本方案旨在解决单一数据库无法处理的复杂关联推理、时序动态关系以及不确定性自然语言查询（如“2016年某党主席”）的难题。采用 GraphRAG（图谱增强检索） 架构，实现从自然语言到结构化情报的高效转化。

1. 总体架构设计

系统采用 混合存储架构，将数据分为“逻辑关系（骨架）”和“属性详情（血肉）”，通过 LLM（大模型）作为认知中枢进行调度。

后端技术栈 核心框架：Spring Boot (Java) 图数据库：Neo4j（处理递归关联、时序逻辑、关联系统） 文档数据库：MySQL（存储人员/机构的完整画像、长文本、图片） 向量数据库：Milvus（处理模糊语义、简历/观点匹配） AI 编排：Spring AI / LangChain4j（负责意图识别与 Text-to-Cypher）

数据分层设计

1. 逻辑层 (Neo4j)：仅存储节点（人、机构、系统）和边（任职、亲属、管理）。核心在于边必须包含 start_date, end_date, role 属性，以支持动态时空查询。
2. 详情层 (MySQL)：存储你提供的人员表和机构表的全量字段（如简历、政治倾向分析、照片Base64）。
3. 索引层 (Milvus)：存储“职务描述”、“政治言论”的向量，用于模糊匹配。

2. 核心 Pipeline 流程详解

整个处理流程是一个闭环的 Agentic Workflow（智能体工作流）。

阶段一：意图识别与路由 (Intent Routing) 系统接收用户的不确定输入，LLM 分析其意图并提取关键参数。

输入处理： 用户输入可能是：“张善政” (实体查询) 可能是：“2016年的民进党主席” (逻辑描述查询) 可能是：“管理行政院公文系统的那个高管” (关联系统查询)

处理逻辑： Router Agent 判断输入类型： 若是明确姓名 -> 路由至 关键词检索。 若是描述性语句（含时间、职位、机构）-> 路由至 逻辑推理引擎 (Text-to-Cypher)。 若是模糊概念（如“反核人士”）-> 路由至 向量检索引擎。

阶段二：实体消歧与定位 (Entity Resolution) 这是解决“2016年主席是谁”的关键步骤。

路径 A：逻辑推理 (Text-to-Cypher) 针对“2016年民进党主席”这类输入，系统动态生成图查询语句。 逻辑：在图数据库中查找 [人] -> [任职] -> [机构]，且 [任职] 关系的时间跨度覆盖 2016 年。 结果：精准返回实体 UID（如：uid_cai_001, Name: 蔡英文）。

路径 B：向量召回 (Vector Search) 针对模糊描述，在 Milvus 中检索简历或职务描述的向量。 结果：返回最相似的 Top-K 实体 UID。

阶段三：递归图谱遍历 (Recursive Graph Traversal) 获取目标 UID 后，系统执行标准的“关联挖掘”。

递归逻辑： 第一层：查找目标人物的所有 1度关联（现任/曾任公司、配偶、管理的系统）。 第二层（递归）：查找上述机构的 核心成员，或上述关联人物的 其他背景。 时序过滤：如果用户指定了时间（如“查他2010年的情况”），则在遍历时过滤边的 start_date/end_date 属性。

阶段四：数据聚合与动态推理 (Hydration & RAG) 系统拿着一堆 UID 去 MySQL 捞取详细数据。

数据聚合：将 Neo4j 的关系数据与 MySQL 的属性数据合并。 动态推理 (On-the-fly Inference)： 检查“政治倾向”、“政策倾向”字段。 如果为空，调用 LLM 读取该人的“简历”、“重大事件”、“公开发言”字段。 LLM 现场分析：“根据其2023年关于两岸的XX言论，判定为亲美、蓝营。” 将分析结果回填到返回结构中。

3. 完整样例演示

为了展示系统如何处理“不确定输入”并生成标准输出，以下演示一次完整的调用周期。

样例场景 用户想查询 2016年担任民进党主席的人，并获取该人的完整画像及其关联机构信息。

3.1 用户输入 用户 Query： "帮我查一下2016年的民进党主席，以及他当时所在的机构详情。"

3.2 系统内部处理 (Pipeline Trace)

Step 1: 意图分析 (LLM Output) { "intent": "logic_search", "target_type": "Person", "constraints": { "role": "主席", "organization": "民进党", "time_point": "2016" }, "requirement": "recursive_profile" }

Step 2: 逻辑转换 (Text-to-Cypher) 系统生成并在 Neo4j 执行如下逻辑（伪逻辑描述）： MATCH (p:Person)-[r:WORKED_AT]->(o:Organization) WHERE o.name LIKE '民进党' AND r.role LIKE '主席' AND r.start_date <= 2016 AND (r.end_date >= 2016 OR r.end_date IS NULL) RETURN p.uid, p.name

执行结果 (Graph DB Return)： [{ "uid": "person_001", "name": "蔡英文" }]

Step 3: 关联扩展 (Recursive Query) 以 "person_001" 为起点，再次查询 Neo4j： 查找：(蔡英文)-[关联]-(实体) 发现：

1. 机构：民进党 (关系：主席, 2014-2018)
2. 机构：中华民国政府 (关系：总统, 2016-2024)
3. 关联系统：国安局情报系统 (关系：最高权限管理者)
4. 人物：某亲属 (关系：...)

Step 4: 数据填充 (MySQL Query) 查询 MySQL 集合 `person_profiles` 匹配 ID `person_001`。 查询 MySQL 集合 `org_profiles` 匹配 ID `org_dpp_001` 等。 触发 LLM 分析其“政治倾向”字段（基于检索到的过往言论）。

3.3 最终输出结果 系统返回符合你要求的两张表结构的 JSON 数据。

返回数据结构 (JSON)： { "status": "success", "resolved_target": "蔡英文 (自动识别结果)", "data": {


```
"person_table": [
  {
    "信源": "维基百科/政府公报",
    "图像": "base64_encoded_string...",
    "姓名": "蔡英文",
    "性别": "女",
    "出生年月": "1956-08-31",
    "家乡": "屏东",
    "电话": "...",
    "证件号": "...",
    "邮箱": "...",
    "工作部门/机构番号": "中华民国总统府",
```

```
      "所在地": "台北",
      "职务": "总统（2016-2024)",
      "职务信息": "...",
      "社交账号": "Twitter: @iingwen",
      "简历": "（长文本）康奈尔大学法学硕士，伦敦政治经济学院博士...",
      "家庭背景": "...",
      "政治背景": "民进党籍",
      "住址": "...",
      "工作经历": "（列表）曾任民进党主席(2014-2018)...",
      "教育经历": "...",
      "护照信息": "...",
      "出国经历": "...",
      "公务活动": "...",
      "重大事件": "2016年当选...",
      "政治倾向": "绿营；支持现状/独立倾向；亲美",
      "政策倾向": "非核家园...",
      "军事人员军衔": "",
      "行业人员类别": "政治人物",
      "岗位重要程度": "极高",
      "关联系统": "国安局情报系统（权限：Top)"
    }
    // 此处可继续列出递归查到的其他关联人信息
  ],

  "organization_table": [
    {
      "机构": "民主进步党",
      "维基百科": "http://...",
      "官网链接": "http://dpp.org.tw",
      "中文名称": "民主进步党",
      "英文名称": "Democratic Progressive Party",
      "曾用名": "",
      "机构类型": "政党",
      "员工数量": "...",
      "核心业务": "政治活动",
      "简介": "台湾主要政党之一...",
      "核心成员列表": "（文本）赖清德、... ",
      "组织架构": "...",
      "关联行为": "目标人物曾任该机构主席（2014-2018)",
      "主要媒体": "...",
      "地理位置": "台北",
      "行业地位": "执政党/主要反对党"
    }
  ]
```


}}