APPENDIX A
QUESTIONARY OF VALIDATION FOR IOTO++ ONTOLOGY

Analogous to the ontology modules to be validated, five subsets of questions are presented to validate the ontology's coverage of the domain. Main drivers to formulate these questions were : aim to evaluate the expressiveness, reasoning support, and flexibility of the ontology. Be focused on practical applicability, not just formal correctness of the ontology. That is shown for example in module 1 and 5, with application-focused metrics like adaptability, accountability, and transparency.

1. Interoperability

(1a) What common data formats are used to ensure interoperability between heterogeneous IoT systems?
(1b) How does the ontology facilitate integration between legacy and modern IoT devices?
(1c) How does the ontology dynamically adapt to newly introduced IoT standards without requiring schema modifications?

2. Privacy Policy

(2a) How is user consent for data collection and sharing represented in the ontology?
(2b) Can the ontology distinguish between public and private data in an IoT network?
(2c) How does the ontology define access control policies for different user roles?
(2d) How does the ontology infer context-dependent access control rules based on real-time device interactions?

3. Security

(3a) How does the ontology model encryption methods for data transmission and storage?
(3b) Can the ontology represent different levels of security clearance for IoT users and devices?
(3c) How does the ontology facilitate anomaly detection in IoT systems?
(3d) How does the ontology manage threat response in IoT systems?

4. Environment Awareness

(4a) How does the ontology define and categorize environmental factors (e.g., temperature, humidity, pollution) in IoT applications?
(4b) How can the ontology support real-time monitoring?
(4c) What relationships exist between IoT devices and sustainability metrics in the ontology?
(4d) How can the ontology model adaptive decision-making for energy efficiency?

5. Ethics

(5a) Can the ontology represent ethical considerations for AI-driven IoT applications?
(5b) How does the ontology support accountability in case of IoT system failures?
(5c) How does the ontology define acceptable and unacceptable uses of IoT data?
(5d) How does the ontology ensure fairness and transparency in IoT decision-making?

APPENDIX B
SPARQL QUERIES

This appendix presents the SPARQL queries designed to answer the questionnaire presented in Appendix A. Below are listed the PREFIX used in the SPARQL queries. Here we have the ontology being validated ioto and its predecessors SOSA, SEAS, Colpri and Ds4iot. Also, we have some general ontology management prefixes such as rdfs and skos.

```
PREFIX ioto:
    <https://github/ioto/EnhacedOntology4IoT/ioto>
PREFIX ssn: <http://www.w3.org/ns/ssn/>
PREFIX sosa: <http://www.w3.org/ns/sosa/>
PREFIX colpri:
   <https://github/ioto/EnhacedOntology4IoT/colpri>
PREFIX ds4iot:
   <https://github/ioto/EnhacedOntology4IoT/ds4iot>
PREFIX seas: <https://w3id.org/seas/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

We present below the questionnaire that corresponds to each of the modules that make up this IoTO++ Ontology.

A. 1. Interoperability

1a) What common data formats are used to ensure interoperability between heterogeneous IoT systems?

```
SELECT DISTINCT ?device ?sensor ?dataFormat
WHERE {
   ?device a ioto:IoTDevice .
   ?device ioto:hasSensor ?sensor .
   ?sensor a sosa:Sensor .
   ?sensor ioto:usesDataFormat ?dataFormat .
}
```

Explanation of the Query: Retrieves IoT devices (*ioto:IoTDevice*) and their associated sensors (sosa:Sensor). Finds the data format (*ioto:usesDataFormat*) used by each sensor for interoperability. Ensures distinct results, listing unique devices, sensors, and their respective data formats.

(1b) How does the ontology facilitate integration between legacy and modern IoT devices?

```
SELECT ?device ?sensor ?dataFormat ?deviceType
   WHERE {
      ?device a ioto:IoTDevice ;
            ioto:hasSensor ?sensor .

      ?sensor a sosa:Sensor ;
            ioto:usesDataFormat ?dataFormat .
      OPTIONAL { ?device a ioto:ModernDevice .
          BIND("ModernDevice" AS ?deviceType) }
      OPTIONAL { ?device a ioto:LegacyDevice .
          BIND("LegacyDevice" AS ?deviceType) }
   }
```

This query explicitly classifies devices into legacy vs modern, making the ontology's role in integration clearer.

(1c) How does the ontology dynamically adapt to newly introduced IoT standards without requiring schema modifications?

```
SELECT DISTINCT ?device ?standard ?protocol \\
WHERE {
   ?device a ioto:IoTDevice .
   OPTIONAL { ?device ioto:hasStandard ?standard . }
   OPTIONAL { ?device ioto:usesProtocol ?protocol .
       }
}
```

This query retrieves newly introduced IoT standards and protocols used by IoT devices dynamically, without modifying the schema.

B. 2. Privacy Policy

(2a) How is user consent for data collection and sharing represented in the ontology?

```
SELECT DISTINCT ?user ?consentType ?consentLabel
   WHERE {
      ?user a ioto:DataController ;
         ioto:makesConsent ?consent .

      ?consent a ?consentType ;
            rdfs:label ?consentLabel .
      FILTER (?consentType IN (colpri:GivenConsent,
          colpri:UngivenConsent))
   }
```

This query retrieves all Privacy Community Members and the consents they made. Also gets consent type (GivenConsent or UngivenConsent). This allows querying which users gave or did not give consent, making it useful for managing privacy policies in IoT environments.

(2b) Can the ontology distinguish between public and private data in an IoT network?

```
SELECT DISTINCT ?dataInstance ?dataType
    (COALESCE(?encryptionStatus, "Not Encrypted")
    AS ?finalStatus)
   WHERE {
      ?dataInstance a ?subType .
      ?subType rdfs:subClassOf* ?dataType .
      OPTIONAL {
         ?dataInstance ioto:mustBeTreatedAs
             ds4iot:EncryptedData .
         BIND("Encrypted" AS ?encryptionStatus)
      }
      FILTER (?dataType IN
          (colpri:SensitivePersonalData,
          colpri:NonSensitivePersonalData))
   }
```

This SPARQL query demonstrates that the ontology can distinguish between public (non-sensitive) and private (sensitive) data by categorizing data under *colpri:SensitivePersonalData* and *colpri:NonSensitivePersonalData*. By retrieving subclasses of these categories and their associated security levels (*ioto:mustBeTreatedAs*), the ontology explicitly defines how different types of data must be handled (e.g., encrypted or not). This structured classification enables IoT systems to enforce appropriate security measures and differentiate between public and private data, ensuring compliance with privacy regulations and interoperability.

(2c) How does the ontology define access control policies for different user roles?

```
SELECT ?dataDomain ?accessControl
WHERE {
  ?dataDomain ds4iot:hasAccessControl
      ?accessControl .
}
```

This query answers the question by identifying which data domains (ioto:DataDomain) are governed by specific access control policies (ds4iot:AccessControl). Since the ontology structures access control policies as a hierarchy of different methods (IBAC, RBAC, OrBAC, RAC), the query ensures that all types of access control policies assigned to different user roles are retrieved.

(2d) How does the ontology infer context-dependent access control rules based on real-time device interactions?

```
SELECT DISTINCT ?device ?dataDomain
     ?accessControlModel
WHERE {
    ?device a ioto:IoTDevice .
    ?device ioto:generatesData ?dataDomain .
    ?dataDomain ds4iot:hasAccessControl
         ?accessControlModel .
}
```

This query identifies IoT devices and their associated data domains and retrieves the access control model that applies to them. It shows how access control rules are applied dynamically based on the type of data being generated. Since different devices interact with various data domains, the system can infer and enforce appropriate access control policies in real-time depending on the situation.

C. 3. Security

(3a) How does the ontology model encryption methods for data transmission and storage?

```
SELECT DISTINCT ?dataItem ?encryptionMethod
     ?algorithm ?securityPolicy ?cryptoManager
WHERE {
    ?dataItem ioto:mustBeTreatedAs
         ds4iot:EncryptedData .
    OPTIONAL { ?dataItem ioto:hasEncryptionMethod
         ?encryptionMethod . }
    OPTIONAL { ?encryptionMethod
         ioto:usesEncryptionAlgorithm ?algorithm . }
    OPTIONAL { ?dataItem ioto:followsSecurityPolicy
         ?securityPolicy . }
    OPTIONAL { ?dataItem ds4iot:hasCryptoManager
         ?cryptoManager . }
}
```

It retrieves all ds4iot:EncryptedData with their corresponding relations to encryption context like: (i) Which encryption method is used (ds4iot:hasEncryptionMethod), (ii) What specific algorithm is applied (ds4iot:usesEncryptionAlgorithm). (iii) What security policies govern encrypted data (ds4iot:followsSecurityPolicy). If available (using OPTIONAL to avoid missing results if not explicitly linked).

(3b) Can the ontology represent different levels of security clearance for IoT users and devices?

```
SELECT DISTINCT ?entity ?accessControl
     ?securityPolicy
   WHERE {
      {
         ?entity a ioto:IoTDevice .
      }
      UNION
      {
         ?entity ioto:followsSecurityPolicy
             ?securityPolicy .
      }
      UNION
      {
         ?entity ioto:triggersAccessControl
             ?accessControl .
      }
      OPTIONAL { ?entity ioto:triggersAccessControl
          ?accessControl . }
```

```
      OPTIONAL { ?entity ioto:followsSecurityPolicy
          ?securityPolicy . }
   }
```

This SPARQL query retrieves entities (security community members, IoT devices, and encrypted data) along with their associated security policies. By doing this, it illustrates how different security levels and policies apply to users and devices in the ontology. This enables distinguishing between different levels of security clearance and access control mechanisms based on the roles of entities within the IoT system.

(3c) How does the ontology facilitate anomaly detection in IoT systems?

```
SELECT ?anomaly ?abnormalBehavior ?device ?monitor
WHERE {
?anomaly a ioto:Anomaly ;
      ioto:representsAbnormalBehavior
          ?abnormalBehavior ;
      ioto:detectedBy ?monitor .
?device ioto:hasBehavior ?abnormalBehavior .
}
```

This query retrieves anomalies, considering the abnormal behavior, associated with each anomaly, and the IoT device exhibiting the abnormal behavior. Additionally, identifies the security monitor that detected the anomaly.

(3d) How does the ontology manage threat response in IoT systems?

```
SELECT ?anomaly ?threatResponse ?incidentManager
WHERE {
   ?anomaly a ioto:Anomaly ;
         ioto:requiresResponse ?threatResponse .
   ?threatResponse a ioto:ThreatResponse ;
               rdfs:label ?responseDescription .
   OPTIONAL {
      ?incidentManager a
          ioto:IncidentResponseManager ;
                  ioto:handlesIncident
                      ?threatResponse .
   }
}
```

For understanding how anomalies are managed, from detection to response execution. we can use this query, that retrieves anomalies detected in the IoT system, linking them to the required threat responses, and the security community members responsible for handling these incidents.

D. Environment Awareness

(4a) How does the ontology define and categorize environmental factors (e.g., temperature, humidity, pollution) in IoT applications?

```
SELECT DISTINCT ?sensor ?observedFactor ?energyForm
WHERE {
   ?sensor a sosa:Sensor ;
         sosa:observes ?observedFactor .
   OPTIONAL {
      ?observedFactor ioto:associatedEnergyForm
          ?energyForm .
   }
}
```

The query retrieves all sensors (*sosa:Sensor*) and the environmental factors (*sosa:ObservableProperty*) they observe, categorizing them under different energy forms (*seas:EnergyForm*) when applicable. This allows us to understand which IoT sensors are responsible for monitoring environmental conditions such as temperature, humidity, and pollution.

(4b) How can the ontology support real-time monitoring?

```
SELECT ?sensor ?observedProperty ?currentValue
    ?resultTime
WHERE {
   ?obs a sosa:Observation ;
       sosa:madeBySensor ?sensor ;
       sosa:observedProperty ?observedProperty ;
       sosa:hasSimpleResult ?currentValue ;
       sosa:resultTime ?resultTime .
}
ORDER BY DESC(?resultTime)
```

This query retrieves real-time energy observations by identifying which sensors (*sosa:Sensor*) are measuring energy-related factors (*sosa:observedProperty*) and their current values (*sosa:hasSimpleResult*). Additionally, *ORDER BY DESC(?resultTime)* fetches the latest readings first (important for real-time)

(4c) What relationships exist between Observed Properties and sustainability metrics in the ontology?

```
SELECT ?evaluation ?property ?metric ?value
WHERE {
   ?evaluation a seas:Evaluation ;
         seas:evaluationOf ?property ;
         seas:evaluatedValue ?value ;
         ioto:relatesToMetric ?metric .
}
```

This query retrieves Evaluations along with their observed sustainability metrics, observed property and recorded values.

(4d) How can the ontology model adaptive decision-making for energy efficiency?

```
SELECT DISTINCT ?controlAction ?energyForm
WHERE {
   ?controlAction a ioto:ControlAction ;
              ioto:adjusts ?energyForm .
   ?energyForm a seas:EnergyForm .
}
```

If an adaptive response is defined, this query retrieves the control action (*ioto:ControlAction*) linked to the energy factor. The presence of *ioto:adjusts* captures dynamic adaptation based on observations.

E. 5. Ethics

(5a) Can the ontology represent ethical considerations for AI-driven IoT applications?

```
SELECT ?system ?ethicCategory ?ethicMode
WHERE {
  ?system a ssn:System ;
        ioto:hasEthicMode ?ethicMode .
}
```

This query retrieves all IoT systems (*ssn:System*) along with their assigned ethical modes (*ioto:hasEthicMode*). It helps analyze how AI-driven IoT systems comply with ethical principles.

(5b) How does the ontology support accountability in case of IoT system failures?

```
SELECT ?dataController ?responsibilityStep
    ?responsibilityThing
WHERE {
  ?dataController a ioto:DataController ;
              ioto:takesResposabilityForStep
                  ?responsibilityStep ;
              ioto:takesResposabilityForThing
                  ?responsibilityThing .
}
```

This query identifies all DataController instances and retrieves the specific *ioto:stepToTakeResponsabilityFor* (such as data collection or transfer) and *ioto:thingToTakeResponsabilityFor* (such as consent or privacy policy evaluation) they are responsible for. This allows for accountability tracking and compliance verification in case of system failures.

(5c) How does the ontology define acceptable and unacceptable uses of IoT data?

```
SELECT ?policy ?acceptableUse ?unacceptableUse
WHERE {
   ?policy a colpri:PrivacyPolicy .
   OPTIONAL { ?policy ioto:definesAcceptableUse
       ?acceptableUse }
   OPTIONAL { ?policy ioto:definesUnacceptableUse
       ?unacceptableUse }
}
```

This query lists what uses are explicitly allowed and which are prohibited. The results provide a structured view of how the ontology governs IoT data usage.

(5d) How does the ontology ensure fairness and transparency in IoT decision-making?

```
SELECT ?system ?ethicCategory ?ethicMode
    ?responsibleEntity ?decision
   WHERE {
      ?system a ssn:System ;
            ioto:hasEthicMode ?ethicMode .

      ?responsibleEntity a ioto:DataController ;
                  ioto:hasEthicCategory
                      ?ethicCategory ;
                  ioto:takesResponsabilityForThing
                      ?decision .
      ?decision rdf:type ioto:Decision .
   }
}
```

This SPARQL query identify systems that follow a stringent ethical mode and data controllers that are responsible for fairness-related decisions. By linking data controllers to specific decisions ($ioto : takesResponsibilityForThing\ ?decision$), the query reveals which ethical decisions are actively managed. This ensures fairness and transparency by showing that responsible entities exist for critical IoT decisions.