APV Technical Assessment and Documentation Strategy

Prepared for: Australia Pacific Valuers Pty Ltd ATF APV Unit Trust
Prepared by: GetCimple Pty Ltd
Date: 25 Feb 2025

Executive Summary

1. Overview

This document outlines APV's technical landscape, documentation needs, and strategic recommendations for enabling third-party collaboration and scalable development. The assessment combines detailed technical findings with practical implementation strategies focused on business continuity and external engagement.

2. Current System Architecture

2.1 Application Architecture

See diagram in Appendix A

The current application architecture consists of:

• Web application built with .NET Core and React
• iOS capture application
• Supporting infrastructure including logging and database systems with point-in-time recovery
• Complete Azure Cloud deployment in Sydney Region

2.2 Development Pipeline

See diagram in Appendix B

The development pipeline includes:

• Bitbucket repository for source control
• Azure Pipelines for continuous integration

• Automated 15-minute deployments to both web and API environments
• Build artifact storage and management

3. Key Business Objectives

The strategic recommendations in this document address the following key business objectives:

• Enable effective third-party collaboration through comprehensive documentation
• Establish clear separation between business processes and technical implementation
• Create scalable resource management approach to reduce key person dependencies
• Ensure business continuity through proper knowledge transfer
• Enable external developer onboarding to reduce reliance on internal resources

4. Strategic Recommendations

4.1 Documentation Framework

The technical audit identifies the need for a three-tiered documentation approach:

External Documentation (Priority: Immediate)

• High-level system architecture overview
• Business process flows and integration points
• Third-party collaboration guidelines
• Security and compliance requirements

Business Process Documentation (Priority: High)

• Mapping of business processes to technical implementation
• Process flow documentation with clear rationales
• Change management procedures using selected project management tool
• Integration points and dependencies

Technical Documentation (Priority: High)

• Detailed technical specifications
• Development and deployment procedures
• Security and infrastructure configurations
• Common issues and resolutions

4.2 Resource Management Strategy

Knowledge Transfer

• Documentation of key system components and their business context
• Structured onboarding process for external resources
• Regular knowledge sharing sessions and documentation reviews
• Specific plans to distribute key personnel knowledge across team and documentation

External Resource Enablement

• Clear documentation hierarchy for different audience types
• Standardized onboarding procedures through Trello/Jira/Asana templates
• Access management and security protocols
• Development environment setup guides
• Streamlined contractor engagement process

5. Implementation Roadmap

5.1 Immediate Actions (0-30 days)

• Establish Trello/Jira/Asana workspace with provided templates
• Begin business process documentation
• Implement critical security recommendations
• Document key personnel responsibilities and knowledge areas

5.2 Short-term Goals (31-90 days)

• Complete initial business process documentation
• Implement backup and security improvements
• Establish external collaboration framework
• Begin systematic knowledge transfer
• Onboard first external developer using new framework

5.3 Long-term Objectives (90+ days)

• Complete technical documentation
• Implement automated testing and deployment
• Establish ongoing documentation maintenance process
• Regular review and updates of all documentation
• Scale development team as needed

6. Tooling Decisions

6.1 Project Management: Trello/Jira/Asana etc

Suggested for:

• Easy adoption and low learning curve
• Flexible workflow customization
• Powerful free tier capabilities
• Built-in templates and checklists
• Integration capabilities

6.2 Documentation Management

• Technical documentation in Bitbucket repository
• Process documentation in Trello/Jira/Asana with templates
• Knowledge base accessible to external parties

7. Next Steps

1. Review and approve documentation framework
2. Set up Trello/Jira/Asan workspace using provided templates
3. Begin capturing critical knowledge in structured format
4. Schedule regular progress reviews
5. Initiate first external developer onboarding

8. Risk Mitigation

• Regular documentation updates and reviews
• Structured knowledge transfer process
• Clear separation of business and technical documentation
• Comprehensive backup and security measures
• Reduced key person risk through documentation and knowledge sharing

9. Success Metrics

The following metrics will be used to track implementation success:

• Documentation completion percentage

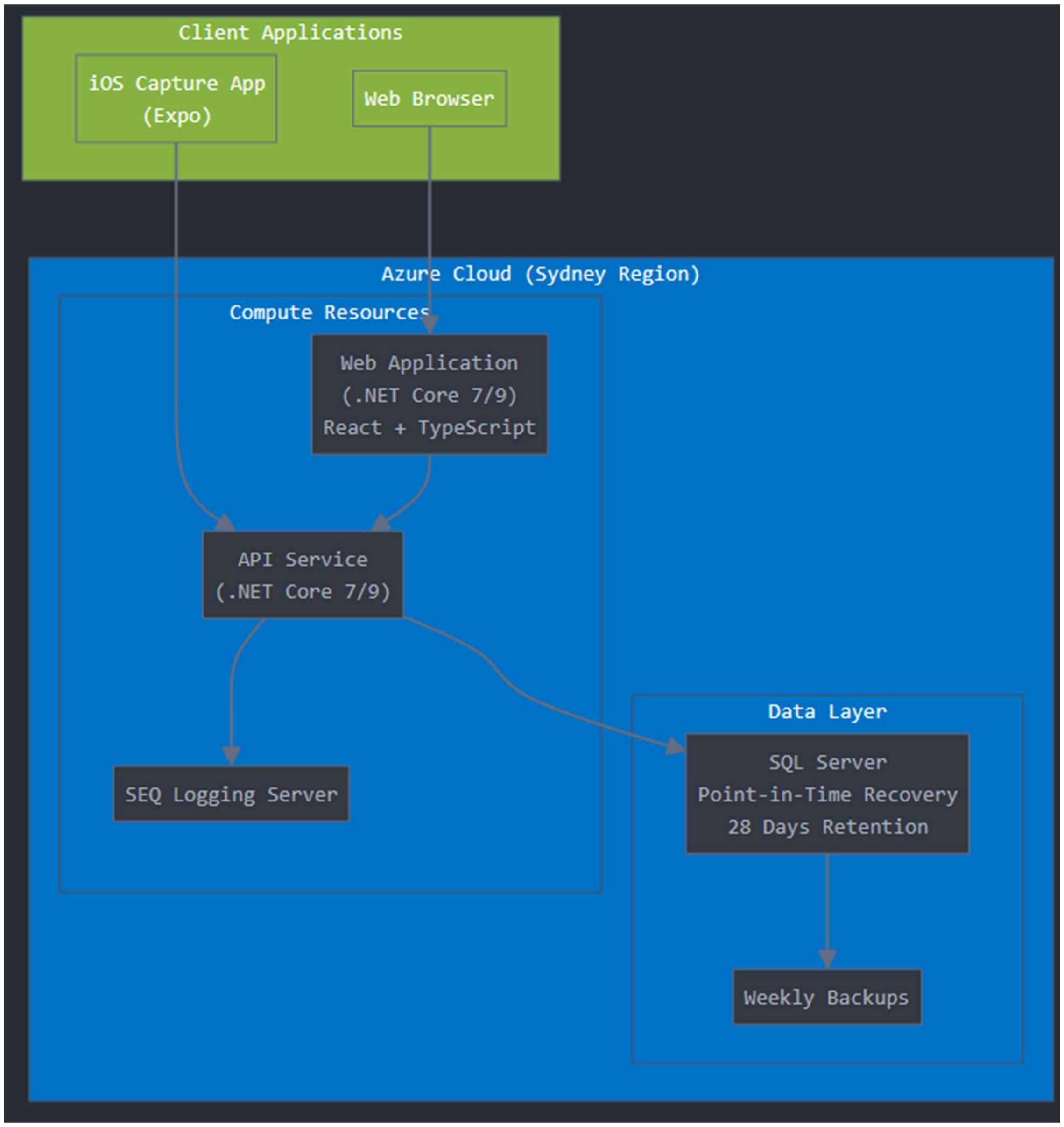
• External developer onboarding time
• Number of processes documented
• Reduction in key personnel direct involvement
• Successful contractor engagements

10. Additional Documentation

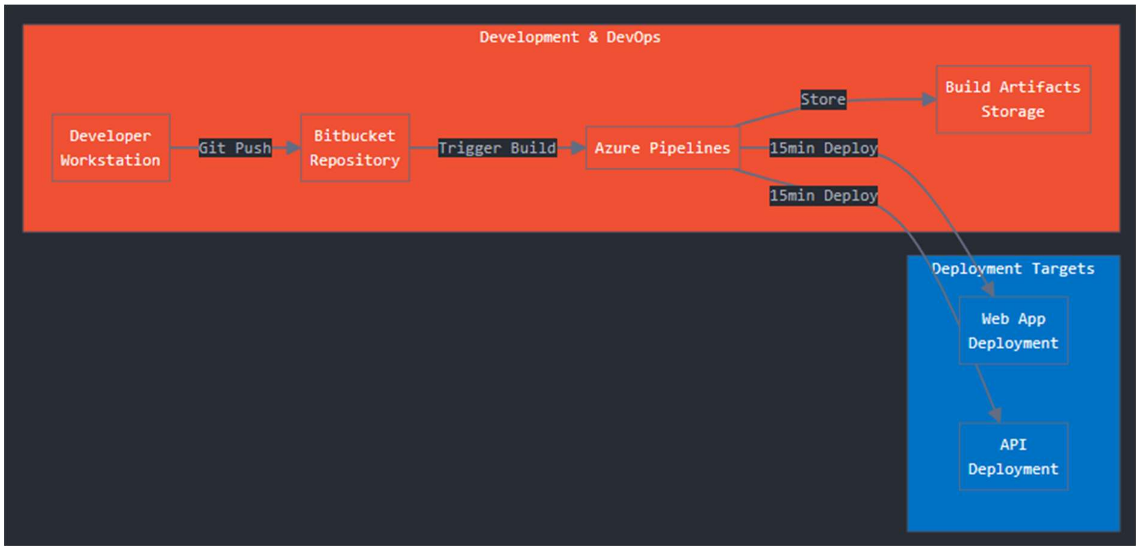
Detailed findings and recommendations can be found in:

• Technical Audit Findings and Recommendations (below)
• External Developer Collaboration Framework (below)
• Project Management Tool Implementation Guide (Separate to this document)

Appendix A: Technical Architecture Diagram


Client Applications
iOS Capture App (Expo)
Web Browser
Azure Cloud (Sydney Region)
Compute Resources
Web Application (.NET Core 7/9) React + TypeScript
API Service (.NET Core 7/9)
SEQ Logging Server
Data Layer
SQL Server Point-in-Time Recovery 28 Days Retention
Weekly Backups

Appendix B: Development Pipeline Diagram



Technical Audit Findings and Recommendations

This technical audit identifies critical areas requiring attention in APV's application infrastructure, development processes, and documentation. The findings are prioritized to enable systematic improvements while maintaining system stability and security.

Application Documentation

Priority: High

Summary

Application code and related processes need to be documented.

Description

All the application processes should be properly documented.

Required Documentation

- Classes and their functions
- Business process representations
- Rationale for non-standard implementations
- Application release process
- High-level module descriptions with business process references

Deliverable

- Comprehensive application documentation

Business Process Documentation

Priority: High

Summary

Business processes need comprehensive documentation.

Description

Business processes represented in the application should be documented as either:

- Formal business process flows
- Flow charts with supplementary process descriptions and rationales

Deliverable

- Business process documentation

Project Management Implementation

Priority: Critical

Summary

Change requests to the application must be managed through a project management tool.

Description

To improve feature implementation, change tracking, and documentation management, all future changes should be tracked in a project management tool.

Internal guidelines for tool usage should be prepared as part of the migration. At minimum, status transition criteria must be defined.

Example Workflow

New → In Development → Testing → Done

Status Definitions:

- New: Feature initially created in the tool
- In Development: Feature is being coded or business process is being refined
- Testing: Feature is deployed to test environment for business validation and bug detection
- Done: Feature is deployed to production and available to end users, or for declined issues, includes explanation of decision

Deliverables

- Project management workflow
- Project management tool implementation

Backup Implementation

Priority: High

Summary

Backup should be co-located.

Description

Current backup system lacks redundancy. Implementation should include:

- Automated redundancy with configurable frequency
- Retention policy (e.g., last 7 days of backups)
- Automated backup process
- Backup monitoring system

Following the 3-2-1 backup rule:

- 3 copies

- 2 different media types
- 1 co-located

This can be fulfilled through office server replication.

Deliverable

- Co-located backup process implementation

Security Preparation

Priority: High

Summary

Prepare for penetration testing.

Description

Required security preparations:

- Review application using static code analysis tools:
  - trunk.io for .NET
  - npm audit for React
- Remediate identified issues
- Verify internet-facing services with SSL Labs (minimum)
  - Use: https://www.ssllabs.com/ssltest/analyze.html

Deliverable

- Security assessment report including:
  - Completed actions
  - Remediated issues
  - Accepted risks

Infrastructure Documentation

Priority: High

Summary

Infrastructure documentation needs to be created.

Description

Comprehensive infrastructure documentation is required for disaster recovery scenarios.

Required Documentation:

- Service inventory
- Infrastructure setup procedures
- Service integration details

- Connection architecture

Deliverable

- Infrastructure documentation OR
- Infrastructure as Code implementation (e.g., Pulumi or Terraform)

Secrets Management

Priority: High

Summary

Secrets management must be reviewed.

Description

All application secrets (e.g., database passwords, third-party API keys) MUST be configured via environment variables and removed from the codebase.

Deliverable

- Verification that all secrets are moved to environment variables

---

Legacy Code Review

Priority: Medium

Summary

Review and potentially remove legacy code.

Description

Legacy components (e.g., DevExpress NuGet host) remain in the repository and should be reviewed for removal.

Deliverable

None specified

Issue Documentation

Priority: Medium

Summary

Document frequent application issues.

Description

Common application issues must be documented to enable resolution by new or external personnel, freeing developers for feature work. Documentation should include any processes requiring manual intervention.

Deliverable

- Common issues documentation

Security Risk Assessment

Priority: Medium

Summary

Create cybersecurity risk matrix.

Description

Matrix should document:

- Potential risks
- Probability
- Impact
- Mitigation approach

Example:

Risk: Ransomware attack

Probability: High

Impact: Low

Mitigation: Risk mitigated through:

- Co-located backups
- Infrastructure as Code deployment capability
- Documented staff device reimaging process
- Incident response procedures
- Authority notification process

Deliverable

- Cybersecurity risk matrix

Insurance Review

Priority: Low

Summary

Review cybersecurity insurance needs.

Description

Evaluate risk versus cost for cybersecurity insurance coverage. Insurance requirements can guide security improvements.

Deliverable

Not specified

Storage Backup

Priority: Low

Summary

Implement blob storage backup.

Description

Consider backup implementation for blob storage due to resource-intensive manual recreation process.

Deliverable

Not specified

CI/CD Implementation

Priority: Low

Summary

Consider implementing CI/CD.

Description

Current single-developer workflow functions adequately, but scaling development team requires:

- Code development procedures
- Deployment processes
- Testing protocols
- Automation for safer deployments

Deliverables

- Development and deployment guidelines
- CI/CD process implementation

.NET Upgrade

Priority: Medium

Summary

Upgrade .NET version.

Description

Upgrade to .NET 8 required as version 7 reached end-of-life 9 months ago.
Reference: https://endoflife.date/dotnet

Deliverable

Not specified

Static Code Analysis

Priority: Medium

Summary

Implement static code analysis in CI.

Description

Configure tool like https://trunk.io/code-quality in repository for:

- Security verification
- Best practice enforcement
- Code change validation

Deliverable

- Trunk code quality report

Developer Documentation

Priority: Low

Summary

Create application startup documentation.

Description

Create README including:

- Required tools
- Repository links
- Configuration steps:

    - Database creation
    - Migration execution
    - Additional setup requirements

Deliverable

- Developer onboarding guide/README

Performance Optimization

Priority: Low

Summary

Review auto-scaling for high-use periods.

Description

Consider auto-scaling infrastructure during high-usage periods (e.g., reporting) to maintain performance.

Deliverable

Not specified

Technical Audit Findings and Recommendations - Summary of Findings

Issue	Priority	Status	Deliverable	Dependencies
Application Documentation	High	Not Started	Documentation Package	None
Business Process Documentation	High	Not Started	Process Documentation	None
Project Management Implementation	Critical	Not Started	PM Tool & Workflow	None
Backup Implementation	High	Not Started	Co-located Backup Process	Infrastructure Documentation
Security Preparation	High	Not Started	Security Assessment Report	Static Code Analysis

Issue	Priority	Status	Deliverable	Dependencies
Infrastructure Documentation	High	Not Started	Infrastructure Documentation/IaC	None
Secrets Management	High	Not Started	Environment Variable Migration	None
Legacy Code Review	Medium	Not Started	N/A	None
Issue Documentation	Medium	Not Started	Common Issues Guide	None
Security Risk Assessment	Medium	Not Started	Risk Matrix	Security Preparation
Insurance Review	Low	Not Started	N/A	Security Risk Assessment
Storage Backup	Low	Not Started	N/A	Backup Implementation
CI/CD Implementation	Low	Not Started	CI/CD Pipeline & Guidelines	Infrastructure Documentation
.NET Upgrade	Medium	Not Started	N/A	None
Static Code Analysis	Medium	Not Started	Code Quality Report	None
Developer Documentation	Low	Not Started	Onboarding Guide	Application Documentation
Performance Optimization	Low	Not Started	N/A	Infrastructure Documentation

External Developer Collaboration Framework

This framework provides a structured approach for onboarding and integrating external developers into APV's development environment. It ensures consistent knowledge transfer, maintains code quality, and enables efficient collaboration while preserving system integrity and security.

1. Development Environment Setup

Local Development Environment

- Required development tools and versions:
  - Visual Studio 2022
  - .NET Core 7/9 SDK
  - Node.js and npm versions
  - Git
- Repository access and credentials
- Local configuration setup
- Environment variables and secrets management
- Database setup and sample data

Cloud Resource Access

- Azure portal access (appropriate role level)
- Connection string configuration
- Logging system access
- Test environment access

2. Codebase Orientation

Architecture Overview

- System component diagram
- Key services and their responsibilities
- Database schema overview
- API documentation
- Frontend architecture (React + TypeScript)

Development Standards

- Code style guide
- Branch naming conventions
- Commit message format
- Pull request process
- Code review requirements

Key Technical Areas

- Authentication/Authorization flow
- API structure and patterns
- Database access patterns
- Logging implementation
- Common utilities and shared code

3. Business Context for Developers

Core Business Processes

- Key business workflows
- Critical business rules
- Validation requirements
- Integration points
- Regulatory considerations

Feature Development Process

- Requirements gathering
- Technical design documentation
- Testing requirements
- Deployment considerations
- Documentation updates

4. Development Workflow

Task Management

- Project management tool access
- Task assignment process
- Status updating
- Time tracking (if required)
- Documentation requirements

Code Management

- Branch strategy
    - Feature branches
    - Release process
    - Hotfix procedure
- Pull request workflow
- Code review process
- Merge requirements

Deployment Process

- CI/CD pipeline overview
- Build validation
- Deployment stages
- Release verification
- Rollback procedures

5. Support and Escalation

Development Support

- Primary technical contact
- Business process questions contact
- Emergency contact procedure
- Common issues documentation
- Known limitations

Quality Assurance

- Testing requirements
- Test environment access
- Test data management
- Bug reporting process
- Performance testing guidelines

6. Documentation Responsibilities

Required Documentation

- Technical design documents
- API changes
- Database changes
- Configuration updates
- Business rule implementations

Knowledge Sharing

- Code comments standards
- README updates
- Wiki contributions
- Common issues documentation

Implementation Plan

Phase 1: Initial Setup (Week 1)

- Development environment setup
- Access provisioning
- Initial documentation review

- Tool access and training

Phase 2: Guided Development (Weeks 2-3)

- Architecture walkthrough
- Business process review
- Paired programming sessions
- Initial task assignment

Phase 3: Independent Development (Week 4+)

- Regular check-ins
- Code review participation
- Documentation contributions
- Increasing task complexity

Next Steps

1. Create detailed environment setup guide
2. Document current codebase structure
3. Set up project management tool templates
4. Establish review and support procedures

[Note: This framework should be customized based on specific project requirements and team structure]