

Authorization for a Service-based Geospatial Data Infrastructure

52nd OGC Meeting, New York, NY

Andreas Matheus

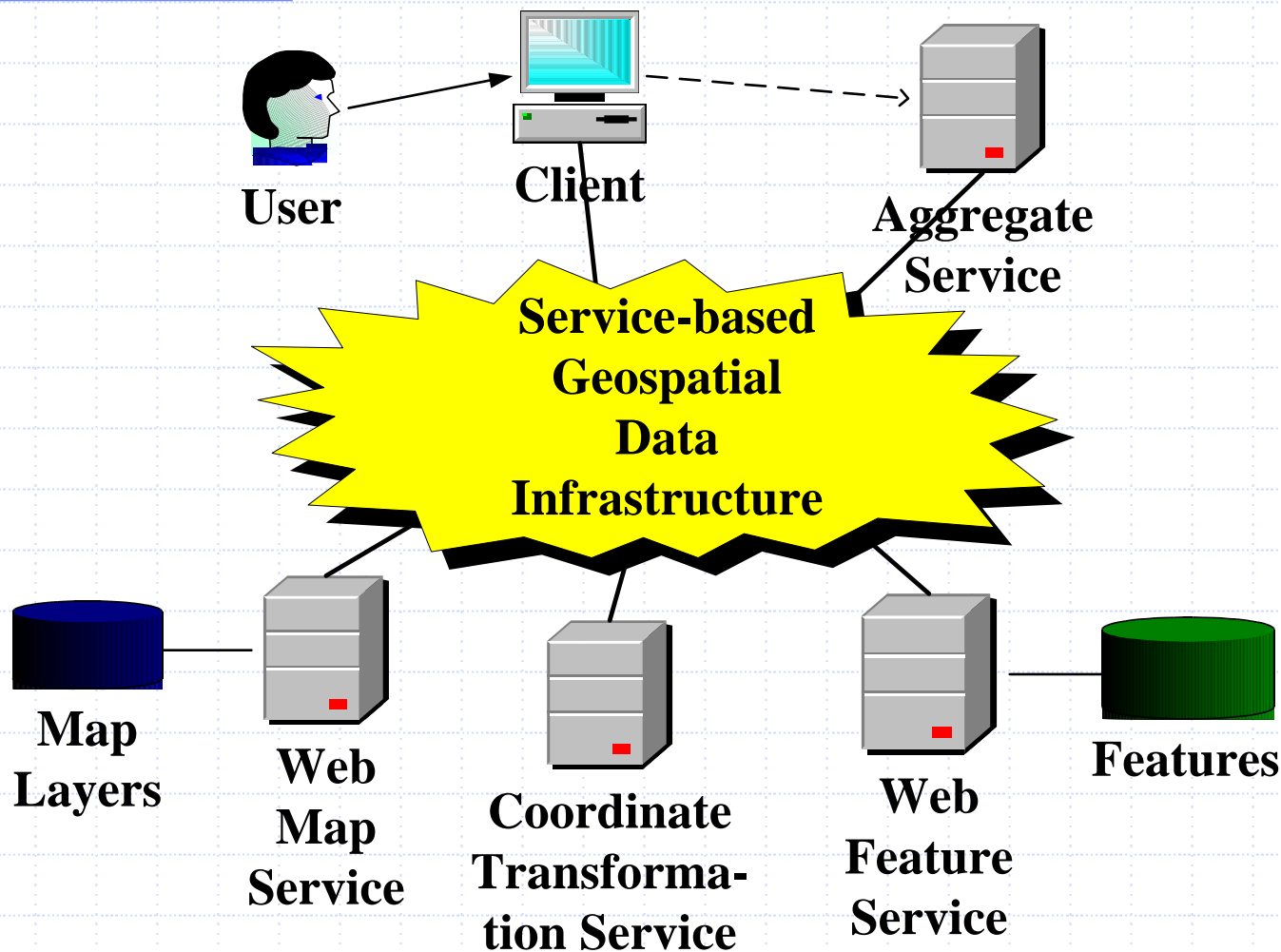
Technische Universität München, Munich

matheus@in.tum.de

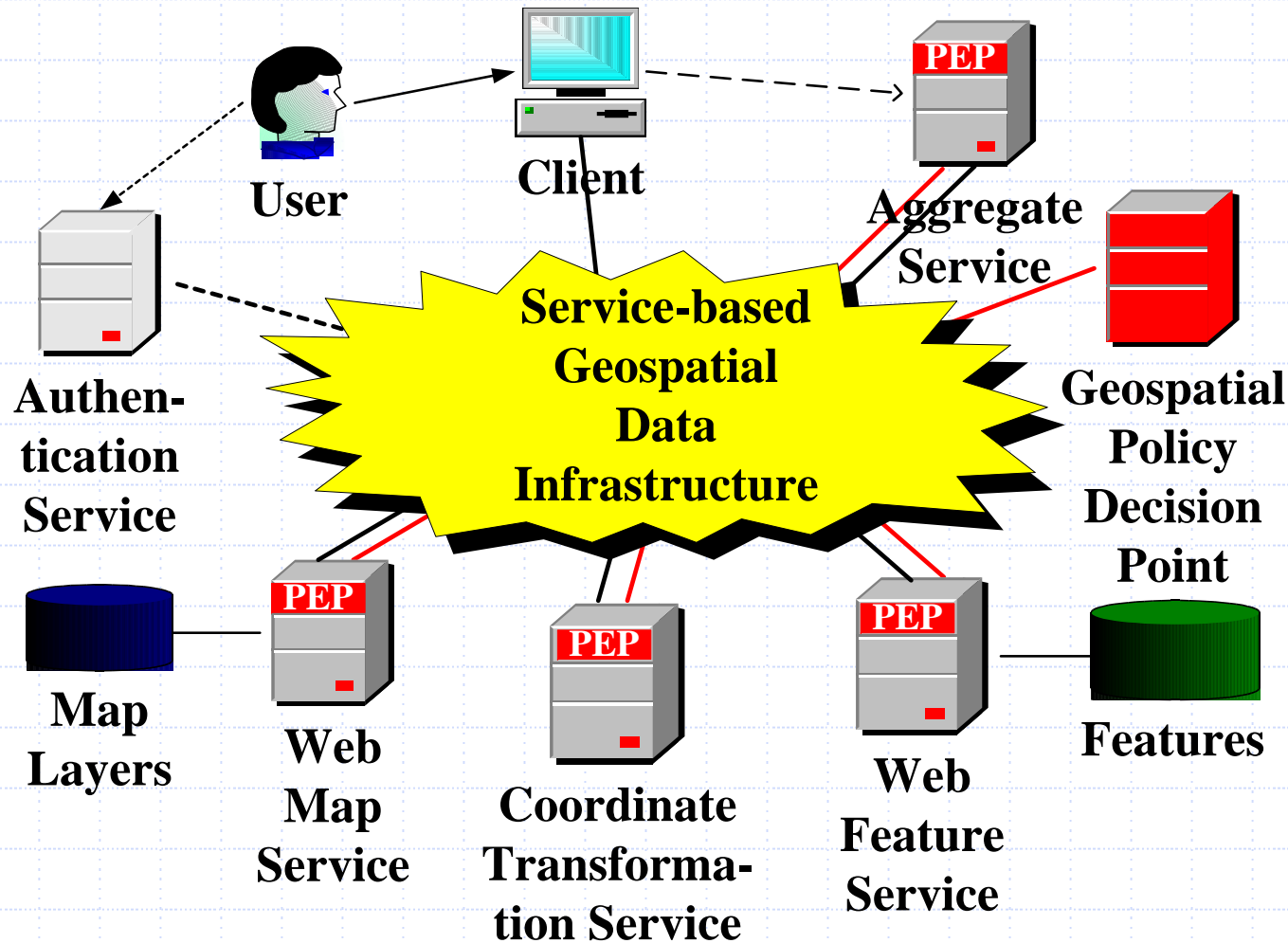
Overview of this Presentation

- ◆ Authorization for a Geospatial Data Infrastructure
- ◆ (Geospatial) Authorization requirements
- ◆ Declaration and enforcement of restrictions
- ◆ Online demonstration for restricting access to a Web Map Service
- ◆ How does this work fit into GeoDRM?
- ◆ Conclusion and (possible) future work

Example Service-based Geospatial Data Infrastructure



Service-based Geospatial Data Infrastructure with Authorization



Authorization Requirements

◆ Class permission

- Applies to all instances of a class
- GML: Applies to all features of the same feature type
- E.g. Alice can read all instances of feature type Building

◆ Object permission

- Applies to individual instances
- GML: Applies to individual features based on values of non-spatial property(ies)
- E.g. Alice can not read the instance of feature type Building, identified with the address "350 Fifth Ave Ste. 3201, New York, NY 10118"

Authorization Requirements

◆ Spatial permission

- Applies to objects that fulfill a specific topological relation with permission geometry
- GML: Applies to individual features according to value of spatial property(ies)
- Alice cannot read the instances of feature type Building, within the area of Manhattan (e.g. PolygonAttribute : = {"EPSG:4326, -74,40.6 -74,40.8 -73,41 -72,41 -73.5 40.6 -74,40.6"})

Declaration of Required Permissions with XACML*

- ◆ Permissions can be encoded in XML
- ◆ Rule-based authorization model
- ◆ Enforcement of permissions for XML encoded resources (e.g. GML feature collection)
- ◆ Supports declaration and enforcement of
 - ✓ **Class permission:** Applies to GML encoded features based on the feature type
 - ✓ **Object permission:** Applies to GML encoded features based on value of non-spatial properties
 - ✗ **Spatial permission:** Not supported ⇒ Extension

*eXtensible Access Control Markup Language by OASIS

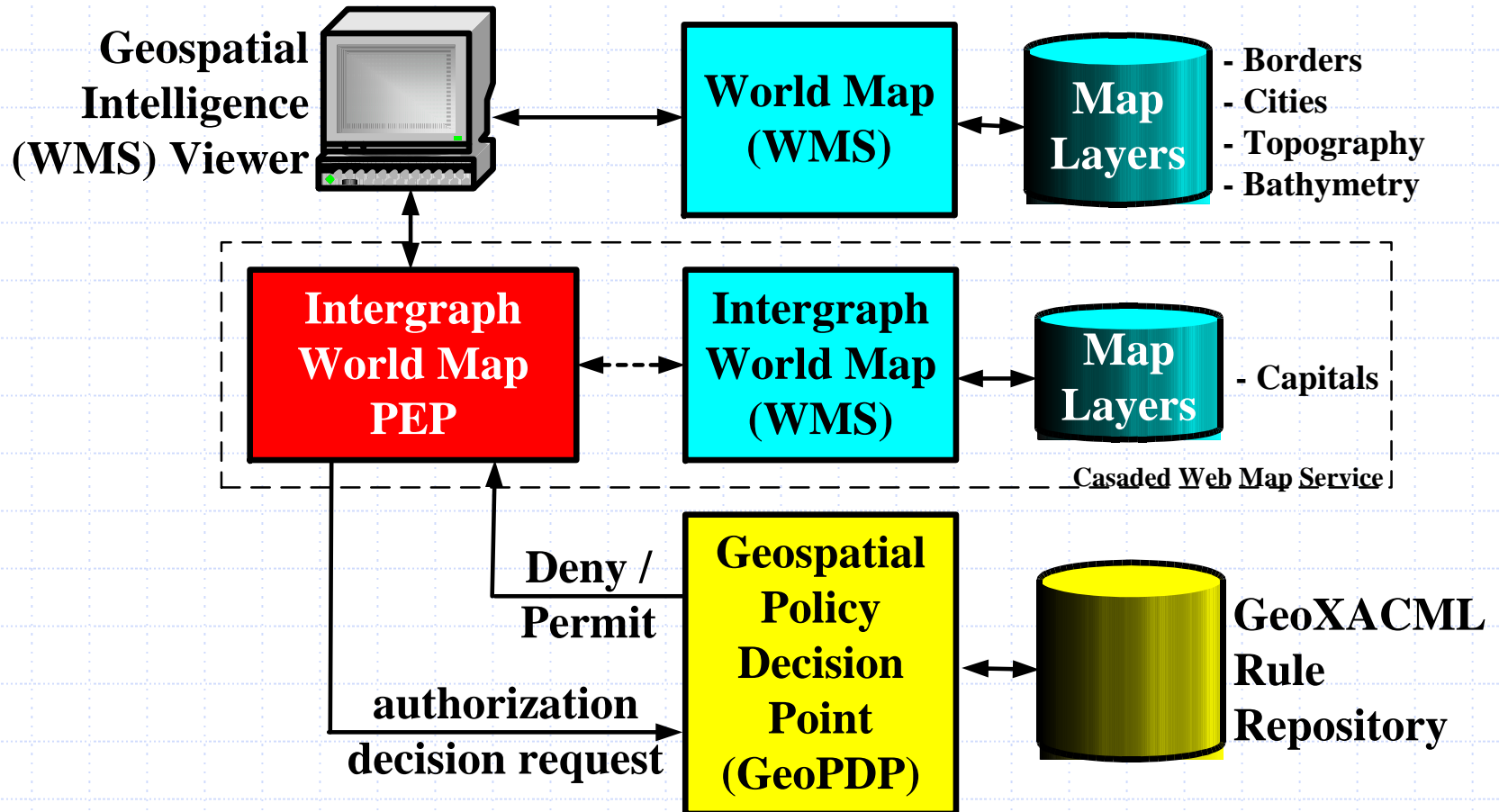
GeoXACML*:

A Geospatial Extension to XACML

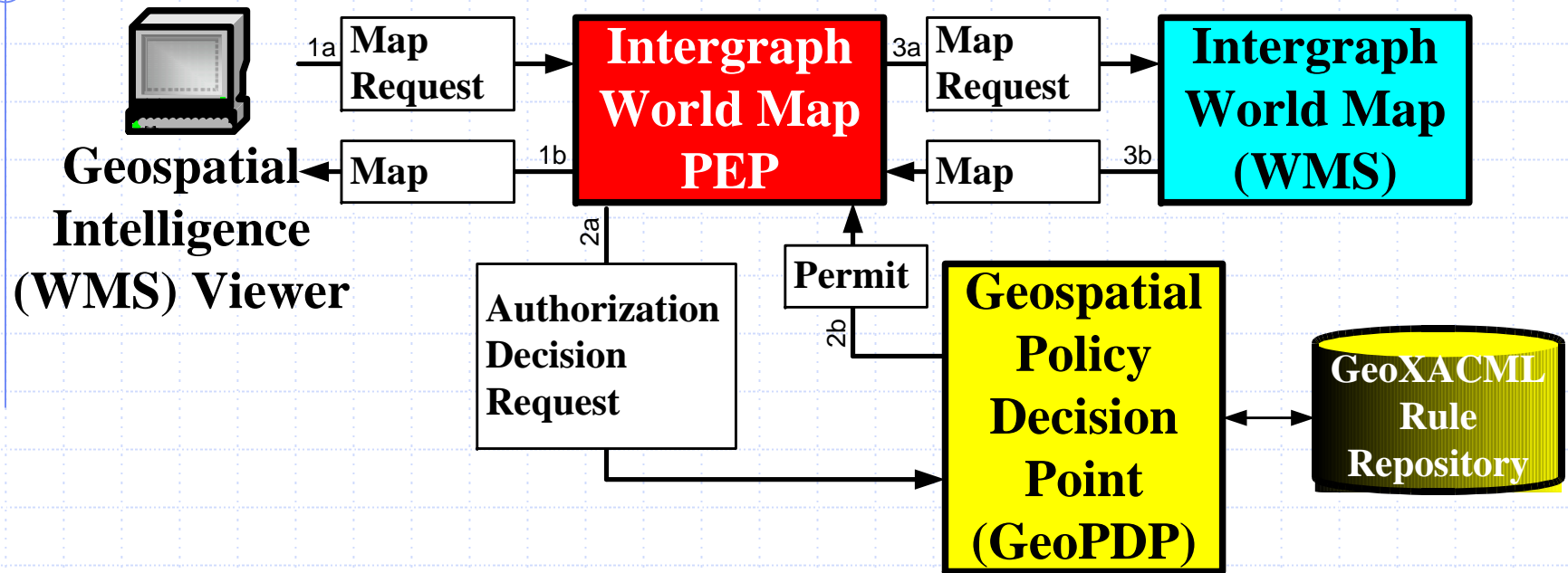
- ◆ Developed during dissertation project
- ◆ Defines constructs for spatial permissions
 - Geospatial data types based on GML 2.1 simple geometry
`PointAttribute`, `LineStringAttribute`,
`LinearRingAttribute`, `BoxAttribute`,
`PolygonAttribute`
 - Topological relational functions
`disjoint`, `touches`, `crosses`, `within`,
`overlaps`, `intersects`, `equals`, `contains`

***Geo**spatial **eX**tensible **A**ccess **C**ontrol **M**arkup **L**anguage

Demonstration Infrastructure



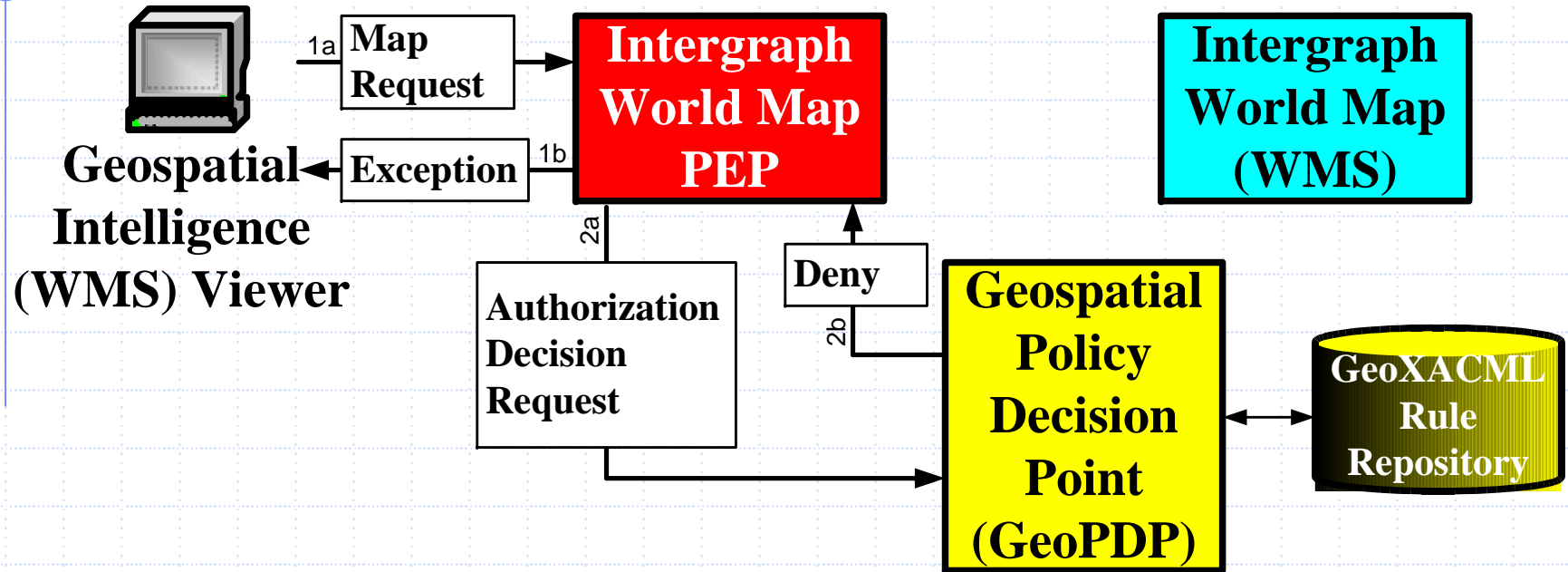
Permitted WMS Request



◆ Authorized request

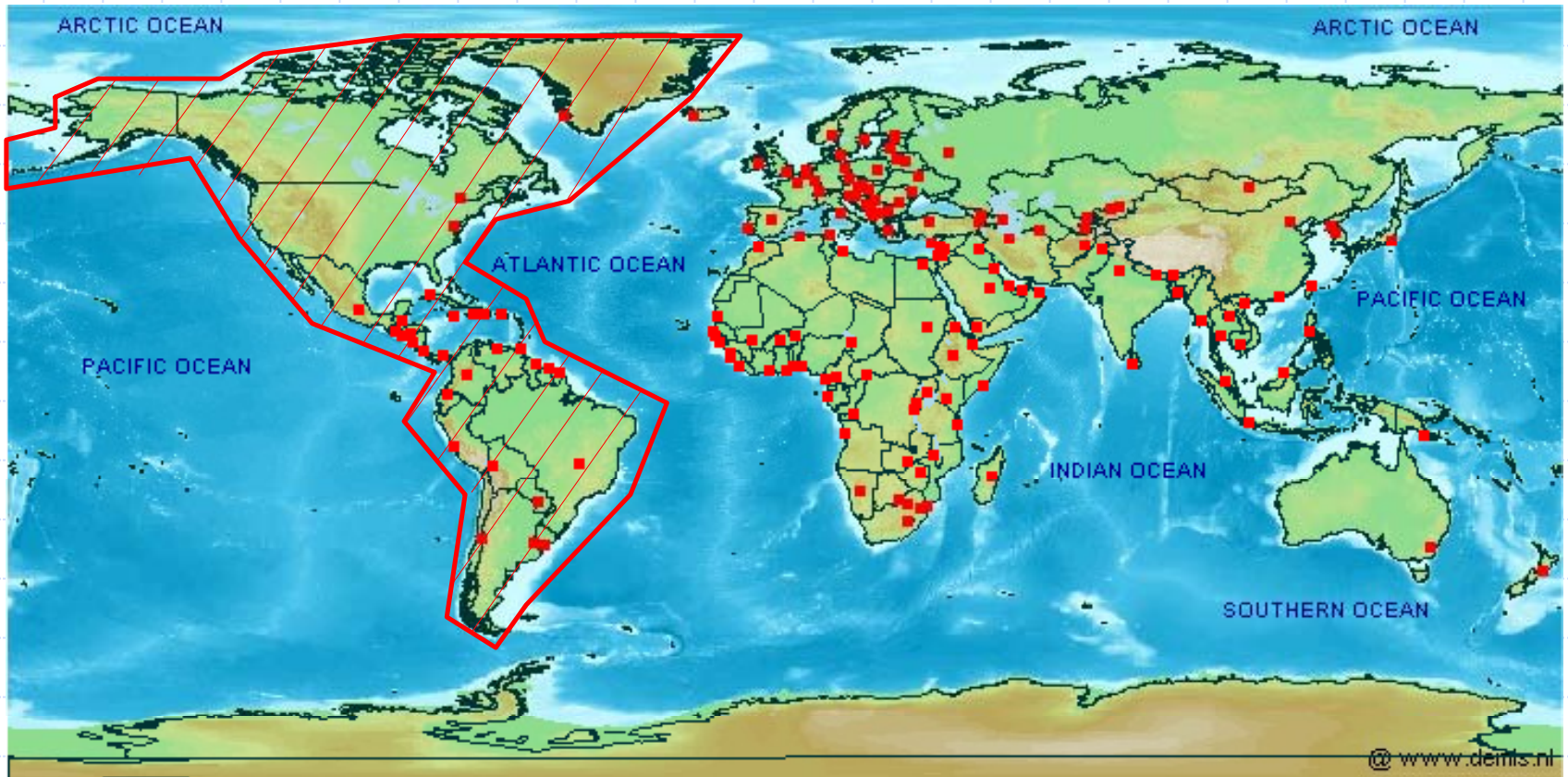
- PEP receives map request
- PEP forwards map (according to request) to Client

Denied WMS Request



- ◆ Not authorized request
 - PEP receives map request
 - PEP returns exception image “Not Authorized”

Map Restrictions



◆ Map access denied to Capitals within red marked area

FeatureInfo Restrictions



◆ **FeatureInfo** access **allowed** to **Capitals** within red marked area

Online Demonstration URLs

- ◆ Geospatial Intelligence Viewer - WMS Client
 - http://ogc.intergraph.com/gi_portal/
- ◆ World Map – WMS Capabilities URL
 - <http://www2.demis.nl/mapserver//request.asp?WMTVER=1.0.0&REQUEST=capabilities>
- ◆ Intergraph World Map – WMS Capabilities URL
 - <http://maps1.intergraph.com/wms/world/request.asp?SERVICE=WMS&VERSION=1.1.0&REQUEST=GetCapabilities>
- ◆ Intergraph World Map PEP – Capabilities URL
 - <http://geo pep.informatik.tu-muenchen.de/WMS-PEP/servlet/WMS?SERVICE=WMS&VERSION=1.1.0&REQUEST=GetCapabilities>

Mission of the GeoDRM WG

- ◆ *“...Trusted infrastructure for purchasing and protecting rights to digital content...”*
- ◆ *“Test ... technologies required for geospatial DRM including ... information security...”*
- ◆ *“...Develop specifications for geospatial DRM that build on the OGC technical baseline...”*
- ◆ *“...protecting IP rights by controlling geodata distribution and use...”*

Citations from GeoDRM homepage

<http://www.opengeospatial.org/groups/?iid=129>

Does the introduced Authorization fit into GeoDRM?

- ◆ **Controlling geodata distribution and use**
 - **Distribution** can be handled through online access:
Restrict online access through **PEP** and **GeoPDP**
 - **Use of offline content** (distributed geodata):
Request license for particular use (e.g. read, write, map) of offline content
- ◆ **License Service** can possibly use XACML
 - Accept license requests for a particular content (e.g. GML feature collection) and usage
 - Query authorization decision from GeoPDP, using GeoXACML
 - Return requested license or "Not Authorized"

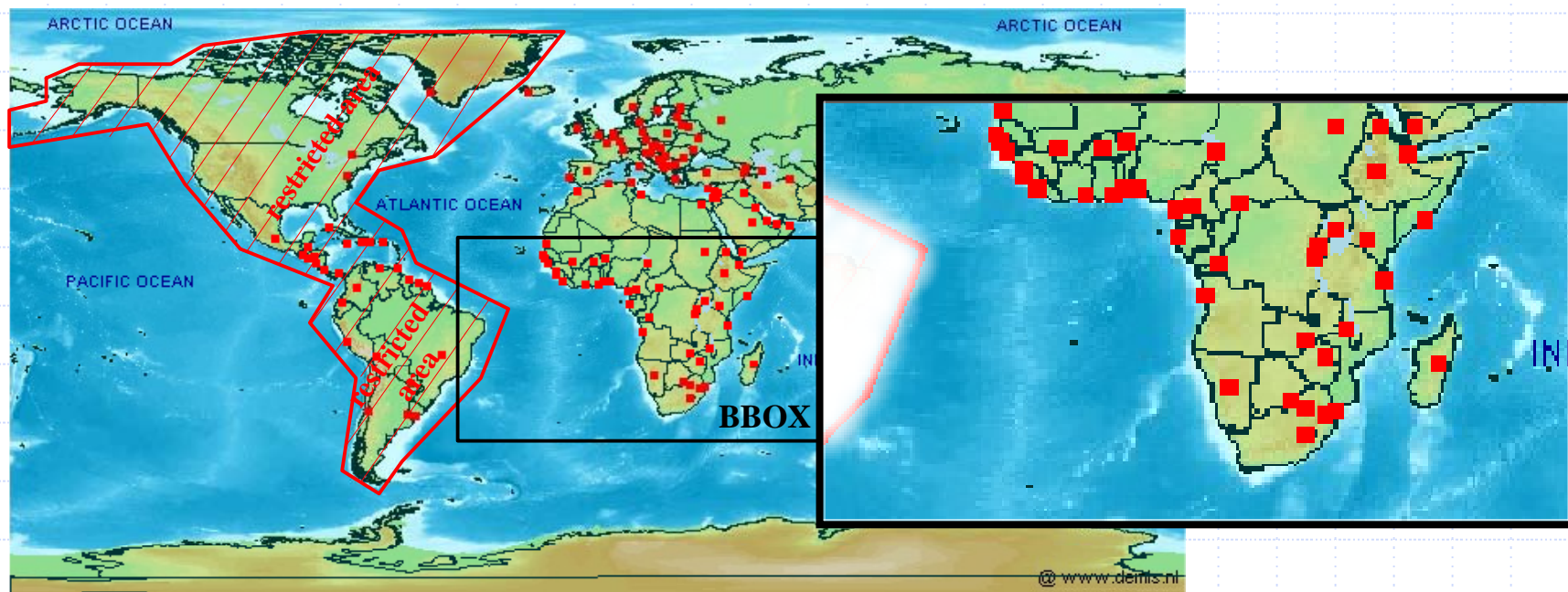
Summary of introduced Authorization

- ◆ **Declaration and enforcement for individual information objects**, encoded in GML (e.g. GML features, GML feature collection, GML geometry)
 - **Class permission**: All features of a particular type
 - **Object permission**: Individual features selected by non-spatial properties
 - **Spatial permission**: Individual features selected through their spatial properties (geometry)
- ◆ **GeoPDP** implementation is independent from usage (one GeoPDP can serve multiple PEPs)
- ◆ **PEP** implementation is specific to the protected service (WMS specific in demonstration)

Possible Future Work (1/3)

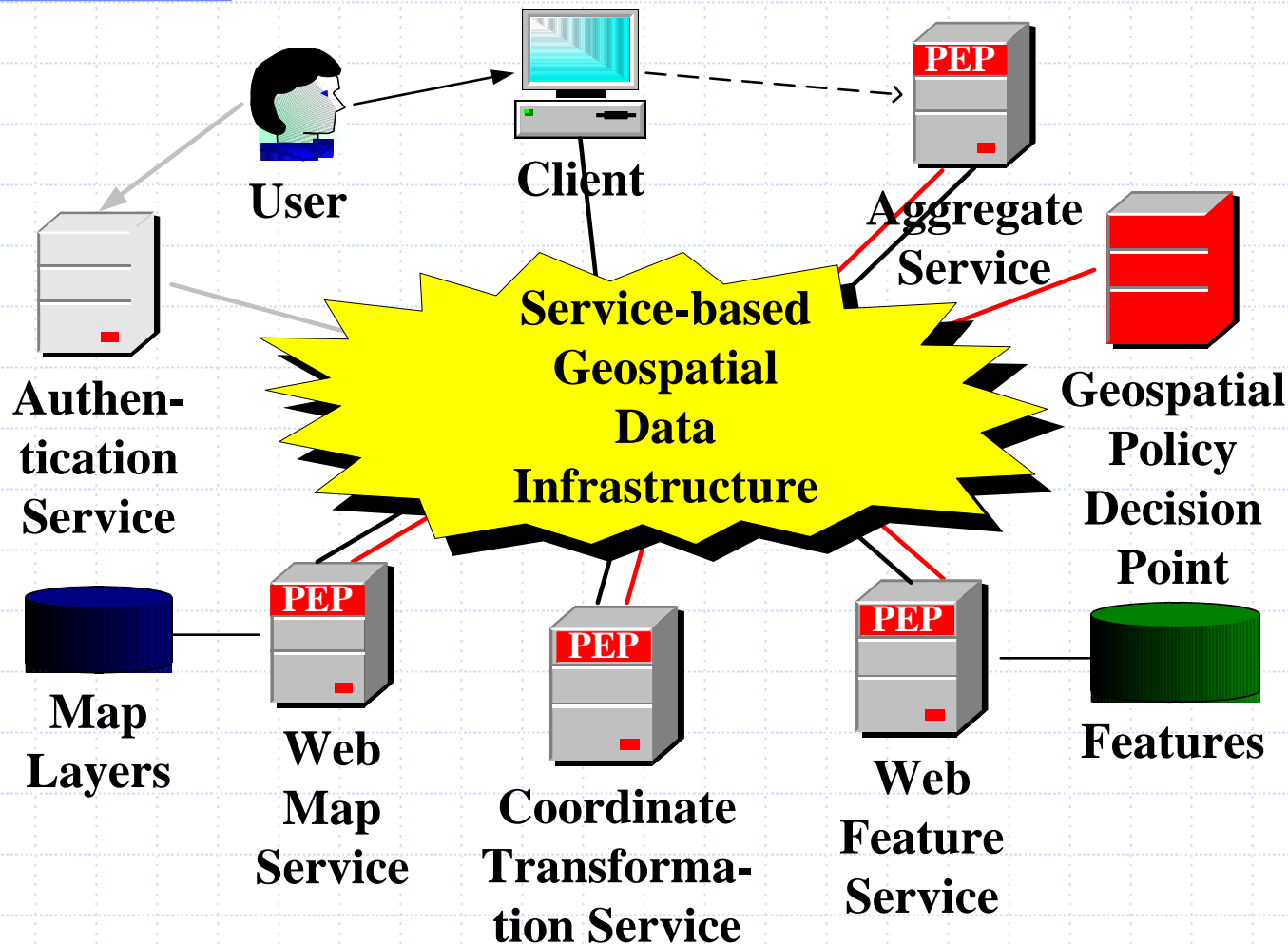
Permit with Constraints

- ◆ Modify original request / filter (XML encoded) result content according to subject's permissions
- ◆ Enable image processing in WMS-PEP for map modification based on subject's permissions



Possible Future Work (2/3)

Authentication Information



Possible Future Work (3/3)

Authentication Information

- ◆ **Standardization** of GeoXACML through OGC?
- ◆ Definition of **SOAP / HTTP-POST** interfaces for **OGC services** such as **WMS** and **WFS**
 - SOAP header can carry authorization information
 - Encoding of authorization information using Security Assertion Markup Language (SAML) from OASIS
 - SOAP body can carry 'original request' in POST format
- ◆ Authorization **test bed** using **SAML** assertions and **SOAP** for restricting access to **WMS** or **WFS** with **PEP** and **GeoPDP**
- ◆ Apply security to communication to achieve integrity, confidentiality, accountability

The Final Slide

Thank you for your attention

Questions, please ...

The XACML Rule

◆ Target

- Defines matching constraints \Rightarrow Applicable permissions
- Matching based on simple string/pattern matching

◆ Condition

- A Boolean expression that evaluates to True or False
- Allows complex matching
- Determines type of encoded permission

◆ Outcome is been used for deriving the authorization decision

- Effect := {Deny, Permit}: Defined at time of writing
- Outcome := {Deny, Permit, NotApplicable, Indeterminate}: Actual value returned at run-time

GeoXACML:

The simplified* model

- ◆ Model is required for validating declared permissions: Incomplete, Unreachable and Contrary permissions
- ◆ Request := {Subject, Action, ResourceContent}
- ◆ Rule := {SM, AM, RM, Condition} → {Deny, Permit}
 - SM: Subject match expression (e.g. "Bob")
 - AM: Action match expression (e.g. "map" or "read")
 - RM: Resource match based on Xpath expression, restricted to match feature types (e.g. //Capitals)

*Detailed model available in dissertation

Condition for Class and Object Permissions

- ◆ Condition := {MF, XM, V} → {True, False}
 - MF: GeoXACML **M**atching **F**unction
 - XM: **X**path **M**atching expression (matching on ResourceContent)
 - V: **V**alue(s) to be compared with result of function MF
- ◆ Class-based permission
 - XM refers to a global element (GML application schema) that represents a feature (e.g. //Building)
- ◆ Object-based
 - XM selects specific features by their non-spatial properties (e.g. //Building/name)
 - V defines the values to be matched with result of XM

Condition for Spatial Permission

- ◆ Condition := {TCF, XM, GP} → {True, False}
 - TCF: **T**opological **C**ondition **F**unction
 - ◆ disjoint, touches, crosses, within, overlaps, intersects, equals, contain
 - ◆ ¬disjoint, ¬touches, ¬crosses, ¬within, ¬overlaps, ¬intersects, ¬equals, ¬contains
 - XM: **X**path **M**atching expression
(e.g. //Building/location)
 - GP: **G**eometry of **P**ermission
(e.g. BoxAttribute, PolygonAttribute)
- ◆ Condition evaluates to True if TCF returns True

Example

Class and Object Permissions

- ◆ **Anyone** can **map** features of type **Building**
 - Rule := {*, map, //Building, Condition} → Permit
 - Condition :=
{string-equal, local-name("//Building"), "Building"}
- ◆ **Bob** can not **read** the feature of type **Building** identified by the **address** "350 Fifth Ave Ste. 3201, New York, NY 10118"
 - Rule := {Bob, read, //Building, Condition} → Deny
 - Condition :=
{string-not-equal, string("//Building/address"),
"350 Fifth Ave Ste. 3201, New York, NY 10118"}

Example Spatial Permissions

- ◆ **Bob** can **map** features of type **Building** if the **location** is **within** the area of Manhattan
 - Rule := {Bob, map, //Building, Condition} → Permit
 - Condition := {within, //Building/location, {EPSG:4326, -74,40.6 ... -74,40.6}}
- ◆ **Anyone** can **map** features of type **Building** if the **location** is **not within** the area of Manhattan
 - Rule := {*, map, //Building, Condition} → Permit
 - Condition := {¬within, //Building/location, EPSG:4326, -74,40.6 ... -74,40.6}