

Sofia 

GESTIONE DELLE ONTOLOGIE IN SOFIA2

DICEMBRE 2014

Versione 2



1 INDICE

1	INDICE	2
2	INTRODUZIONE	3
3	ENTITA' DI SOFIA2	4
3.1	ASSETS.....	4
3.2	ONTOLOGIE	5
4	NOMENCLATURA.....	6
5	TIPI E FORMATI	7
6	TEMPLATES PREDEFINITI	10
6.1	FEEDS (MISURAZIONI).....	10
6.2	COMMANDS (COMANDI).....	13
6.3	ALLARMI	14
6.4	EVENTI.....	15
6.5	AUDIT.....	16



2 INTRODUZIONE

La piattaforma Sofia2 offre tutto i meccanismi per interconnettere qualsiasi tipo di rete di sensori, offrendo vari meccanismi per il suo funzionamento.

In questo documento sono contenute delle **raccomandazioni** sulle regole e i processi per l'uso delle applicazioni e dei verticali, al fine di definire una comune base di lavoro fra di essi.

L'obiettivo di queste raccomandazioni è di fornire agli sviluppatori e ai sistemi che useranno le informazioni le definizioni delle strutture dati e dei tipi usati nella Piattaforma, **in modo che essi possano usarli uniformemente ed indipendentemente dalla loro implementazione e dagli sviluppi di integrazione.**

Si raccomanda caldamente la lettura preliminare della documentazione di riferimento, per comprendere i concetti che verranno usati.

La Piattaforma **ottiene, registra e gestisce tutte le informazioni dei sensori** (misure, comandi, allarmi, eventi, etc.) usando delle **ontologie JSON** per la rappresentazione dei dati.

Poiché i meccanismi forniti da Sofia2 per la gestione delle ontologie sono molto flessibili ed adattabili, sono state stabilite delle regole di governance, che vengono qui descritte. Il fine di questi vincoli è **garantire che tutta l'informazione contenuta nella piattaforma abbia una struttura uniforme fra le applicazioni e le soluzioni verticali**, e che si possa fare un **uso orizzontale di tutta l'informazione generata.**



3 ENTITA' DI SOFIA2

La Piattaforma di Sofia2 gestisce due tipi di entità:

- **Asset.** Un asset è qualsiasi elemento (fisico o virtuale) in grado di generare o consumare informazioni sensoriali e gestirle mediante la piattaforma SOFIA2.
- **Ontologie.** Gli asset generano informazioni che vengono modellate mediante ontologie (JSON) nella Piattaforma.

3.1 Asset

Si stabilisce una classificazione degli asset a seconda della loro natura e di come essi si connettono alla Piattaforma:

- **Asset dell'Inventario** della Piattaforma. Questi asset sono gestiti direttamente dalla piattaforma. E' necessario conoscerne a priori tutte le informazioni per potersi connettere ad essi. **Tutti gli asset dell'inventario devono essere scaricati nell'Inventario degli Asset della Piattaforma.** Si dividono in:
 - **Managed.** Sono Asset connessi alla Piattaforma, ossia gestiti dai KP. La loro caratteristica è che i KP, per gestirli, necessitano di conoscere come si accede fisicamente ad essi.
 - **Unmanaged.** Sono Asset conosciuti dall'inventario della Piattaforma, generano e consumano le sue informazioni. Non è però responsabilità del modulo di interconnessione la loro gestione finale, cioè sono gestiti da altri sistemi informativi integrati nella Piattaforma.
- **Asset Virtuali.** Questi Asset non sono conosciuti, generano informazione che arriva alla Piattaforma mediante integrazioni, generalmente con altri servizi d'informazione nel Cloud, per esempio con le reti sociali.

3.2 Ontologie

Gli asset connessi generano e consumano informazioni sensoriali in forma di ontologie JSON.

Al momento di creare un'ontologia si possono usare i seguenti template:

- **Feed** (misurazioni). Si tratta dell'informazione di misurazioni emessa dagli asset connessi. Si può trattare di misurazioni istantanee, aggregazioni per periodo di tempo, calcoli, etc. Un feed può raccogliere più misurazioni simultaneamente. **Tutti i feed devono essere georeferenziati** (punti, linee, aree) perché possono provenire da dispositivi mobili.
- **Command** (comandi o ordini). Sono **ordini inviati agli asset** con capacità di azione. Sono ordini asincroni (non c'è risposta immediata) per i quali esistono due tipi di comando:
 - **Comandi request**. Le istruzioni inviate ad un Asset.
 - **Comandi response**. Le risposte o ACKs inviate dagli Asset ai comandi corrispondenti. Queste risposte indicano solo l'avvenuta ricezione dell'istruzione di comando.
- **Alert** (allarmi). Gli allarmi sono messaggi di notificazione di situazioni (allarmi, incidenti, messaggi, notificazioni, etc.) ed hanno la particolarità che **il loro stato ed il livello di criticità cambiano con il passare del tempo**, dal messaggio iniziale che genera l'allarme fino alla chiusura dell'allarme. Gli allarmi non sono conosciuti fino a quando non si verificano.
- **Schedule** (eventi). Gli eventi riflettono situazioni conosciute nel tempo e nello spazio, ossia ci si aspetta che esse si verifichino in un luogo ed ad un momento concreto, in genere conosciuti fin dall'inizio.
- **Audit** (log). I messaggi di audit o log sono messaggi interni alla piattaforma e sono usati per esplicitare o nascondere situazioni gestite dagli asset. Per esempio l'accensione di un fanale, il malfunzionamento di un sensore, etc.
- **KPI** (indicatore). Gli indicatori sono un tipo speciale di misurazione. La loro caratteristica è che essi non vengono generati da un solo asset. Sono il risultati di calcoli su dati, serie storiche ed anche famiglie di dati spaziali che si memorizzano come misurazioni nella Piattaforma.

Regole speciali da considerare:

- Ogni volta che viene integrato un nuovo insieme di asset (tutti dello stesso tipo) alla Piattaforma, devono essere scaricate al meno le ontologie **feed** ed **audit** e, se è necessario, il **command** corrispondente.
- Una volta che viene ricevuto un comando e che viene correttamente eseguito, l'asset (il KP che lo gestisce) deve generare una nuova misurazione perché sulla Piattaforma sia visibile il suo nuovo stato.

4 NOMENCLATURA

Le regole di base della nomenclatura:

- Si usano nomi in inglese e si segue lo standard Java (aka "camelCase").
- Per la definizione di **template**:
 - Nomi corti e autoesplicativi. La prima lettera è maiuscola. Le prime lettere indicano il tipo di ontologia:
 - Feed: **Feed**
 - Command: **Cmd**
 - Alert: **Alrt**
 - Schedule: **Schdl**
 - Audit: **Adt**
 - KPI: **Kpi**
- Per la definizione di **ontologie**:
 - La prima lettera è minuscola, iniziano sempre con il tipo di ontologia. Per esempio, a proposito di una spira: feedSpira, cmdSpira, adtSpira, etc.
- Per la definizione di **attributi** dell'ontologia:
- La prima lettera è minuscola, non ci sono spazi né caratteri speciali.
- Per la definizioni di **costanti** usate come valori possibili degli attributi dell'ontologia: tutte le lettere sono maiuscole. Per esempio: MOBILE, FIXED, VIRTUAL.

5 TIPI E FORMATI

Per la definizione di ontologie vengono usate stringhe di testo UTF-8 con lo schema JSON stabilito nel template corrispondente (al momento si adotta JSON Schema 0.4 <http://json-schema.org/draft-04/schema#>).

Poi si definiscono le regole di tipizzazione e formato per i tipi supportati:

- **UUIDs:**
 - Stringa di testo. Standard Universally Unique Identifier.
- **Numeri Interi:**
 - Interi di lunghezza 64 bit
 - Esempio: `{'contatore': 10}`
- **Numeri a virgola mobile:**
 - Notazione semplice. Decimale con punto. 64 bits
 - Esempio: `{'valore': 10.5}`
- **Stringhe di testo:**
 - Stringa di testo. UTF-8. Caratteri speciali esclusi
 - Esempio: `{'comment': 'next station'}`
- **URLs e URIs:**
 - Stringa di testo. Codificate con lo standard RFC-1738
 - Esempio: `{'url': 'http%3A%2F%2Fwww.coruna.es%2Fmedioambiente%2F'}`
- **Timestamps:**
 - Data. Stringa di testo secondo il formato ISO-8601. RFC 3339
 - Oggetto contenente l'attributo "\$date"
 - Esempio: `{"timestamp":{"$date":"2014-01-27T11:14:00Z"}}`
- **Date ed intervalli di date:**
 - Stringa di testo secondo il formato ISO-8601. RFC 3339
 - Oggetto contenente l'attributo "\$date"
 - Esempio di data: `{"created":{"$date":"2014-01-27T11:14:00Z"}}`
 - Esempio di intervallo fra due date: `{"period":{"$date": "2010-07-02T11:44:09Z/2010-07-02T11:47:00Z"}}`
- **Indirizzi:**
 - Notazione semplificata per facilitare le operazioni di integrazione:

```
{"address": {  
  "location": "cadena de texto",  
  "number": "cadena de texto"  
}}
```

- Unità di misura
 - Stringa di testo string secondo la notazione JScience library (<http://jscience.org/api/javax/measure/unit/Sl.html>) (<http://jscience.org/api/javax/measure/unit/NonSl.html>)
 - Esempio { 'unit': 'A' } # Amperes
- Coordinate geografiche:
 - Seguono le definizioni OGC GeoJson. Schema di coordinate WGS84. Non ci sono coordinate Z. Array [longitudine, latitudine]
 - Punti:
 - GeoJson Point
 - Esempio:

```
{ "geometry": {
  "type": "Point",
  "coordinates": [-8.410161625142807, 43.360463863501934]
}
```

- Linee:
 - GeoJson LineString
 - Esempio:

```
{ "geometry": {
  "type": "LineString",
  "coordinates": [
    [-8.410161625142807, 43.360463863501934],
    [-8.410161625142807, 43.360463863501978]
  ]
}
```

- Aree:
 - GeoJson Polygon
 - Esempio:

```
{ "geometry": {
  "type": "Polygon",
  "coordinates": [
    [
      [-8.410161625142807, 43.360463863501934],
      [-8.410161625142807, 43.360463863501978],
      [-8.41016162514290, 43.360463863501978],
      [-8.410161625142807, 43.360463863501934]]
    ]
}
```

[NOTA]: Per chiudere il poligono il primo e l'ultimo valore di ogni anello devono essere identici.

[NOTA]: Un poligono può avere due anelli (esterno ed interno).



6 TEMPLATES PREDEFINITI

6.1 Feed (Misurazioni)

Per la definizione di templates di misurazioni si usa una semplificazione dello standard dati AMON (<http://amee.github.io/AMON/>):

```
{ "Feed":
  {
    "asset":
      {
        "assetId" : string, (required)
        "assetType": string, (required)
        "assetSource": string, (required)
        "assetName": string (optional)
      },
    "type": string, (required) [FIXED, MOBILE, VIRTUAL]
    "timestamp": (required)
      {
        "$date": "RFC 3339 DATETIME"
      },
    "attribs": (optional)
      [
        { "name": "value" }
      ],
    "geometry": geojson [Point, LineString, Polygon], (optional)
    "measures": (required)
      {
        "timestamp" : (required)
          {
            "$date": "RFC 3339 DATETIME"
          },
        "type"      : string, (required) [INSTANT, CUMULATIVE, PULSE]
        "period"    : number, (optional)
        "periodUnit": string, (optional) [m, s, h, d]
        "values"    : (required)
          [
            {
              "name": string, (optional)
              "desc": string, (optional)
              "unit": string, (required)
              "measure": string, (required)
              "method": string, (required)
              "modifications": (optional)
                [ {
                  "oldMeasure": string, (required)
                  "changeTimestamp": (required)
                    {
                      "$date": "RFC 3339 DATETIME"
                    }
                  "changeDesc": string, (optional)
                }
              ]
            }
          ]
        }
      }
  }
}
```

L'oggetto **asset** si riferisce all'asset che emette la segnalazione di misura:

- **assetId**: identificatore dell'asset nel sistema di riferimento che lo gestisce (definito nel campo
- **assetSource**).
- **assetType**: tipo di asset (fanale, sensore di umidità, etc.)
- **assetSource**: Sistema informativo che gestisce l'asset.
- **assetName**: attributo facoltativo per associare un nome all'asset, se necessario.

Il tipo del sensore (**type**) si riferisce alla sua natura, i tipi validi sono:

- **FIXED**. Sensori conosciuti a priori (gestiti da un inventario conosciuto) e sono posizionati in una postazione geografica conosciuta e fissa.
- **MOBILE**. Sensori conosciuti a priori che si muovono. La loro posizione si aggiorna ad ogni misurazione.
- **VIRTUAL**. Sensori che non sono conosciuti (per esempio di reti sociali).

Il **timestamp** si riferisce alla **data e ora di arrivo del feed**.

NOTA1: da non confondere con il timestamp che viene automaticamente generato quando si invia l'ontologia al modulo di integrazione

NOTA2: da non confondere con il timestamp che si riferisce al momento di ricezione della misurazione.

L'oggetto degli **attributi** (attribs) ha come scopo la raccolta di una lista arbitraria di attributi nella forma chiave:valore. Può essere utile dalla raccolta di chiavi secondarie fino alla memorizzazione di attributi aggiuntivi necessari.

L'oggetto **geometry** raccoglie la posizione (punto, linea o poligono) alla quale il feed fa riferimento. **Comunque, ammesso che la posizione del feed sia conosciuta, essa deve figurare nel feed, a prescindere dal fatto che sia già stata scaricata nell'inventario. Se invece essa non è conosciuta non deve esserci.**

L'oggetto **measures** si riferisce a caratteristiche di riferimento comuni a tutte le misurazioni prese ed elenca le misurazioni fatte:

- **timestamp**: data di riferimento di esecuzione della misurazione.
- **type**: tipo di misurazione realizzata: misurazione istantanea, cumulato, pulsazione.
- **period**: se esiste, periodo di tempo per il calcolo delle misurazioni.
- **periodUnit**: unità di tempo ('s', 'm', 'h', 'd') usata per definire il periodo di tempo.
- **values**: lista di misurazioni prese.

- **name**: se esiste, nome parlante della misurazione.
- **desc**: se esiste, descrizione della misurazione presa.
- **unit**: unità di misura.
- **measure**: valore della misurazione nella versione più attuale. Cioè, l'attributo measure conterrà sempre la misurazione valida. Quando la misurazione viene modificata i valori storici verranno salvati nella lista di attributi "modifications".
- **method**: metodo usato per ottenere la misurazione (media, min, max, etc..).
- **modifications**: lista di modifiche fatte sulla misurazione a partire dal primo valore raccolto.



6.2 Commands (Comandi)

Per definire i comandi si usa:

```
{ "Command":
  {
    "commandId": string, (required)
    "asset":
      {
        "assetId" : string, (required)
        "assetType": string, (required)
        "assetSource": string, (required)
        "assetName": string (optional)
      },
    "timestamp": (required)
      {
        "$date": "ISO 3339 DATETIME"
      },
    "desc": string, (optional),
    "type": string, (required) [REQUEST, RESPONSE],
    "command":
      {
        "type": string, (required) [SWITCH, DIM, SET, EXECUTE, SEND]
        "value1": string, (optional)
        "value2": string, (optional)
        "value3": string, (optional)
        "msg": string (optional)
      }
    "rule":
      {
        "type": string, (required) [ASAP, DATE]
        "date": (optional)
          {
            "$date": "ISO 3339 DATETIME"
          }
      }
  }
}
```



6.3 Allarmi

Per la definizione degli allarmi si usa una semplificazione dello standard CAP 1.2 (<http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>)

```
{
  "Alert": {
    {
      "id" : {
        "alertId": string, (required)
        "alertSource": string (required)
      },
      "timestamp": {
        "$date": "ISO 3339", (required)
      },
      "asset": {
        {
          "assetId" : string, (required)
          "assetType": string, (required)
          "assetSource": string, (required)
          "assetName": string (optional)
        },
        "alert": {
          {
            "sourceAlertId": identifier,
            "subject": string required,
            "description": string optional,
            "source": string required,
            "type": [ALARM, WARNING, MESSAGE, NOTIFICATION, INFO],
            "status": [OPEN, CLOSED, UNKNOWN],
            "affectedLocations": (optional)
              [
                {
                  "desc": string, (optional)
                  "geometry": geojson, (optional) [Point, Line, Polygon],
                  "locationUri": string, (optional)
                }
              ]
          }
        }
      }
    }
    "info": {
      {
        "action": [CREATE, CLOSE, UPDATE, ACK, FOLLOW, SCALATION, REMINDER, CANCEL],
        "sender": string, (required)
        "contact": string, (optional)
        "description": string, (optional)
        "parameters": string, (optional)
        "urgency": [EXPECTED, FUTURE, IMMEDIATE, PAST, UNKNOWN],
        "severity": [EXTREME, MINOR, MODERATE, SEVERE, UNKNOWN],
        "certainty": [LIKELY, OBSERVED, POSSIBLE, UNLIKELY, UNKNOWN],
        "resources": optional
          [
            {
              "name": string, (required)
              "description": string, (optional)
              "uri": string, (required)
              "mimeType": string (optional)
            }
          ]
      }
    }
  }
}
```

6.4 Eventi

La definizione di eventi segue questo schema:

```
{
  "Event": {
    "id": {
      "eventId": string, (required)
      "eventSource": string (required)
    },
    "timestamp": {
      "$date": "ISO 3339", (required)
    },
    "asset": {
      "assetId" : string, (required)
      "assetType": string, (required)
      "assetSource": string, (required)
      "assetName": string (optional)
    },
    "eventInfo": {
      "subject": string, (required)
      "description": string, (optional)
      "type": string, [INFO, PROGRAM, EVENT]
      "affectedLocations": (optional)
      [
        {
          "desc": string, (required)
          "geometry": geojson optional [Point, Line, Polygon],
          "locationURI": string optional
        }
      ],
      "resources": (optional)
      [
        {
          "name": string, (required)
          "description": string, (optional)
          "uri": string, (required)
          "mimeType": string (optional)
        }
      ]
    }
  },
  "eventRule": {
    "type": [SINGLE, PERIOD, RULE],
    "period": (required)
    {
      $date: "RFC 3339 INTERVAL"
    },
    "repeatEach": entero, (opcional)
    "repeatUnit": string (opcional) [s, m, h, d, w, m]
  }
}
```

6.5 Audit

La definizione di messaggi di audit segue questo schema:

```
{ "Adt":
  {
    "id" : {
      "auditId": string, (required)
      "auditSource": string (required)
    },
    "timestamp": (required)
      {
        "$date": "ISO 3339 DATETIME"
      },
    "asset":
      {
        "assetId" :      string, (required)
        "assetType":    string, (required)
        "assetProvider": string, (required)
        "assetName":    string (optional)
      },
    "message": {
      "source": string required,
      "sender": string optional,
      "subject": string required,
      "body": string optional,
      "level": [INFO, WARNING, ERROR, DEBUG]
    }
  }
}
```