AMU

Technical Architecture & Infrastructure Compatibility Brief

Asset Management Unit Internal Software Solution

PURPOSE

To provide Government IT with a clear view of the proposed AMU application architecture, technical requirements and security expectations, and to obtain the infrastructure standards needed before the production architecture is finalized.

Prepared by
Joycelyn Charlton
Buying Bahamian
Digital Solutions Architect

Technical review document • Not a hosting mandate

1. Executive Summary

The AMU solution is proposed as a secure, browser-based internal application for managing assets, cases, documents, movements, custody information, reporting, user access and auditable activity. The prototype defines business workflows and user experience; this brief is intended to align the production implementation with Government IT's approved infrastructure before the final production architecture is locked.

The application is being designed so that core components - application runtime, PostgreSQL database, protected file storage, identity integration, logging and deployment - can be mapped to Government-approved services rather than assuming a particular hosting provider.

2. What AMU Needs the System to Do

- Provide authenticated internal access to authorized AMU personnel.
- Maintain structured records for assets, cases, source agencies, movements, custody, contacts and related operational data.
- Store and retrieve protected documents and media, including PDFs, photographs, scans and other approved attachments.
- Enforce role-based access so users see and perform only the functions authorized for their role.
- Maintain audit records for significant user and system activity.
- Generate operational and compliance reports and support controlled export/printing where authorized.
- Support secure integration with approved identity, email, storage, logging and infrastructure services where required.

3. Proposed Application Technology Stack

The following describes the current technical direction. Items marked as development tooling do not automatically become production dependencies.

Layer	Proposed Technology	Purpose
Frontend	React / Next.js, TypeScript, HTML, CSS	Production application interface, routing, forms and reusable components.
Application runtime	Node.js-compatible runtime	Executes server-side application logic and approved API endpoints.
Database	PostgreSQL	Relational system of record for AMU operational data.
Database language	SQL	Schema definition, queries, reporting and controlled data operations.
Application/API layer	Developer-built server/API endpoints within the Next.js/Node.js application	Controlled communication between the user interface, PostgreSQL, protected storage and approved

Layer	Proposed Technology	Purpose
		Government integrations; production exposure and security standards are aligned with Government IT.
Document/media storage	Protected object or file storage	Stores PDFs, images, scans, videos and other authorized attachments.
Authentication	Government-approved identity provider preferred	Identifies authorized users; production method to be confirmed by IT.
Authorization	RBAC plus server/database controls	Enforces Director/Admin/Inspector/restricted access models subject to AMU approval.
Audit logging	Application audit trail + infrastructure logging integration	Records significant access, changes and actions for accountability.
Source/version control	Git; private repository during development	Change history, controlled development and release management.
Development platform	Supabase may be used for development/prototyping	Development convenience only; not intended to dictate Government production hosting.

Important production distinction: PostgreSQL is the proposed relational database. Supabase may support development/prototyping, but the production database, storage and authentication can be deployed using Government-approved infrastructure and services.

4. High-Level Architecture

AUTHORIZED AMU USERS
↓
GOVERNMENT-APPROVED IDENTITY / MFA
↓
AMU WEB APPLICATION
React / Next.js / TypeScript
↓
AMU SERVER / API LAYER
Developer-built application logic, validation & authorization
↓
PostgreSQL Database | Protected File Storage | Audit Logging
↓
APPROVED GOVERNMENT SYSTEM APIs / INTEGRATIONS (where required)
↓
GOVERNMENT-APPROVED HOSTING, NETWORK, SECURITY,
BACKUP, MONITORING & RECOVERY CONTROLS

5. Production Capabilities Required from Government IT

The final implementation needs approved services or standards for each of the following. Government IT may satisfy these with its existing products and infrastructure.

Capability	Information / Service Needed
Hosting / compute	Approved environment capable of hosting the web application and supported runtime.
Database	Supported PostgreSQL deployment, version, connectivity, administration and backup standards.
Identity	Approved organizational authentication, SSO and MFA method.
File storage	Protected storage appropriate for AMU documents/media, with access controls and retention rules.
Networking	Approved internal/private access model, DNS, certificates and connectivity between application components.
Encryption	Approved TLS/HTTPS, encryption-at-rest and key/certificate management requirements.
Secrets	Approved mechanism for passwords, connection strings, API credentials and other secrets.
Logging / monitoring	Approved application, infrastructure and security logging/monitoring destinations.
Backup / recovery	Database and file backup, restore testing, retention and disaster-recovery requirements.
Email	Approved SMTP/API service if system notifications or report delivery are required.
Security review	Required vulnerability scanning, code review, testing and production release controls.
Environments	Standards for development, UAT/testing and production separation.
API / integration standards	Approved standards for internal APIs, network exposure, authentication, credentials, service-to-service access and connections to existing Government systems.

6. Application Security Controls in the Build

- Authentication integrated with the approved identity mechanism.
- Role-based authorization enforced on protected operations, not only hidden in the user interface.
- Least-privilege access for application roles and service identities.
- Server-side input validation and safe database access practices.
- Controlled file upload, file type/size validation and authorization before document access.
- Secure session handling and protection of authentication tokens/cookies.
- Secrets kept out of source code and repositories.
- Audit logging for defined sensitive and administrative actions.
- Dependency and source-code security scanning during development where available.
- Separate development/testing and production configuration.
- Security requirements reviewed against an accepted application-security baseline such as OWASP ASVS, subject to Government IT standards.

7. Development vs. Government Production

Area	Development Position	Production Alignment Needed
Source code development	Developer-managed using Git/version control and private development repository.	Government may specify the approved repository or transfer process for final source.
Prototype / development database	May use Supabase/PostgreSQL for development.	Production database should use Government-approved PostgreSQL service/deployment.
Development storage	May use development storage for non-production/test data.	Production files must use Government-approved protected storage.
Authentication	Development/test identities may be used before integration.	Production should integrate with the approved Government identity/MFA solution.
Deployment	Developer prepares application and deployment documentation.	Government IT confirms hosting, networking, security controls and production release process.
AMU API / backend	Developer designs and implements AMU server/API endpoints, validation, authorization and backend business logic.	Government IT approves production hosting, network exposure, identity/service authentication, secrets, monitoring and deployment standards.
Existing Government-system integrations	Developer implements the AMU-side connector/integration against approved specifications.	Government/system owner provides or approves the API/integration method, documentation, access, credentials and security/network requirements.

8. API & Integration Responsibility

The AMU application will include its own server-side API layer. This API is part of the application code and will be designed and developed by the Digital Solutions Architect/Developer. It is not a separate Government product that IT must build.

Government IT / Security is responsible for approving or providing the production environment in which that API operates, including hosting, network exposure, identity/service authentication, certificates, secrets, firewall/private-access rules, monitoring, logging and deployment/security standards.

Area	Developer / AMU Application	Government IT / System Owner
AMU application API	Design and develop endpoints, backend business logic, validation, authorization, database/storage operations and application audit actions.	Approve/provide production hosting, networking, identity/service authentication, certificates, secrets, monitoring and deployment/security requirements.
Existing Government API	Build the AMU-side integration/connector according to approved specifications.	The owning Government system/IT team provides or approves the API/integration method, endpoint documentation, access process, credentials and applicable security/network requirements.
Data exchanged through integrations	Implement only approved fields, validation and application handling.	Confirm authorized data scope, connectivity, retention, logging and any inter-agency/security controls.

Plain-language boundary: We build AMU's API. Government IT approves where and how it runs. If AMU needs to connect to another Government system, that system's owner provides the approved connection/API details and we build AMU's side of the connection.

9. Information Requested from Government IT

Please provide or confirm the following before the AMU production architecture is finalized.

1. What hosting environment is approved for the AMU application (on-premises, Azure, private cloud, other approved environment, or a combination)?

2. Is PostgreSQL supported? If yes, what versions and deployment model are approved?

3. Is a Node.js-compatible application runtime supported? Are there approved versions or runtime restrictions?

4. Are containerized applications (for example, Docker containers) supported or preferred, or is direct server deployment required?

5. What identity provider is used for staff authentication (for example, Microsoft Entra ID/Active Directory or another approved service)?

6. Can the AMU application integrate with that identity provider for Single Sign-On (SSO) and Multi-Factor Authentication (MFA)?

7. What storage service is approved for application documents, images, scans, videos and related attachments?

8. What encryption, certificate, key-management and secrets-management standards must the application follow?

9. What network access model is required: internal network only, VPN, approved remote access, private cloud access, or another model?

10. What backup, retention, restore-testing and disaster-recovery requirements apply to the database and file storage?

11. What application/security logging and monitoring platforms must AMU integrate with?

12. What approved email/SMTP/API service should be used for application notifications or internally generated reports?

13. What vulnerability scanning, application security testing, code review or security approval is required before production?

14. What is the required process for moving releases through Development → UAT/Testing → Production?

15. Are there Government coding, API design/security, accessibility, database, records-retention or application-security standards that must be followed?

16. Where must final source code, database migration scripts and technical documentation be stored or transferred?

17. Are external development services permitted for non-production work, and what restrictions apply to data used in those environments?

18. Are there restrictions on open-source frameworks, third-party packages, licenses or software dependencies?

19. What are the expected availability, performance, storage capacity and file-size requirements for the AMU system?

20. Who should serve as the Government IT/security technical contact for architecture review, environment provisioning and deployment?

21. What standards must the AMU application follow for internal API authentication, authorization, endpoint security, versioning and logging?

22. Must AMU application APIs remain private/internal only, and what network controls or gateway/WAF requirements apply?

23. If AMU needs to connect to an existing Government system, who owns that integration and who will provide the approved API/interface specifications, credentials and technical documentation?

24. Are service accounts, managed identities, API keys, OAuth/OIDC tokens, mutual TLS or another method required for system-to-system authentication?

10. IT Compatibility Response Sheet

Government IT may use this section as a concise response checklist. Additional technical notes can be attached where necessary.

Item	IT Response	Notes / Requirement
Approved hosting environment		
PostgreSQL supported / approved version		
Approved application runtime		
Container support / deployment method		
Identity provider / SSO / MFA		
Approved document/file storage		
Secrets / key management		
Network access model		
Logging / monitoring platform		
Backup / disaster recovery standard		
Approved email service		
Development/UAT/Production process		
Security testing / release requirements		
Source-code repository / handover standard		
Primary IT technical contact		

Item	IT Response	Notes / Requirement
API network exposure / gateway requirement		
Government-system integration owner / contact		
API design / security standard		
System-to-system authentication method		

11. Proposed Responsibility Boundary

Request to Government IT **Before the AMU production architecture is finalized, please provide the approved application-hosting, API/integration, technology and security requirements applicable to this solution. Where AMU must connect to an existing Government system, please identify the system owner and approved integration method so the AMU-side connector can be designed against authoritative specifications.**	**Primary Responsibility**
Digital Solutions Architect / Developer	Application architecture; frontend/backend implementation; database schema/migrations; application-level authorization; secure coding; application audit features; tests; technical documentation; deployment package/configuration requirements.
AMU Business Owners	Approve workflows, roles, records, reports, operational rules, access needs and acceptance criteria.
Government IT / Security	Approve/provide production hosting, network, identity, certificates/keys, infrastructure security, production database/storage services, monitoring, backups/recovery and release/security standards.
Joint responsibility	Architecture alignment, identity integration, UAT, security remediation, deployment validation, operational handover and change-management process.

12. Requested Next Step

Request to Government IT

Before the AMU production architecture is finalized, please provide the approved application-hosting standards, technology requirements and security requirements applicable to this solution. The development architecture will then be mapped to those specifications so that the production build is designed for compatibility with the approved environment.

13. Technical Review Sign-Off / Comments

Reviewer	Role / Department	Date	Comments / Follow-up

END OF UPDATED TECHNICAL ARCHITECTURE & INFRASTRUCTURE COMPATIBILITY BRIEF