**1. Conditional Access Based on User Attributes (Recommended)**

This approach introduces an **Attribute-Based Access Control (ABAC)** layer without changing RSAM.

**How It Works**

* Instead of relying solely on RSAM entitlements, access decisions are made dynamically using **user attributes** such as:
  + **User location** (e.g., India, Malaysia, or Global)
  + **Department** (e.g., Finance, HR, IT)
  + **Project/Business Unit**
  + **Explicit country-level approvals**
* When a user tries to access **Malaysia/India data**, an access check is performed in **a middleware layer** or **API gateway** (without altering RSAM).
* This check ensures that the entitlement alone is not enough—**additional context is required for access.**

**Implementation Steps**

1. **Extend identity management (IDM) or Active Directory (AD)** to store relevant attributes for users.
2. **Introduce a policy engine (middleware)** that verifies these attributes at runtime.
3. **Modify access logic in APIs or applications** to enforce the attribute-based access rule.
4. **Log access attempts for auditability**, allowing Malaysia/India teams to monitor access.

**Pros**

✔️ No changes to RSAM structure.  
✔️ Allows dynamic access rules.  
✔️ Transparent enforcement without user intervention.

**Cons**

❌ Requires middleware or policy engine integration.  
❌ May need additional identity attribute management.

**2. Dynamic Approval at Data Access Level**

Rather than blocking access outright, introduce a mechanism where **a one-time or recurring approval is needed before accessing Malaysia/India data.**

**How It Works**

* Users with global entitlement can **see** the data but cannot access it without explicit approval.
* When access is attempted, a **just-in-time approval request is triggered**:
  + If pre-approved (within a defined period), access is granted.
  + If not, a request is sent to a designated **approval authority** in Malaysia/India.
* This is **independent of RSAM** and only affects data access.

**Implementation Steps**

1. **Implement a lightweight approval system** at the application level.
2. **Integrate with existing workflows** (email, notification, ticketing systems).
3. **Store approval records** for audit and compliance.
4. **Implement auto-revalidation** (e.g., re-approval every 6 months).

**Pros**

✔️ Easy to implement without changing RSAM.  
✔️ Provides an audit trail.  
✔️ Gives Malaysia/India control over their data.

**Cons**

❌ Delays access when approvals are pending.  
❌ Needs an escalation mechanism for urgent cases.

**3. Country-Specific Access Logs & Review Mechanism**

This approach does **not block access** but ensures **strict monitoring** of global entitlement users accessing Malaysia/India data.

**How It Works**

* A **real-time access logging system** is introduced, tracking when users with global entitlement access country-specific data.
* The **Malaysia/India teams can review logs** and raise concerns if necessary.
* Alerts are triggered for **suspicious or excessive access attempts.**
* Can be implemented via **SIEM (Security Information and Event Management)** tools or **data warehouse logging**.

**Implementation Steps**

1. **Modify logging infrastructure** to capture country-level data access.
2. **Create real-time dashboards** for Malaysia/India teams.
3. **Set up anomaly detection rules** (e.g., if a user downloads excessive Malaysia data).
4. **Define an escalation workflow** for reviewing suspicious access.

**Pros**

✔️ No access restriction—only monitoring.  
✔️ Provides transparency to concerned countries.  
✔️ Helps with audit and compliance.

**Cons**

❌ Does not prevent access, only tracks it.  
❌ Requires Malaysia/India teams to actively monitor logs.

**4. Hybrid Role-Based Access Control (RBAC) with Global Entitlement Override**

Instead of a blanket global entitlement, introduce **country-level override roles** that selectively limit data access.

**How It Works**

* Global entitlement remains unchanged, but additional **country-specific restrictions are layered** on top.
* Users **must be assigned a secondary approval-based role** (Malaysia/India role) to access country-specific data.
* These roles **can be managed separately** without affecting RSAM structure.

**Implementation Steps**

1. **Create country-level override roles** (e.g., Malaysia\_Access\_Approved).
2. **Integrate these roles with existing authentication mechanisms**.
3. **Set up automatic provisioning of these roles** based on approvals.
4. **Ensure role assignments are periodically reviewed**.

**Pros**

✔️ Works within RBAC without major structural changes.  
✔️ Allows granular control over access.  
✔️ Easily scalable for future access needs.

**Cons**

❌ Needs manual intervention to assign override roles.  
❌ Can become complex if too many country overrides are required.

**5. Just-in-Time (JIT) Access for Country Data**

Instead of providing continuous access to Malaysia/India data, introduce a **Just-in-Time (JIT) access model**, where users can request access for a **limited duration**.

**How It Works**

* Users with **global entitlement do not have permanent access** to Malaysia/India data.
* When access is required, they **submit a request**, which is either:
  + **Auto-approved** if criteria are met.
  + **Manually approved** for sensitive data.
* Access is granted **only for a limited period** (e.g., 24 hours).

**Implementation Steps**

1. **Develop an access request workflow** integrated into the existing system.
2. **Configure time-bound access grants** for Malaysia/India data.
3. **Ensure automatic revocation** after access duration expires.

**Pros**

✔️ Minimizes unauthorized long-term access.  
✔️ Provides temporary, controlled access.  
✔️ Helps meet strict compliance needs.

**Cons**

❌ Requires a system to track access requests.  
❌ Adds an additional step before data access.

**6. API Gateway Rule Enforcement**

If data access happens through **APIs**, introduce an API gateway **that enforces additional country-level rules.**

**How It Works**

* The API gateway inspects every **data request** and **enforces country-specific rules**.
* Even if the entitlement allows access, **API policies add an extra verification layer**.
* Example rules:
  + **Deny access to Malaysia/India data unless a secondary flag is present.**
  + **Throttle excessive data requests to prevent bulk downloads.**
  + **Mask sensitive data unless explicitly authorized.**

**Implementation Steps**

1. **Modify API gateway policies** to check country-level access.
2. **Introduce logging and alerting** for country-specific requests.
3. **Integrate with an entitlement validation service**.

**Pros**

✔️ Works **transparently at the backend** without modifying RSAM.  
✔️ Prevents unauthorized bulk data access.  
✔️ Can be easily adjusted for future policies.

**Cons**

❌ Only applies to API-based access, not UI-level access.  
❌ Requires API gateway customization.

**Conclusion: Choosing the Best Approach**

* **If you need strict access control** → **Option 1 (ABAC) or Option 4 (Hybrid RBAC)**.
* **If monitoring is enough** → **Option 3 (Access Logs & Review Mechanism)**.
* **If temporary access is okay** → **Option 5 (JIT Access)**.
* **If API access control is required** → **Option 6 (API Gateway Rules)**.