

License and Template Access Control for Geospatial Linked Data

Alan Meehan¹, Kaniz Fatema², Rob Brennan¹, Eamonn Clinton³,
Lorraine McNerney³ and Declan O'Sullivan¹

¹ADAPT Research Centre, Trinity College Dublin, Ireland

²Department of Computing, University of Derby, United Kingdom

³Ordnance Survey Ireland, Dublin, Ireland

- Motivation
- Requirements and Use Cases
- Research Objective
- Access Control Approach
 - Overview of proposed components and how it works
- Case study
- Conclusion

- Geospatial Linked Data space expanding
- Benefits:
 - Data enrichment through links
 - Serve fine grain data on the web
- As time goes on - more institutions will utilize Linked Data to serve users/customers
- Institutions have proprietary (closed) data:
 - Geospatial data especially painstaking to collect
- Need for an access control to ensure restricted data is not accessed by users/customers



- Irelands national mapping agency
- Authoritative geospatial information
- OSi holds both open data and proprietary data
- Building data particularly valuable that they serve to customers
- Current process of serving data to customers inefficient
- New process:
 - Convert building data to LD
 - Serve data on the web to customers via an access control approach

- **Requirements:**

- Customers wants modelled declaratively to facilitate easy management (license)
- A set of customer wants have limited use and duration
- Allow customers to check status of their license(s)
- Perform 5 specific use cases but flexible enough to accommodate future data access use cases

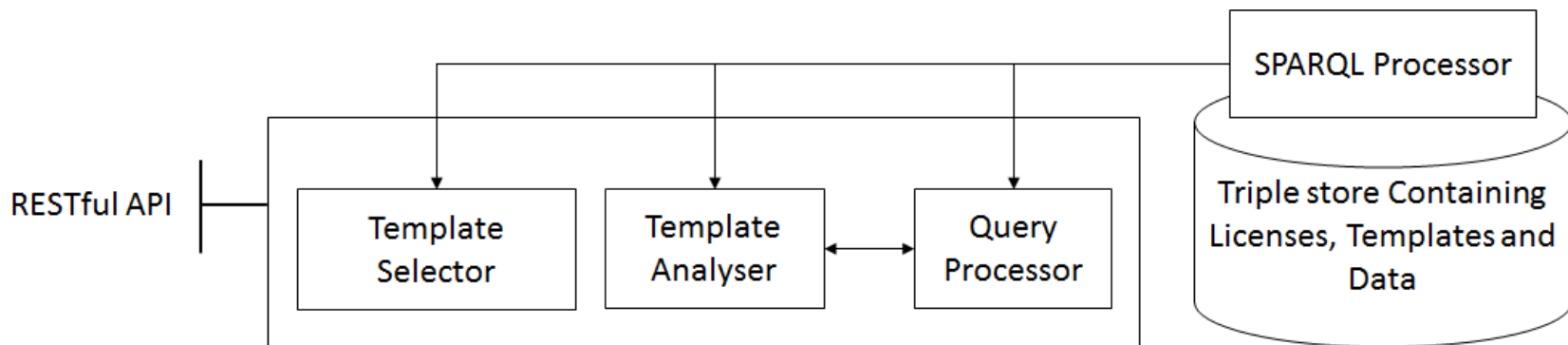
- **Use Cases:**

1. Retrieve the nearest X number of buildings around a point.
2. Retrieve the nearest X number of commercial buildings around a point
3. Retrieve the nearest X number of buildings around another building.
4. Retrieve all buildings of a certain type in a polygon.
5. Retrieve all buildings of a certain type in a county.

How can we develop an access control approach that is fine grain enough to capture the details of what a customer is allowed to access and flexible enough to meet the (potential) different data retrieval use cases of each customer, in a geospatial data retrieval scenario?

- **Propose five parts to the approach:**

- Access Control Model
 - Licenses
 - Templates
- RESTful API
- Template Selector
- Template Analyser
- Query Processor



Vocabulary used to model:

- **License:**
 - Use to model what a user is allowed to access
 - Expiry date
 - Number of uses
- **Template:**
 - Used to model how data can be accessed
 - Contains a SPARQL query with variable placeholders
 - Models what each variable can contain


```
ex:License1 a acon:License ;
  acon:hasLicenseField _:b1 ;
  acon:hasLicenseField _:b2 ;
  acon:hasLicenseField _:b3 ;
  acon:hasLicenseField _:b4 ;
  acon:licenseOwner ex:user1 ;
  acon:queryExecutionNumber "100"^^xsd:integer ;
  acon:licenseExpiryDate "2018-10-10"^^xsd:date .

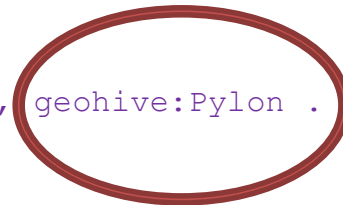
_:b1 a acon:FeaturesAllowed ;
  acon:licenseFieldValue geohiveb:Building .
_:b2 a acon:GeographicalPoint ;
  acon:licenseFieldValue "POINT(-6.35 53.37)" .
_:b3 a acon:Radius ;
  acon:licenseFieldValue "100" .
_:b4 a acon:FeatureNumber ;
  acon:licenseFieldValue "10" .
```

```
ex:Template1 a acon:Template ;
  acon:hasVariable _:b1 ;
  acon:hasVariable _:b2 ;
  acon:hasVariable _:b3 ;
  acon:hasVariable _:b4 ;
  acon:templateDescription "This template will..." ;
  acon:query """
PREFIX geo:<http://www.opengis.net/ont/geosparql#> PREFIX
geof:<http://www.opengis.net/def/function/geosparql/> PREFIX
units:<http://www.opengis.net/def/uom/OGC/1.0/> SELECT ?feature WHERE { ?feature a
<$variable2> ; geo:hasGeometry ?g1. ?g1 geo:asWKT ?g1_wkt .
BIND(geof:distance("$variable4"^^geo:wktLiteral, ?g1_wkt, units:metre) as
?distance) FILTER( ?distance <= "$variable3"^^xsd:double) } LIMIT
$variable1""""^^xsd:string .

_:b1 a acon:TemplateVariable ;
  acon:variableOrder "1"^^xsd:integer ;
  acon:variableExpression acon:FeatureNumber .
_:b2 a acon:TemplateVariable ;
  acon:variableOrder "2"^^xsd:integer ;
  acon:variableExpression geohiveb:Building .
_:b3 a acon:TemplateVariable ;
  acon:variableOrder "3"^^xsd:integer ;
  acon:variableExpression acon:Radius .
_:b4 a acon:TemplateVariable ;
  acon:variableOrder "4"^^xsd:integer ;
  acon:variableExpression acon:GeographicalPoint .
```

```
ex:Template1 a acon:Template ;
  acon:hasVariable _:b1 ;
  acon:hasVariable _:b2 ;
  acon:hasVariable _:b3 ;
  acon:hasVariable _:b4 ;
  acon:templateDescription "This template will..." ;
  acon:query """
PREFIX geo:<http://www.opengis.net/ont/geosparql#> PREFIX
geof:<http://www.opengis.net/def/function/geosparql/> PREFIX
units:<http://www.opengis.net/def/uom/OGC/1.0/> SELECT ?feature WHERE { ?feature a
<$variable2> ; geo:hasGeometry ?g1. ?g1 geo:asWKT ?g1_wkt .
BIND(geof:distance("$variable4"^^geo:wktLiteral, ?g1_wkt, units:metre) as
?distance) FILTER( ?distance <= "$variable3"^^xsd:double) } LIMIT
$variable1""""^^xsd:string .

_:b1 a acon:TemplateVariable ;
  acon:variableOrder "1"^^xsd:integer ;
  acon:variableExpression acon:FeatureNumber .
_:b2 a acon:TemplateVariable ;
  acon:variableOrder "2"^^xsd:integer ;
  acon:variableExpression geohiveb:Building, geohive:Pylon .
_:b3 a acon:TemplateVariable ;
  acon:variableOrder "3"^^xsd:integer ;
  acon:variableExpression acon:Radius .
_:b4 a acon:TemplateVariable ;
  acon:variableOrder "4"^^xsd:integer ;
  acon:variableExpression acon:GeographicalPoint .
```



Access through a RESTful API – Two calls:

- **Status Call:**
 - Used to check the status of license(s)

```
/acon/status/{userID}
```

- **Query Call:**
 - Used to get data

```
/acon/query/{userID}/{LicenseID}/{TemplateID}?variable1={variable_1_value}&variable2={variable_2_value}&variableN={variable_N_value}
```

```
/acon/query/{userID}/{LicenseID}/{TemplateID}?variable1={variable_1_value}&variable2={variable_2_value}&variableN={variable_N_value}
```

```
/acon/query/user1/License1/Template1?variable1=10&variable2=http://ontologies.geohive.ie/osi/building#Building&variable3=100&variable4=POINT(-6.35 53.37)
```

- Invoked by a **Status Call**:
 - Purpose to discover which templates allowable by user based on their license(s)
 - Return that information to user

Checks made by **Template Selector**:

- FOR each user **license**:
 - Check **expiry date** and **execution number**
 - FOR each **template**:
 - FOR each **template variable**:
 - Check **template variable values** against **license field values**
- RETURN description of templates usable with each license

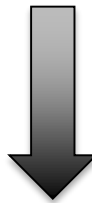
- Invoked by a **Query Call**:
 - Purpose to validate a query call

Checks made by **Template Analyser**:

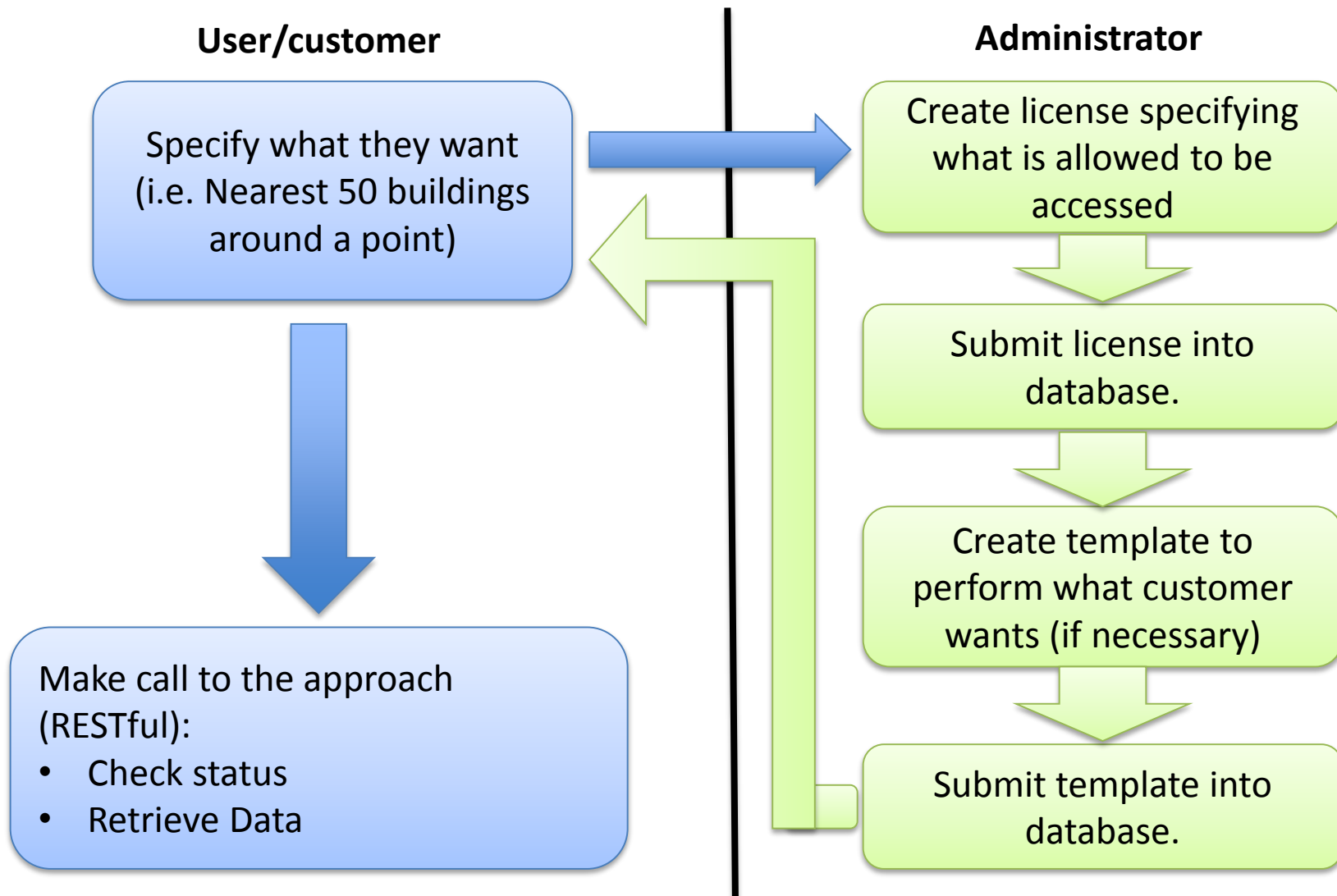
- Check **license** belongs to **customer**
- Check **expiry date** and **execution number**
- FOR each **template variable**:
 - Check **template variable values** against **license field values**
 - Check **license field values** against **variables (from query call)**
- RETURN template and variables which is prepared for execution by **Query Processor**

- Invoked by **Template Analyser**:
 - Purpose to substitute variables from call into SPARQL query from template
 - Execute query and return result

/acon/query/**user1**/**License1**/**Template1**?**variable1=10**&**variable2=**
http://ontologies.geohive.ie/osi/building#Building&**variable3=100**&**variable4=PO**
INT(-6.35 53.37)



```
SELECT ?feature WHERE { ?feature a <$variable2> ;  
geo:hasGeometry ?g1. ?g1 geo:asWKT ?g1_wkt .  
BIND(geof:distance("$variable4"^^geo:wktLiteral, ?g1_wkt,  
units:metre) as ?distance) FILTER( ?distance <=  
"$variable3"^^xsd:double) } LIMIT $variable1
```

Created:

- Implemented of our access control approach:
 - Implemented in Python – hosted on Apache Web server
 - Using Parliament triple store

Performed:

- Previous OSi customer wants modelled as licenses
- Templates created for data access use cases
- Made multiple calls using the approach against OSi building data

Checked:

- Could model the customer wants and the use cases
- Reject a query call when non-valid values specified
- Allowed a query call with valid values specified
- Check returned results

Observed:

- No problems

- Research objective:
How can we develop an access control approach that is fine grain enough to capture the details of what a customer is allowed to access and flexible enough to meet the (potential) different data retrieval use cases of each customer, in a geospatial data retrieval scenario?
- Access control approach for Geospatial Linked Data
- Proposed implementation of our approach
- Access Control Model: Flexible approach for modelling user/customer wants and data retrieval use cases.
- Case study showed usefulness in OSi data publication scenario

Questions?

Contact Email: alan.meehan@adaptcentre.ie