# 📋 Meeting Report – OSB Hub System Engineering

**Date:** April 28, 2025
**Time:** 14:02 CET
**Duration:** ~1 hour
**Participants:** Pascal Bouquet, Marius Conjeaud, Gerard Castillo, Kim Zachariassen, Imad Bousaid, Katja Glass, Josef Hartmann, Muharrem Daja, and others

---

## 1. Introduction and Objectives (Pascal Bouquet)

- This is the fourth meeting in the OSB System Engineering trail, under the OSB Hub community.
- OSB Hub aims to foster community contributions, feedback, and guideline development.
- The session's agenda:
    1. Data loading steps for OSB (by Marius Conjeaud)
    2. Migration strategy
    3. Deployment approach using Helm charts (by Boehringer Ingelheim)

---

## 2. Initial Data Loading in OSB (Marius Conjeaud)

### 🔷 Key Concepts

- OSB deployments require pre-loaded data: standards, sponsor libraries, optional legacy protocols.
- Two major data lifecycles:
    - **Initial Loading:** Controlled, file-based or scripted.
    - **Live Input:** User-generated via front-end/API (non-scriptable).

### 🔷 Initial Loading Phases

1. **MDR DB (FHM-drtb)** – Initializes the Neo4j graph DB:
    - Creates schema (indexes, constraints)
    - Can start from backup
    - Adds utility nodes
    - Adds NeoDash dashboards for data exploration
2. **MDR Standards Import:**
    - Loads CDISC standards: control terminology, data models, IGs
    - Requires your CDISC API Key

    - Two-step process: staging → final import
    - Incremental: does not re-import already existing packages
    - Note: CDSIC standard import is being re-engineered
3. **Study Builder Import:**
    - Loads sponsor library data (activities, criteria, units, models)
    - Requires the API to be live
    - Reads CSVs → makes API calls to populate the DB
    - Incremental logic avoids duplicate entries
    - Allows for loading mock studies for performance testing

---

# 3. Data Migration Strategy

### 🔷 Motivation

- Live data cannot be re-imported from scratch
- New OSB releases (e.g., changes in control terminology structure) necessitate schema changes

### 🔷 Migration Mechanism

- Implemented via migration scripts (in Cypher or Python)
- Located in the `migrations` folder
- Each script has a README explaining:
    - Scope of the migration
    - Steps performed (e.g., renaming properties, schema tweaks)
    - Data correction examples (e.g., converting usernames from initials to emails)

### 🔷 Good Practices

- Migrations are independant(safe to run multiple times)
- No formal versioning yet for the data model — versions tracked manually
- Users should test migrations on a data subset prior to production
- Migration must match frontend/API version to avoid feature breaks

### 🔷 Questions

- **Q:** Can migrations be skipped? **A:** Yes, but only if you don't upgrade front-end/API. Otherwise, features may break.

---

# 4. OSB Deployment via Helm Charts (Gerard Castillo, Boehringer Ingelheim)

### 🔷 Overview

- Presentation from Boehringer Ingelheim to the community: Creation of Helm Charts to deploy OSB
- In Boehringher, OSB is being deployed in OpenShift using Helm (industry standard for Kubernetes packaging)
- Chart includes:
    - Neo4j (as a Helm dependency with minor overrides)
    - API, Consumer API, Frontend, Docs, NeoDash
    - Persistent volume configurations

### 🔷 Customizations for OpenShift

- Adapted Neo4j chart to support OpenShift's security context (random UID at runtime)
- Disabled hardcoded security contexts
- Modified base images and Dockerfiles for OSB components (Python base → Bookworm preferred)
- Configured resource constraints and single-threaded NGINX for performance and stability

### 🔷 Job Handling (Kubernetes Jobs)

- Repackaged initial loading and maintenance tasks (e.g., standards import) as Kubernetes jobs
- Supports both manual and scheduled execution
- Examples: automatic CT updates, regular backups

### 🔷 Configuration

- Central `values.yaml` file drives configuration
- Supports environment-specific overrides
- Handles external dependencies (e.g., Azure auth) via environment variables/secrets

---

## 5. Q&A and Next Steps

### 🔷 Questions Answered

- **Q:** Are the Helm charts customizable? **A:** Fully configurable using `values.yaml`.
- **Q:** How are external dependencies (e.g., Azure, secrets) handled? **A:** Through `.env`, config maps, and secrets injection mechanisms.

### 🔷 Next Meeting

- **Date:** June 2, 2025, at 4:00 PM CET
- **Proposed topics:** Testing strategies, USDM ingestion in OSB

---