Okay, this is a critical document! Separating a module like IAT (likely standing for Internal Audit Management or something similar, within IBM OpenPages) requires a detailed design.

Here's a comprehensive template for your design document, tailored for a module separation project. Remember to adapt it heavily to your specific context, technologies, and the actual functionalities of your IAT module.

# **Design Document: Separation of OpenPages IAT Module**

**Document Version:** 1.0

**Date:** YYYY-MM-DD

**Author(s):** [Your Name/Team Name]

**Project Sponsor:** [Sponsor Name]

**Project Manager:** [PM Name]

## **1. Introduction**

### **1.1. Purpose**

This document details the design for separating the [Full Name of IAT Module, e.g., Internal Audit Tracking] module from the existing IBM OpenPages platform into a standalone or loosely coupled application/service. It outlines the architectural decisions, technical specifications, integration points, and migration strategy required for this separation.

### **1.2. Scope**

This document covers:

* The rationale and objectives for separating the IAT module.
* The high-level and detailed design of the new, standalone IAT module.
* The definition of interfaces and integration points with the remaining OpenPages environment.
* The data migration strategy for existing IAT data.
* The testing and deployment considerations for the separated module.

This document explicitly **does not** cover:

* [Any functionality or system explicitly out of scope, e.g., detailed UI/UX mockups, specific infrastructure provisioning details beyond architectural needs.]

### **1.3. Audience**

This document is intended for:

* Project Stakeholders (Business & IT)
* Architects
* Developers
* Quality Assurance Testers
* Operations and Support Teams

### **1.4. Definitions and Acronyms**

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| IAT | Internal Audit Tracking (or relevant full name) |
| OpenPages | IBM OpenPages GRC Platform |
| API | Application Programming Interface |
| SaaS | Software as a Service |
| [Add others] | [Add definitions] |

Export to Sheets

### **1.5. References**

* [Link to Project Charter/Initiation Document]
* [Link to Business Requirements Document (BRD) for IAT module separation]
* [Link to Existing OpenPages Architecture Diagram]
* [Any relevant technical specifications or standards]

## **2. Current State Analysis (OpenPages IAT Module)**

### **2.1. Overview of Current IAT Functionality in OpenPages**

* Brief description of what the IAT module currently does within OpenPages.
* Key business processes supported.
* Primary users and their interactions.

### **2.2. Current Technical Implementation (within OpenPages)**

* **Data Model:** Describe how IAT data is currently stored (e.g., OpenPages objects, fields, relationships).
* **Workflows:** Describe OpenPages workflows tied to IAT.
* **Reporting:** How is IAT data reported today? (e.g., OpenPages Cognos/standard reports).
* **Integrations:** Any existing integrations with other systems from within OpenPages related to IAT.
* **Customizations:** List significant customizations, scripts, or custom objects specific to IAT.

### **2.3. Rationale for Separation**

* **Pain Points/Challenges:** (e.g., performance issues, scalability limitations, difficulty in customization, licensing costs, platform dependencies, maintenance overhead, technology stack mismatch, need for greater agility).
* **Business Drivers:** (e.g., strategic shift, specific compliance requirements, need for specialized features not easily supported by OpenPages).
* **Expected Benefits:** (e.g., improved performance, reduced TCO, enhanced scalability, independent release cycles, improved user experience, adoption of modern technologies).

## **3. High-Level Design of Separated IAT Module**

### **3.1. Architectural Overview**

* **Diagram:** Provide a high-level architectural diagram showing the new IAT module, its components, and its interaction points with OpenPages and other relevant systems.
* **Architectural Style:** (e.g., Microservices, Monolithic Application, Serverless, Event-Driven). Justify the choice.

### **3.2. Component Breakdown**

Identify the major logical or physical components of the new IAT module.

* **[Component 1 Name]:** Brief description of its responsibility.
* **[Component 2 Name]:** Brief description.
* ... (e.g., UI Layer, API Gateway, Business Logic Service, Data Access Layer, Database, Integration Service)

### **3.3. Technology Stack Selection**

* **Programming Language(s):** [e.g., Java, Python, .NET Core, Node.js]
* **Framework(s):** [e.g., Spring Boot, Django, React, Angular, Vue.js]
* **Database:** [e.g., PostgreSQL, SQL Server, MongoDB, Oracle]
* **Cloud Platform (if applicable):** [e.g., AWS, Azure, GCP]
* **Other Key Technologies:** [e.g., Kafka, Kubernetes, Docker, Elasticsearch]
* **Justification:** Briefly explain why these technologies were chosen.

### **3.4. Key Design Principles**

* **Modularity:** Emphasize clear separation of concerns.
* **Scalability:** How will the new module handle growth?
* **Security:** Principles guiding secure design.
* **Performance:** How will responsiveness be ensured?
* **Maintainability:** How will the code be easy to understand and modify?
* **Extensibility:** How will future enhancements be accommodated?
* **Observability:** How will the system be monitored and debugged?

## **4. Detailed Design**

### **4.1. Functional Design**

Describe the core functionalities of the new IAT module, potentially mapping them to their OpenPages equivalents.

* **[Function 1]:** (e.g., Audit Creation and Planning)
  + User Interface/Interaction Flow.
  + API Endpoints (if applicable).
  + Associated Business Rules.
* **[Function 2]:** (e.g., Finding and Recommendation Management)
* **[Function 3]:** (e.g., Action Plan Tracking)
* **[Function N]:** (e.g., Reporting & Dashboards for IAT)

### **4.2. Data Model Design**

* **Schema Diagrams:** Provide Entity-Relationship Diagrams (ERDs) for the new IAT database.
* **Table Definitions:** For key tables, define columns, data types, primary keys, foreign keys, and constraints.
* **Data Dictionary:** (Optional, but good practice) List all entities and attributes with descriptions.

### **4.3. Interface Design / Integration with OpenPages**

This is crucial for a separation project.

* **Authentication & Authorization:**
  + How will users authenticate with the new IAT module? (e.g., SSO via SAML/OAuth, OpenPages integrated auth, separate auth).
  + How will authorization (permissions) be managed, especially for users who still interact with OpenPages for other functions?
* **Data Synchronization / Communication:**
  + Identify data that needs to be synchronized between IAT and OpenPages (e.g., Organization structure, User profiles, specific GRC objects).
  + **Integration Patterns:** (e.g., REST APIs for synchronous calls, Message Queues for asynchronous events, Database replication, File-based transfers).
  + **Specific APIs/Endpoints:** Define the contract for each API (request/response format, parameters, error codes).
    - POST /api/v1/audits (to create an audit in IAT from an OpenPages trigger)
    - GET /api/v1/users/{id} (to retrieve user details from OpenPages or a common identity store)
  + **Triggers & Events:** How will changes in one system trigger updates in the other? (e.g., OpenPages event listeners, webhooks).
* **UI Integration (if applicable):**
  + Will there be deep links from OpenPages to the new IAT module?
  + Will any IAT UI elements be embedded back into OpenPages (e.g., iframes)?

### **4.4. Error Handling and Logging**

* **Error Codes and Messages:** Standardized error responses for APIs and internal errors.
* **Logging Strategy:**
  + What information will be logged? (e.g., request/response, errors, warnings, info).
  + Logging levels.
  + Where will logs be stored? (e.g., file system, centralized logging service like Splunk/ELK).
  + Log retention policies.

### **4.5. Security Considerations**

* **Authentication & Authorization:** (detailed technical implementation)
* **Data Encryption:** (in transit and at rest)
* **Input Validation:** Prevention of common vulnerabilities (e.g., XSS, SQL Injection).
* **Vulnerability Management:** Plan for regular security scanning and patching.
* **Access Control:** Role-Based Access Control (RBAC) definitions.
* **Audit Trails:** What user actions will be logged for auditing purposes?

### **4.6. Performance Considerations**

* **Performance Goals:** Define key performance indicators (KPIs) (e.g., response times, throughput, concurrent users).
* **Scalability Mechanisms:** (e.g., horizontal scaling, load balancing, caching strategies, database sharding/replication).
* **Bottleneck Identification:** Anticipated areas of high load and how they will be addressed.

### **4.7. Deployment Strategy**

* **Deployment Environment(s):** (e.g., Dev, Test, Staging, Production).
* **Deployment Process:** (e.g., CI/CD pipeline, manual steps).
* **Rollback Plan:** How to revert to a previous stable state if deployment fails.

### **4.8. Backup and Recovery**

* **Data Backup Strategy:** (frequency, retention, storage location).
* **Disaster Recovery:** Plan for restoring service in case of a major outage.

## **5. Migration Strategy**

### **5.1. Data Migration Plan**

* **Data Scope:** What IAT data (objects, fields, attachments) needs to be migrated from OpenPages?
* **Migration Tooling/Approach:** (e.g., custom scripts, ETL tools, OpenPages API exports).
* **Migration Phases:** (e.g., test migration, dry runs, cutover).
* **Data Transformation:** Any data cleansing, mapping, or restructuring required.
* **Validation:** How will data integrity be verified after migration?

### **5.2. Cutover Plan**

* Detailed steps for switching from the old IAT in OpenPages to the new module.
* Downtime considerations and communication plan.
* Responsibilities during cutover.

### **5.3. Rollback Plan**

* Steps to revert to the old OpenPages IAT module if the cutover fails.
* Data rollback strategy.

## **6. Testing Strategy**

### **6.1. Unit Testing**

* Tools and Frameworks.
* Code coverage targets.

### **6.2. Integration Testing**

* Focus on the interfaces between the new IAT module and OpenPages, as well as other integrated systems.
* Test scenarios for data synchronization, API calls, and authentication.

### **6.3. System Testing**

* End-to-end testing of the separated IAT module's functionalities.

### **6.4. Performance Testing**

* Testing against the defined performance goals.
* Load, stress, and scalability testing.

### **6.5. User Acceptance Testing (UAT)**

* Business involvement and sign-off.
* Key UAT scenarios.

### **6.6. Security Testing**

* Vulnerability scanning, penetration testing.

## **7. Operational Considerations**

### **7.1. Monitoring and Alerting**

* Metrics to monitor (e.g., CPU, memory, disk, network, application errors, response times).
* Alerting thresholds and notification channels.
* Monitoring tools (e.g., Prometheus, Grafana, ELK Stack, Splunk).

### **7.2. Support and Maintenance**

* Support model (e.g., L1, L2, L3).
* On-call procedures.
* Maintenance windows and procedures.
* Runbooks for common operational tasks and issues.

### **7.3. Documentation**

* Operational runbooks.
* API documentation.
* Troubleshooting guides.

## **8. Risks and Assumptions**

### **8.1. Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Impact** | **Likelihood** | **Mitigation Strategy** |
| R-001 | Data migration complexity | High | Medium | Thorough data mapping, phased migration, extensive validation. |
| R-002 | OpenPages API limitations | Medium | Medium | Early prototyping, detailed vendor discussions. |
| [Add others] | [Add description] | [Low/Medium/High] | [Low/Medium/High] | [Add mitigation] |

Export to Sheets

### **8.2. Assumptions**

* [e.g., "The OpenPages APIs will provide sufficient access to all necessary historical IAT data for migration."]
* [e.g., "The existing OpenPages environment will remain stable during the separation project."]
* [e.g., "Sufficient resources (personnel, budget) will be allocated for the project duration."]

## **9. Open Issues and Future Considerations**

### **9.1. Open Issues / Deferred Decisions**

* [e.g., "Final decision on specific cloud provider region is pending."]
* [e.g., "Detailed UI design to be covered in a separate UX document."]

### **9.2. Future Enhancements / Roadmap**

* [e.g., "Integration with external risk management systems."]
* [e.g., "Advanced analytics and AI capabilities."]

## **10. Approvals**

This design document has been reviewed and approved by the undersigned:

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Signature** | **Date** |
| Project Sponsor |  |  |  |
| Business Lead |  |  |  |
| Lead Architect |  |  |  |
| Security Officer |  |  |  |
| Operations Lead |  |  |  |
| [Add other roles] |  |  |  |