Software Requirements Specification - FLUXEON

An internal document for software design and development

1. Introduction

Purpose

The purpose of this Software Requirements Specification (SRS) is to define, in a precise and unambiguous way, all functional and non-functional requirements for **FLUXEON**, an AI-agent-based system designed to help a Distribution System Operator (DSO) **detect, predict and mitigate demand spikes** at the feeder level.

FLUXEON integrates:

- Time-series (TS) analytics and classification.
- AI agents for orchestration and decision making.
- The **Beckn Protocol** for interoperable discovery and activation of DER-based flexibility.
- Regulatory-grade logging for **P444** settlement using **OBP IDs** and cryptographic verification of DER identities.

This SRS targets product owners, DSOs, DER aggregators, architects, data scientists, developers, QA, and regulators.

Scope

FLUXEON addresses **Problem Statement 1: Utility Interface with Agentic Orchestration for Grid-Scale Demand Flexibility**, offering a **command and control layer** between:

- Feeder-level telemetry and time-series models.
- Distributed Energy Resources (DERs) and their aggregators.
- A Beckn-based flexibility marketplace.
- DSO control room operators and auditors.

Core capabilities:

- Simulate and ingest **feeder load curves**.
- Detect spikes and predict near-term risk windows using TS models with **4×15-minute prediction horizon** and at least **1 hour of historical context**.
- Extract fast features (RMS, kurtosis, crest factor, exogenous variables such as temperature) and classify operating state as:
  - **0 = Normal**,
  - **1 = Alert**,
  - **2 = Critical**.
- Optimize and orchestrate flexibility actions (batteries, EVs, HVAC, etc.).
- Use **Beckn workflows**: DISCOVER → SELECT → INIT → CONFIRM → STATUS → COMPLETE.
- Respect a **sub–5 second SLA** from state change detection (Alert/Critical) to Beckn dispatch decision.
- Maintain cryptographically verifiable identities for DERs and settlement-grade audit logs with **OBP IDs**.

Definitions, Acronyms & Abbreviations

Acronym	Definition
AI Agent	Autonomous software component that perceives, decides and acts on behalf of the DSO under given policies.
Beckn Protocol	Open protocol enabling interoperable discovery, selection, ordering and fulfillment across a decentralized network. Used to orchestrate DER flexibility.
DER (Distributed Energy Resource)	Behind-the-meter resource such as battery storage, EV charging, HVAC loads, solar-plus-storage, etc.
DER Owner	Individual or entity that owns or directly controls a DER asset.
Aggregator	Party that aggregates multiple DERs and exposes flexibility services via Beckn.
DSO (Distribution System Operator)	Utility responsible for operating and maintaining the distribution grid and feeders.
Feeder	Distribution circuit or line segment serving a set of customers; FLUEXEON models and controls risk at feeder level.
Feeder Load	Time-series of electrical load (kW, kVA, current) measured at feeder level
SLA (Service Level Agreement)	Contractual requirement specifying maximum allowed processing times; here, detection-to-dispatch under 5 seconds.
TS (Time-Series)	Sequence of time-ordered measurements, e.g., feeder load.
RMS (Root Mean Square)	Statistical measure of signal magnitude used as a feature in spike detection.
Kurtosis	Fourth standardized moment of a distribution; measures tail heaviness, used as a feature.

Crest Factor	Ratio of peak value to RMS value; used to characterize spikes and distortions.
Exogenous Variables	External variables influencing load (temperature, calendar effects, holiday flags, etc.).
OBP ID (Offer/Bid Pair ID)	Identifier used for settlement under P444, linking offered and delivered flexibility.
P415	UK code modification enabling Virtual Lead Party (VLP) / aggregator participation in balancing and settlement.
P444	Settlement mechanism specifying how flexibility events are recorded and settled, relying on OBP IDs and verified identities.
Universal DER Identity	Unique, cryptographically verifiable identifier for each DER or DER portfolio.
DEG	Distributed Energy/Grid platform (assumed external system providing generation/flexibility simulation or control endpoints).

References

1. **Problem Statement 1** – "Utility Interface with Agentic Orchestration for Grid-Scale Demand Flexibility", provided as: Problem statement.pdf
2. IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
3. Beckn Protocol Core Specification (latest).
4. Industry documents relating to P415 and P444.
5. Ofgem publications on flexibility, settlement and DER programs.

Overview

The remainder of this SRS is structured as follows:

- Section 2 – Overall system description and context.
- Section 3 – Detailed system features with requirement IDs (R1, R2, ...).
- Section 4 – External interface requirements (UI, APIs, Beckn schemas).
- Section 5 – System architecture and agent interaction.
- Section 6 – Non-functional requirements.
- Section 7 – Data requirements.
- Section 8 – Risks and constraints.
- Section 9 – Future enhancements.
- Section 10 – Appendices.

2. Overall Description

Product Perspective

FLUXEON is a **middleware / command centre platform** positioned between:

- **Grid side**: Feeder-level telemetry and existing DSO systems (SCADA, historians, DEG).
- **Market side**: DER owners/aggregators exposed via Beckn networks.
- **Human users**: DSO operators and auditors/regulators.

FLUXEON:

- Ingests real or simulated feeder load curves.
- Runs TS analysis to predict risk windows.
- Classifies feeder state (0/1/2).
- When Alert/Critical, orchestrates flexibility via Beckn.
- Logs everything in a regulatory-grade audit trail (OBP IDs, cryptographic proofs, payloads).

The system is modular, implemented as **microservices / agents** (simulator, TS engine, classifier, optimizer, Beckn client, dashboard, audit logger) communicating through secure APIs and an event bus.

General Product Functions

- At a high level, FLUXEON shall:
- Simulate and/or ingest feeder load curves and exogenous data.
- Detect spikes and classify real-time feeder states.
- Generate short-term predictions using TS models.
- Extract feature vectors in near real-time.
- Classify states and trigger actions when thresholds are crossed.
- Query Beckn network for DER flexibility (DISCOVER, SELECT).
- Place and manage flexibility orders (INIT, CONFIRM, STATUS, COMPLETE).
- Log events, OBP IDs, signatures, payloads and decision rationales.
- Provide operator dashboard for monitoring and control.
- Expose APIs for DEG integration and external consumption.

User Classes & Characteristics

User Class	Description	Characteristics
DSO Operator	Control-room operator monitoring feeders and triggering interventions.	Experienced with SCADA/EMS; expects concise, actionable UI and alerts.
DER Owner	Individual or organization owning a DER; usually acts via Aggregator.	Sees individual asset view and activation history through external systems; indirectly affected by FLUXEON.
Aggregator	Entity aggregating DERs and exposing flexibility services via Beckn.	Operates Beckn-compatible endpoints; needs clear order lifecycle and settlement data.

Auditor / Regulator	Internal or external auditor validating compliance (P415, P444, Ofgem rules).	Needs access to immutable logs, OBP IDs, cryptographic proofs and decision context, with a non-technical view.

Operating Environment

- **Backend**: Linux-based servers or containers (Kubernetes), running microservices.
- **Frontend**: Browser-based dashboard (modern browsers, 1080p+).
- **Network**:
  - Secure connections to DSO data sources (SCADA, historian, DEG).
  - Secure connections to Beckn networks and aggregator endpoints.
- **Data Stores**:
  - Time-series database for load curves and predictions.
  - Relational/NoSQL store for configuration, DER catalog snapshots, logs, OBP IDs.
  - Object storage for raw payload archival.

Constraints

- **C1 (SLA)**: For any feeder whose classification changes from 0 → 1 or 1 → 2, the time from event detection to Beckn INIT/CONFIRM dispatch must be **< 5 seconds** under nominal conditions.
- **C2 (Beckn)**: All DER discovery and activation must use **Beckn Protocol workflows** (DISCOVER → SELECT → INIT → CONFIRM → STATUS → COMPLETE).
- **C3 (Regulatory / P444)**: Audit logs must be compatible with **P444 settlement**, relying on OBP IDs and cryptographically verifiable DER identities.
- **C4 (Identity)**: DERs must have **universal unique identities** bound to cryptographic keys or proofs.
- **C5 (Data retention)**: Settlement and audit logs must be retained for a regulatory-defined period (e.g., ≥ 7 years).

Assumptions

- A **simulator module** or external DEG environment is available to generate or replay realistic load curves for testing and training.
- Feeder data may arrive in **batch (historical)** and **real-time (streaming)** forms, with synchronized timestamps.
- DER aggregators and Beckn participants have implemented minimal required Beckn APIs and semantics for flexibility.
- DSO provides necessary feeder configuration (limits, topology, criticality), and local regulations do not prevent agentic orchestration as described.
- Clocks are synchronized via NTP or equivalent, to guarantee temporal consistency of logs.

3. System Features

3.1 Load Curve Simulator

3.1.1 Description and Priority

Simulates feeder load curves with configurable profiles to support training, testing and sandbox operation.

- Priority: **High**

3.1.2 Functional Requirements

- **R1** – The system shall provide a simulator capable of generating synthetic feeder load curves with configurable patterns (base load, peak windows, noise).
- **R2** – The simulator shall support configuration of time resolution (e.g., 1-min, 5-min, 15-min samples).
- **R3** – The simulator shall allow injection of artificial spike events at specified times and magnitudes.
- **R4** – The simulator shall be able to run in **offline mode** (historical generation for training) and **online mode**(streaming to the TS engine).
- **R5** – The simulator shall emit load curves in the same format as real telemetry for seamless substitution.

3.2 Spike Detection Engine

3.2.1 Description and Priority

Real-time module that inspects incoming load curves and identifies potential spikes.

- Priority: **Critical**

3.2.2 Functional Requirements

- **R6** – The system shall continuously ingest feeder load measurements from real-time and/or simulated sources.
- **R7** – The system shall detect spikes using configurable thresholds and statistical rules (e.g., deviation from moving average).
- **R8** – The system shall emit a **Spike Event** when load exceeds pre-defined limits or exhibits anomalous characteristics (based on RMS/crest factor).
- **R9** – The Spike Event shall include feeder ID, timestamp, measured value, reference thresholds and basic feature values.
- **R10** – The Spike Detection Engine shall expose its events to the TS Prediction Engine and Classification Model via an internal API or event bus.

3.3 TS Prediction Engine (4×15 min Horizon)

3.3.1 Description and Priority

Time-series prediction engine that produces near-term forecasts over a window of **4×15 minutes** based on at least 1 hour of historical data.

- Priority: **Critical**

3.3.2 Functional Requirements

- **R11** – The engine shall maintain for each feeder a rolling historical window of at least **1 hour** of measurements.
- **R12** – The engine shall compute forecasts for the next **4 time steps of 15 minutes each** (total 60 minutes) at a configurable frequency (e.g., every 5 minutes).
- **R13** – The prediction model shall accept exogenous variables (e.g., temperature, calendar flags, event flags) as optional inputs.
- **R14** – The engine shall output predicted load curves and associated confidence intervals.
- **R15** – The prediction results shall be exposed to the Classification Model, Flexibility Optimization Engine, and dashboard.

3.4 Feature Extraction Module (RMS, Kurtosis, Crest Factor, etc.)

3.4.1 Description and Priority

Transforms raw TS windows into feature vectors for classification and optimization.

- Priority: **High**

3.4.2 Functional Requirements

- **R16** – The module shall compute for each sliding window (e.g., 15 minutes, configurable):
  - RMS,
  - kurtosis,
  - crest factor,
  - mean, variance,
  - min, max.
- **R17** – The module shall enrich feature vectors with exogenous variables such as:
  - ambient temperature,
  - weekday/weekend/holiday flags,
  - time-of-day encoding,
  - special event indicators.
- **R18** – The module shall normalize or standardize features according to configured rules to support robust model performance.
- **R19** – The module shall generate feature vectors for both historical and predicted segments to support classification of current and upcoming states.
- **R20** – The feature vectors shall be passed to the Classification Model and logged for audit if requested.

3.5 Classification Model (0 = Normal, 1 = Alert, 2 = Critical)

3.5.1 Description and Priority

AI model that classifies the operating state of each feeder per time step.

- Priority: **Critical**

3.5.2 Functional Requirements

- **R21** – The classification model shall take as input feature vectors derived from current and predicted data.
- **R22** – The model shall output a discrete state label per feeder and time slice: **0 (Normal), 1 (Alert), 2 (Critical)**.
- **R23** – The model shall output a confidence score or probability distribution over the three classes.
- **R24** – The model shall be configurable for different feeders (model instance or parameters) to accommodate local characteristics.
- **R25** – When a feeder transitions to state 1 or 2, the model shall publish a **State Change Event** including previous state, new state and confidence.
- **R26** – The Classification Model shall expose its outcomes to the Flexibility Optimization Engine, Dashboard and Audit Logger.

3.6 Flexibility Optimization Engine

3.6.1 Description and Priority

Determines the optimal flexibility actions (batteries, EV, HVAC, etc.) required to mitigate Alert/Critical situations.

- Priority: **Critical**

3.6.2 Functional Requirements

- **R27** – Upon receiving a State Change Event (state 1 or 2), the engine shall compute the **required flexibility volume** per feeder and time slot over the prediction horizon.

- **R28** – The engine shall accept DER catalog data (capacity, activation time, duration, price, constraints) as an input.
- **R29** – The engine shall generate a **Flexibility Plan** specifying which DERs/aggregators to activate, volumes, start/end times, and priorities.
- **R30** – The engine shall support multiple optimization objectives (minimize curtailment, minimize cost, maximize reliability) with configurable weights.
- **R31** – The engine shall consider DER and feeder-level constraints, including response time, minimum activation duration and participation limits.
- **R32** – The engine shall produce plans within a bounded computation time so that the end-to-end SLA of **< 5 seconds** is maintained when combined with Beckn dispatch.
- **R33** – The Flexibility Plan shall be exposed to the Beckn DER Discovery and Beckn Order Lifecycle modules.

3.7 Beckn DER Discovery

3.7.1 Description and Priority

Uses Beckn DISCOVER and SELECT to find DER flexibility suitable for the Flexibility Plan.

- Priority: **High**

3.7.2 Functional Requirements

- **R34** – The system shall issue **Beckn DISCOVER** requests when flexibility is required for a feeder, specifying location, required volumes, time window and any program constraints.
- **R35** – The system shall process **ON_SEARCH** responses and build a normalized **DER Flexibility Catalog**.
- **R36** – The system shall issue **Beckn SELECT** requests to shortlist suitable offers in accordance with the Flexibility Plan.
- **R37** – The system shall support filtering and ranking of offers based on criteria such as capacity, cost, historical reliability and DER type (battery, EV, HVAC).
- **R38** – The resulting selection shall be linked to universal DER identities and cryptographic verification where provided.
- **R39** – All Beckn DISCOVER/ON_SEARCH/SELECT payloads shall be logged for audit and replay.

3.8 Beckn Order Lifecycle (INIT → CONFIRM → STATUS → COMPLETE)

3.8.1 Description and Priority

Executes the Flexibility Plan via Beckn orders and manages lifecycle states until completion.

- Priority: **Critical**

3.8.2 Functional Requirements

- **R40** – The system shall generate Beckn **INIT** requests corresponding to selected offers, referencing DER identity, feeder, time window and quantities.
- **R41** – The system shall process **ON_INIT** responses and, where acceptable, issue **CONFIRM** requests to finalize orders.
- **R42** – The system shall ensure that the time span from State Change Event (to state 1 or 2) to sending CONFIRM is less than **5 seconds** in 95% of nominal cases.
- **R43** – The system shall poll or subscribe to **STATUS**/ON_STATUS updates and maintain an up-to-date view of order state (pending, active, completed, failed).
- **R44** – The system shall send **COMPLETE** messages and process ON_COMPLETE to capture delivered flexibility volumes and performance metrics.
- **R45** – The system shall handle partial failures (some DERs decline or fail) by either reselecting via Beckn or escalating (as per policies).
- **R46** – Each Beckn order shall be associated with:

    - a universal DER identity,
    - cryptographic verification info (where available),
    - an **OBP ID** generated or supplied for P444 settlement.
- **R47** – The system shall expose order lifecycle state to the Operator Dashboard and Audit Logging module in near real time.

3.9 Audit Logging (OBP ID / P444)

3.9.1 Description and Priority

Captures all data necessary for P444-compliant settlement and regulatory audit.

- Priority: **Critical**

3.9.2 Functional Requirements

- **R48** – The system shall create a unique **OBP ID** for each pair of offer and bid resulting in a flexibility activation event, or ingest it if provided externally.
- **R49** – For every OBP ID, the system shall log:
    - associated feeders,
    - associated DER identities,
    - requested vs delivered volumes,
    - timestamps (initiation, confirmation, activation, completion),
    - prices/payment parameters (if available).
- **R50** – The system shall store raw Beckn payloads (requests and responses) linked to OBP IDs for full reconstructability.
- **R51** – The system shall store cryptographic proofs or signatures used to verify universal DER identities.
- **R52** – Audit logs shall be **append-only**, tamper-evident and retained for a configurable period aligned with regulatory requirements.
- **R53** – The system shall expose query APIs and dashboard views allowing auditors to retrieve events filtered by OBP ID, feeder, date ranges, DER identity or aggregator.
- **R54** – All classification and optimization decisions contributing to an activation shall be linked to the corresponding OBP ID for full traceability.

3.10 Operator Dashboard

3.10.1 Description and Priority

Single pane of glass for DSO operators to monitor feeder states, predictions, DER activations and audit information.

- Priority: **High**

3.10.2 Functional Requirements

- **R55** – The dashboard shall display for each feeder:
    - current state (0/1/2),
    - current load and predicted load over the 4×15-min horizon,
    - color-coded risk indicator (e.g., green/amber/red).
- **R56** – The dashboard shall present **real-time charts** of load, prediction, activated flexibility and residual risk.
- **R57** – The dashboard shall show active and upcoming flexibility orders, including DER type (battery, EV, HVAC), aggregator, and order state.
- **R58** – The dashboard shall expose an **audit view** where operators and auditors can drill down by OBP ID and see associated logs, payload excerpts and identity proofs.
- **R59** – The dashboard shall allow authorized operators to:
    - override or disable automatic orchestration for specific feeders,

    - modify optimization weights (cost vs reliability),
    - trigger test simulations.
- **R60** – All operator actions (overrides, parameter changes) shall be logged and linked to user identity and timestamp.
- **R61** – The dashboard shall refresh key metrics at least every 2 seconds during active events.

4. External Interface Requirements

UI/UX Interfaces

- Web-based dashboard accessible via secure HTTPS.
- Main panels:
    - Global overview map/table of feeders with states and alerts.
    - Time-series charts for selected feeder.
    - Order/DER activation panel.
    - Audit/OBP panel.
- Basic UI rules:
    - Color codes: green (0), amber (1), red (2).
    - Snack-bar or modal alerts for new Alert/Critical states.
    - Responsive layout for control room screens (1080p, 4K).

4.2 APIs

FLUXEON shall expose and consume:

- **Internal REST/gRPC** APIs:
    - /ingest/load
    - /prediction/horizon
    - /features/vector
    - /classification/state
    - /optimization/plan
    - /beckn/orders
    - /audit/obp/{id}
- **External APIs**:
    - To DEG: e.g., /deg/simulate, /deg/control.
    - To DSO systems: e.g., /telemetry, /config/feeders.
- All APIs shall enforce authentication (tokens, mTLS) and role-based access control.

4.3 Inputs

- **Load Curves**:
    - Feeder ID, timestamp, load (kW/kVA), optional voltage/current.
- **Exogenous Data**:
    - Temperature, weather, calendar flags, event flags.
- **DER Catalog**:
    - From Beckn ON_SEARCH/SELECT: DER identity, type, capacity, response time, location, price, constraints.
- **Configuration**:
    - Feeder limits, classification thresholds, optimization weights, SLA parameters.

4.4 Outputs

- **Flexibility Orders**:
    - Beckn INIT/CONFIRM/STATUS/COMPLETE payloads.
- **Logs**:
    - Event logs, audit logs, error logs, OBP ID records.

- **Metrics**:
  - SLA statistics, number of events, volumes activated, success/failure rates.
- **Dashboard Views**:
  - HTML/JSON APIs for UI front-end consumption.

4.5 Beckn JSON/YAML Schemas

- The system shall validate input/output payloads against **Beckn JSON schemas** for:
  - DISCOVER, ON_SEARCH, SELECT, ON_SELECT,
  - INIT, ON_INIT, CONFIRM, ON_CONFIRM,
  - STATUS, ON_STATUS, COMPLETE, ON_COMPLETE.
- Internal YAML schema definitions may be used for configuration and code generation of Beckn client stubs.

---

5. System Architecture

5.1 Data Flow Diagram (Textual)

High-level flow:

1. Load curves (real or simulated) → **Ingestion Service**.
2. Ingestion → **Spike Detection Engine** → Spike Events.
3. Spike Events + historical data → **TS Prediction Engine** → Forecasts.
4. Forecasts + historical data → **Feature Extraction Module** → Feature Vectors.
5. Feature Vectors → **Classification Model** → State labels (0/1/2).
6. State Change Events (1/2) → **Flexibility Optimization Engine** → Flexibility Plan.
7. Flexibility Plan → **Beckn DER Discovery** → DER Catalog & Selection.
8. Selection → **Beckn Order Lifecycle** → Orders/Status/Completion.
9. All events + payloads → **Audit Logging (OBP IDs)**.
10. All states + metrics + logs → **Operator Dashboard** and external reporting.

5.2 Agent Architecture

Key agents/services:

- **Simulator Agent** – Generates synthetic load curves or interfaces with DEG.
- **TS Agent** – Performs spike detection, prediction and feature extraction.
- **Classifier Agent** – Applies classification model and publishes state changes.
- **Orchestrator Agent (agent_core)** – Coordinates optimization, Beckn calls and policies.
- **Beckn Client Agent** – Implements Beckn workflows and handles connectivity to aggregators.
- **Audit Agent** – Manages OBP IDs, logging, cryptographic verification and queries.
- **Dashboard Agent** – Serves UI and aggregates data for visualization.
- **Integration Agent (DEG)** – Interfaces with DEG for control or co-simulation if required.

Agents communicate via:

- Shared message bus (e.g., Kafka, NATS) for events.
- Synchronous APIs for certain queries (e.g., config).

5.3 Component Overview

Pipeline overview:
simulator → ingest → ts_features → model → agent_core → beckn_client → audit + dashboard

- **simulator**: Synthetic load generator and test scenario runner.
- **ingest**: Normalizes telemetry from SCADA/DEG and simulator.
- **ts_features**: Combines Spike Detection, TS Prediction, Feature Extraction.

- **model**: Performs classification and publishes states.
- **agent_core**: High-level orchestrator; decides when to trigger Beckn flows and optimization.
- **beckn_client**: Encapsulates all Beckn interactions.
- **audit**: Handles OBP IDs, logs, verifications.
- **dashboard**: Exposes monitoring and control UI.

5.4 Integration with DEG

- FLUXEON shall integrate with DEG as:
  - **Source**: DEG provides detailed simulation or actual generation/flexibility states.
  - **Sink**: FLUXEON may send control suggestions or activation signals that DEG can map to plant-side actions.
- Integration interface:
  - APIs for requesting **scenario simulations** (pre-event training).
  - APIs/events for receiving **real-time DEG status**.
  - Optional: DEG as an alternative or additional data source for exogenous variables.

---

6. Non-functional Requirements

6.1 Performance

- **NFR1** – Detection-to-CONFIRM for a new Alert/Critical state must be **< 5 seconds** for at least 95% of events under nominal load.
- **NFR2** – The system shall support processing up to **N feeders** (configurable, e.g., 1,000) without violating SLA when average event concurrency is below a defined threshold.
- **NFR3** – Dashboard updates (for active feeders) shall reflect new data within **2 seconds**.

6.2 Scalability

- **NFR4** – Microservices shall be horizontally scalable across nodes.
- **NFR5** – Storage solutions for time series and logs shall support growth in data volume and retention period without requiring major redesign.
- **NFR6** – Beckn client components shall support scaling out to handle multiple concurrent activation events.

6.3 Auditability

- **NFR7** – All events affecting real-world or simulated grid actions must be logged with sufficient context (inputs, parameters, outputs, identities, timestamps).
- **NFR8** – OBP ID logs must be queryable with typical queries executing under **5 seconds** for standard time ranges (e.g., one month).

6.4 Reliability

- **NFR9** – Target availability of the core orchestration path (ingest → classification → Beckn) is **≥ 99.9%** monthly.
- **NFR10** – The system shall support automatic failover for critical services.
- **NFR11** – In case of partial failures (e.g., Beckn network outage), system shall degrade gracefully and provide clear alerts.

6.5 Regulatory Compliance

- **NFR12** – Data models and logs must be compatible with **P415** and **P444** requirements regarding event and settlement records.
- **NFR13** – Logs and data exports must be structured to be usable as evidence for regulatory investigations without additional transformations.

- **NFR14** – Configurations affecting settlement (e.g., pricing rules) must be versioned and auditable.

6.6 Security

- **NFR15** – All external communications must use TLS 1.2+.
- **NFR16** – Authentication and authorization must be enforced using role-based access control and secure token issuance.
- **NFR17** – Sensitive data must be encrypted at rest using industry-standard algorithms.
- **NFR18** – Security-relevant events (login, config changes, API keys) must be logged.

6.7 Usability

- **NFR19** – The dashboard shall be usable by trained operators with minimal additional training (1–2 days).
- **NFR20** – Critical information (red/amber state) must be visible at a glance without deep navigation.
- **NFR21** – The system shall support light and dark theme variants for 24/7 operation environments (optional but recommended).

---

7. Data Requirements

Data Models

Feeder Data:

Field	Type	Description
feeder_id	String	Unique feeder identifier.
timestamp	DateTime	Measurement time.
load_kw	Float	Feeder load.
voltage_v	Float	Voltage.
current_a	Float	Current.

Exogenous Data:

Field	Type
temperature_c	Float
holiday_flag	Boolean
event_flag	Boolean
dow	Integer (0–6)

DER Catalog Structure

Field	Type	Description
der_id	String	Universal DER identity.
aggregator_id	String	Aggregator/vendor identifier.
type	Enum	battery/EV/HVAC/other.
location	JSON	Simplified location/feeder mapping.
capacity_kw	Float	Available flexibility.
response_time_s	Float	Activation latency.
duration_min	Integer	Max activation time.
price_per_kwh	Float	Price or incentive.
constraints	JSON	Operational constraints.
crypto_proof	JSON	Public key or proof metadata.

Feature Vector Format

Field	Type
feeder_id	String
window_start	DateTime
window_end	DateTime
rms	Float

kurtosis	Float
crest_factor	Float
mean	Float
variance	Float
min	Float
max	Float
temperature_c	Float
holiday_flag	Boolean
dow	Integer
other_exogenous	JSON

OBP ID Log Format

Field	Type	Description
obp_id	String	OBP ID.
feeder_id	String	Affected feeder.
der_ids	String	DER identities involved.
becky_order_ids	String	Related Beckn orders.
requested_kwh	Float	Requested volume.
delivered_kwh	Float	Delivered volume.

price	Float	Price/incentive.
timestamps	JSON	start/confirm/activate/complete.
classification_snapshot	JSON	Classification model outputs at decision time.
optimization_snapshot	JSON	Optimization params and outcomes.
crypto_evidence	JSON	Signatures/keys/proofs.

Time-Series Structure

- Stored in a TS database with indexes on:
  - feeder_id,
  - timestamp.
- Supports:
  - range queries (time window),
  - downsampling (e.g., hourly aggregates),
  - retention policy management.

---

8. Risks & Constraints

8.1 Technical Risks

Inaccurate TS model predictions leading to false positives/negatives in classification.

Performance bottlenecks in optimization or Beckn communication preventing SLA compliance.

Data quality issues (missing or noisy telemetry).

8.2 Regulatory & Market Risks

Evolution of P415/P444 requirements may require schema and process adjustments.

Insufficient explainability might reduce regulator trust in AI-driven decisions.

8.3 Operational Risks

Operator overreliance on automation without sufficient oversight.

Misconfiguration of thresholds or optimization weights causing under/over-activation of flexibility.

DER participation variability (opt-outs, degraded availability) affecting reliability.

8.4 Environmental Constraints

Dependence on connectivity to Beckn networks and DER aggregators.

Dependency on DEG or equivalent platforms for accurate simulation and/or control.

---

9. Future Enhancements

Multi-agent negotiation:

- Agents representing grid constraints, DER portfolios, and markets negotiating in real time.

Carbon-aware orchestration:

- Incorporating carbon intensity signals to prioritize low-carbon resources.

Advanced explainability:

- Counterfactual simulations: “What if different DERs had been chosen?”
- Natural language explanation for regulators.

Bidirectional DER markets:

- Support for bid/ask mechanisms and auction-based allocation.

ML Ops & continuous learning:

- Online learning for TS and classification models using feedback from real events.

---

10. Appendices

Glossary

Term	Definition
• **Alert State (1)**	• Elevated risk; proactive flexibility may be triggered based on policy.
• **Critical State (2)**	• High risk; aggressive or emergency flexibility activation is expected.
• **DEG**	• External platform providing distributed energy generation or simulation capabilities; treated as a data/control peer.

Example Beckn Workflow

1. FLUXEON detects feeder F enters state 2 (Critical).
2. Flexibility Optimization Engine computes a plan for next 60 minutes.
3. Beckn Client sends DISCOVER for F, receives ON_SEARCH.
4. Beckn Client sends SELECT for selected offers, receives ON_SELECT.
5. Beckn Client sends INIT, receives ON_INIT.
6. Beckn Client sends CONFIRM, receives ON_CONFIRM.
7. STATUS/ON_STATUS track progress; COMPLETE/ON_COMPLETE finalize.
8. Audit Agent records OBP IDs and all associated data.

Mock Data Payloads

- **Feature Vector (JSON)**:


```
{
  "feeder_id": "FDR_001",
  "window_start": "2025-11-22T10:00:00Z",
  "window_end": "2025-11-22T10:15:00Z",
  "rms": 138.4,
  "kurtosis": 4.1,
  "crest_factor": 1.8,
  "temperature_c": 27.5,
  "holiday_flag": false,
  "dow": 6
}
```