<项目名称>

软件架构文档

版本 <1.0>

[注：用方括号括起来并以蓝色斜体（样式=InfoBlue）显示的文本，它们用于向作者提供指导，在发布此文档之前应该将其删除。按此样式输入的段落将被自动设置为普通样式（样式=Body Text）。]

[要定制 Microsoft Word 中的自动字段（选中时显示灰色背景），请选择 File>Properties，然后将 Title、Subject 和 Company 等字段替换为此文档的相应信息。关闭该对话框后，通过选择 Edit>Select All（或 Ctrl-A）并按 F9，或只是在字段上单击并按 F9，可以在整个文档中更新自动字段。对于页眉和页脚，这一操作必须单独进行。按 Alt-F9，将在显示字段名称和字段内容之间切换。有关字段处理的详细信息，请参见 Word 帮助。]

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <2022/6/27> | <1.0> | <初步制定软件架构文档> | <刘骏霖、李忱泽、汪志远、姜凯> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 4

3.1 概述 4

3.2 在构架方面具有重要意义的设计包 4

4. 进程视图 4

5. 部署视图 4

6. 实现视图 5

7. 技术视图 5

8. 数据视图（可选） 5

9. 核心算法设计（可选） 5

10. 质量属性的设计 5

软件架构文档

# 简介

## 目的

本文档的目的是为“事件搜索引擎”提供软件架构层面的概括与介绍，包括各类视图与相关算法的设计介绍。

## 参考资料

软件需求规约，软件项目计划项目

# 用例视图

用户通过登录系统登录后，搜索想要了解的内容，经系统过滤后以思维导图、时间轴的方式将内容呈现给用户以供浏览。

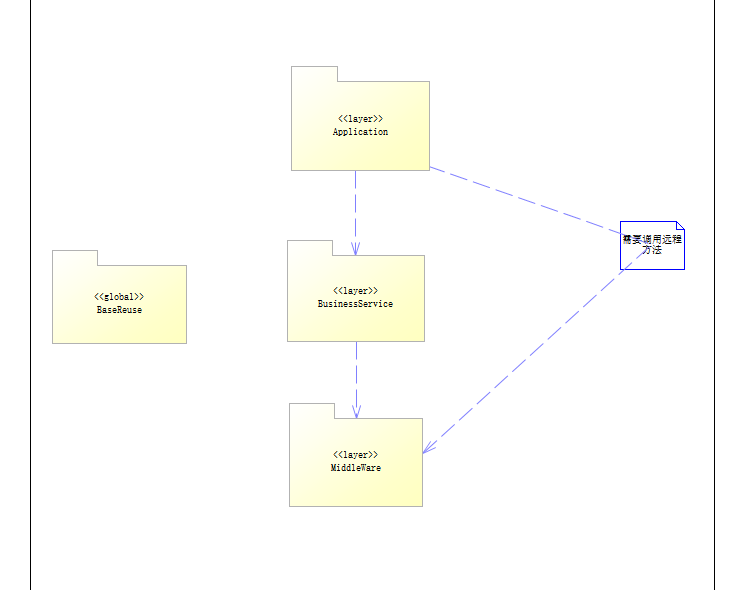
# 逻辑视图

## 概述

逻辑架构：

Application依赖于BusinessService依赖于Middleware，Application同时依赖于Middleware的远程调用

选择分层架构模式

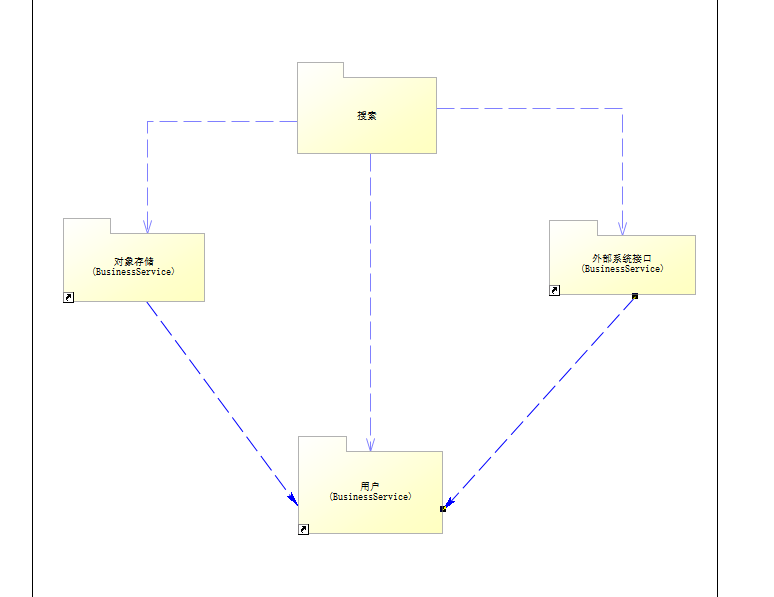


[本节说明所采用的架构风格以及整个软件的模块分解和协同情况，画出逻辑架构图，并进行文字描述。]

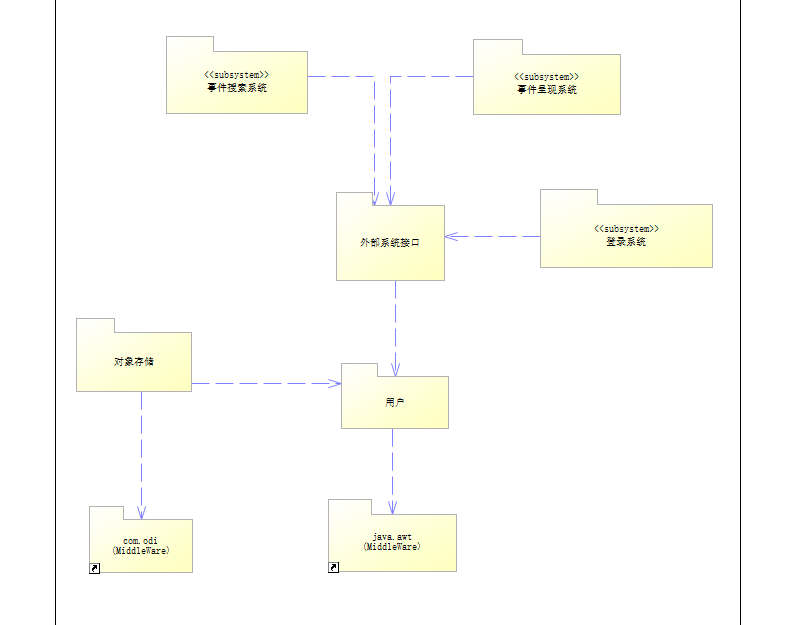
## 在构架方面具有重要意义的设计包

1、Application：

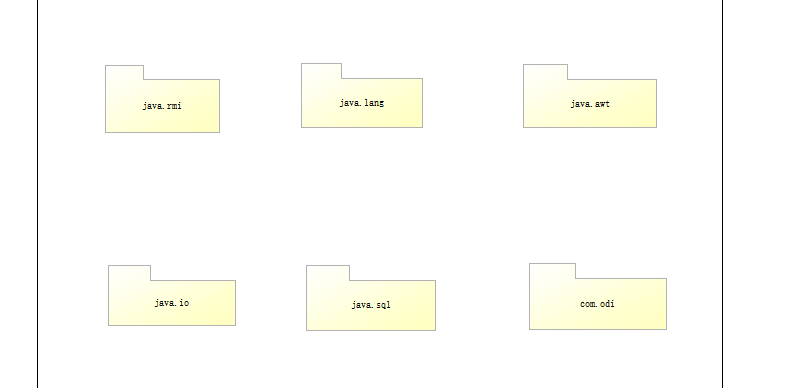
搜索依赖于对象存储和外部系统接口



2、Business Service：



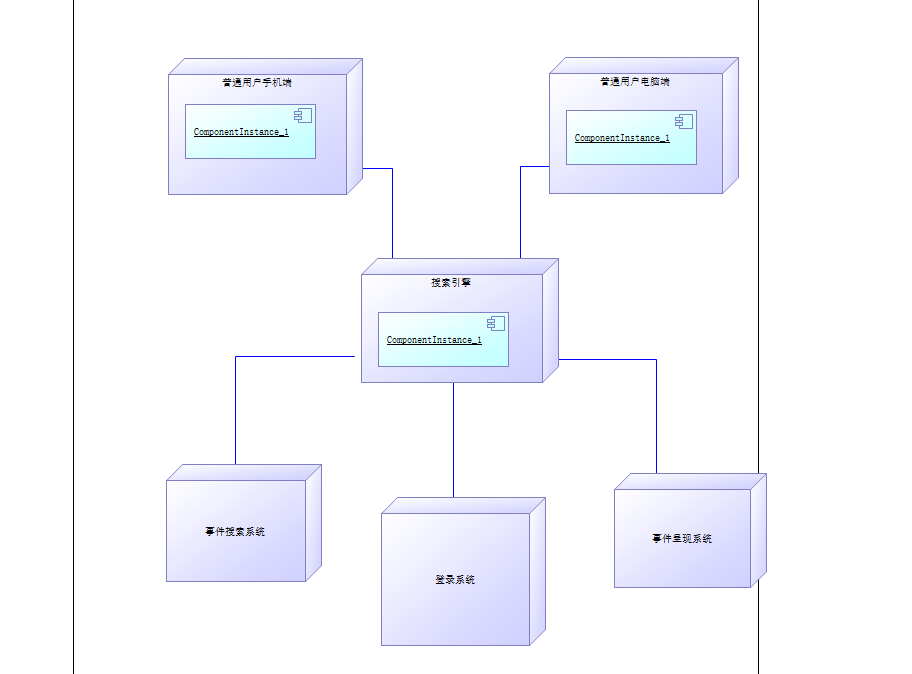
3、Middleware：



[对于每个重要的包，都用一个小节来加以说明，其中应包括该包的名称、简要说明以及显示该包中所有重要的类和包的图。

对于该包中的每个重要类，应包括其名称、简要说明，还可选择包括对其部分主要职责、操作和属性的说明。]

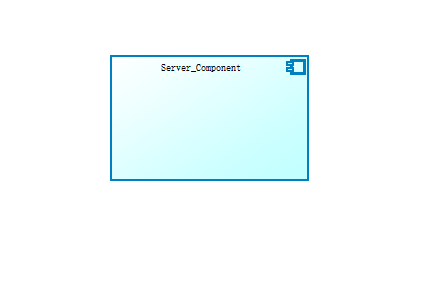
# 部署视图



用户通过搜索引擎搜索想要了解的内容，搜索引擎借由登录系统、事件搜索系统、事件呈现系统为用户提供服务。

[本节说明用来部署和运行该软件的一种或多种物理网络（硬件）配置。对于每种配置，它至少应该指出执行该软件的物理节点（计算机、CPU）及其互连情况（总线连接、LAN 连接、点到点连接等）。另外还要包括**进程视图**中的各进程到物理节点的映射。]

# 实现视图



采用BS架构，仅需server端提供服务。

# 技术视图

编程语言：java,javaScript,python

开发工具：idea

框架：React，Spring Boot

数据库：mysql

中间件：Tomcat、JDBC

# 质量属性的设计

软件架构明确了整个系统的结构组成与详细分工，其用例视图明确了系统面向用户的核心功能，使得研发过程中对功能的开发更具有针对性；逻辑视图的设计将系统分解为从问题域中得到的一系列关键抽象，即对象、类和包，这些抽象为系统提供了功能分析，在本次设计中采用分层架构模式进行设计，将不同层次的功能部分抽象出来并分别加以详细说明，从而使得整个架构更加清晰，更具有可扩展性；部署视图的设计则从物理层面明确了本次开发中系统的详细结构，保证在将软件系统部署到不同环境时不必大量修改源代码，提高了可移植性。