# Product Requirements Document (PRD)

## Deliver AI-Driven Energy Data Access Portal

## Purpose

To enable Users and business stakeholders to interact with energy data stored in Azure ADX through a secure, user-friendly interface, and to provide intelligent insights via an AI assistant that leverages both structured and unstructured data.

## Use Cases

### Use Case 1: Energy Data Access Portal

**Goal:** Allow users to preview, confirm, and export energy data without direct access to Azure ADX.

**Features**:

* Sample data preview
* Site/date range selection
* Hybrid RAG pipeline
* Context-aware responses
* Compliance-aware recommendations
* Citations and traceability
* Export to CSV/JSON
* Role-based access control

### Use Case 2: Energy AI Assistant (Enhance existing Energy Assistant)

**Goal:** Provide intelligent answers and recommendations using energy data, VPA documents, and contextual metadata.

**Features**:

* Natural language query interface
* Hybrid RAG pipeline (structured + semi structured + unstructured)
* Context-aware responses
* Compliance-aware recommendations
* Citations and traceability

## Functional Requirements

|  |  |
| --- | --- |
| **Feature** | **Description** |
| ADX Query API | Secure backend service to query Azure ADX |
| Preview Engine | Show sample data before full fetch |
| Export Service | Convert and export data to CSV/JSON |
| Autogen or Lang flow Agents | Modular agents for data retrieval, reasoning, compliance |
| MCP Layer | Context routing, contract enforcement, logging |
| Neo4j & Knowledge graphs | Store metadata, relationships, context graphs |
| Embedding Store (Azure AI search or Neo4j) | Vector search for VPA documents |
| UI Layer | Web portal or Teams bot interface |

## Non-Functional Requirements

* **Security**:
* RBAC & managed identities - Existing NZOS supports
* Scoped queries –
  + MCP can enforce query scoping rules before execution.
  + Queries are dynamically scoped based on user context (e.g., site, date range).
* **Scalability**: Support multiple sites and users
* **Traceability**:
  + traceability Every response from the AI assistant should be traceable to:
    - Source data (ADX, Neo4j, VPA docs)
    - Agent decisions
    - Context used
  + MCP should log context contracts and model usage per interaction.
* **Compliance**:
  + Data governance
    - Define data access policies in MCP and enforce via RBAC and scoped queries
  + Audit trails
    - Who accessed what data
    - What queries were run
    - What recommendations were made

## Success Metrics

- % of IT users successfully exporting data  
- Accuracy of AI Assistant response to data extraction

# Development Plan (Sequenced)

## Phase 1: Foundation Setup (Weeks 1–4)

1. Define MCP Contracts for both use cases  
2. Set up Azure ADX access layer (API or data fetch?)  
3. Create Neo4j schema for energy metadata  
4. Prepare VPA document store (Blob/SharePoint/upload + embedding pipeline)

## Phase 2: IT Data Access Portal (Weeks 4–7)

1. Build UI for query preview and confirmation – **depends on UI**   
2. Implement ADX query templating and execution  
3. Add CSV/JSON export and SFTP integration  
4. Integrate MCP for context scoping and logging  
5. Test with sample users and refine

## Phase 3: Energy AI Assistant

1. Define Autogen/Langlfow agents: Advisor, Retriever, Compliance -   
2. Implement RAG pipeline (Neo4j + embeddings + ADX)  
3. Build chat interface  
4. Integrate MCP for context routing  
5. Add response traceability and citations  
6. Conduct UAT and feedback loop

## Phase 4: Governance & Observability

1. Add logging and monitoring (Azure Monitor, ADX)  
2. Implement RBAC and audit trails  
3. Finalize documentation and training materials