**Technical Implementation Document: Country-Based Access Control Solution**

**1. Overview**

This document outlines the technical implementation of a country-based access control system using **Attribute-Based Access Control (ABAC)**, **ServiceNow (SNOW) Approval Workflows**, and a **Dynamic Country List Service**. The goal is to restrict access to sensitive data based on geographical entitlements while maintaining compliance and automation.

**2. System Architecture**

**2.1 Key Components**

* **Authentication Layer**: Single Sign-On (SSO) via OAuth2, SAML, or Active Directory.
* **Access Control Layer**:
  + ABAC Engine (Open Policy Agent / XACML-based policies)
  + Dynamic Country List Service (Database-driven updates)
* **Approval & Workflow Layer**:
  + ServiceNow (SNOW) for access approval requests
* **API Gateway & Middleware**:
  + Secure request validation before granting access
* **Audit & Compliance Layer**:
  + Logging, Monitoring, and Reporting

**2.2 Interaction Flow**

1. **User requests access** to a restricted country’s data.
2. **API Gateway intercepts the request** and forwards it to the **ABAC Engine**.
3. **ABAC Engine checks policies**:
   * Does the user have a global entitlement?
   * Is the requested country restricted?
   * Does the user have an approved SNOW request?
4. **If approved**, access is granted; otherwise, the request is denied with an option to raise a **SNOW ticket**.
5. **For new country restrictions**, the **Dynamic Country List Service** updates the ABAC policy automatically.

**3. Implementation Details**

**3.1 Database Schema**

**User Access Table**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| user\_id | UUID | Unique User Identifier |
| role | STRING | User Role (Admin, Employee, etc.) |
| global\_access | BOOLEAN | True if user has global entitlement |
| approved\_countries | ARRAY | List of countries user is allowed to access |

**Restricted Countries Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| country\_code | STRING | ISO 3166-1 Alpha-2 Country Code |
| restricted | BOOLEAN | True if country is restricted |
| last\_updated | TIMESTAMP | Last update timestamp |

**SNOW Approvals Table**

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type | Description |
| user\_id | UUID | User requesting access |
| country\_code | STRING | Country for which access is requested |
| status | ENUM | PENDING, APPROVED, DENIED |
| request\_id | STRING | ServiceNow Ticket ID |

**4. API Endpoints**

**4.1 User Access Validation API**

**Endpoint:** POST /access/validate

**Request Payload:**

{

"user\_id": "abc-123",

"requested\_country": "IN"

}

**Response:**

{

"access\_granted": false,

"reason": "Country restricted. Please raise a SNOW request.",

"snow\_ticket\_url": "https://service-now.com/request?id=xyz"

}

**4.2 ServiceNow Approval Integration**

**Endpoint:** POST /snow/request

**Request Payload:**

{

"user\_id": "abc-123",

"country\_code": "IN"

}

**Response:**

{

"request\_id": "snow-456",

"status": "PENDING"

}

**4.3 Dynamic Country List Update**

**Endpoint:** PUT /country-list/update

**Request Payload:**

{

"country\_code": "SG",

"restricted": true

}

**Response:**

{

"message": "Country restriction updated successfully."

}

**5. Policy Enforcement Using ABAC**

**Sample ABAC Policy (OPA/Rego):**

package access

default allow = false

allow {

input.global\_access == true

}

allow {

not restricted[input.requested\_country]

}

allow {

approved[input.user\_id][input.requested\_country] == true

}

**6. Security & Performance Considerations**

* **Cache approved access requests** to reduce frequent DB queries.
* **Implement rate-limiting** at API Gateway to prevent abuse.
* **Regular audits** of SNOW requests to prevent access creep.
* **Monitoring & Alerts** for unauthorized access attempts.

**7. Scalability Considerations**

* **Dynamic Country List Service** allows automatic rule enforcement without code changes.
* **Decentralized policy management** via ABAC ensures flexibility.
* **Cloud-based ServiceNow integration** enables scalable approval workflows.

**8. Conclusion**

This approach ensures: ✅ **Compliance with Data Sovereignty Laws** (e.g., GDPR, ISO 27001).  
✅ **Minimal manual intervention** for access rule updates.  
✅ **ServiceNow Integration** ensures approval auditability.  
✅ **Scalability for future country additions**.

**9. Next Steps**

1. **Develop API Gateway policies** and integrate with ABAC.
2. **Set up ServiceNow request automation**.
3. **Deploy Dynamic Country List Service**.
4. **Test end-to-end workflow and monitor access logs.**