# Impact Matrix: e-Postcode SOAP vs. Experian REST API Integration

This matrix provides a comparison of key differences between the existing \*\*e-Postcode SOAP service\*\* and the new \*\*Experian REST API\*\*, highlighting their impact on your \*\*Spring Boot microservice\*\* and integration via \*\*IBM API Connect\*\*.

|  |  |  |  |
| --- | --- | --- | --- |
| Aspect | e-Postcode SOAP API | Experian REST API | Impact & Required Actions |
| API Type | SOAP (XML-based) | REST (JSON-based) | ✅ Convert SOAP XML to JSON where necessary. |
| Data Format | XML | JSON | ✅ Implement data transformation logic in Spring Boot. |
| Integration Protocol | SOAP over HTTP | REST over HTTPS | ✅ Update the API client in Spring Boot (RestTemplate/WebClient/Feign). |
| Authentication | Basic Auth / WS-Security | API Key / OAuth 2.0 | ✅ Update authentication mechanisms. |
| Response Structure | Complex, nested XML | Flat JSON with optional nested fields | ✅ Update response parsing logic in microservice. |
| Request Structure | Requires SOAP envelope | Simple HTTP request with query parameters | ✅ Modify request formation logic. |
| Data Fields Returned | Address details (basic fields) | Address details (enhanced with metadata) | ✅ Update database schema if needed. |
| Performance | Typically slower due to XML processing | Faster due to lightweight JSON | ✅ Performance benchmarking needed before deployment. |
| Rate Limits | Not applicable or minimal restrictions | API rate limits may apply | ✅ Implement rate limiting and caching. |
| Fault Handling | SOAP Faults (Detailed error response) | HTTP Status Codes + JSON error messages | ✅ Update exception handling and logging. |
| Security Compliance | Legacy security controls | Modern security (OAuth 2.0, encrypted payloads) | ✅ Implement OAuth-based token management in IBM APIC. |
| IBM API Connect Changes | Minimal configuration | Requires endpoint, security, and caching setup | ✅ Configure IBM API Connect policies for security and logging. |
| Deployment Complexity | Moderate, existing integration | High, due to new authentication & response handling | ✅ Plan phased rollout and testing. |
| Monitoring & Logging | Basic logs | Advanced logging and monitoring available | ✅ Enable logging for request tracking and failures. |

## Action Plan for Integration

### Develop Data Mapping Strategy

* ✅ Map XML response from \*\*e-Postcode\*\* to JSON format of \*\*Experian REST API\*\*.
* ✅ Ensure all required fields are available in Experian API.

### Update Spring Boot Microservice

* ✅ Implement REST client (FeignClient, WebClient, or RestTemplate).
* ✅ Modify request and response handling.
* ✅ Implement caching for frequent lookups.

### IBM API Connect Configuration

* ✅ Configure \*\*endpoint security policies\*\* (API Key or OAuth).
* ✅ Implement \*\*logging and monitoring\*\* for API usage.
* ✅ Apply \*\*rate limiting\*\* if required by Experian.

### Testing & Validation

* ✅ Perform \*\*unit testing\*\* for API calls.
* ✅ Conduct \*\*integration testing\*\* with IBM API Connect.
* ✅ Validate \*\*response times and error handling\*\*.

### Deploy & Monitor

* ✅ Deploy changes to \*\*staging environment\*\* first.
* ✅ Monitor API performance and error logs.
* ✅ Roll out to production gradually.

### ****Impact Assessment for Integrating Experian REST API with Spring Boot via IBM API Connect****

#### ****1. Overview****

The integration of the **Experian REST API** with the existing **Spring Boot microservice** via **IBM API Connect (APIC)** requires an impact assessment to evaluate changes, dependencies, and potential risks. This document outlines the key considerations and areas affected by this integration.

#### ****2. Key Areas of Impact****

##### **2.1. Functional Impact**

* The new integration will **replace** or **augment** the existing **e-Postcode SOAP service** for address retrieval.
* API response structure differences between **SOAP (e-Postcode)** and **REST (Experian)** need to be handled in the application layer.
* Potential **data format mismatches** (e.g., XML vs. JSON) may require transformation logic.
* Experian API may provide **additional address metadata** that was not available in e-Postcode, requiring updates in the database or service layer.

##### **2.2. Architectural Impact**

* **IBM API Connect** will serve as the API gateway, managing authentication, rate limiting, and security.
* Existing **Spring Boot microservice** will need to integrate a new **REST client** (e.g., RestTemplate, WebClient, Feign Client) to communicate with Experian API.
* **Configuration changes**: API base URLs, authentication mechanisms, request parameters, and headers need to be configured dynamically.
* **Logging and monitoring**: Additional logs should be enabled to track API usage and performance.

##### **2.3. Data Impact**

* **Database schema changes** may be required if the Experian API returns additional address attributes that should be persisted.
* **Data mapping** between e-Postcode and Experian API responses must be defined.
* **Handling null or missing values**: Experian’s API might return different response formats based on input, which requires fallback logic.

##### **2.4. Performance & Scalability Impact**

* **Rate limits** imposed by Experian API should be considered to prevent throttling.
* **Response times** of Experian API should be assessed against current system performance SLAs.
* Potential need for **caching address responses** to reduce repeated external API calls.

##### **2.5. Security & Compliance Impact**

* **Authentication mechanism**: API Key, OAuth, or mutual TLS needs to be implemented securely.
* **Encryption**: Ensure address data is handled securely in transit and at rest.
* **PII Data Handling**: Ensure compliance with **GDPR, CCPA**, and other data protection regulations.
* **IBM APIC security policies**: Validate the security policies and access control mechanisms applied to the API gateway.

##### **2.6. Operational Impact**

* **Deployment pipeline changes**: CI/CD pipelines must be updated to include the new API dependency.
* **Service monitoring**: Configure **API response monitoring and failure alerts** in IBM API Connect.
* **Testing Strategy**: Unit tests, integration tests, and performance tests need to be updated.

#### ****3. Next Steps****

✅ Compare e-Postcode SOAP vs. Experian REST API response structures.  
✅ Define **data transformation logic** between old and new address formats.  
✅ Implement **rate-limiting mechanisms** to handle Experian API restrictions.  
✅ Validate **security policies** in IBM APIC for Experian API integration.  
✅ Conduct **end-to-end testing** before deployment.

Would you like a **detailed impact matrix** comparing old vs. new behavior? 🚀

##### You said:

YES

###### ChatGPT said:

ChatGPT

### ****Impact Matrix: e-Postcode SOAP vs. Experian REST API Integration****

This matrix provides a comparison of key differences between the existing **e-Postcode SOAP service** and the new **Experian REST API**, highlighting their impact on your **Spring Boot microservice** and integration via **IBM API Connect**.

| **Aspect** | **e-Postcode SOAP API** | **Experian REST API** | **Impact & Required Actions** |
| --- | --- | --- | --- |
| **API Type** | SOAP (XML-based) | REST (JSON-based) | ✅ Convert SOAP XML to JSON where necessary. |
| **Data Format** | XML | JSON | ✅ Implement data transformation logic in Spring Boot. |
| **Integration Protocol** | SOAP over HTTP | REST over HTTPS | ✅ Update the API client in Spring Boot (RestTemplate/WebClient/Feign). |
| **Authentication** | Basic Auth / WS-Security | API Key / OAuth 2.0 | ✅ Update authentication mechanisms. |
| **Response Structure** | Complex, nested XML | Flat JSON with optional nested fields | ✅ Update response parsing logic in microservice. |
| **Request Structure** | Requires SOAP envelope | Simple HTTP request with query parameters | ✅ Modify request formation logic. |
| **Data Fields Returned** | Address details (basic fields) | Address details (enhanced with metadata) | ✅ Update database schema if needed. |
| **Performance** | Typically slower due to XML processing | Faster due to lightweight JSON | ✅ Performance benchmarking needed before deployment. |
| **Rate Limits** | Not applicable or minimal restrictions | API rate limits may apply | ✅ Implement rate limiting and caching. |
| **Fault Handling** | SOAP Faults (Detailed error response) | HTTP Status Codes + JSON error messages | ✅ Update exception handling and logging. |
| **Security Compliance** | Legacy security controls | Modern security (OAuth 2.0, encrypted payloads) | ✅ Implement OAuth-based token management in IBM APIC. |
| **IBM API Connect Changes** | Minimal configuration | Requires endpoint, security, and caching setup | ✅ Configure IBM API Connect policies for security and logging. |
| **Deployment Complexity** | Moderate, existing integration | High, due to new authentication & response handling | ✅ Plan phased rollout and testing. |
| **Monitoring & Logging** | Basic logs | Advanced logging and monitoring available | ✅ Enable logging for request tracking and failures. |

### ****Action Plan for Integration****

1. **Develop Data Mapping Strategy**:
   * Map XML response from **e-Postcode** to JSON format of **Experian REST API**.
   * Ensure all required fields are available in Experian API.
2. **Update Spring Boot Microservice**:
   * Implement REST client (FeignClient, WebClient, or RestTemplate).
   * Modify request and response handling.
   * Implement caching for frequent lookups.
3. **IBM API Connect Configuration**:
   * Configure **endpoint security policies** (API Key or OAuth).
   * Implement **logging and monitoring** for API usage.
   * Apply **rate limiting** if required by Experian.
4. **Testing & Validation**:
   * Perform **unit testing** for API calls.
   * Conduct **integration testing** with IBM API Connect.
   * Validate **response times and error handling**.
5. **Deploy & Monitor**:
   * Deploy changes to **staging environment** first.
   * Monitor API performance and error logs.
   * Roll out to production gradually.