OBINexus Phenomenologic Data Specification (PDS)

1. Purpose

The **Phenomenologic Data Specification (PDS)** defines how objective data structures are extended with **phenomenological overlays** that capture subjective, contextual, and experiential variance in system behavior, interpretation, or use.

2. Terminology

- **Objective Layer (OL):** The canonical, structured data model — factual, invariant, machine-verifiable.
- **Phenomenologic Layer (PL):** The subjective/contextual metadata overlay — variant, interpretive, human/machine relational.
- **Consicione:** The binding rule where OL and PL are joined into a complete record.
- **Phenomenohog Block:** The formal structure for representing phenomenological subjectivity in a machine-readable way.

3. Structure

3.1 Base Schema

```
record:
  id: <UUID>
  objective:
    service: <string>
    operation: <string>
    division: <string>
    department: <string>
    country: <ISO code>
  phenomenohog:
    session: <session-id>
    variance:
      - scope: person|instance|context
```

type: <variance-type>

description: <free-text>

timestamp: <UTC>

3.2 Example

```
record:
  id: "c5d2-8af3-44e9-bc29"
  objective:
    service: education
    operation: render
    division: publishing
    department: curriculum
    country: uk
  phenomenohog:
    session: "sess-42"
    variance:
      - scope: person
        type: preference
        description: "User prefers tabular format over narrative prose."
        timestamp: 2025-08-29T18:30:00Z
      - scope: context
        type: semantic-shift
        description: "Term 'curriculum' mapped to 'syllabus' in US-English context."
        timestamp: 2025-08-29T18:30:02Z
```

4. Rules (Consicione)

1. Every **objective record** may have zero or more **phenomenohog blocks**.
2. A **phenomenohog block** must declare its scope (person, instance, or context).
3. All subjective declarations must remain **append-only** — no overwriting.

4. Systems must treat the **objective layer** as the canonical baseline; the **phenomenohog layer** augments but does not erase.

5. Rendering & Documentation

- In Markdown/Gitbook, **objective fields** are rendered as fixed tables.
- **Phenomenohog entries** are rendered as collapsible annotations, so readers can expand subjective layers.
- This ensures a clear separation of **fact vs. perception** in system documentation.