Architectural Decision Record (ADR) for Role-Based Access Control (RBAC) for Fish Watch System

Decision Summary

Utilize the architectural decisions for role-based access control for fish watch system.

Context

The Fish Watch System, our real-time monitoring and analytics platform for fisheries management, requires a robust access control mechanism to ensure that only authorized users have appropriate permissions to access, modify, and manage sensitive data and system functionalities. We need to decide on the implementation of role-based access control (RBAC) for the Fish Watch System to enforce fine-grained access control based on users' roles and responsibilities.

Decision:

We have decided to implement role-based access control (RBAC) for the Fish Watch System to manage user permissions and access rights effectively.

Rationale:

Several factors influenced this decision:

1. **Granular Access Control**: RBAC enables us to define roles and assign permissions based on users' job functions, responsibilities, and organizational hierarchy. This allows for granular control over who can perform specific actions within the system, reducing the risk of unauthorized access and data breaches.
2. **Scalability and Maintainability**: RBAC provides a scalable and maintainable access control model, especially in large and complex systems like the Fish Watch System. As the number of users and roles grows, RBAC simplifies user management by centralizing permissions and access policies, reducing administrative overhead and ensuring consistency across the organization.
3. **Flexibility and Adaptability**: RBAC allows for flexibility in defining roles and permissions, accommodating changes in user roles, organizational structure, and business requirements over time. This flexibility ensures that the access control model can evolve with the Fish Watch System's needs and adapt to new user roles or functionalities as the system matures.

4. **Auditability and Compliance**: RBAC provides audit trails and accountability by logging user actions and permissions changes, facilitating compliance with regulatory requirements such as GDPR, HIPAA, and CCPA. This auditability enables administrators to track user activities, detect anomalies, and investigate security incidents effectively.

5. **Integration with Identity Providers:** RBAC can be integrated with identity providers (IdPs) such as AWS Identity and Access Management (IAM), Active Directory (AD), or LDAP for centralized user authentication and authorization. This integration streamlines user provisioning, authentication, and single sign-on (SSO) across different systems and applications, enhancing user experience and security.

Consequences:

By implementing RBAC for the Fish Watch System, we anticipate the following consequences:

1. **User Management Overhead:** RBAC requires careful planning and administration to define roles, assign permissions, and manage user access effectively. Proper documentation, training, and governance processes are essential to ensure consistency and accuracy in role assignments and access policies.
2. **Role Definition Complexity:** Defining roles and permissions can be complex, especially in systems with diverse user roles and complex workflows. We need to conduct thorough analysis and stakeholder consultations to identify user roles, responsibilities, and required permissions accurately.
3. **Access Control Matrix Maintenance:** As the Fish Watch System evolves and new features are introduced, the access control matrix may need to be updated and maintained to reflect changes in user roles, permissions, and system functionalities. Regular reviews and audits are necessary to ensure the integrity and effectiveness of the RBAC model.
4. **User Experience Considerations:** While RBAC enhances security and access control, it's essential to balance security requirements with usability considerations to avoid overly restrictive access policies that impede user productivity. Providing clear documentation, role-based dashboards, and self-service access request workflows can improve the user experience and reduce support overhead.

In summary, implementing RBAC for the Fish Watch System provides a scalable, flexible, and auditable access control mechanism to manage user permissions effectively, enforce security policies, and ensure compliance with regulatory requirements. With proper planning, governance, and user engagement, RBAC will contribute to the overall security and integrity of the Fish Watch System while supporting the needs of users and stakeholders.