

地市电力局 AI 产品设计思路交流方案

基于现有 MySQL 8.0 电力业务数据结构的智能化应用与 Agent 设计建议

核心观点

不把 AI 做成“会查数据库的聊天机器人”，而是构建能够理解电网对象、事件、拓扑、运行状态与业务闭环的 Power Operations Intelligence Platform。

交流版本 | 2026 年 9 月

01 执行摘要

本方案基于 Volt Mind Technology 当前获取的某地市电力局 MySQL 8.0 表结构文件进行产品设计分析。所提供结构共定义 65 张表，覆盖设备与资产、GIS/馈线、配电自动化、遥测、遥信、SOE、故障停电、缺陷、巡视、检修计划及部分专题管理数据。

从产品视角看，这套数据已经具备构建“电力运行智能体”的基础：实时数据负责感知，SOE/遥信负责事件还原，设备与拓扑负责定位上下文，故障/缺陷/停电/巡视/计划数据负责解释与闭环。

建议把 Volt Mind 的产品定位为：AI-native Power Operations Platform（电力运行智能平台）。第一阶段优先打通“事件感知 -> 故障研判 -> 影响分析 -> 处置建议 -> 复盘闭环”这一主链，再逐步扩展到设备健康、馈线风险、巡视检修与管理决策。

建议优先验证的 3 个场景

① AI 故障研判 Agent；② SOE/遥信/遥测事件重构；③ 馈线与设备风险智能分析。三个场景共享同一底层语，能形成可持用的品能力。

本材料建议的交流路径

章节	讨论重点
02 数据资产判断	现有数据库能够支撑什么，哪些能力最有价值
03 产品愿景	从数据平台升级为电力运行智能平台
04 核心 Agent	故障、SOE、风险、设备、停电、巡视、检修等
05 业务语义层	
06 技术架构	实时湖仓/OLAP + Event Store + Feature + Agent Tool
07 落地路线	从 1 个 PoC 到多 Agent 平台
08 与局方交流问题	用于现场确认业务痛点和数据边界
09 附录	重点表据域系

02 现有数据资产与 AI 价值判断

2.1 数据域全景

数据域	代表性表/字段	AI 可用价值
设备与资产	accounting、dev_accounting、automation_switch_ledger、arrester_equipment_ledger	设备主数据、生命周期、厂家/型号、运维组织、资产画像
GIS / 馈线 / 拓扑	automated_gis_devices、automated_switches、feeder/line 字段	设备空间归属、上下级关系、故障影响范围、转供分析
实时遥测	dms_data_gzdy、telemetry_data、telemetry_data_latest	序常、、荷/电压/电流特征、设备运行画像
实时遥信	remote_signaling_data、telesignaling_data	保护动作、闭锁、终端状态、自动化动作解释
SOE 事件	soe_data、soe_data_latest	毫秒级事件顺序、动作链、故障时序重构
故障与停电	dispatch_logs、fault_records、outage_log、outage_analysis、outage_plans	故障原因、影响范围、复电效果、事件复盘
缺陷与异常	defects、anomaly_records、device_abnormal_state	缺陷闭环、重复异常、风险聚合
巡视与检修	inspection_plan_completion_status、inspection_record、work_plans	风险驱动巡视、计划协同、任务推荐
专题资产	避雷器、接地电阻、雷击区段、盯跳等	专业领域风险模型与专项 Agent

2.2 现有结构中最值得关注的信号

- 实时侧不仅有数值遥测，还有三相电流/电压、零序量、功率因数及数据质量码，适合构造设备运行特征。
- SOE 记录包含 ocs_time、soe_time、设备 OID/GIS ID、馈线、事件名称与状态，具备构造毫秒级事件序列的基。
- 遥信数据已经覆盖过流、零序、闭锁、终端异常、失压、得电合闸、失电分闸、智能分布式等典型配电自动化状态。
- 故障分类数据已经存在“遥信 ID + 多条遥测 ID + 模型版本 + 一级/二分”的形式，天然适合造督本。
- 离线数据包含故障、停电、缺陷、巡视、工作计划以及部分处置/闭环结果，能够形成“事件 -> 处置 -> 结果”的业务。

重要判断

真正的产品壁垒不是“数据量大”，而是把不同时间粒度、不同系统来源、不同业务对象的数据统一成可供 AI

推理的力。

03 产品愿景：从数据平台走向电力运行智能平台

3.1 产品定位

Volt Mind 不建议以“电力 AI 问答”作为最终定位。更适合的方向是构建一个 AI-native Power Operations Platform，让 AI 能够回答并持续执行四类问题：

问题	对应能力
发生了什么？	事件发现、SOE 重构、异常检测、停电识别
为什么发生？	故障研判、保护动作解释、历史案例检索、设备/缺陷关联
可能产生什么影响？	拓扑影响范围、用户/配变影响、停电时长、风险传播
下一步做什么？	巡视建议、检修建议、复盘任务、处置步骤、计划协同

3.2 产品形态

层级	产品形态	说明
底层	Power Semantic Layer	统一成业务语义
中层	Power Event Intelligence	负责事件关联、特征计算、故障研判和风险模型
上层	Power Agents	按角色提供故障检修、管理等 Agent
交互层	Power Copilot	与现有系统交互

3.3 产品设计原则

1. AI 不直接替代调度与运检人员的专业判断，而是提供可解释、可追溯、可复核的分析结果。
2. 大模型负责理解、编排与解释；规则引擎、时序模型、统计模型和拓扑算法负责专业计算。
3. 所有重要量定据：据源、口、算、相及史案例。
4. 先高价、强景切入，不以“大而全”的知或聊天界面作第一段目。

04 重点 AI Agent 设计

4.1 AI 故障研判 Agent - 第一优先级

目标：用户输入“分析 XX 馈线刚才的故障”，系统自动完成事件聚合、动作时序重构、影响范围分析、历史案例检索和处置建议生成。

输入	核心工具/数据	输出
SOE / 遥信	soe_data、remote_signaling_data、 telesignaling_data	动作时序、保护动作链
遥测	dms_data_gzdy、telemetry_data	故障前后电气特征
设备与馈线	dev_accounting、 automated_gis_devices、 automated_switches	对象上下文与拓扑
历史故障/缺陷	dispatch_logs_backfilling、 fault_records、defects	相似案例与历史问题
停电与计划	outage_log、outage_analysis、 work_plans	影响与后续处置建议

典型输出示例

事件发生时间、保护动作、疑似故障区段、受影响范围、可能原因、自动化动作是否合理、相似历史案例、现场优先检查对象，以及结论证据链。

4.2 SOE 事件重构 Agent

把十“7”戴一事件。例如把 “流 -> 跳闸 -> 下游失压 -> 联络开关动作”理解为一个故障与恢复过程。该能力将成为多个 Agent 共用的底层能力。

4.3 馈线风险 Agent

结合高危馈线、故障次数、近期异常、未消缺缺陷、自动化水平、设备健康、巡视状态等因素，形成动态风险画像。现有结构已存在 high_risk_feeders 表及相关排名/管控字段，可作为现有业务逻辑的入口。

4.4 设备健康 Agent

围绕“设备现在是否值得关注”构建画像：投运年限、异常次数、离线、缺陷、故障、厂家/型号、近期运行趋势等。避雷器领域已具备台账、故障、接地电阻和健康评分等数据基础，可作为专业设备 Agent 的示范场景。

4.5 停电复盘 Agent

以停电事件为中心，自动回答：为什么停电、影响多大、自动化是否有效、复电用了多久、问题是设备还是网架还是处置流程，并自动生成复盘材料。

4.6 巡视与检修 Agent

把风险、故障、缺陷、巡视历史和工作计划联动起来，形成“风险驱动巡视”和“计划协同”的智能建议，而不是只做静态计划查询。

4.7 领导/管理 Copilot

以自然语言支持“最近一个月故障主要集中在哪里”“哪些区县问题在上升”“哪些馈线需要重点关注”等管理问题，并支持逐级钻取到设备和事件证据。

05 核心底座：可被 AI 调用的电力业务语义层

建议把业务语义层作为 Volt Mind 的核心产品能力，而不是把 SQL 生成器作为核心。它是连接“数据”和“Agent”的中间层。

语义对象	典型内容	Agent 用法
Object Model	城市、、站、、分支、、FTU、配变、避雷器、用户	确定“我在谈谁”
Relation Model	于、接、上下、供、供、影、替代	确定“谁和谁有关”
Event Model	故障、跳闸、SOE、遥信变化、遥测异常、停电、复电、遥控	确定“发生了什么”
Metric Model	故障率、重复故障率、复电时长、健康度、风险度、自动化成功率	确定“怎么算”
Rule Model	故障判据、重复停电、风险分级、闭环准	确定“什么时候算”
Case Model	历史故障、原因、处理方式、结果	确定“以前怎么处理”
Lineage Model	源表、字段、口、算、版本	确定“这个结论从哪里来”

5.1 最关键的技术动作：统一实体

从现有结构看，设备、馈线与开关在不同表中存在 dev_oid、device_id、switch_oid、switch_id、switch_gis_id、feeder_id、feeder_gid、line_id、line_gis_id 等多种标识。建议建设 Canonical Entity ID，将这些系统标识映射到统一的电网对象。

实体统一的价值

一旦一个开关/馈线拥有统一身份，遥测、遥信、SOE、资产台账、故障、缺陷、巡视和工作计划才能真正被 AI 放在同一上下文中推理。

5.2 业务语义 API，而非直接让 Agent 查询原始表

- get_device_context(device_id, time_window) - 取行、常、缺陷上下文。
- get_event_timeline(event_id) - 返回事件相关 SOE、遥信、遥测及动作链。
- get_feeder_topology(feeder_id) - 返回馈线拓扑、开关与上下级关系。
- get_feeder_risk(feeder_id, horizon) - 返回风险因子、趋势与证据。
- find_similar_faults(event_features) - 从历史案例中检索相似故障。
- get_outage_impact(event_id) - 返回影响设备、用户、时长及复电情况。
- recommend_inspection_targets(filters) - 根据、缺陷和史巡生成建象。

06 技术架构建议

6.1 总体架构



6.2 为什么不建议 Agent 直接查询遥测大表

- 实时遥测是高频时序数据，原始明细并不是最适合 LLM 直接消费的形态。
- Agent 直接生成 SQL 以保性能、限界、口和口一致。
- 应在语义层预先形成可复用指标和特征，让 Agent 查询“设备当前异常程度”“过去 30 天重复故障次数”等业务对象。
- 原始数据仍保留作为证据源，AI 出中提供可追溯的口和源。

6.3 推荐的数据服务分工

组件	建议职责	说明
MySQL	主数据与业务事务	台账、缺陷、计划、配置等结构化业务数据
OLAP / 时序存储	大规模遥测/遥信分析	适合口、聚合、特征算和史分析
Event Store	事件中心	按设备/馈线/时间窗口组织故障与动作链
Feature Store	模型特征	设备/馈线的滚动统计、异常、趋势与风险因子

组件	建议职责	说明
Search / Vector	案例与文本知识	历史故障描述、专业规则、处置经验等
Agent API	能力封装	向 Agent 提供安全、可解释、可审计的工具调用

架构原则

在电力行业，Agent 的“聪明”不是来自无限访问数据，而是来自经过业务语义约束、专业模型计算和证据链包后的工具用。

07 重点技术专题

7.1 实时遥测智能分析

能力	输入	建议算法/实现
单点异常	电压/电流/功率	robust statistics
三相不平衡	Ia/Ib/Ic、Va/Vb/Vc	不平衡度特征 + 聚类/阈值
时序漂移	7/30/90 天曲线	基线模型 + change point
群体异常	同馈线/同型号设备	peer-group anomaly detection
事件关联	遥测 + 遥信 + SOE	时间窗口 join + event graph
故障先兆	历史异常 + 后续故障	survival / classification / ranking

7.2 故障诊断的建议链路

建议采用“规则 + 模型 + 案例 + LLM”的组合，而不是让 LLM 单独判断故障。

6. 特征层：提取事件前后电压、电流、功率、状态变化。
7. 模型层：进行故障分类、相似度计算、风险排序。
8. 案例层：检索历史相似故障并返回证据。
9. LLM 层：把机器结果组织成电网专业人员可复核的故障研判报告。

7.3 AI 输出的“证据链”标准

字段	示例
结论	疑似故障位于 XX 至 XX 开关区段
时间窗口	2026-09-16 10:21:34.000 - 10:21:40.000
关键证据	SOE 动作 6 条、遥信异常 3 条、遥测突变 2 组
历史案例	相似事件 4 条
计算版本	fault_model_v1.3 / risk_rule_v2
可信度	建议采用置信等级，而非未经验证的“确定性”措辞
人工确认	支持调度员/专业人员采纳、修改、驳回并记录原因

08 建议实施路线

8.1 第一阶段：1 个高价值 PoC

段	目标	主要工作	交付物
0-1 月	数据与事件打通	统一设备/馈线 ID；梳理 SOE/遥信/遥测；定义事件对象	Power Semantic v0.1、事件模型、数据质量报告
1-2 月	故障研判 PoC	事件聚合、规则、历史案例检索、AI 告	AI 故障研判 Agent v0.1
2-3 月	真实业务验证	选取 1 个区县/专业线试用，沉淀人工反馈	效果评估、样例库、推广清单

8.2 第二阶段：从 Agent 到平台

- 扩展馈线风险、设备健康、停电复盘 Agent。
- 建设 Feature Store 与 Event Store，统一时序特征和事件模型。
- 把专业规则、历史案例与人工反馈纳入版本化知识体系。
- 接入巡视与工作计划，形成建议 -> 执行 -> 反馈 -> 学习闭环。

8.3 第三阶段：形成电力运行智能平台

- 形成可复用的 Power Agent Runtime 和业务 Tool Catalog。
- 逐步覆盖度、配自化、管理等景。
- 形成跨系统的指标语义、事件语义和知识语义标准。
- 在确保安全、权限、审计和人工确认的前提下，再考虑半自动或闭环执行能力。

落地建议

不要第一天就建设“全局大模型”。先证明一个真实故障场景能被 AI 稳定复盘，再向风险、设备、巡视和管理扩展。这样最容易在局方内部形成信任。

09 与局方交流时建议重点确认的问题

主题	现场建议确认的问题
真实痛点	当前故障分析、停电复盘、馈线风险、巡视计划中，哪个环节最耗人力？
实时数据	遥测/遥信采样周期、数据到达延迟、历史保留周期、质量码含义是什么？
SOE	SOE 与遥信、遥测、调度日志之间是否存在稳定的事件关联键？
拓扑	是否有完整 CIM/拓扑图模型？开关常开/常闭、上下游关系如何获取？
主数据	设备 OID/GISID/资产编码/设备编号之间是否存在统一主数据映射？
故障闭环	哪些表能够代表“最终真实原因”和“最终处置结果”？
知识资产	是否存在历史故障报告、典型案例、规程、工作票/处置经验等文本知识？
AI 边界	哪些环节可以自动分析，哪些必须由人工确认？是否允许 AI 给出处置建议？
安全与部署	数据是否允许出域？是否需要本地化/私有化模型和部署？

10 附录：重点数据表与产品能力映射

数据表/数据域	在 AI 产品中的角色
dms_data_gzdy	实时电气量；故障前后特征与异常检测
telemetry_data / telemetry_data_latest	FTU 遥测；设备/馈线运行画像
remote_signaling_data / telesignaling_data	保护、闭锁、终端状态及自动化动作
soe_data / soe_data_latest	毫秒级事件顺序与事件重构
automated_gis_devices / automated_switches	终端关联与拓扑上下文
dev_accounting / accounting	设备与馈线资产画像
defects / anomaly_records	缺陷、异常及闭环状态
dispatch_logs / dispatch_logs_backfilling	故障事件及人工回填的真实原因
fault_classification_data	故障分类训练/验证样本结构
outage_log / outage_analysis / outage_plans	停电影响、复电和计划分析
high_risk_feeders	现有风险管控结果，可作为风险 Agent 输入
inspection_plan_completion_status / inspection_record	巡视计划与完成状态
work_plans	工作计划及班组/时间/地点信息
arrester_* / lightning_* / grounding_*	专业设备与雷击风险专题

10.1 数据结构依据说明

本报告对数据库结构的判断严格以所提供的 MySQL 8.0 DDL 为依据。DDL 显示数据库 smart_dms_db，MySQL 版本为 8.0.46；完整结构文件共包含 65 个 CREATE TABLE 定义。

需要特别说明：本报告尚未对生产数据样本进行完整质量评估，也未验证所有表之间的实际主外键一致性、实时延迟、拓扑完整性和业务口径。因此，报告中的“可做”表示“从现有表结构看具备合理基础”，最终产品设计仍应在 PoC 阶段通过真实样本进行验证。

建议下一步

以 1 个真实故障案例为入口，现场共同定义“事件 -> 证据 -> 结论 -> 建议”的标准，然后反向确定语义层、数据服务和 Agent Tool 的最小集合。

资料来源：用户提供的 structure.sql（MySQL 8.0.46，数据库 smart_dms_db），以及本次产品交流中确认的实时遥测规模背景。