# OGC Testbed 13
# QGIS Security

Andreas Matheus - Secure Dimensions

and

Frank Terpstra – Geonovum

GEONOVUM

# Overview

- Background Testbed 13
- Why QGIS security?
- Approach
- Modern security standards
  - SAML
  - OAuth/OpenID Connect
- Lessons so far
- What’s ahead

GEONOVUM

# Background Testbed 13

- Testbed 13 is part of the Open Geospatial Consortium Interoperability Program
- OGC uses Testbeds for prototyping new standards
- >56 organizations >133 individuals & $2.4 million is involved in the whole of Testbed 13
- Geonovum (together with Kadaster & Digitaal Stelsel Omgevingswet) is one of the sponsors
- Secure Dimensions delivers the work package that Geonovum is sponsoring

# Why security in QGIS

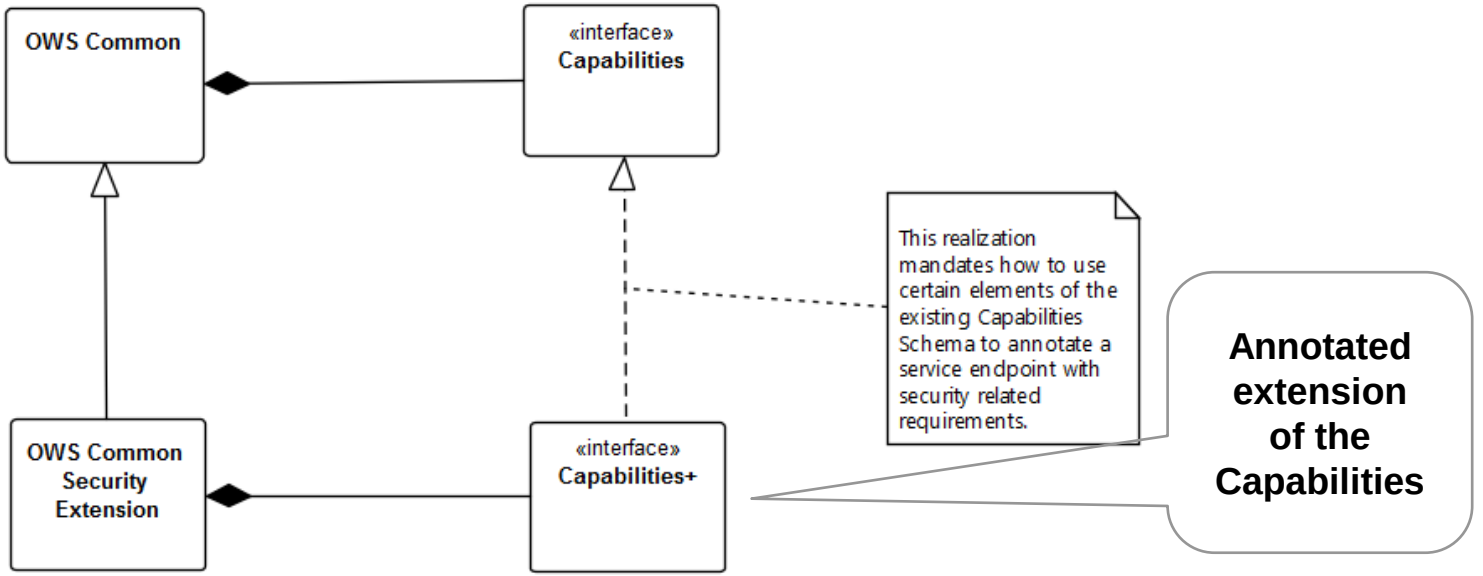
- Security in OGC standards (WMS,WFS etc…) is non-existent (not even basic authentication over HTTP as status codes 4xx results in XML ExceptionReport with "NoApplicableCode" [OGC06-121r9, page 47, table 28])
- Work is underway on new Common Security standard supporting modern methods
- In previous Testbeds security using modern methods on the serverside was demonstrated (SAML, GeoXACML, X.509)
- The big remaining hurdle is interoperable support for modern security methods in client software

GEONOVUM

# Approach

- Build a plugin in QGIS (2.18.4)
- Support SAML2 & OAuth2 for WMS & WFS
- Test against endpoints from Kadaster(OAuth), Secure Dimensions(SAML) and NASA(OAuth)
- Use draft OWS Common Security standard to annotate endpoints such that client QGIS (and other clients) can use them with minimal input from the user

# Approach

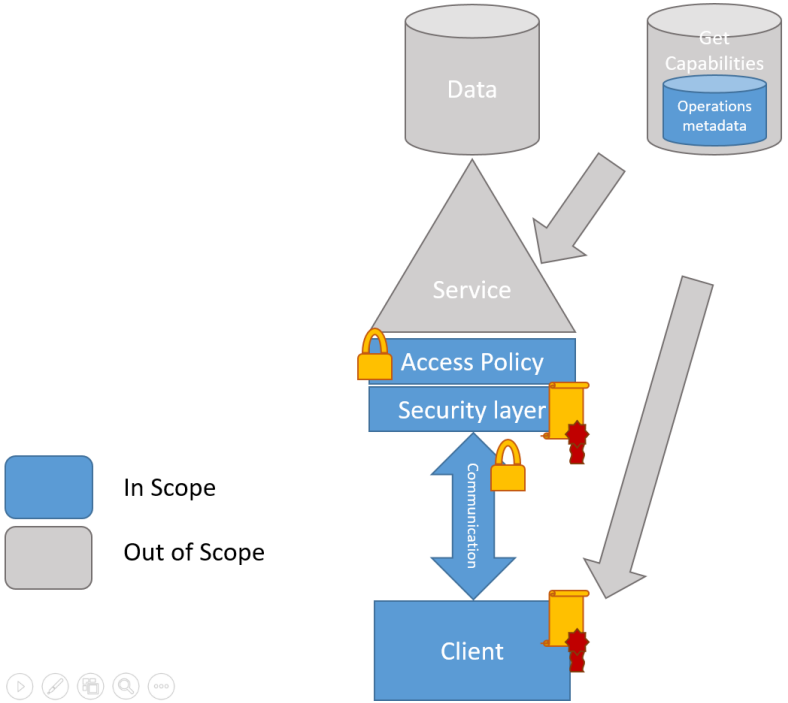

OWS Common
«interface»
Capabilities
This realization mandates how to use certain elements of the existing Capabilities Schema to annotate a service endpoint with security related requirements.
OWS Common Security Extension
«interface»
Capabilities+
Annotated extension of the Capabilities

# Example Annotated Capabilities for WFS

```
</ows:Operation>
<ows:Operation name="Transaction">
  <ows:DCP>
    <ows:HTTP>
      <ows:Get xlink:href="http://tb12.secure-dimensions.com/basic/wfs">
        <!--
        This option to state a constraint is on this GET operation, I have proposed two options.
        Not sure which one or if both are valid. We need to consult domain experts in OWS Common
        Here OWS Common 1.1.0
        -->
        <ows:Constraint name="ogc:urn:def:security:authentication">
          <!--
            Option one uses the ValuesReference to the codelist for authentication
          -->
          <!-- The ValuesReference is the machine readable part of the defintion? -->
          <ows:ValuesReference ows:reference="http://www.opengis.net/security/authCodeList#HTTP_BASIC"/>
          <!-- The ows:Meaning is the human readable definition pointing to the IETF RFC 2617 -->
          <ows:Meaning ows:reference="http://ietf.org/rfc/2617.html#BASIC"/>
        </ows:Constraint>
      </ows:Get>
      <ows:Post xlink:href="http://tb12.secure-dimensions.com/basic/wfs">
        <!--
          This option to state a constraint is on this GET operation, I have proposed two options.
          Not sure which one or if both are valid. We need to consult domain experts in OWS Common
          Here OWS Common 1.1.0
        -->
        <ows:Constraint name="ogc:urn:def:security:authentication">
          <!--
            Option one uses the ValuesReference to the codelist for authentication
          -->
          <!-- The ValuesReference is the machine readable part of the defintion? -->
          <ows:ValuesReference ows:reference="http://www.opengis.net/security/authCodeList#HTTP_BASIC"/>
          <!-- The ows:Meaning is the human readable definition pointing to the IETF RFC 2617 -->
          <ows:Meaning ows:reference="http://ietf.org/rfc/2617.html#BASIC"/>
        </ows:Constraint>
      </ows:Post>
    </ows:HTTP>
```

```
<!--================================================================================-->
<!--================================================================================-->
<!--============================== Codelists ======================================-->
<!--=== AuthenticationCode ===-->
<gmx:codelistItem>
  <gmx:CodeListDictionary gml:id="AuthenticationCode">
    <gml:description>identification of authentication methods</gml:description>
    <gml:identifier codeSpace="SD">urn:sd:authentication</gml:identifier>
    <gmx:codeEntry>
      <gmx:CodeDefinition gml:id="urn_ietf_rfc_2617_basic_authentication">
        <gml:description>The "basic" authentication scheme is based on the model that the
          client must authenticate itself with a user-ID and a password for
          each realm.  The realm value should be considered an opaque string
          which can only be compared for equality with other realms on that
          server. The server will service the request only if it can validate
          the user-ID and password for the protection space of the Request-URI.
          There are no optional authentication parameters.</gml:description>
        <gml:identifier codeSpace="IETF">BASIC</gml:identifier>
      </gmx:CodeDefinition>
    </gmx:codeEntry>
    <gmx:codeEntry>
      <gmx:CodeDefinition gml:id="urn_ietf_rfc_2617_digest_authentication">
        <gml:description>Like Basic Access Authentication, the Digest scheme is based on a
          simple challenge-response paradigm. The Digest scheme challenges
          using a nonce value. A valid response contains a checksum (by
          default, the MD5 checksum) of the username, the password, the given
          nonce value, the HTTP method, and the requested URI. In this way, the
          password is never sent in the clear. Just as with the Basic scheme,
          the username and password must be prearranged in some fashion not
          addressed by this document.</gml:description>
        <gml:identifier codeSpace="IETF">DIGEST</gml:identifier>
      </gmx:CodeDefinition>
    </gmx:codeEntry>
```

# Scope of OWS Common Security standard



# Modern security standards

- Support for modern security in OGC standards requires new standard
- OWS Common Security (draft) standard defines required changes
  - Implementation on server side is easily added through additional security layer (toegangslaag Kadaster, Knooppunt DSO)
  - Implementation in mobile and web clients is straightforward through mobile platform libraries & standard Web Browser functionality
  - Implementation for desktop clients requires more work
    - Support HTTP**S** and HTTP (secure) Cookies
    - Support all HTTP status codes especially 3xx (redirects)

GEONOVUM

# SAML

- Standard for authentication
- Supports multiple levels of assurance
- Mainly used for web based logon, Single-Sign-On
- Also has Enhanced Client Proxy Profile (ECP) for standalone (desktop) applications
- Underlying standard for DigiD/eHerkenning

# OAuth/OpenID Connect

- OAuth2 is standard for Rights Delegation
- OpenID Connect is an extension to OAuth2 for adding user claims
- Mainly used for social media login
  - Log into app/site using your facebook/google+ account
  - OAuth allows user to grant application access to his personal information that is registered elsewhere (google/facebook …)

# Lessons so far

- Implementing OAuth on the server side was easy (few hours of work at Kadaster)
- OAuth not a natural fit for desktop clients (especially if open source)
- Authorization Code Grant (as used by NASA) fits better than Resource Owner Password grant (as implemented at Kadaster)
- OAuth makes sense when OGC services are used:
  - Within a larger ecosystem where OAuth is the norm (e.g. Digitaal Stelsel Omgevingswet)
  - When you want to do more than just authenticate (e.g. grant access to a user owned shared folder for storing results)
  - Other technical (deployment) reasons of your OGC services

GEONOVUM

# Lessons so far

- OAuth requires Client registration
  - In open source only possible in binary distributions
  - Registration for large Identity providers (Google, Facebook etc…) can be “pre registered”
  - Registration for smaller identity providers (DSO, Kadaster, NASA) requires extra steps by user and/or extra effort by identity provider to get clients registered
    - Not a seamless user experience unless identity provider & client support **“dynamic client registration”**

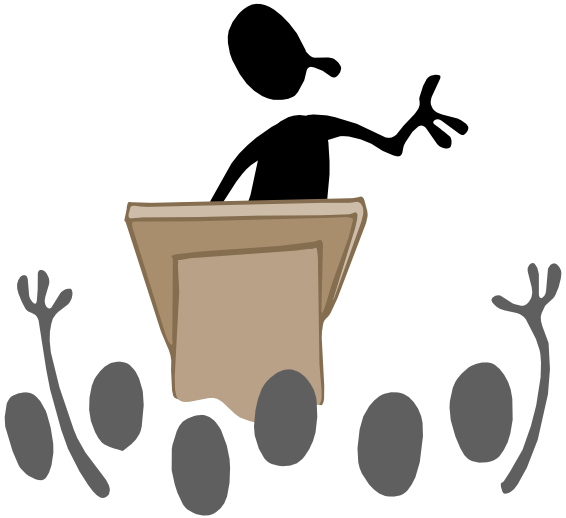
GEONOVUM

# What's ahead

- Today-November: Testbed execution
  - Implementing QGIS client plugin
  - Testing with endpoints
  - Testing by clients from other work packages in Testbed 13
- December: Testbed demonstration event
- Early 2018 public release of final engineering reports

GEONOVUM

# Questions?



- Or ask Frank Terpstra later (f.terpstra@geonovum.nl)

***It is important,***
***to do security right...***


**Secure Dimensions GmbH**
**Holistic Geosecurity**
Dr. Andreas Matheus

Leopoldstr. 244
D-80807 München, Germany

Email am@secure-dimensions.de
Web www.secure-dimensions.de