数据实体“账户信息”和“账户变动”都放在“账户”主题域中，将与之相关的“购买并查询”任务放到“账户管理”组件。
5. 将数据实体“用户反馈”和“需求信息”放在“运维管理”主题域中，将与之相关的“产品改进”任务放到“运维管理”组件。

### 4.3 业务架构图

最终得到的智能小投产品的业务架构图如下所示。

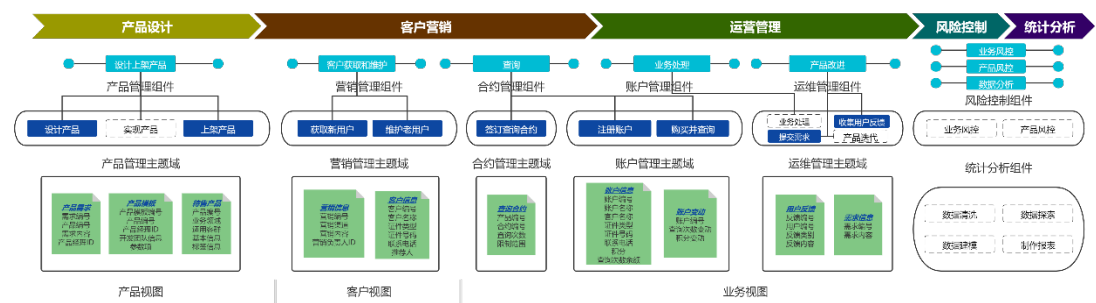


图 4-10 业务架构图

# 五、应用架构

## 5.1 逻辑应用架构

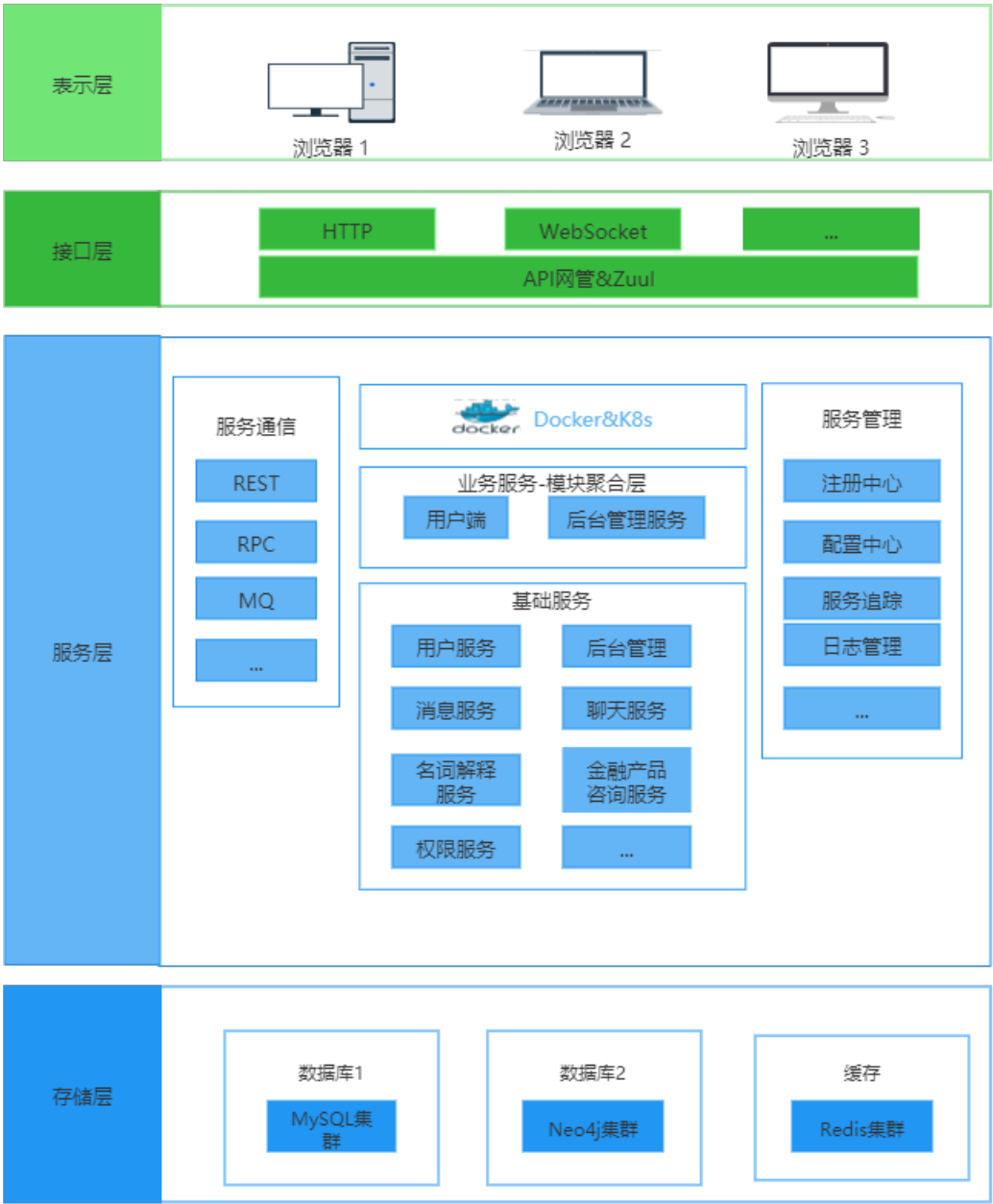
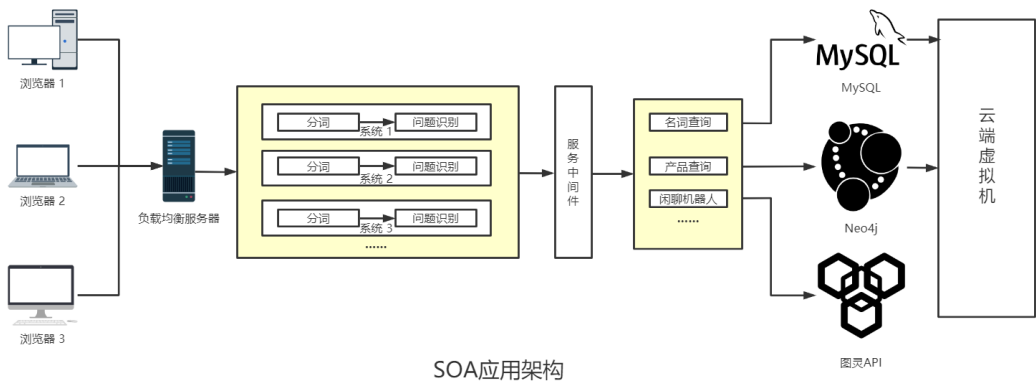


图 5-1 逻辑应用架构

我们的产品“智能小投”的逻辑应用架构分为四层，分别是表示层，接口层，服务层和存储层。表示层负责将我们的产品展示给用户，与用户进行交互，我们的产品主要聚焦于 PC 端。接口层，负责的是前后端之间的通信，在用户登录注

册方面，我们前期使用的是 HTTP 协议，但后面为了安全，我们使用了 HTTPS 协议进行了加密，保证了用户的用户名和密码传输的安全；在用户咨询问题方面的问题传输，我们使用的 WebSocket 协议，因为 WebSocket 一方面可以双向发起连接请求，另一方面不需要像 HTTP 和 HTTPS 协议一样，每次连接需要三次握手，WebSocket 只需一次连接，且连接后不主动断开连接是不会断开连接的，这样非常适合我们需要与用户频繁的交互这一需求。存储层，我们使用了三种数据库集群，且这三种数据库集群，都在远端虚拟机上面，MySQL 主要负责用户信息和一些金融名词等的存储，Neo4j 主要负责金融产品信息的保存，因为金融产品设计很多属性之间的关系，因此我们使用 Neo4j 图数据库来保存。Redis 负责缓存，提升系统的可用性。

产品的主要功能的 SOA 应用架构如下图所示。



SOA应用架构

图 5-2 SOA 架构

## 5.2 应用技术观点

我们整个项目所使用的的主要软件和硬件技术元素如下图所示。



图 5-3 应用技术观点

# 5.3 序列流

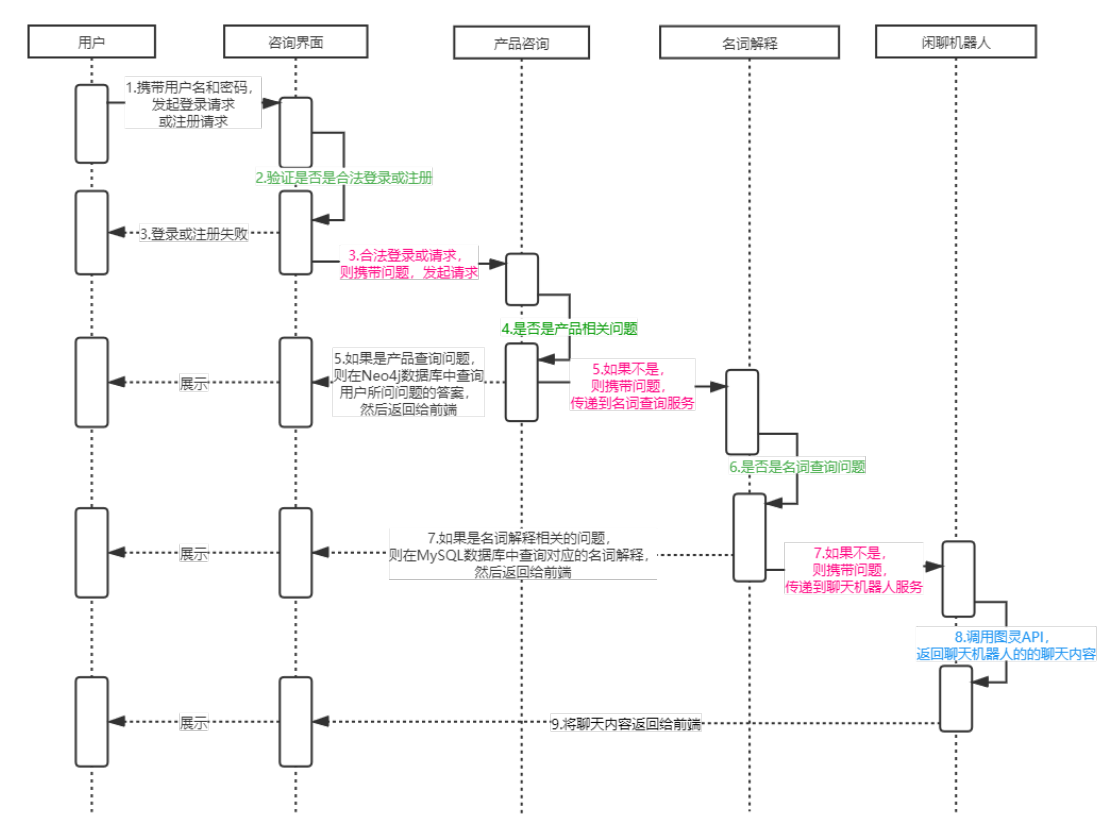


图 5-4 序列流

整个产品的运行序列流图如上图所示。用户首先是进行登录或注册，在这一步需要验证用户的登录和注册是否合法，当验证通过后，进入咨询界面。用户可以在咨询界面咨询自己想要咨询的问题，拿到用户输入的问题后，我们需要进行一系列的判断，判断用户的问题是否是金融产品相关的问题或者是金融名词相关的问题，如果是则进入对应的模块，进行查询，然后返回对应的答案，如果两次判断都为没有通过，则调用聊天机器人与用户进行对话，以避免尴尬，提升用户体验。

# 5.4 Interfaces-APIs（以微服务划分）

## 5.4.1 用户服务 API

**登录接口**，参数：userName 用户名、password 加密过后的密码。返回值：True or False。

**注册接口**，参数：userName 用户名、password 加密过后的密码。返回值：True or False。

**反馈接口**，参数：userName 用户名、feedback 建议或者意见。返回值：True or False。

### 5.4.2 日志服务 API

**日志记录接口**，参数：message 日志信息、time 时间。返回值：无。

### 5.4.3 自然语言处理服务 API

**NLP 接口**，参数：sentence 用户输入的语句。返回值：result 自然语言处理得到的结果，以 Json 格式封装，主要有 code、query 两个键值对。

### 5.4.4 闲聊服务 API

**闲聊接口**，参数：sentence 用户输入的语句。返回值：reply 闲聊机器人的回复。

### 5.4.5 金融名词查询服务 API

**金融名词查询接口**，参数：result 自然语言处理得到的结果。返回值：explanation 对应的金融名词的解释。

### 5.4.6 理财产品查询服务 API

**理财产品查询接口**，参数：result 自然语言处理得到的结果。返回值：data 理财产品的相关信息。

## 5.5 微服务

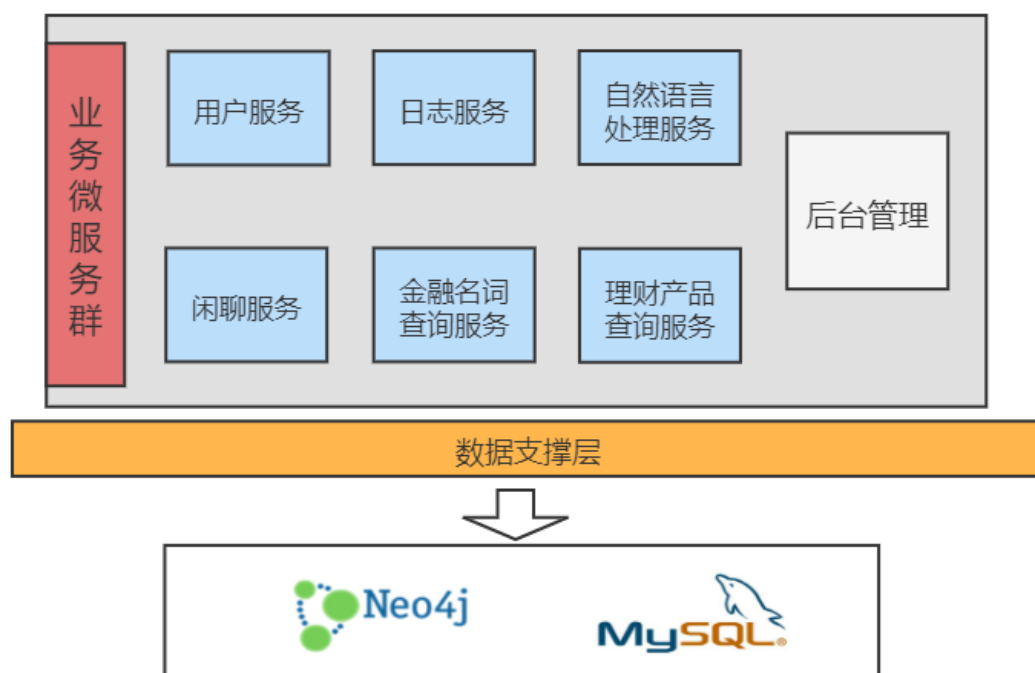


图 5-5 微服务

如上图所示，项目中的业务模块划分成微服务的形式，主要有六类微服务：用户服务、日志服务、自然语言处理服务、闲聊服务、金融名词查询服务、理财产品查询服务，后台管理负责整体的流程控制，调度各个模块以提供服务。

数据支撑层主要有两类数据库，一是 MySQL 数据库，供用户服务、日志服务、金融名词查询服务三个模块使用。二是 Neo4J 数据库，供理财产品查询服务模块使用。

## 5.6 容器化

容器是一种轻量级、可移植、自包含的软件打包技术，使应用程序可以在几乎任何地方以相同的方式运行。开发人员在自己计算机上创建并测试好的容器，无需任何修改就能够在生产系统的虚拟机、物理服务器或公有云主机上运行。容器使软件具备极强的可移植性。



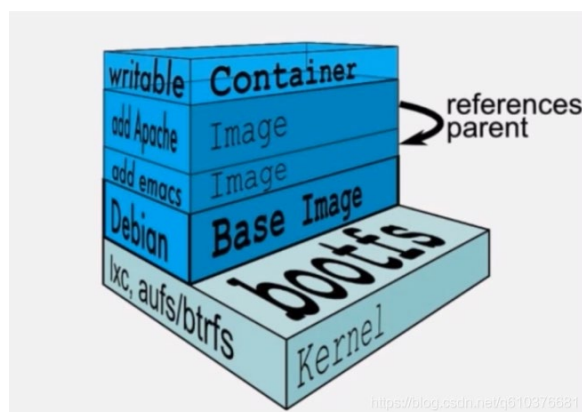


图 5-6 容器化

本项目采用 Docker 对各个微服务进行容器化。

下载相应的镜像文件，并安装微服务中所需要的软件或应用程序。如安装 python3、django、pymysql 等。

编写 Dockerfile 编译脚本。如指定基础镜像信息，镜像操作指令，容器启动执行指令等。

通过 docker build 以制作新镜像。docker build 命令会根据 Dockerfile 文件及上下文构建新 Docker 镜像。

测试运行制作的容器。

# 六、数据架构

## 6.1 数据模型

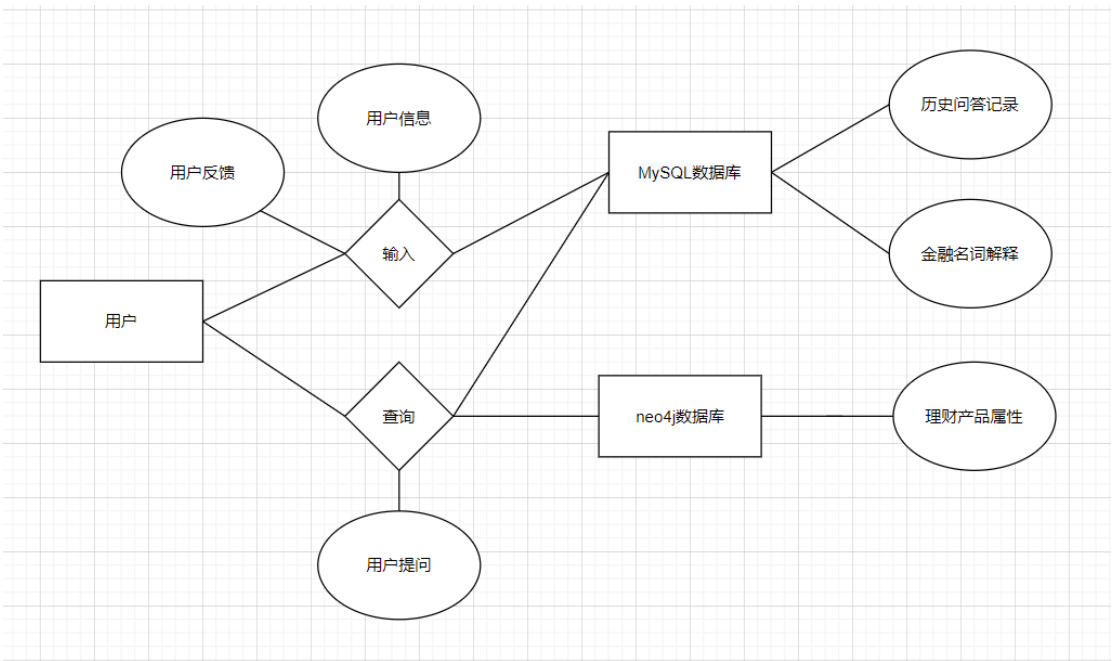


图 6-1 数据模型

上图为项目的数据逻辑模型图，可以看出项目中分为三个实体、六个属性。实体分别为项目中负责存储的 MySQL 数据库、neo4j 数据库以及使用产品的用户，属性为用户反馈、用户信息、金融名词解释、理财产品属性、用户提问以及历史问答记录。

用户与两个数据库分别以输入、查询两个关系连接。用户通过输入自己的用户信息以及对产品的反馈与 MySQL 数据库连接，并将输入的信息保存在数据库中。用户在使用产品时，通过提问的方式查询 MySQL 以及 neo4j 数据库，同时用户的提问也将作为历史问答记录保存在 MySQL 数据库中。

## 6.2 数据范围

本项目中涉及到的数据全是结构化的数据，分别存在 MySQL 数据库以及 neo4j 数据库中。

MySQL 数据库中存放的大部分数据是对各种金融名词的解释，有少部分是用户的用户名密码信息、用户的反馈信息以及用户的历史问答数据。其中的大部分

数据是文本型数据，长度从几个字符到上百个字符不等，也有少数用户密码是数值型数据，长度在 6 个字符到 14 个字符之间。

Neo4j 数据库中存放的数据是各种理财产品的属性。其中的大部分是文本型数据，长度从几个字符到上百个字符不等，也有许多日期型数据与少数数值型数据。

### 6.3 数据流

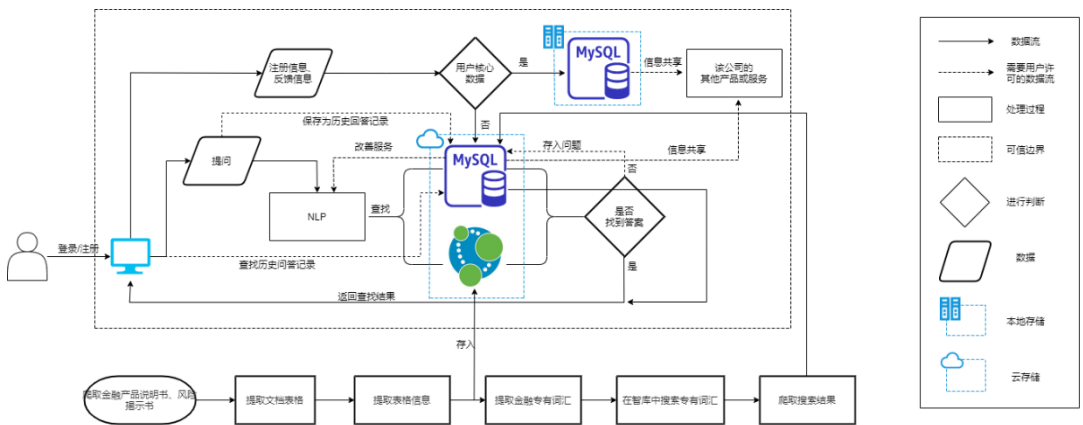


图 6-2 数据流

- 上图为本项目的逻辑数据流程图，可以看出，本项目大概分为四个数据流：
1. 用户的注册信息及反馈信息经用户输入后，将用户的核心信息（身份证、电话）存储在本地的 MySQL 数据库中，其他非核心信息直接流入到云上的 MySQL 数据库中存储。通过内部的用户编号和用户名称进行链接。在用户许可的情况下，用户的核心个人数据和查询数据将被共享给该公司的其他产品或服务使用以改善用户体验。
  2. 用户登录后，进入聊天机器人页面，在用户许可的情况下，提出的问题将被记录在云上的 MySQL 数据库中作为用户的历史记录。同时，问题经过 NLP 解析后再对 MySQL 或 neo4j 数据库中的信息进行查询，查询出的结果传到聊天机器人页面中进行显示供用户查看。在用户许可的情况下，查不出的问题被记录在云上的 MySQL 数据库中。用户的历史记录和未查询出结果的问题在用户许可的情况下将被用于改善 NLP 模型以提供更好的服务。
  3. 用户登陆后，在用户许可的情况下，前端会给后端发送请求，经 MySQL 数

数据库查询历史问答记录，查询出的信息传到聊天机器人页面中进行显示供用户查看。

4. 云上数据库中的信息来源于我们对外部数据如理财产品说明书、风险揭示书的爬取与分析，最终提取出金融名词的相关解释并流入 MySQL 数据库中存储；提取出的理财产品属性流入 neo4j 数据库中存储。

# 七、基础设施架构

## 7.1 业务分析

项目的最终实现目标是建立一个智能的线上金融产品指导 AI，以对话的形式解答用户所提出的金融产品领域的相关疑问，从而使没有金融领域经验的用户也能获取金融产品最基础的专业知识，降低其进行金融产品投资时的理解性损失。

具体业务流程如下：

### 1. 前后端建立与连接

具体目标功能：

- 1) 后端（由基于 Python 的 Django 框架实现）
- 2) 前端页面（由 vue 框架，Element ui 组件实现）
- 3) 前后端对话（由 websocket 与前端建立长链接）

### 2. 客户的网页端注册与登录

具体目标功能：

- 1) 登录、注册模块
- 2) 访问控制权限管理

### 3. 客户以对话的形式提问有关金融产品问题

具体目标功能：

- 1) 对话模块
- 2) 金融产品数据信息获取（爬虫，文本提取）
- 3) 数据存储（MySQL, neo4j）
- 4) 语言 NLP 主体、属性值识别及相关处理
- 5) 数据调用（API）
- 6) 数据库查询并返回结果

### 4. 客户以对话的形式提问其他问题

具体目标功能：

- 1) 闲聊模块
- 2) 图灵 API 的调用

### 5. 用户反馈产品体验

具体目标功能：

1) 反馈模块

最终实现业务流程的各类技术选择将由下一板块进行介绍。

## 7.2 技术选择

分析了项目实现的业务流程，接下来我们具体阐述所使用的的基础技术。我们将从智能小投的基础服务关键实现部分进行精炼介绍，如下图：

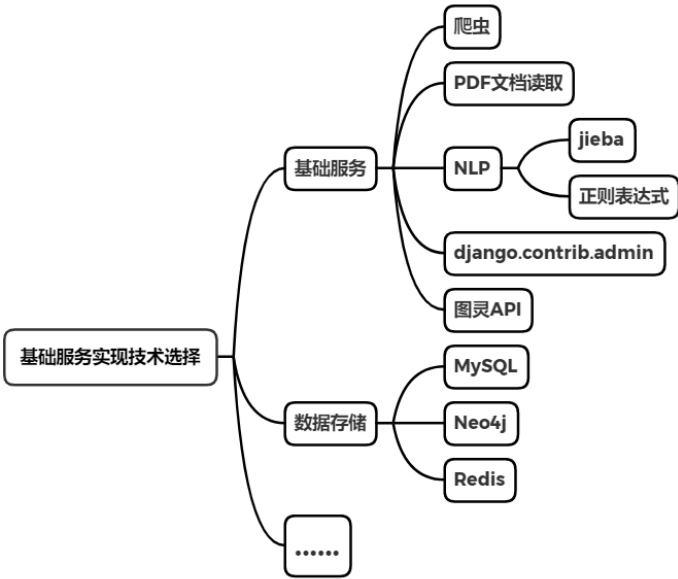


图 7-1 技术选择

### 1. 数据获取——爬虫部分

使用 python 爬虫来获取理财产品《说明书》和《风险揭示书》。

使用 python 爬虫获取 MBA 智库百科的金融词汇解释。

### 2. 数据获取——文档读取部分

文档表格提取，产品文档中均存在多个表格，并且表格内的信息也大部分涵盖了该理财产品说明书的主要内容，包括理财计划要素表和投资比例表等。

表格信息提取，在获取表格后对表格中的数据进行了相应的处理，以便更好地进行后续的存储和查询操作。

文档金融专有词汇提取，对产品文档中的非表格数据进行金融词汇专有名词的提取。主要采用 NLP 自然语言等处理方式。

### 3. 数据获取——对话的 NLP 自然语言处理

通过 jieba、正则表达式等技术，利用 NLP 自然语言处理将用户对话进行拆分，获取需求并将其转化为规范字段。

### 4. 数据存储——MySQL 数据库

金融名词专有词汇将被储存在 MySQL 数据库中。

### 5. 数据存储——Neo4j 数据库

将产品属性写入 Neo4j 图数据库中，用以直观、高效的查询并展示关联数据。

## 7.3 Hosting 网络托管

在实现了基础服务技术的基础上加入 Hosting 网路托管技术，采用基于 B/S 架构的 SaaS 应用模式，提高项目的部署效率。

### 1. Hosting 的定义

Hosting，作为 ASP 的一个分支，是指通过外包方式承揽信息技术应用、软硬件系统服务和网络基础服务。Hosting 可分网络托管、网站托管和应用托管，主要提供的服务包括域名注册、虚拟主机、网站建设和主机托管租用等一系列互联网基础托管服务。

### 2. SaaS 模式

ASP 应用软件服务供货商，应用软件服务供货负责开发、执行与维护的软件，而企业只要透过网络租赁、使用所需服务。

SaaS 软件即服务，与 ASP 的概念有些类似，是透过网络提供从商业应用软件、中间件软件、数据库到应用服务器底层基础架构的一种服务模式。

比较 SaaS 与 ASP：ASP 以提供一对一服务为主，而以 SaaS、PaaS 为主的平台服务则提供一对多服务，实现了透过资源共享降低 IT 营运支出的好处。且相较于 ASP，SaaS 更富弹性与多样化。因此 SaaS 很快代替 ASP 成为了热门的信息服务模式。

### 3. 使用 B/S 架构的 SaaS 应用

B/S (Browser/Server)：浏览器/服务器架构，在这种架构下，用户工作界面是通过浏览器来实现，极少部分事务逻辑在前端（Browser）实现，主要事务逻辑在服务器端 (Server) 实现，形成三层 3-tier 结构，如图 3-1 所示。

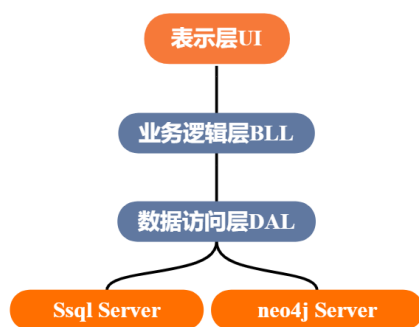


图 7-2 三层 3-tier 结构

基于 B/S 架构的 SaaS 应用，由运营商提供软硬件等基础设施和技术服务，用户由 Internet 通过本地浏览器即可实现应用，如图 3-2 所示。

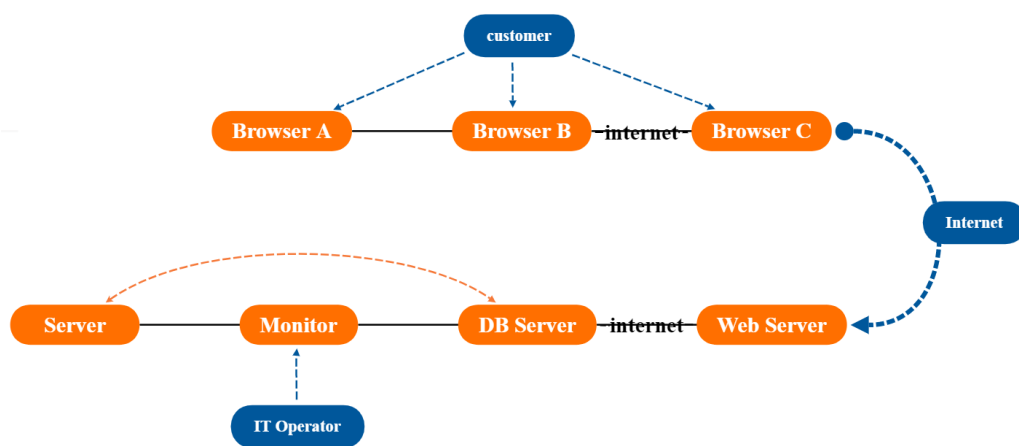


图 7-3 基于 B/S 架构的 SaaS 应用

这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本（TCO）。以目前的技术看，局域网建立 B/S 结构的网络应用，并通过 Internet/Intranet 模式下数据库应用，相对易于把握、成本也是较低的。它是一次性到位的开发，能实现不同的人员，从不同的地点，以不同的接入方式（比如 LAN，WAN，Internet/Intranet 等）访问和操作共同的数据库；它能有效地保护数据平台和管理访问权限，服务器数据库也很安全。

## 7.4 性能分析

在进行了技术选择和架构设计后，下面对方案设计的优势和性能进行分析展示：



## 7.4.1 数据存储采用 Neo4j 数据库与 SQL 结合

### 1、高性能表现

Neo4j 是一个原生的图数据库引擎，它存储了原生的图数据，因此，可以使用图结构的自然伸展特性来设计免索引邻近节点遍历的查询算法，即图的遍历算法设计。图的遍历是图数据结构所具有的独特算法，即从一个节点开始，根据其连接的关系，可以快速和方便地找出它的邻近节点。这种查找数据的方法并不受数据量的大小所影响，因为邻近查询始终查找的是有限的局部数据，不会对整个数据库进行搜索。所以，Neo4j 具有非常高效的查询性能，相比于 RDBMS 可以提高数倍乃至数十倍的查询速度。而且查询速度不会因数据量的增长而下降，即数据库可以经久耐用，并且始终保持最初的活力。不像 RDBMS 那样，因为不可避免地使用了一些范式设计，所以在查询时如果需要表示一些复杂的关系，势必会构造很多连接，从而形成很多复杂的运算。并且在查询中更加可怕的是还会涉及大量数据，这些数据大多数与结果毫无关系，有的可能仅仅是通过 ID 查找它的名称而已，所以随着数据量的增长，即使查询一小部分数据，查询也会变得越来越慢，性能日趋下降，以至于让人无法忍受。

### 2、设计的灵活性

在日新月异的互联网应用中，业务需求会随着时间和条件的改变而发生变化，这对于以往使用结构化数据的系统来说，往往很难适应这种变化的需要。图数据库结构的自然伸展特性及其非结构化的数据格式，让 Neo4j 的数据库设计可以具有很大的伸缩性和灵活性。因为随着需求的变化而增加的节点、关系及其属性并不会影响到原来数据的正常使用，所以使用 Neo4j 来设计数据库，可以更接近业务需求的变化，可以更快地赶上需求发展变化的脚步。

大多数使用关系型数据库的系统，为了应对快速变化的业务需求，往往需要采取推倒重来的方法重构整个应用系统。而这样做的成本是巨大的。使用 Neo4j 可以最大限度地避免这种情况发生。虽然有时候，也许是因为最初的设计考虑得太不周全，或者为了获得更好的表现力，数据库变更和迁移在所难免，但是使用 Neo4j 来做这项工作也是非常容易的，至少它没有模式结构定义方面的苦恼。

### 3、开发的敏捷性

图数据库设计中直观明了的数据模型，从需求的讨论开始，到程序开发和实

现，以及最终保存在数据库中的样子，它的模样似乎没有什么变化，甚至可以说本来就是一模一样的。这说明，业务需求与系统设计之间可以拉近距离，需求和实现结果之间越来越接近。这不但降低了业务人员与设计人员之间的沟通成本，也使得开发更加容易迭代，并且非常适合使用敏捷开发方法。

Neo4j 本身可伸缩的设计灵活性，以及直观明了的数据模型设计，还有其自身简单易用的特点等，所有这些优势充分说明，使用 Neo4j 很适合以一种测试驱动的方法应用于系统设计和开发自始至终的过程之中，通过迭代来加深对需求的理解，并通过迭代来完善数据模型设计。

#### 4、与其他数据库的比较

与当前一些主流的数据库相比，不管是传统的关系型数据库，还是 NoSQL 数据库，或者同类的图数据库，Neo4j 都是出类拔萃的。

在传统的 RDBMS 中，如果要表现一个部门的用户，即 1.2 节提到的例子，按照第三范式的设计要求，至少需要三张表格来表示，即部门表、用户表和部门-用户关系表，这样实体和关系就被人为地隔开了，它们是完全分离的，存在于不同的表中，这就给查询带来了一定的难度，从而影响了查询的性能。而 Neo4j 所表现的是实体的联系本身，它表现了现实世界中事物联系的本质，它的联系在节点创建时就已经建立，所以在查询中能以快捷的路径返回关联数据，从而表现出非常高效的查询性能。

Key-Value 的数据库虽然能提供高性能的查询，但它所能表示的内容是有限的。实际上，Neo4j 节点的属性就是一些 Key-Value 的数据集合。而 Neo4j 通过节点和关系的属性可以表现更为丰富多彩的内容，这是其他 Key-Value 的数据库所无法比拟的。

对于 Key-Document 文档数据库来说，相对于 Key-Value 数据库，内容是丰富了些，但美中不足的是，一个文档经不起内容的变更或修改。如果用 Neo4j 的节点及其属性来表示，则处理这种类似的变更却是轻而易举的。

在图数据库领域，除 Neo4j 之外，还有其他如 OrientDB、Giraph、AllegroGraph 等各种图数据库。跟所有这些图数据库相比，Neo4j 的优势表现在以下两个方面。

(1) Neo4j 是一个原生图计算引擎，它存储和使用的数据自始至终都是使

用原生的图结构数据进行处理，不像有些图数据库，只是在计算处理时使用了图结构数据，而在存储时还将数据保存在关系型数据库中。

(2) Neo4j 是一个开源的数据库，其开源的社区版吸引了众多第三方的使用和推广，如开源项目 Spring Data Neo4j 就是一个做得很不错的例子，同时也得到了更多开发者的拥趸和支持，聚集了丰富的可供交流和学习的资源与案例。这些支持、推广和大量的使用，反过来会很好地推动 Neo4j 的发展。

## 5、综合表现

Neo4j 查询的高性能表现、易于使用的特性及其设计的灵活性和开发的敏捷性，以及坚如磐石般的事务管理特性等特点，都充分说明了使用 Neo4j 是一个不错的选择。有关它的所有优点，总结起来，主要表现在以下几个方面。

- (1) 闪电般的读/写速度，无与伦比的高性能表现。
- (2) 非结构化数据存储方式，在数据库设计上具有很大的灵活性。
- (3) 能很好地适应需求变化，并适合使用敏捷开发方法。
- (4) 很容易使用，可以用嵌入式、服务器模式、分布式模式等方式来使用数据库。
- (5) 使用简单框图就可以设计数据模型，方便建模。
- (6) 图数据的结构特点可以提供更多更优秀的算法设计。
- (7) 完全支持 ACID 完整的事务管理特性。
- (8) 提供分布式高可用模式，可以支持大规模的数据增长。
- (9) 数据库安全可靠，可以实时备份数据，很方便恢复数据。
- (10) 图的数据结构直观而形象地表现了现实世界的应用场景。

## 7.4.2 VUE 前端

### 1、轻量级框架

只关注视图层，是一个构建数据的视图集合，大小只有几十 kbVue.js 通过简洁的 API 提供高效的数据绑定和灵活的组件系统。

### 2、双向数据绑定

所谓的响应式数据绑定。这里的响应式不是 @media 媒体查询中的响应式布局，而是指 vue.js 会自动对页面中某些数据的变化做出同步的响应。

也就是说，vue.js 会自动响应数据的变化情况，并且根据用户在代码中预

先写好的绑定关系，对所有绑定在一起的数据和视图内容都进行修改。而这种绑定关系，就是以 `input` 标签的 `v-model` 属性来声明的，因此你在别的地方可能也会看到有人粗略的称 `vue.js` 为声明式渲染的模版引擎。

这也就是 `vue.js` 最大的优点，通过 MVVM 思想实现数据的双向绑定，让开发者不用再操作 dom 对象，有更多的时间去思考业务逻辑。

### 3、组件化

在前端应用，我们是否也可以像编程一样把模块封装呢？这就引入了组件化开发的思想。`Vue.js` 通过组件，把一个单页应用中的各种模块拆分到一个一个单独的组件（component）中，我们只要先在父级应用中写好各种组件标签（占坑），并且在组件标签中写好要传入组件的参数（就像给函数传入参数一样，这个参数叫做组件的属性），然后再分别写好各种组件的实现（填坑），然后整个应用就算做完了。

### 4、视图, 数据, 结构分离

使数据的更改更为简单, 不需要进行逻辑代码的修改, 只需要操作数据就能完成相关操作。

### 5、虚拟 DOM

现在的网速越来越快，很多人家里都是几十甚至上百 M 的光纤，手机也是 4G 起步，按网页才几百 K，而且浏览器本身还会缓存很多资源文件，那么几十 M 的光纤为什么打开一个之前已经打开过，已经有缓存的页面还是感觉很慢呢？这就是因为浏览器本身处理 DOM 也是有性能瓶颈的，尤其是在传统开发中，用 JQuery 或者原生的 JavaScript DOM 操作函数对 DOM 进行频繁操作的时候，浏览器要不停的渲染新的 DOM 树，导致页面看起来非常卡顿。

而 Virtual DOM 则是虚拟 DOM 的英文，简单来说，他就是一种可以预先通过 JavaScript 进行各种计算，把最终的 DOM 操作计算出来并优化，由于这个 DOM 操作属于预处理操作，并没有真实的操作 DOM，所以叫做虚拟 DOM。最后在计算完毕才真正将 DOM 操作提交，将 DOM 操作变化反映到 DOM 树上。

### 6、运行速度更快

比较与 react 而言, 同样都是操作虚拟 dom, 就性能而言, vue 存在很大的优势。

## 7.5 敏感数据保护

作为一个金融产品信息指导项目，敏感数据的保护是项目的重中之重，基于数据的存储、查询、平台选择、用户选择，我们采用了以下几大数据保护措施。

### 1、基于数据保护套件的企业级数据保护

基于企业级数据保护套件，通过连续复制、快照、重复数据删除和传统备份以及归档等技术进行数据保护，企业版利用单一、综合性的产品满足各种规模企业的需求。建立完善的数据保护架构和产品相结合，为用户提供端到端方案。

#### 1. 个性化高效备份方案

用于 Neo4j 数据库和 MySQL 数据库备份的数据保护套件高效实现对物理和虚拟环境的数据保护，支持多种部署模式和保护技术（包括重复数据删除备份、备份到磁盘、基于快照的备份和备份到磁带），可灵活地组合和匹配组件，在 IT 演进过程持续优化数据保护性能和使用效率以降低数据保护成本，提高数据保护服务水平。

#### 2、应用的数据保护方案

针对应用的数据保护，可满足关键任务应用要求。通过从主存储或应用服务器直接备份到云端，备份速度可提高 20 倍，恢复速度可提高 10 倍；同时，保留和保护数据所需的磁盘存储量至少可缩减为原来的 1/30 至 1/10。可以设置存储的数据保护利用精简快照，提供快速备份，最大限度降低数据备份对资源的消耗。可以设置存储基于 iSCSI SAN 架构，可通过快照自动从中央存储池中备份，减少备份窗口，从而减轻备份和恢复的对业务的影响。PS 系列的所有备份和保护功能(包括快照、克隆、复制和调度)，不需要额外的成本，降低了数据保护开支。

#### 3、VMware 的数据保护方案

针对 VMware 的数据保护套件为基于 VMware 的环境（包括备份和恢复、连续复制、监控和分析，以及搜索功能）提供端到端的数据保护。此产品是纯软件的数据保护解决方案，可在虚拟化和云环境中简化部署和管理。可以实现与 VMware vCenter 集成为数据保护功能做了高度整合，以支持运行 VMware 虚拟机(VMs)的 VM 一致性回放。

#### 4、归档的数据保护方案

针对归档的数据保护套件是考虑针对所有事件归档和电子发现的最佳解决

方案。作为数据保护套件系列的一部分，此解决方案为企业提供了对其信息完全的所有权和控制权。客户可以通过以下方式来降低成本：回收有价值的主要数据存储空间、优化服务器和操作性能、符合公司合规性规定、满足电子发现和诉讼需求等。

## **5、支持云计算**

数字经济下，用户业务会持续在混合 IT 环境运营。设置的基础架构既能实现私有云高效保护，也能助力用户对公有云中的数据和应用进行备份。管理员可以本机控制数据在云层面的备份数据重复删除和移动策略，界定哪些客户端需进行本地备份，哪些客户端需进行云备份。多与各云平台进行战略合作，为用户提供私有云和混合云备份归档方案和服务。

## **7.6 访问控制**

### **7.6.1 由安全问题衍生而来的访问控制**

不同于需求和技术层面的隐私数据保护，在系统和云层面的访问控制管理也是隐私保护和用户管理的一大要点。

云计算结合分布式技术，为用户提供超大规模计算、存储及软件等服务，大大提高了应用中对大数据的使用、分析和管理的效率。但在服务过程中，数据由本地存储转向为存储到云端，这在带来便利的同时，也引起了相应的数据安全及隐私信息泄露的忧虑。

针对这个问题，结合基于角色的访问控制、基于属性的访问控制方法，配置了应用的访问控制权限管理。

### **7.6.2 以具体应用举例安全问题**

为了解决云服务环境下，访问控制中存在的数据安全性问题，本文提出了基于客体属性匹配的访问控制方法。

为了更直观的说明为什么要进行访问控制管理，以具体应用为例：用户甲向云端服务器上传一份金融产品业务信息，乙、丙、丁等用户作为数据使用者可以向云端提出申请访问、查看并使用数据文件，如图 3-3 所示。

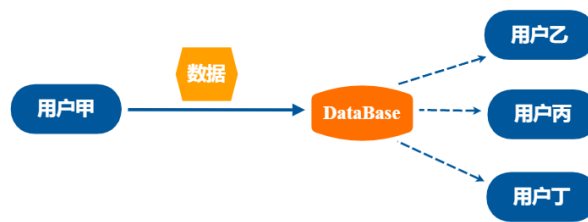


图 7-4 数据上传与用户访问

在以上过程中存在以下问题：

- 1) 数据拥有者甲担心上传信息中包含的关键数据被泄露；
- 2) 甲对访问者乙、丙、丁等的真实身份和其对数据操作行为是否为恶意的担忧；
- 3) 乙、丙、丁对云端数据的真实性、可用性的忧虑

综合分析以上场景，从用户甲的角度出发，保障数据贡献者共享数据的安全性、保障合法用户进行合理访问等是需要解决的问题。

从使用数据的用户乙、丙、丁等角度出发，保障所请求的数据的完整性、可用性等是主要关心的问题。

### 7.6.3 访问权限控制

#### 1、基于角色的访问控制

基于角色的访问控制（Role Based Access Control, RBAC）模型，每一种角色对应一组相应的权限，用户组、用户、角色、权限和数据组这五者之间的关系结构如图 3-4 所示，“角色”的引入简化了授权，方便了系统管理。

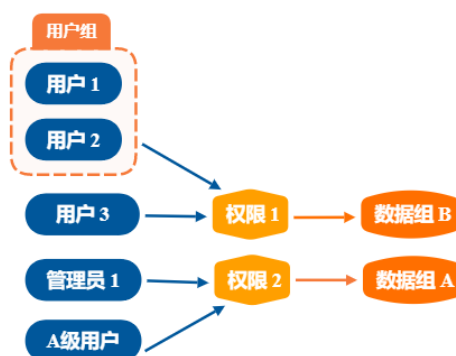


图 7-5 基于角色的访问控制

RBAC 具有以下优点：

- 1) 简化了权限管理，只需将不同角色授权给相应用户；
- 2) 实现责权分离原则；
- 3) 支持最小特权原则；
- 4) 符合内部管理结构，方便实用。

但与此同时 RBAC 的缺点也很明显：它无法做到对未知访问需求的虚无角色的权限设置，在超大规模的用户场景中，角色的管控、权限授予的复杂度也在呈指数级别的增长。

## 2、基于属性的访问控制

基于属性的访问控制（Attribute-Based Access Control, ABAC）为解决上述问题提供了新的方法，更直观的基于属性的访问控制机制展示如图 3-5。



图 7-6 基于属性的访问控制机制

ABAC 机制：通过定义一种访问控制范式，使用将属性组合在一起的策略，为不同用户提供动态、上下文感知和风险智能防护的访问权限。策略可使用各类属性，且允许简单的与非逻辑判别，例如：“IF 请求者是管理者，THEN 允许对敏感数据的读/写访问”。

ABAC 机制的核心是主体的请求发起后，联合主体的属性、客体的属性与环境状态作为输入，从 PEP 策略决策点（Policy Decision Point, PDP）获取规则，PDP 计算，最后判定主体的请求是否合理予以执行。

## 7.7 可扩展性

在可扩展性部分，将分别针对前端和后端使用的基础设施技术和架构进行分



析。

### 7.7.1 前端可扩展性

#### 1、向上扩展

Vue 提供了强大的路由来应对大型应用。React 社区在状态管理方面非常有创新精神（比如 Flux、Redux），而这些状态管理模式甚至 Redux 本身也可以非常容易的集成在 Vue 应用中。实际上，Vue 更进一步地采用了这种模式(Vuex)，更加深入集成 Vue 的状态管理解决方案。Vuex 能为你带来更好的开发体验。两者另一个重要差异是，Vue 的路由库和状态管理库都是由官方维护支持且与核心库同步更新的。React 则是选择把这些问题交给社区维护，因此创建了一个更分散的生态系统。

Vue 提供了 CLI 脚手架，能让你通过交互式的脚手架引导非常容易地构建项目。你甚至可以使用它快速开发组件的原型。React 在这方面也提供了 create-react-app，但是现在还存在一些局限性：

- a. 它不允许在项目生成时进行任何配置，而 Vue CLI 运行于可升级的运行时依赖之上，该运行时可以通过插件进行扩展。
- b. 它只提供一个构建单页面应用的默认选项，而 Vue 提供了各种用途的模板。

#### 2、向下扩展

就像 Vue 向上扩展好比 React 一样，Vue 向下扩展后就类似于 jQuery。你只要把如下标签放到页面就可以运行：

```
<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
```

图 7-7 Vue 向下扩展

然后你就可以编写 Vue 代码并应用到生产中，你只要用 min 版 Vue 文件替换掉就不用担心其他的性能问题。由于起步阶段不需学 JSX，ES2015 以及构建系统，所以使用 Vue 的开发者只需不到一天的时间阅读指南就可以建立简单的应用程序。

### 7.7.2 后端可扩展性

#### 1、Neo4j 图数据库

集群实现了高可用性，对于高并发的读任务可以分散到各个节点上。

## 2、MySQL 数据库

通过主键和索引，设置主键或者索引的查找，主键唯一标识数据库表中的每个记录。这意味着每一行都具有用于主键和并且主键的值不允许为空。由于 MySQL 能够比较快速的处理整数，所以主键列的类型通常是具有自动递增属性的整数类型。自动递增生成的主键值中，下一行的主键值大于前一行的主键值。

主键影响 MySQL 数据库性能的关键因素。例如，以主键为限定条件选取一条记录(例如 `SELECT * FROM 消息 WHERE news_id = 40000`)只需要 0.0953s 了，而没有主键限定条件的类似查询需要 3.6120s 的时间才能返回结果。其他类型的索引，如唯一性和关键字约束，也提供了性能上的改进。

对于扩展性，有三种方式：数据分片、应用拆分、数据备份。Mysql 也一样，首先是数据备份。

### a. 主从数据库

数据备份这里体现的是添加主从的方式。即将数据库分成两个数据库，用于读写分离，一般主库用于写，从库用于读，如图 3-5 所示。

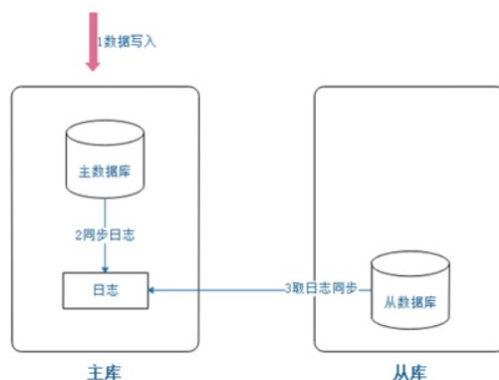


图 7-8 主从数据库

### b. 数据分区

分区方式是物理级别的分区，它会把同一个表通过指定的算法分离存储到不同位置，最终达到提高表的量级的目的。更深一层说，一个表的数据通过物理分区分离到数个不同的物理区域，每个分区都是独立的。这样数据的扩展性通过数据分区而更加独立。

### c. 分库分表

再进一步，就是分库分表了。先讨论一下分库。分库很简单，把不同的表分

到不同的库，或者把相同的表不同的数据分到不同的库。第一，不同的表分到不同的库。把主库里的表分离出来，分到各个不同业务逻辑上的库。第二，相同的表不同的数据分到不同的库。这种与分表挂钩，即把数据分离到不同的库，不同库的表都一致。本产品所采用的方式即为分库分表形式，使存储产品更加具有扩展性，如图 3-6。

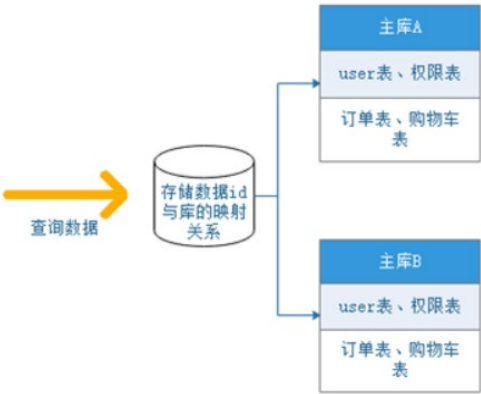


图 7-9 数据分库分表

## 7.8 可用性

### 1、云计算带来的高可用性

在传统 IT 系统中，个人计算机的软、硬件故障导致计算不可用的概率接近 95%，其中又以软件故障最为普遍，而因停电、断网导致计算不可用的概率微乎其微，服务器故障导致计算不可用的概率还不到 5%。而当项目采用云计算后，云终端出故障的概率几乎可以忽略不计。云端采用多路供电、引入集群技术、容错技术及负载均衡技术等措施确保计算持续可用，这样由云端、网络、终端组成的云计算系统具备极高的可靠性和安全性，计算可用性非常高。

### 2、数据库读写分离带来的高可用性

MHA (Master High Availability) 目前在 MySQL 高可用方面是一个相对成熟的解决方案，是一套优秀的作为 MySQL 高可用性环境下故障切换和主从提升的高可用软件。在 MySQL 故障切换过程中，MHA 能做到在 10~30 秒之内自动完成数据库的故障切换操作，并且在进行故障切换的过程中，MHA 能在最大程度上保证数据的一致性，以达到真正意义上的高可用。

该软件由两部分组成：MHA Manager (管理节点) 和 MHA Node (数据节点)。

MHA Manager 可以单独部署在一台独立的机器上管理多个 master-slave 集群，也可以部署在一台 slave 节点上。MHA Node 运行在每台 MySQL 服务器上，MHA Manager 会定时探测集群中的 master 节点，当 master 出现故障时，它可以自动将最新数据的 slave 提升为新的 master，然后将所有其他的 slave 重新指向新的 master。整个故障转移过程对应用程序完全透明。其集群架构如图 3-7 所示。

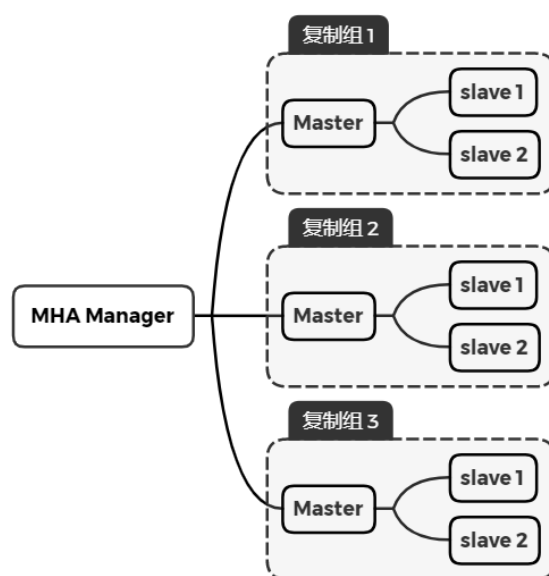


图 7-10 MHA Manager 管理多组主从复制

## 7.9 信息生命周期管理

ILM (Information Lifecycle Management), 根据 META 集团的定义, 信息 (或数据) 生命周期管理是, 信息在储存媒介网络之内流动的过程, 而这种过程需要确保企业获取需要的商业信息, 并向客户提供一个良好的服务水平, 同时把单位成本降到最低。ILM 还要满足日益增长的对于成熟和自动化存储管理的需求, 这可以在保持企业对于商业环境变化作出快速反应的能力的同时, 提高个人的工作效率。这一种定义强调的是过程的概念。

智能小投获取用户数据的来源主要从智能问答和反馈上。从智能问答中, 我们可以记录用户询问的信息和点击获取的信息内容, 以达到能够准确将用户画像作为信息数据获取内在价值的作用。而随着时间变化, 用户在各种金融知识层面的理解会随着他提出问题的深入而不断增加, 因此当用户长久性使用问答, 用户

画像信息需根据问题不断更新，这样才能获得更加准确的用户数据。更新周期可以是任意时间，不同用户存在不同更新周期。

信息技术人员在一段时间结束以后，是不能随便清除所有数据库中信息的。其中的冗余数据，当然可以马上删除，但是那些个属于业务交易、人力资源和财务等重要数据内容却应当被保留，作进一步的分类、储存和保护，为了更长期间内的使用。同时，由于这一信息可能在一定时期内都会有人要查阅，所以它也不应该被简单地移动到后备数据系统。在这里，关键的问题是识别的流程、制度规定以及方便实用的技术支持能力。

### 信息的价值随时间变化

信息生命周期管理就是…

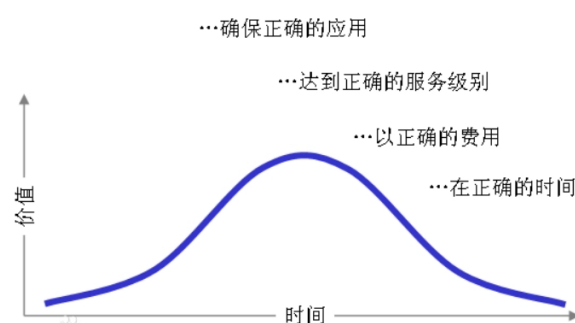


图 7-11 信息生命周期

## 7.10 部署

### 7.10.1 Neo4j 数据库云端部署

Linux 下部署 Neo4j

#### (1) 在 Linux 系统下安装 java 8

首先上官网下载自己系统对应的 jdk1.8，之后配置环境变量。

#### (2) 安装 Neo4j

首先进入 Neo4j 的官网 <https://neo4j.com/>，找到下载界面，点击 Download Neo4j Server，在社区版中选择适合 Linux/Mac 系统的版本，进行下载。如果没有图形化界面的话，可以使用 wget 命令来下载：

```
wget https://neo4j.com/artifact.php?name=neo4j-community-3.5.11-unix.tar.gz
```

接下来，解压 tar.gz 为后缀的压缩包，然后就可以进入 Neo4j 的文件夹了。

### (3) 配置 Neo4j 允许远程访问

进入 config 文件夹，修改 neo4j.conf 文件中的配置：

```
vim conf/neo4j.conf
```

我们需要取消第 54 行的注释，即删除

#dbms.connectors.default\_listen\_address=0.0.0.0 前面的#即可

否则 Neo4j 只能通过本地来访问。Neo4j 的默认 bolt 协议的端口是 7687，http 和 https 的端口分别是 7474 和 7473。如果想要修改端口的话，可以去掉 71，75 和 79 行的注释，并且修改端口号，如下：

dbms.connector.bolt.listen\_address=:7687 #去掉 71 行注释，把 7687 改成想要的端口号

dbms.connector.http.listen\_address=:7474 #去掉 75 行注释，把 7474 改成想要的端口号

dbms.connector.https.listen\_address=:7473 #去掉 79 行注释，把 7473 改成想要的端口号

仅仅这样设置是不够的，我们需要在系统中让防火墙开启这几个端口，如下：

firewall-cmd --zone=public --add-port=7474/tcp --permanent #开启 7474 端口

firewall-cmd --zone=public --add-port=7687/tcp --permanent #开启 7687 端口

firewall-cmd --zone=public --add-port=7473/tcp --permanent #开启 7473 端口

firewall-cmd --reload # 生效配置

通过以下语句查看这几个端口是不是打开了：

```
firewall-cmd --zone=public --list-ports
```

接着进入 bin 文件夹，启动 Neo4j 服务：

```
./bin/neo4j start
```

我们输入 curl localhost:7474 命令，可以看到能够获取到数据，说明 neo4j 确实启动了，curl 返回的数据如下：

```
{  
  "data" : "http://localhost:7474/db/data/",  
  "management" : "http://localhost:7474/db/manage/",  
  "bolt" : "bolt://localhost:7687"  
}
```

#### (4) 配置端口映射来让外网访问 Neo4j

实际操作中，我目前项目的适合需要让外网能够访问 Neo4j，所以在云服务中申请了一个公网 ip，并且增加了相关的端口映射。这样可以将公网 IP 地址的某个端口的请求转发到我们安装了 Neo4j 的云主机的 7474，7687 和 7473 端口中。

#### (5) 端口映射遇到的问题

在端口映射完成了之后，我还是无法从外网直接访问搭建在云服务器上的 Neo4j。后来发现这是因为云服务商一般是不会让 7474，7687 这些不常用的端口通过的，所以还需要在云平台上添加安全组规则，来允许这几个端口的 TCP 连接。

#### (6) 登录 Neo4j

通过以上操作，环境就全部配置好了，可以通过浏览器从外网访问 Neo4j 数据库。假设公网 ip 是 x1，外网 y1 端口映射到了内网的 7474 端口，那么可以在浏览器中输入 <http://x1:y1/browser>，就可以进入云服务器上部署的 Neo4j 的管理页面。数据库的默认账号密码都是 neo4j，第一次登录会让你设置新的密码，之后通过账户密码即可登录。

### 7.10.2 MySQL 数据库云端部署

同上在 linux 中部署 Mysql 数据库

#### (1) 在 Linux 系统下安装 mysql 数据库

#### (2) 将本地数据转移到 linux 数据库中

#### (3) 设置本地登录

先检查是否有 mysql 用户组和 mysql 用户，如果没有则添加。再启动 Mysql 修改权限，使不同的用户有不同的权限等级。

#### (4) 增加远程登陆权限

上一步即可本地登录，但远程登录会报错，如图：



解决这一问题，需要本地登陆 MySQL 后执行如下命令：

```
grant all privileges on *.* to root@'%' identified by 'root';  
flush privileges;
```

执行之后即可远程登录，需要注意的是云服务器默认是没有开 3306 端口的，所以你要先开启 3306 端口。

#### （5）登录 Mysql

通过以上操作，环境就全部配置好了，可以通过其他机器访问某地址下固定端口开放的数据库。通过远程登录，就可以进入云服务器上部署的 Mysql 的管理页面。



# 八、安全架构

## 8.1 关键资产



图 8-1 关键资产

上图为智能小投产品的关键资产。本产品的关键资产共分为两大类，一类是是以数据形式存在的无形资产，另一类是以程序形式存在的无形资产。

在数据资产中，存在三个大类：用户数据、产品说明书数据和专有名词数据。用户的个人信息和查询记录是用户数据中最关键的两项资产，它们的机密性、完整性、可用性以及如何被使用是需要得到保障的。在产品说明书数据和专有名词数据中，说明书 PDF 本身以及专有名词解释不属于关键资产，而从说明书中提取出的理财产品属性以及用于存储的 Neo4j 数据库和 MySQL 数据库则属于我们的关键资产。

在程序资产中，爬虫、PDF 信息提取和 NLP 分析三个模块对应的程序文件（或程序项目包）属于智能小投产品的关键资产。

## 8.2 威胁模型

因为缺乏专业的威胁模型构建知识和经验，并且银行产品相对于其他行业的产品有更多的合规要求，需要更充分的隐私保护机制。所以，我们使用 LINDDUN GO<sup>2</sup>来帮助我们以一个初学者的简单方式为智能小投产品建立一个相对专业的威胁模型。

<sup>2</sup> <https://www.linddun.org/go>

LINDDUN 是一种隐私威胁建模方法，支持分析师系统地引出和减轻软件架构中的隐私威胁。LINDDUN GO 是一种根据 LINDDUN 威胁类别构建的威胁建模方法，旨在为威胁建模提供结构化但轻量级的支持。

首先，使用已经建立好数据流图来表示我们的系统模型。我们初始的数据流图如下。

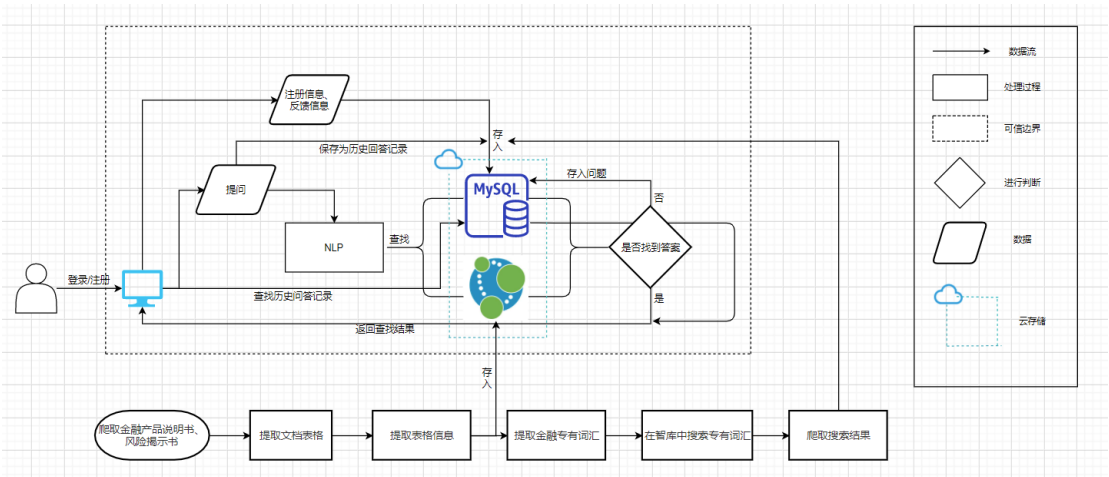


图 8-2 基础数据流图

之后，以 LINDDUN GO 中的 5 个主要热点为目标，使用其提供的威胁类型卡来发现每个热点可能存在的威胁。5 个热点说明如下图所示。

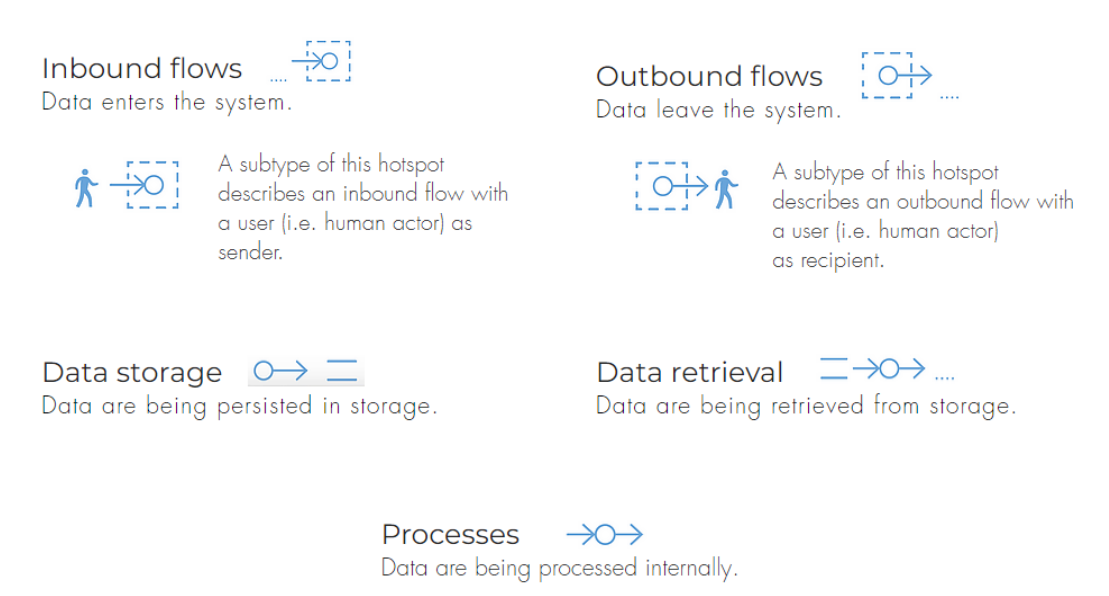


图 8-3 LINDDUN GO 威胁模型

根据 LINDDUN GO 提供的威胁类型卡，共识别出来了 6 种威胁，其中 2 个威胁位于入站流中，4 个威胁位于数据存储中。具体的说明如下表所示，详细的威

胁类型卡请参见附录。

表 8-1 威胁类型卡

	威胁目标	类型	威胁描述
数据存储	MySQL（用户信息）	个人数据	被存储的个人信息 包含真实的信息 （比如姓名、地址等）
	MySQL（用户信息、历史记录、没有找到答案的问题）		数据主体（用户） 不能获得他们的个人数据或者不能导出他们的个人数据
	MySQL（用户信息、历史记录、没有找到答案的问题）		数据主体（用户） 不能删除或修改他们的个人数据
	MySQL（用户信息、历史记录、没有找到答案的问题）	许可	在使用数据前没有 获得数据主体（用户）的许可或当用户撤销许可后数据仍被使用
进站流	用户注册和登录	用户发送许可	用户在注册和登陆时无法得到有效的提示（提示是否存在该用户、密码错误、忘记密码等）
	用户隐私设置	带有个人信息的进站流	系统未能向用户提供友好、便捷的隐私设置服务

## 8.3 安全解决方案设计

针对由 LINDDUN 构建得到的威胁模型中识别出来的 6 种威胁，我们提出以下解决方案来缓解威胁。

**方案一：**将用户核心个人信息（身份证、手机号等）存储在本地 MySQL 数据库中，并用内部的用户编号和用户名称与云上 MySQL 中用户的其他非核心个人信息进行链接，在保证信息一致性、可用性的同时提高用户核心信息的机密性。

**方案二：**在用户个人信息页面和历史记录页面添加“导出”、“修改”、“删除”（仅在历史记录页面添加）和“注销账户并删除所有个人信息和搜索信息”（仅在用户个人信息页面添加）四个按钮，让用户能够对他们的个人数据进行获取、修改或删除。

**方案三：**在用户注册时，对于用户的核心个人信息（身份证、手机号等），给予用户两个隐私保护的简单选项进行选择——①仅可将其用于本产品的用户身份验证 ②可将其共享给用户正在使用的本公司的其他业务，并依此改善用户体验。在用户历史记录页面，用户可以选择开启或关闭历史记录，并且在开启历史记录时，仍会给予用户两个隐私保护的简单选项进行选择——①仅记录用户的搜索记录，不能用于其他用途。②可以将搜索记录、未找到答案的问题用于协助产品问答系统的改善。用户可以随时更改系统提供的隐私保护的简单选项。在用户进行更改并确认后，系统将从问答机器人训练语料库中自动删除该用户的历史记录并且不再向本公司的其他产品提供用户的关联信息。对于已经提供的用户关联信息将进行删除处理。在此基础之上，在用户注册和开启历史记录时均会附加该产品完整的隐私保护协议。

**方案四：**在用户注册和登陆时，需要检查用户名称是否已被占用、用户名称是否存在、密码和用户名称是否匹配等并将对应的情况反馈给用户。在登陆界面提供“忘记密码”选项。

各个安全解决方案与威胁之间的对应关系如下表所示。

表 8-2 安全解决方案与威胁

	方案描述	解决威胁的描述
方案一	分离用户核心信息与非核心信息	被存储的个人信息包含真实的信息（比如姓名、地址等）
方案二	用户可以导出、修改、删除个人信息	数据主体（用户）不能获得他们的个人数据或者不能导出他们的个人数据
		数据主体（用户）不能删除或修改他们的个人数据
方案三	提供便捷的隐私设置以及允许客户随时修改隐私设置	在使用数据前没有获得数据主体（用户）的许可或当用户撤销许可后数据仍被使用
		系统未能向用户提供友好、便捷的隐私设置服务
方案四	在注册和登录时提供提示	用户在注册和登陆时无法得到有效的提示（提示是否存 在该用户、密码错误、忘记 密码等）

以数据流图（DFD）为基本模型的安全架构模型如下图所示。（和 6.3 介绍的数据流一致）

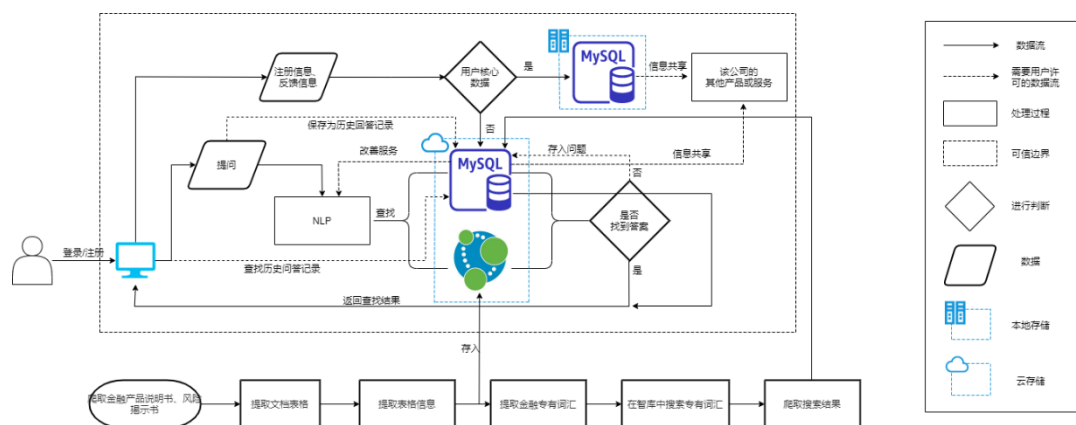


图 8-4 安全架构模型

# 九、云原生架构

## 9.1 云原生架构四要点

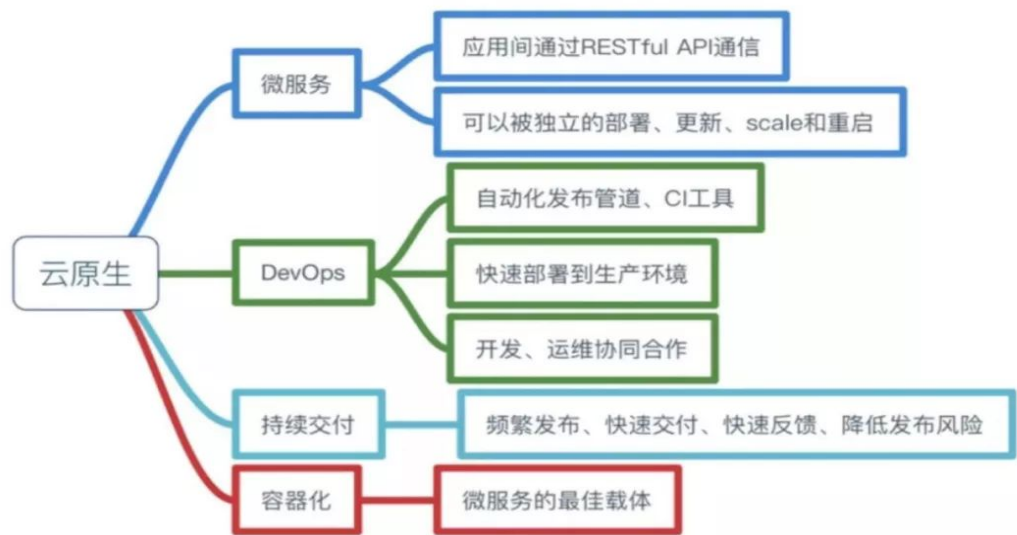


图 9-1 云原生架构四要点

云原生架构有 4 个要点：DevOps+持续交付+微服务+容器化。

**微服务：**本项目按功能划分微服务，服务解耦，内聚更强，变更更易。

**容器化：**本项目采用 Docker 技术进行微服务的容器化，容器化为微服务提供实施保障，起到应用隔离作用，K8S 是容器编排系统，用于容器管理，容器间的负载均衡。

**DevOps：**Dev+Ops，就是开发和运维合体。DevOps 是一个敏捷思维，是一个沟通文化，也是组织形式，为云原生提供持续交付能力。

**持续交付：**持续交付是不误时开发，不停机更新，小步快跑，反传统瀑布式开发模型，这要求开发版本和稳定版本并存。

## 9.2 云原生架构部署

### 9.2.1 总体架构

项目后端架构使用阿里云服务搭建，ECS 可根据业务量动态弹性伸缩，其余服务均采用单实例的方式远程调用。RDS 为主从集群，并配备灾备实例。

## 9.2.2 业务数据层

### RDS

阿里云提供稳定可靠、可弹性伸缩的关系型云数据库 RDS，支持 MySQL，供项目中的金融名词释义数据的存储。主实例配置一个灾备实例，防止意外发生。

### Redis

阿里云 Redis 用做数据缓存，提高问答流程中的查询环节的响应速度。

### Neo4j

Neo4j 服务器主要保存理财产品的相关信息，Neo4j 服务器本身可配置远程访问，通过 REST 进行。若响应速度太慢，也可考虑转为阿里云的图数据库 GDB，图数据库 GDB 是一种支持 Property Graph 图模型，用于处理高度连接数据查询与存储的实时，可靠的在线数据库服务。

## 9.2.3 服务器安全

### 运维层面

阿里云的 web 防火墙和态势感知的服务。这两个服务可以实时监控服务器状态，识别并跟踪攻击来源和类型。另外，可以配置 firewalld。

### 业务层面

针对接口访问的安全性，采取如下措施：

1. 签名验证，防止伪造请求。

2. 访问频次限制。

3. https 访问。

4. 部分敏感数据，如用户的历史查询、用户的个人资料等，使用 RSA 非对称加密。



## 9.2.4 服务器集群

### 主 ECS

通过这台 ECS，可以管理其它从属的 ECS，并查看状态。

### 从属 ECS

这类 ECS 服务器只存放逻辑代码，所以当需求量增加时，只需增加此类服务器的个数即可。而且，在增加个数时，可以使用之前制作好的镜像，创建多台相同环境的 ECS 服务器。

### 负载均衡

购买阿里云的负载均衡实例，由该负载均衡实例接收请求后，会分发到内部服务器，以提高应用的响应性能和可靠性。


## 9.2.5 根据业务量提高性能

http 请求的并发性能可以通过增加 ECS 实现，针对部分耗时较长且无须即时回调的请求，可以用 gearman 异步处理。

数据库的并发连接数可以通过增加配置来提高，也可以通过创建只读实例进行读写分离，提高数据处理能力。后期可能需要搭建 hadoop 管理数据库集群。

其它还可以采用优化 nginx 配置，优化 linux 内核，采用高速固态硬盘等等的手段。

# 附录 1 LINDDUN GO 威胁卡片



## INSUFFICIENT CONSENT SUPPORT

Hotspot

Threat source

STORE  
CONSENT

CONSENT  
O → =

ORGANIZATIONAL

**Data subject consents are not properly taken into account by the relevant processes and data are still being processed with a missing or withdrawn consent.**

**?**

1. Does the system require user consent to process personal data? Does the system fail to take the consent into account?
2. Are means lacking for the data subject to explicitly provide or withdraw consent or are the consents not taken into account for processing operations (e.g. access control)?

**💡**

Wearables data are being used for a research study, but

- The data subject has never given his consent
- The data subject decides to revoke his consent, but there is no technical revocation support
- The system only stops collecting new data but continues its analysis with the previously collected data.

**⚠️**

A consent should always be freely given and thus also be revocable. The system should thus support the consequences of a newly obtained or revoked consent.

**i**

This can be a feature directly available to the data subject or it can be done indirectly (e.g. helpdesk). In both cases, an internal process should be in place to support this.

U5

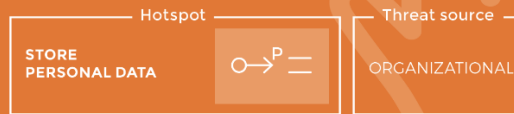
LINDDUN

Copy Game URL

prev. card



## NO ACCESS OR PORTABILITY



**The data subject does not have access to their personal data or is not able to port personal data to another platform/vendor/...**

- ?
1. Are personal data being stored?
  2. Is a process lacking that can extract data (in both a human understandable and computer interpretable format) for an individual data subject?

- 💡
- A wearable device's sensor data are sent to a lifestyle tracking app, but the user is unable to access the statistics and deduced information based on his data that the app has collected and processed.
  - A data subject does not have the means to request their data, neither directly through the system, or indirectly (e.g. a request to a helpdesk which generates the requested data set and forwards it to the data subject.).

- ⚠️
- Access and data portability is a data subject right (GDPR).
  - Does not apply to data that infringes other data subjects' privacy, corporate secrets, etc.
  - This access can also exist outside of the system. (e.g. a helpdesk request)
  - Data portability only involves personal data that was provided directly by the data subject.
- i

U3

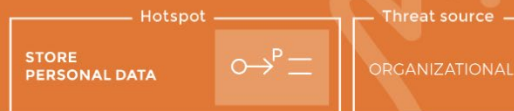
LINDDUN

Copy Game URL

prev. card



## NO ERASURE OR RECTIFICATION



**The data subject cannot request erasure or rectification of personal data.**

- ?
1. Are personal data being stored?
  2. Is a process lacking that can delete and rectify (a subset of) data related to a specific data subject?

- 💡
- A data subject requests deletion of his social media data, but only his account is revoked, the actual data remain.
  - The data subject moved and wants to update their address in the system, but is unable to.

- ⚠️
- Request of erasure and rectification is a data subject right (under certain conditions) (GDPR).
  - Deletion can only be requested 'within reason'.
  - The request can also be made outside of the system (e.g. helpdesk), it however should always be technically feasible to delete the data.
  - A data subject can only request rectification of data to increase accuracy.
- i

U4

LINDDUN

Copy Game URL

prev. card



## NO USER-FRIENDLY PRIVACY CONTROL



**The system does not provide user-friendly privacy control.**  
(e.g. default settings, feedback & awareness tools, user-friendly privacy preferences support)

- ?
1. Does the system process personal data?
  2. Are there no privacy-preserving default settings and/or is there no user-friendly support for the data subject to set privacy preferences or provide awareness information?

- 💡
- When visiting a website for the first time, it requires navigation through several tabs and slide several switches to set the cookie settings and other privacy preferences (no privacy by default, no user-friendliness).
  - When posting on social media, the post is made public by default (no privacy by default, no feedback and awareness tools to educate the data subject on the consequences of these privacy settings).

- ⚠️
- Mainly relevant for systems directly collecting personal data from users (or indirectly through communication metadata) and systems targeted at sharing personal data (e.g. social media).
  - Privacy-friendly settings should be the default.
  - The data subject should be able to easily control his privacy settings.
  - Raising privacy awareness can nudge the data subject into a more privacy-aware behavior.
- i

U2

LINDDUN

Copy Game URL

prev. card



## DETECTABLE CREDENTIALS



**Response of a request allows detection of existence of a user (without actually accessing any data).**

- ?
1. Does the system provide feedback w.r.t. credentials (wrong password, forgot password, ...)?
  2. Would it be a problem for a user if his use of the system is known? (i.e. does the system have a sensitive context?)

- 💡
- When signing in to a service, it is possible to detect that a user exists (i.e. error message of wrong password) or does not exist (i.e. error message of invalid user ID).

- Detecting user accounts also results in security threats (information disclosure/spoofing).
- Often easy to fix by making the responses more privacy-friendly.



D1

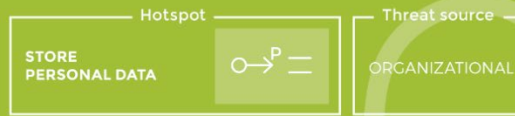
LINDDUN

Copy Game URL

prev. card



## IDENTIFYING STORED DATA



**Personal data being stored can be identified (because they are insufficiently minimized/de-identified before storage).**

- ?
1. Are data stored with identifiable attributes? (i.e. do the data contain identifiers, quasi-identifiers, or links to identified data?)
  2. Can identifying data item(s) be minimized (e.g. removed, de-identified, decentralized)?

- 💡
- The data are being de-identified by replacing identifying attributes (e.g. name, address) by an internal identifier. A link to the actual identity is however being kept, which still allows identifiability.
  - Data are being stored with username, email address or SSN as (internal) identifier [TRIM].

- ⚠️
- Identified data, when they are meant to be anonymous and/or when they do not have to be identified, cause a serious privacy violation.
  - If data cannot be de-identified (because required in the system), they might be de-centralized.
  - Closely related to minimization (Nc5).
- i