GDPR SAR Compliance & Nexus Search Formal Specification

Domain: sar.project/operation.obinexus.department.division.county.org
Primary Domain: obinexus.org
Compliance: ICO Law 30-Day SAR Response Deadline
Version: 1.0 - Milestone Seed Disclosure Framework

Executive Summary

This document establishes the formal specification for GDPR Subject Access Request (SAR) compliance integrated with the OBINexus Nexus Search architecture, ensuring phonological data structure integrity while meeting ICO regulatory requirements.

---

1. Project Classification Schema

Tier Classifications

- **T1**: Open Access - Research Only (Public Score-Based Access)
- **T2**: Business Nexus Access Partnership (Commercial Integration)
- **T3A**: Research Lead Project (Non-Descriptive, Academic Focus)
- **T3B**: Research & Development (Transitional to Operational)

Access Control Matrix

```
T1 → Open Access Created Score
T2 → Business Nexus Access Partnership
T3A → Research Lead Project (Eze Knowledge King - Igbo Research Focus)
T3B → Research & Development → Operational Deployment
```

---

2. GDPR SAR Technical Architecture

2.1 Schema Implementation

```
sar.project/operation.obinexus.department.division.county.org
├── project: SAR request classification
├── operation: Data retrieval workflow
├── obinexus: Core search engine
├── department: Organizational unit
```

```
├── division: Sub-unit classification
└── county: Geographical jurisdiction
```

2.2 30-Day Compliance Pipeline

```pseudo
FUNCTION process_sar_request(request_id: String, deadline: 30_days) {
    // Milestone-based data disclosure
    milestone_1: data_discovery(7_days)
    milestone_2: data_compilation(14_days)
    milestone_3: legal_review(21_days)
    milestone_4: response_delivery(30_days)

    return compliance_status
}
```

---

3. Nexus Search Core Architecture

3.1 Nexus Search Memory Structure

```pseudo
STRUCT NexusSearchMemory {
    phonological_index: TrieNode<char>
    identity_resolver: HashMap<NI_Number, PersonRecord>
    aura_segments: Array<256_byte_blocks>
    temporal_cache: LRU<SearchQuery, ResultSet>
}

CLASS TrieNode {
    character: char
    data_tremor_id: UUID
    children: HashMap<char, TrieNode>
    is_terminal: boolean
    person_refs: Array<PersonID>
}
```

3.2 Nexus Search Token Type System

```pseudo
```

```
ENUM NexusTokenType {
   NAME_TOKEN(phonological_signature: String)
   DATE_TOKEN(immutable_timestamp: ISO8601)
   NI_TOKEN(national_insurance: String)
   AURA_TOKEN(psychological_marker: 256_bit)
   GDPR_TOKEN(compliance_flag: Boolean)
}

STRUCT NexusTokenValue {
   token_type: NexusTokenType
   raw_value: Any
   encrypted_value: HexString
   access_tier: ProjectTier
   retention_policy: RetentionRule
}
```

3.3 Identity Resolution Algorithm

```pseudo
FUNCTION resolve_identity(query: SearchQuery) -> PersonRecord {
   // Prevent NHS-type matching failures
   phonological_match = trie.search_phonological(query.name)
   temporal_match = verify_dob(query.dob, phonological_match.records)
   ni_verification = validate_ni_number(query.ni, temporal_match)

   // Ensure isomorphic data structure
   IF ni_verification.is_unique() {
      return build_complete_record(ni_verification)
   } ELSE {
      trigger_manual_resolution(query)
   }
}
```

---

4. Phonological Data Structure Specification

4.1 Character Segmentation

```pseudo
```

```
FUNCTION segment_name(name: String) -> Array<PhonologicalUnit> {
  // Example: "NNAMDI" → [N,n,a,m,d,i]
  segments = []
  FOR each character IN name {
    unit = PhonologicalUnit {
      character: character,
      unique_tremor: generate_tremor_id(character, position),
      aura_signature: calculate_aura(character, context)
    }
    segments.append(unit)
  }
  return segments
}
```

4.2 Data Integrity Preservation

```pseudo
FUNCTION ensure_data_continuity(person_record: PersonRecord) {
  // Even after GDPR deletion, maintain identity links
  immutable_identifiers = [
    person_record.ni_number,  // Cannot change without HQ authority
    person_record.dob_hash,   // Cryptographic birth date
    person_record.phonological_signature
  ]

  archive_essential_identifiers(immutable_identifiers)
  return gdpr_compliant_deletion_proof()
}
```

---

5. Implementation Roadmap

Phase 1: Core NSC Development (Rust)

- Implement `nsc` CLI tool
- Build phonological trie structures
- Develop identity resolution engine

Phase 2: GDPR Integration

- SAR request processing pipeline
- 30-day compliance automation
- ICO reporting mechanisms

Phase 3: Operational Deployment

- Multi-tier access control (T1-T3B)
- Cross-system adapter framework
- Performance optimization

---

6. Compliance Validation

6.1 GDPR Article 15 Requirements

- ✅ Identity verification through phonological matching
- ✅ Data portability via structured export
- ✅ 30-day response timeline automation
- ✅ Retention policy enforcement

6.2 ICO Guidelines Adherence

- ✅ Accurate data retrieval (no NHS-type errors)
- ✅ Complete disclosure within legal timeframes
- ✅ Secure data transmission protocols
- ✅ Audit trail maintenance

---

7. Technical Validation Framework

```pseudo
FUNCTION validate_sar_response(response: SARResponse) -> ComplianceReport {
    identity_accuracy = verify_no_cross_contamination(response.person_data)
    completeness_check = ensure_all_data_categories_covered(response)
    timeline_compliance = verify_30_day_deadline_met(response.timestamp)

    return ComplianceReport {
        ico_compliant: identity_accuracy && completeness_check && timeline_compliance,
        obinexus_score: calculate_system_reliability_score(response),
        next_review_date: response.timestamp + 12_months
    }
}
```

Document Authority: OBINexus.org
Classification: T3A Research Lead Project

Last Updated: 2025-08-30

Next Review: Milestone-dependent