

# MPA (S-122)

## Provider side implementation

Changyun Lee



**AIVN**

Innovating Navigation



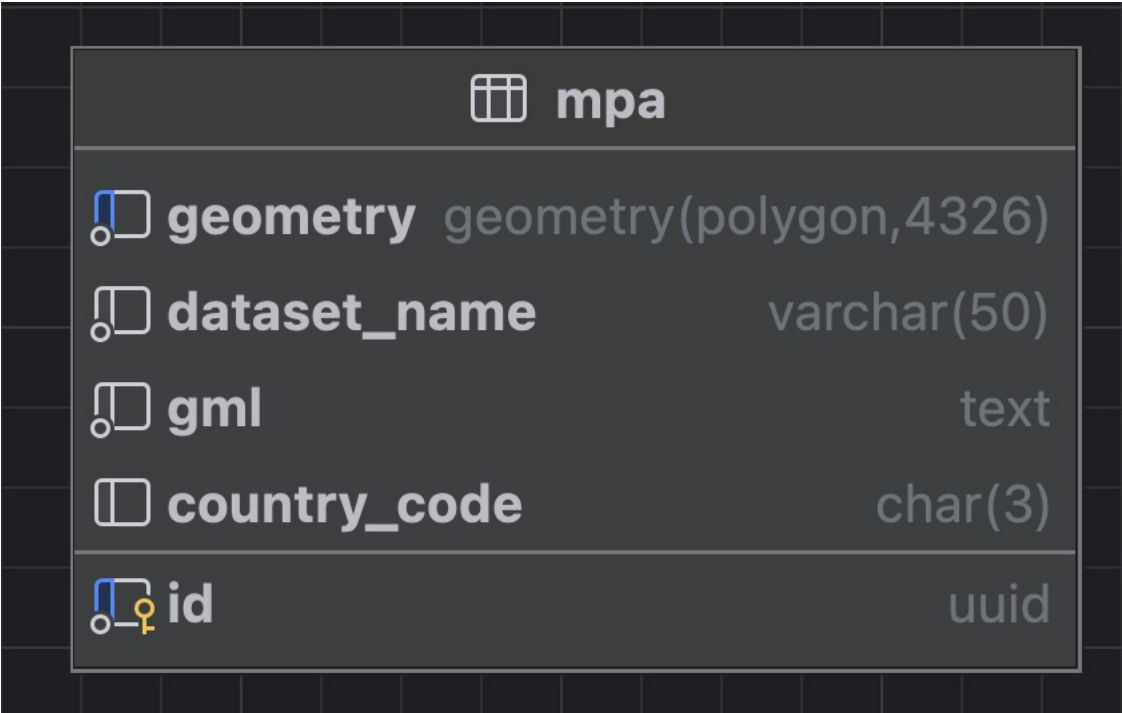
# Table of contents






- Designing a SECOM-compliant service
  - Sequence Diagram Planning
  - SECOM Component Mapping for MPA Service
- Implementing SECOM Services using SECOMLib
  - SECOM Communication Channel Security
  - Service Interface Implementation
- MPA insertion pipeline

# Designing SECOM-based MPA Service Architecture

- One of the simplest SECOM services to implement
- The main function is to deliver MPA data in a **unidirectional** manner
- No bidirectional messaging or subscription mechanism is required — only **request–response** interactions
- The **Get** and **Get Summary** interfaces were the most natural starting point for this service

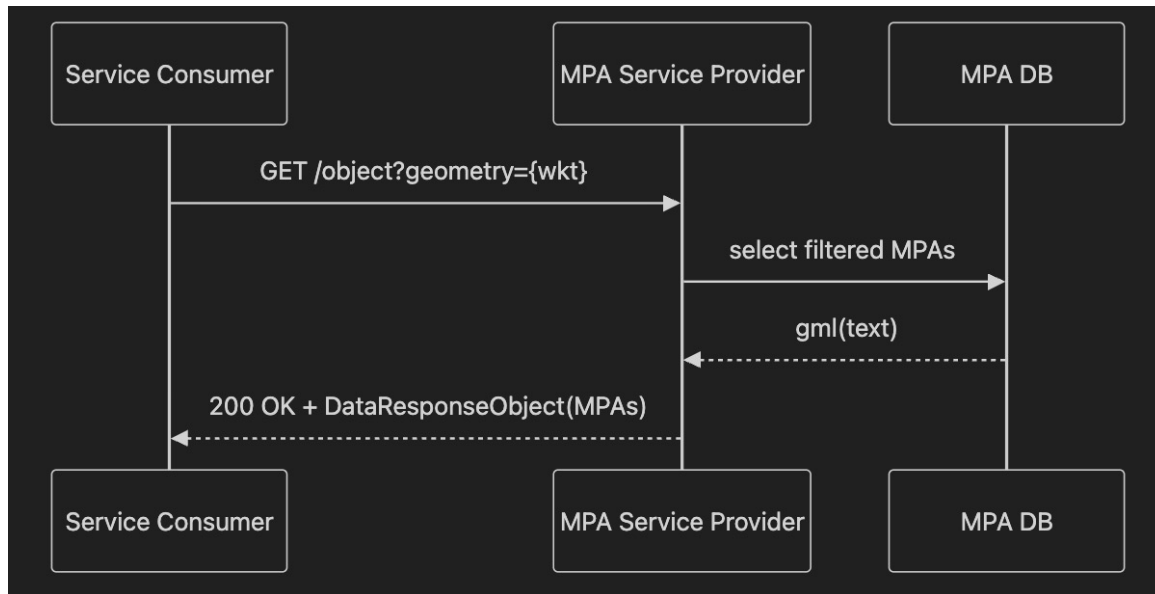
# Designing SECOM-based MPA Service Architecture



mpa	
 <b>geometry</b>	geometry(polygon,4326)
 <b>dataset_name</b>	varchar(50)
 <b>gml</b>	text
 <b>country_code</b>	char(3)
 <b>id</b>	uuid

# Sequence Diagram : Get

By Geometry- Returns all MPAs whose geometries intersect with the requested area (geometry={WKT})

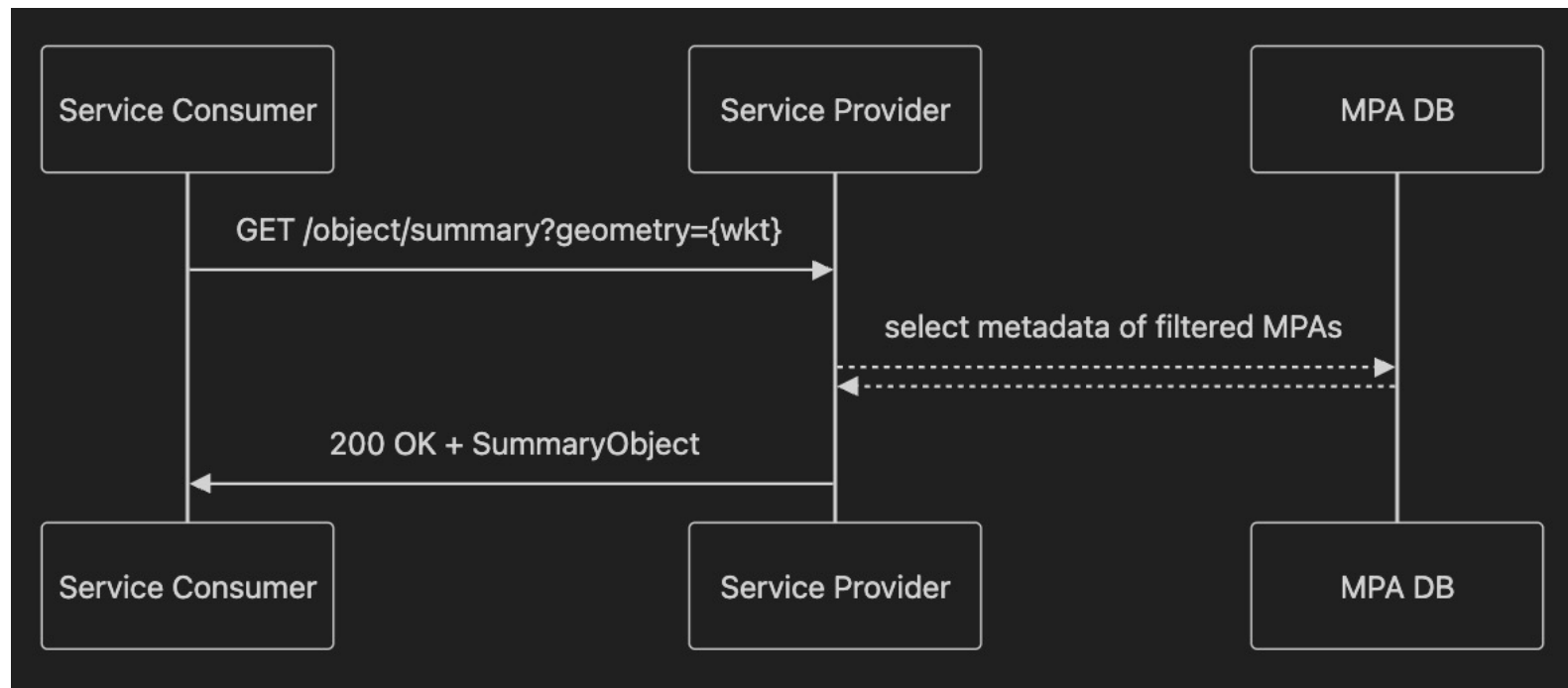


By dataReference - Returns one specific MPA identified by its unique ID (dataReference={UUID})



# Sequence Diagram : Get Summary

By Geometry - Retrieves summary information (metadata) of all MPAs whose geometries intersect with the given area (geometry={WKT}).



# SECOM Component Mapping – GetFilterObject

GetFilterObject				
Attribute	Mult	Processing	Type	Definition
dataReference	0..1	Mandatory	UUID	Reference to information object, e.g. from Get Summary
containerType	0..1	Mandatory	ContainerTypeEnum	Container type requested, see Table 7
dataProductType	0..1	Mandatory	SECOM_DataProduct Type.Name	The name column of SECOM_DataProductType in Table 8
productVersion	0..1	Optional	CharacterString	S-100 based Product type version, e.g. 1.0.0
geometry	0..1	Optional	CharacterString	Geometry condition for geo-located information objects, see Table 12
unlocode	0..1	Optional	CharacterString	Code of defined object see 5.6.13
validFrom	0..1	Optional	DateTime	Valid from time
validTo	0..1	Optional	DateTime	Valid until time
page	0..1	Mandatory	PositiveInteger	Requested pagination page
pageSize	0..1	Mandatory	PositiveInteger	Requested pagination page size

- dataReference: single mpa
- containerType: dataset only
- dataProductType: S-122 only
- unlocode: not supported
- validFrom, validTo: not supported
- page: default 0
- pageSize: 0~50, default 20

# Geometry Field

Supported WKT geometric objects		
Object	Definition	Example
Point	A zero-dimensional geometry that represents a single location in coordinate space on a map. The (x,y) coordinate is represented in order (longitude, latitude) and expressed as real value, degree/degree decimal format.	POINT (30 10)
LineString	A curve with linear interpolation between points. Each consecutive pair of points defines a line segment.	LINestring (30 10, 10 30, 40 40)
Polygon	A planar surface, i.e. a two-dimensional geometry object, defined by 1 exterior boundary and 0 or more interior boundaries. Each interior boundary defines a hole in the polygon.	POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
MultiPoint	An array of point objects	MULTIPOINT ((10 40), (40 30), (20 20), (30 10))
MultiLineString	An array of LineString objects	MULTILINestring ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon	An array of polygon objects	MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))
GeometryCollection	A collection of 1 or more geometry objects	GEOMETRYCOLLECTION (POINT (40 10), LINestring (10 10, 20 20, 10 40), POLYGON ((40 40, 20 45, 45 30, 40 40)))

Defines the spatial filter for MPA data

- Used to select MPAs whose geometries **intersect** with the given area
- Accepts geometric objects in **WKT format**

# SECOM Component Mapping – GetSummaryFilterObject

GetSummaryFilterObject				
Attribute	Mult	Processing	Type	Definition
containerType	0..1	Mandatory	ContainerTypeEnum	Container type requested, Table 7
dataProductType	0..1	Mandatory	SECOM_DataProductType.Name	The name column of SECOM_DataProductType in Table 8
productVersion	0..1	Optional	CharacterString	S-100 based Product type version, e.g. 1.0.0
geometry	0..1	Optional	CharacterString	Geometry condition for geo-located information objects, see Table 12
unlocode	0..1	Optional	CharacterString	Code of defined object, see 5.6.13
validFrom	0..1	Optional	DateTime	Valid from time
validTo	0..1	Optional	DateTime	Valid until time
page	0..1	Mandatory	PositiveInteger	Requested pagination page
pageSize	0..1	Mandatory	PositiveInteger	Requested pagination page size

- containerType: dataset only
- dataProductType: S-122 only
- unlocode: not supported
- validFrom, validTo: not supported
- page: default 0
- pageSize: 0~500, default 100

# Implementing SECOM Services using SECOMLib

- <https://github.com/gla-rad/SECOMLib>

```
repositories {
    mavenCentral()

    maven { MavenArtifactRepository it ->
        name = "grad"
        url = uri("https://rnavlab.gla-rad.org/mvn/repository/grad/")
    }

    maven { MavenArtifactRepository it ->
        name = "grad-snapshots"
        url = uri("https://rnavlab.gla-rad.org/mvn/repository/grad-snapshots/")
        mavenContent { MavenRepositoryContentDescript... it ->
            snapshotsOnly()
        }
    }
}
```

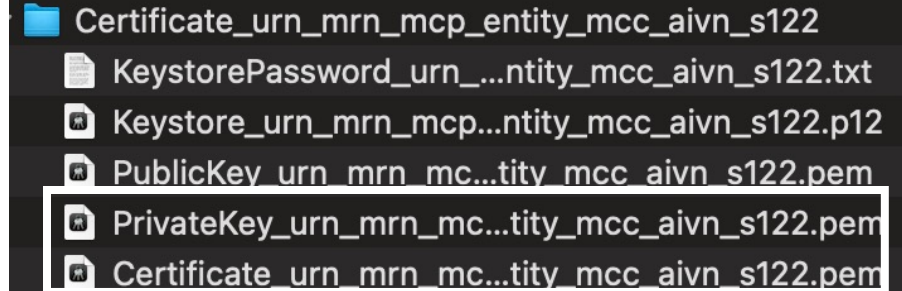
**SECOMLib** is a library that automatically handles **data signing, exchange metadata generation, Base64 encoding, signature verification, etc.** that is required for SECOM communication.

all you need to do is **implement the provider interfaces**

## Service Interface Implementation – Provider setup

- |                                     |   |
|-------------------------------------|---|
| <b>SecomSignatureProviderImpl</b>   | Generates and validates digital signatures  |
| <b>SecomCertificateProviderImpl</b> | Supplies the certificate chain and public key for signing operations                    |
| <b>SecomTrustStoreProviderImpl</b>  | Provides access to the CA root certificate and trusted certificates from the truststore |
- ~~SecomEncryptionProvider~~
  - ~~SecomCompressionProvider~~

# Keystore & TrustStore setup



## Keystore

- `openssl pkcs12 -export -in {Certificate~~.pem} -inkey {PrivateKey~~.pem} -name {privateKeyEntry alias} -out {keystore.p12} -passout pass:{password}`

## Truststore

- `keytool -importcert -alias {trustedCertificateEntry alias} -file {root CA / intermediate cert.pem} -keystore {truststore.p12} -storetype {PKCS12} -storepass {password}`

# Provider setup (1)

application.yml

```
secom:  
  security:  
    ssl:  
      keystore: classpath:certificate/keystore.p12  
      keystore-type: pkcs12  
      keystore-password:  
      keystore-alias: mpa  
      truststore: classpath:certificate/truststore.p12  
      truststore-type: pkcs12  
      truststore-password:
```



SecomConfigProperties

```
@ConfigurationProperties(prefix = "secom.security.ssl")  
public class SecomConfigProperties {  
  
    // Configuration Variables  
    private String keystore;  
    private String keystoreType;  
    private String keystorePassword;  
    private String truststore;  
    private String truststoreType;  
    private String truststorePassword;  
    private Boolean insecureSslPolicy;  
    private Integer clientMaxMemorySize;  
}
```

## Provider setup (2)

**SecomSignatureProviderImpl** : override generateSignature() & validateSignature

```
public class SecomSignatureProviderImpl implements SecomSignatureProvider {
    private final SecomConfigProperties config;

    @Override 4개 사용 위치 & Changyun Lee
    public DigitalSignatureAlgorithmEnum getSignatureAlgorithm() {...}

    @Override 2개 사용 위치 & Changyun Lee *
    public byte[] generateSignature(DigitalSignatureCertificate signatureCertificate, DigitalSignatureAlgorithmEnum algorithm, byte[] payload) {
        try {
            Signature sign = Signature.getInstance(algorithm.getValue());

            KeyStore keyStore = KeyStoreUtils.getKeyStore(config.getKeyStore(), config.getKeyStorePassword(), config.getKeyStoreType());

            PrivateKey privateKey = (PrivateKey) keyStore.getKey(keyStoreAlias, config.getKeyStorePassword().toCharArray());

            sign.initSign(privateKey);
            sign.update(payload);

            return sign.sign();
        } catch (Exception e) {...}
    }

    @Override 2개 사용 위치 & Changyun Lee *
    public boolean validateSignature(String signatureCertificate, DigitalSignatureAlgorithmEnum algorithm, byte[] signature, byte[] content) {
        try {
            Signature sign = Signature.getInstance(algorithm.getValue());
            sign.initVerify(SecomPemUtils.getCertFromPem(signatureCertificate));
            sign.update(content);

            return sign.verify(signature);
        } catch (NoSuchAlgorithmException | CertificateException | SignatureException | InvalidKeyException ex) {...}
    }
}
```

# Provider setup (3)

## SecomCertificateProviderImpl

override  
getDigitalSignatureCertificate()

```
@Slf4j @Component
@RequiredArgsConstructor
public class SecomCertificateProviderImpl implements SecomCertificateProvider {

    @Value("mpa")
    private String keystoreAlias;

    private final SecomConfigProperties config;
    private final SecomTrustStoreProvider trustStoreProvider;

    @Override 2개 사용 위치 @Changyun Lee *
    public DigitalSignatureCertificate getDigitalSignatureCertificate() {
        DigitalSignatureCertificate signature = new DigitalSignatureCertificate();
        try {
            KeyStore keyStore = KeyStoreUtils.getKeyStore(config.getKeyStore(), config.getKeyStorePassword(), config.getKeyStorePassword());
            KeyStore trustStore = trustStoreProvider.getTrustStore();

            Certificate cert = keyStore.getCertificate(keystoreAlias);
            Certificate rootCert = trustStore.getCertificate(trustStoreProvider.getCARootCertificateAlias());

            PublicKey publicKey = cert.getPublicKey();
            signature.setCertificate((X509Certificate) cert);
            signature.setCertificateAlias(keystoreAlias);
            signature.setRootCertificate((X509Certificate) rootCert);
            signature.setPublicKey(publicKey);
        } catch (Exception e) {...}
        return signature;
    }
}
```

# Provider setup (4)

## SecomTrustStoreProviderImpl

Override getTrustStore()

```
@Component  👤 Changyun Lee *  
@RequiredArgsConstructor  
public class SecomTrustStoreProviderImpl implements SecomTrustStoreProvider {  
  
    private final SecomConfigProperties config;  
  
    @Override  3개 사용 위치 👤 Changyun Lee *  
    public KeyStore getTrustStore() {  
        try {  
            return KeyStoreUtils.getKeyStore(config.getTruststore(), config.getTruststorePassword(), config.getTruststoreType());  
        } catch (Exception ex) {...}  
    }  
}
```

# SECOM Communication Channel Security - Mutual TLS Setup

## Spring Boot (application level)

```
server:
  ssl:
    keystore: classpath:certificate/keystore.p12
    keystore-type: pkcs12
    keystore-password:
    truststore: classpath:certificate/truststore.p12
    truststore-type: pkcs12
    truststore-password:
    client-auth: need
    signature-algorithm: SHA384withECDSA
```

## NGINX reverse proxy level

```
server {
  listen 443 ssl;
  server_name mpa.aivn-dev.com;

  ssl_certificate /etc/nginx/certs/server.crt; # keystore
  ssl_certificate_key /etc/nginx/certs/server.key;

  ssl_client_certificate /etc/nginx/certs/ca.crt; # truststore
  ssl_verify_client on;

  location / {
    proxy_pass https://127.0.0.1:8080;
  }
}
```

## K8s Ingress annotations

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: mpa-ingress
  annotations:
    kubernetes.io/ingress.class: "nginx"
    nginx.ingress.kubernetes.io/auth-tls-secret: "default/mpa-ca-secret" # truststore
    nginx.ingress.kubernetes.io/auth-tls-verify-client: "on"
spec:
  tls:
    - hosts:
      - mpa.aivn-dev.com
      secretName: mpa-server-cert-secret # keystore
  rules:
    - host: mpa.aivn-dev.com
      http:
        paths:
```

# SECOM Communication Channel Security – OCSP validation

## JSSE(Java Secure Socket Extension) OCSP Check

Built-in validation in JVM (during TLS handshake)

Performing OCSP validation during the TLS handshake is the **standard approach**

,  
But it runs synchronously inside the JVM

```
static {  
    System.setProperty("com.sun.net.ssl.checkRevocation", "true"); // Enables certificate revocation checking  
    java.security.Security.setProperty("ocsp.enable", "true"); // Enables OCSP as the primary revocation check method  
    System.setProperty("com.sun.security.enableCRLDP", "true"); // fallback  
}
```

## OCSP Stapling (NGINX)

Server caches OCSP response and “staples” it into handshake

## Custom OCSP Client

Post-handshake validation via code

# Service Interface Implementation

Layer	Role / Responsibility	Key Implementation Details
<b>Controller</b>	Handles SECOM requests and implements the SecomInterfaces	Implements SECOM standard interface → builds FilterObject and Pageable → delegates to Service.
<b>Service</b>	Creates and returns SECOM-compliant response objects (GetResponseObject, DataResponseObject, etc.).	Uses repository results to populate SECOM model objects and adds pagination handling and validation.
<b>Repository</b>	Data access layer interacting with DB. Executes spatial queries based on FilterObject	applies pagination and spatial conditions dynamically.

# Service Interface Implementation – Ping

PingResponseObject				
Attribute	Mult	Processing	Type	Definition
lastPrivateInteractionTime	0..1	Optional	DateTime	Describes the time for last interaction with end user application

```

@Component  @Changyun Lee +1
@RequiredArgsConstructor
public class LastInteractionRequestListener implements ServletRequestListener {

    private final CacheManager cacheManager;

    @Override  0개의 사용위치  @Changyun Lee +1
    public void requestDestroyed(ServletRequestEvent servletRequestEvent) {
        Objects.requireNonNull(cacheManager.getCache("lastInteractionCache"))
            .put("lastInteractionTime", LocalDateTime.now(ZoneOffset.UTC));
    }
}

```

# Service Interface Implementation – Capability

```

public class CapabilitySecomController implements CapabilitySecomInterface {

    private final ApplicationContext applicationContext;

    @Override
    public CapabilityResponseObject capability() {
        CapabilityObject capabilityObject = new CapabilityObject();
        capabilityObject.setServiceVersion("0.1");
        capabilityObject.setImplementedInterfaces(getImplementedInterfaces());
        capabilityObject.setDataProductType(SECOM_DataProductType.S122);
        capabilityObject.setContainerType(ContainerTypeEnum.S100_DataSet);

        CapabilityResponseObject capabilityResponseObject = new CapabilityResponseObject();
        capabilityResponseObject.setCapability(List.of(capabilityObject));

        return capabilityResponseObject;
    }

    private ImplementedInterfaces getImplementedInterfaces() {
        ImplementedInterfaces interfaces = new ImplementedInterfaces();

        interfaces.setUpload(!applicationContext.getBeansOfType(UploadSecomInterface.class).isEmpty());
        interfaces.setUploadLink(!applicationContext.getBeansOfType(UploadLinkSecomInterface.class).isEmpty());
        interfaces.setGet(!applicationContext.getBeansOfType(GetSecomInterface.class).isEmpty());
        interfaces.setGetSummary(!applicationContext.getBeansOfType(GetSummarySecomInterface.class).isEmpty());
        interfaces.setGetByLink(!applicationContext.getBeansOfType(GetByLinkSecomInterface.class).isEmpty());
        interfaces.setSubscription(!applicationContext.getBeansOfType(SubscriptionSecomInterface.class).isEmpty());
        interfaces.setAccess(!applicationContext.getBeansOfType(AccessSecomInterface.class).isEmpty());
        interfaces.setEncryptionKey(!applicationContext.getBeansOfType(EncryptionKeySecomInterface.class).isEmpty());

        return interfaces;
    }
}

```

Checks which SECOM interfaces are implemented as Spring Beans in the current application context

CapabilityResponseObject				
Attribute	Mult	Processing	Type	Definition
capability	1..*	Optional	CapabilityObject	List of capability objects

CapabilityObject				
Attribute	Mult	Processing	Type	Definition
containerType	1	Optional	ContainerTypeEnum	The container type of the data attribute, see Table 7
dataProductType	1	Optional	SECOM_DataProductType.Name	The name column of SECOM_DataProductType in Table 8
productSchemaUrl	1	Optional	URL	The schemaUrl attribute of the S100 Product
implementedInterfaces	1	Optional	ImplementedInterfaces	List of service interfaces implemented with the payload (product) in focus
serviceVersion	1	Optional	CharacterString	Version of service instance, for example New Editions denoted as n.0.0 Revisions denoted as n.n.0 Clarifications denoted as n.n.n

ImplementedInterfaces				
Attribute	Mult	Processing	Type	Definition
upload	1	Optional	Boolean	True if service interface is implemented
uploadLink	1	Optional	Boolean	True if service interface is implemented
get	1	Optional	Boolean	True if service interface is implemented
getByLink	1	Optional	Boolean	True if service interface is implemented
getSummary	1	Optional	Boolean	True if service interface is implemented
subscription	1	Optional	Boolean	True if service interface is implemented
access	1	Optional	Boolean	True if service interface is implemented
encryptionKey	1	Optional	Boolean	True if service interface is implemented



# Service Interface Implementation – Get Summary interface

SummaryObject				
Attribute	Mult	Processing	Type	Definition
dataReference	1	Mandatory	UUID	Reference to data
dataProtection	1	Optional	Boolean	Flag indicating if data is encrypted or not
dataCompression	1	Optional	Boolean	Flag indicating if data is compressed or not
containerType	1	Optional	ContainerTypeEnum	Container type, see Table 7
dataProductType	1	Optional	SECOM_DataProductType.Name	The name column of SECOM_DataProductType in Table 8
info_identifier	0..1	Optional	CharacterString	Identifier of the information object, e.g. <S100XC:identifier>, gml:id
info_name	0..1	Optional	CharacterString	Name of the information object, e.g. <S100XC:exchangeCatalogueName>, <S100:datasetFileIdentifier>
info_status	0..1	Optional	CharacterString	Status of the information object, e.g. active, inactive
info_description	0..1	Optional	CharacterString	Description of the information object, e.g. <S100XC:description>, <S100:datasetTitle>
info_lastModifiedDate	0..1	Optional	DateTime	Date for last modified, e.g. <S100XC:date>, <S100:datasetReferenceDate>
info_productVersion	0..1	Optional	CharacterString	S-100 based Product type version, e.g. 1.0.0
info_size	0..1	Optional	PositiveInteger	Size of the data in kB expressed as unsigned long or unsigned int64 with max value = 2^64

- **dataReference:** id of MPA
- dataProtection, dataCompression: false
- containerType: dataset
- dataProductType: S-122
- **Info\_identifier:** dataset's gml:id
- **Info\_size:** size of gml in kb (integer)

```

{
  "summaryObject": [
    {
      "dataReference": "1814f335-f177-34f4-b9b5-e400531954ab",
      "dataProtection": false,
      "dataCompression": false,
      "containerType": 0,
      "dataProductType": "S122",
      "info_identifier": "AIMPA20250306_AIAG4_1_0_0",
      "info_name": null,
      "info_status": null,
      "info_description": null,
      "info_lastModifiedDate": null,
      "info_productVersion": null,
      "info_size": 5
    },
    {
      "dataReference": "5cc56799-6e74-3832-9b47-dc7e7f126536",

```

# MPA Insertion Pipeline (Spring Batch)

## Reader:

- Reads multiple .gml files
- StaxEventItemReader (with **JAXB**) parses XML into java objects

```

@XmlRootElement(name = "Dataset", namespace = "http://www.iho.int/S122/gml/1.0") 14개 사용 위치
@XmlAccessorType(XmlAccessType.FIELD)
@NoArgsConstructor
@Getter
public class Dataset {

    @XmlAttribute(name = "id", namespace = "http://www.opengis.net/gml/3.2")
    private String datasetId;

    @XmlElement(name = "member")
    private List<Member> member;

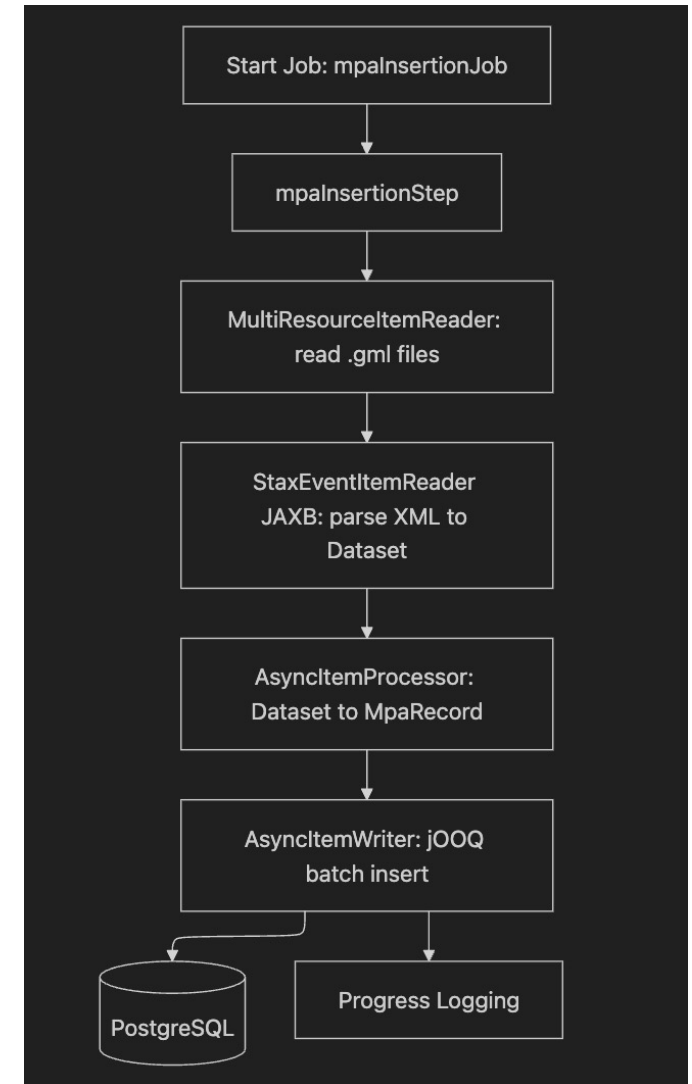
    @Getter @XmlAttribute(XmlAccessType.FIELD) 1개 사용 위치
    public static class Member {
        @XmlElement(name = "MarineProtectedArea", namespace = "http://www.iho.int/S122/gml/1.0")
        private MarineProtectedArea mpa;
    }

    @Getter @XmlAttribute(XmlAccessType.FIELD) 1개 사용 위치
    public static class MarineProtectedArea {
        @XmlElement(name = "geometry")
        private Geometry geometry;
    }

    @XmlAttribute(XmlAccessType.FIELD)
    @Getter
    public static class Geometry {
        @XmlElement(name = "surfaceProperty", namespace = "http://www.iho.int/s100gml/1.0")
        private SurfaceProperty surface;
    }

    @XmlAttribute(XmlAccessType.FIELD) 1개 사용 위치
    @Getter
    public static class SurfaceProperty {
        @XmlElement(name = "Polygon", namespace = "http://www.opengis.net/gml/3.2")
    }

```



# MPA Insertion Pipeline (Spring Batch)

## Processor:

- Converts each objects to MpaRecord (DTO)
- Extracts country code from file name
- Builds **WKT polygon geometry** from XML's posList (polygon attribute)

## Writer:

- Uses AsyncItemWriter with **jOOQ batch insert**
- Persists data into **Database**

## 💡 Future Usage

A **monthly MPA update** using this batch configuration

- automatically loading the latest dataset from the **Protected Seas**

```
s122.config.BatchConfig : Progress: saved 49000/49389 files (99.21%)
s122.config.BatchConfig : Progress: saved 49250/49389 files (99.72%)
s122.config.BatchConfig : Progress: saved 49389/49389 files (100.00%)
step.AbstractStep       : Step: [mpaInsertionStep] executed in 2m20s126ms
```

# Spatial Query - PostGIS

```
@Override 1개 사용 위치 yun-aivn *
public Page<String> findGmlByFilter(FilterObject filter, Pageable pageable) {
    Field<Integer> totalCount = DSL.count().over().as( alias: "total_count");

    Condition where = buildSpatialWhere(filter.geometry());
    var records = dsl.select(MPA.GML, totalCount) SelectSelectStep<Record2<...>>
        .from(MPA) SelectJoinStep<Record2<...>>
        .where(where) SelectConditionStep<Record2<...>>
        .orderBy(MPA.ID.asc()) SelectSeekStep1<Record2<...>, UUID>
        .limit(pageable.getPageSize()) SelectLimitPercentStep<Record2<...>>
        .offset((int) pageable.getOffset()) SelectForUpdateStep<Record2<...>>
        .fetch();

    long total = records.isEmpty() ? 0L : records.getFirst().get(totalCount);
    List<String> content = records.map( Record2<String, Integer> r -> r.get(MPA.GML));

    return new PageImpl<>(content, pageable, total);
}
```

SQL generated by jOOQ

```
SELECT
    mpa.gml,
    COUNT(*) OVER () AS total_count
FROM mpa
WHERE ST_Intersects( geog1 mpa.geometry, geog2 ST_GeomFromText(wkt, 4326))
ORDER BY mpa.id
LIMIT pageSize OFFSET page;
```

```
CREATE INDEX idx_mpa_geometry_gist ON mpa USING GIST (geometry);
```

```
Index Scan using idx_mpa_geometry_gist on mpa (cost=0.28..15.00 rows=1 width=18) (actual time=0.091..0.164 rows=5 loo...
Index Cond: (geometry && '0103000020E610000001000000050000003333333333F34EC09A99999999931403333333333F34EC0333333333...
Filter: st_intersects(geometry, '0103000020E610000001000000050000003333333333F34EC09A99999999931403333333333F34EC03...
Buffers: shared hit=15
Planning Time: 0.249 ms
Execution Time: 0.187 ms
```

# Quick test (1)

## Test Geometry :

Country : AG(Antigua & Barbuda)

POLYGON((-61.9000 17.6000, -61.9000 17.7000, -61.8000 17.7000, -61.8000 17.6000, -61.9000 17.6000))

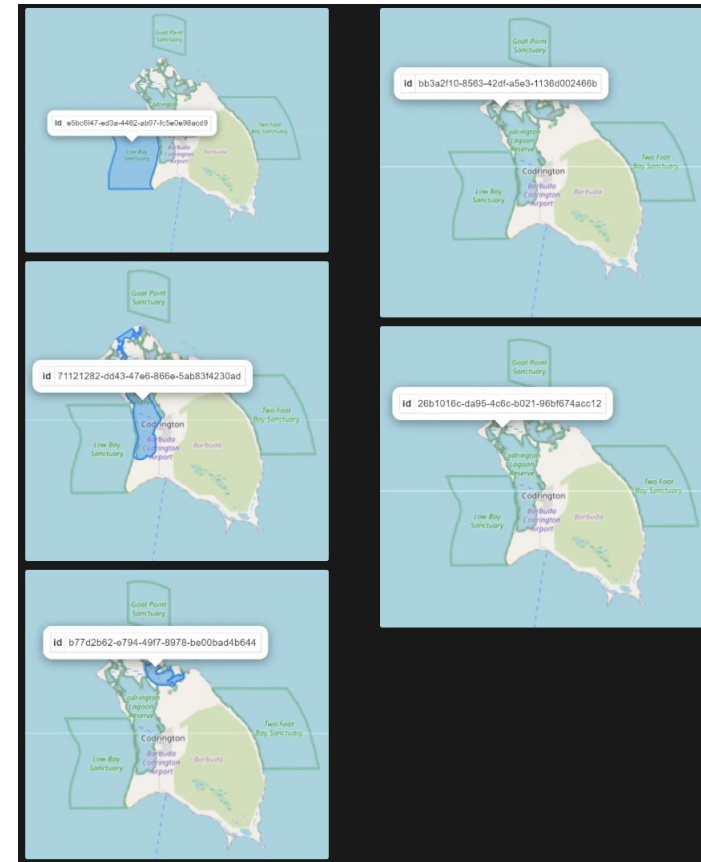


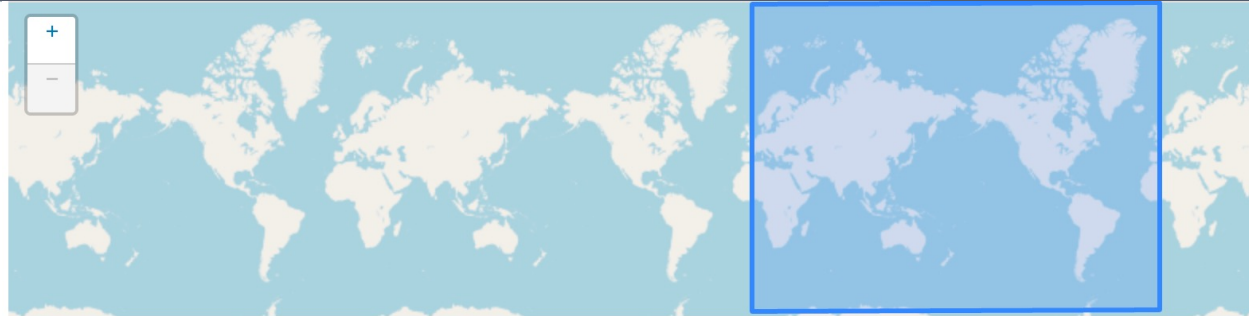
# Quick test (2)

## Test Geometry :

Country : AG(Antigua & Barbuda)

POLYGON((-61.9000 17.6000, -61.9000 17.7000, -61.8000 17.7000, -61.8000 17.6000, -61.9000 17.6000))





<b>Name</b>	AIVN MPA Service
<b>Version</b>	0.1
<b>Service type</b>	<input type="radio"/> MS 11 - Nautical chart service <input checked="" type="radio"/> MS 12 - Nautical publications service
<b>Data product type</b>	<input checked="" type="radio"/> S-122 Marine Protected Areas (MPAs)
<b>Status</b>	PROVISIONAL
<b>Endpoint URI</b>	<a href="https://mpa.aivn-dev.com/api/secom">https://mpa.aivn-dev.com/api/secom</a>
<b>Organization ID</b>	urn:mrn:mcp:org:mcc:aivn
<b>Keywords</b>	<input checked="" type="radio"/> S-122 <input checked="" type="radio"/> MPA
<b>Instance ID</b>	urn:mrn:mcp:entity:dmc:aivn:s122
<b>Technical design ID</b>	
<b>Technical design version</b>	
<b>Comment</b>	Prototype SECOM-service for MPAs - certificate required

**Thank You**