



AIRPORTS COMPANY
SOUTH AFRICA

API Management

Scope of Work

Glossary

Acronym	Description
ACSA	Airports Company South Africa
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CUPPS	Common Use Passenger Processing Service
CUSS	Common Use Self-Service
CUTE	Common Use Terminal Equipment
CUWS	Common Use Web Services
EDI	Exchange Data Interface
ESB	Enterprise Service Bus
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IATA	International Air Transport Association
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation

JMS	Java Message Service
ODBC	Open Database Connectivity
MQ	Message Queuing
MQI	Message Queuing Interface
REST	Representational state transfer
RFP	Request for Proposals
SOAP	Simple Object Access Protocol
SOW	Scope of Work / Statement of Work
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
XML	eXtensible Markup Language

TABLE OF CONTENTS

1.	INTRODUCTION	6
1.1.	PURPOSE.....	6
1.2.	OBJECTIVE.....	6
1.3.	BACKGROUND.....	6
1.4.	CHALLENGES	7
1.5.	KEY FEATURES & CAPABILITIES	7
2.	RFP SCOPE.....	8
2.1.	IN SCOPE.....	8
2.2.	OUT OF SCOPE.....	8
3.	REQUIREMENTS.....	8
3	CONCEPTUAL DESIGN	16
4	REQUIRED SERVICES FROM THE BIDDER	17

1. Introduction

1.1. Purpose

Airports Company South Africa SOC Ltd (ACSA) hereby invites Bidders to submit proposals for Procuring, Implementation and Support of an API Management solution.

1.2. Objective

ACSA requires an API Management Solution that will enable it to seamlessly integrate with other Applications and Services, i.e., both On-premises and Cloud. The solution should enable ACSA to publish and consume APIs from different application interfaces.

1.3. Background

ACSA currently utilises the Enterprise Integration Bus (ESB) and Oracle, SQL Database.

ACSA requires an API Management solution that will enforce ACSA's policies, control access, collect and analyze usage statistics and report on performance. The solution must also integrate with the existing solutions specified above. The solution must have the following core components:

- **API Gateway:** a server that acts as an API front-end, receives API requests, enforces throttling and security policies, passes requests to the back-end service and then passes the response back to the requester.
- **API Lifecycle Management:** ability to define or publish APIs.
- **API Developer portal:** community site that is a source of information (including documentation, tutorials, sample code, software development kits, an interactive API console and sandbox to trial APIs and obtain support from the API provider and user community).
- **API Analytics:** functionality to monitor API usage as well as diagnostics.

1.4. Challenges

- **Agility.** Integrations between ACSA and third parties take time, resulting in project delays.
- **Complexity.** The complexity of the Integrated products necessitates advanced technical skills to use them successfully.
- **Reporting and analytics.** No visibility into different interfaces and integration failures.

1.5. API Management Solution Key Features & Capabilities

- Support different third-party connectors and adaptors to a wide variety of applications, either hosted On-Premises or Cloud.
- Faster development of complex integration flows and scenarios using pre-built connectors/integration templates, i.e., to improve time to market.
- Enable data transformation and transmission effortlessly, e.g., CSV, XML, JSON, Low-code, etc.
- Support for real-time and batch integration.
- Security features such as access management, LDAP integration and encryption.
- Centralized platform for administration and monitoring, e.g., resource utilisation and system health.
- Ability to provision, manage and monitor APIs, i.e., API Management.
- Keep APIs up to date with API changes and services.
- Ability to manage API to discover all available services.
- Control and manage API access authentication.
- Scalable to be able to manage additional traffic and requests.

2. RFP Scope

2.1. In scope – API Management Solution must be able to do the following:

- Integration to current and future ACSA solutions and services.
- Pre-built connectors to integrate different applications and data sources.
- Handover to ACSA IT Technical team.
- Testing of the solution to demonstrate conformance to requirements.
- Support and maintenance to be in place for five years to assist ACSA in resolving identified issues. The Service Provider also needs to assist ACSA with complex integrations that it cannot achieve independently.

2.2. Out of Scope

- Any requirement that is not explicitly described in this document.

3. Requirements

3.1 API Management Solution must be able to do the following :

API management supports APIs' planning, design, implementation, publication, operation, versioning, and retirement. API management implements key governance features and manages access to any endpoint that exposes an API (i.e., whether On-Premises, in the Cloud, Mobile or IoT).

3.1.1 API Lifecycle Management

- 3.1.1.1 The solution must support the entire lifecycle of API delivery (build, deploy and manage) for both On-Premises and Cloud applications.
- 3.1.1.2 The solution must be able to provide APIs for internal and external consumption.
- 3.1.1.3 The solution must be able to set up and administer integrations using a user-friendly interface that requires low coding.

3.1.2 Communication Protocol Connectors

- 3.1.2.1 The solution must be able to interact with applications and data structures using different protocols (such as HTTP, HTTPS, FTP, TCP/IP, AMPQ, JMS, ODBC, etc.).

3.1.3 Application and Data Connectors

- 3.1.3.1 The solution must be able to establish connectivity to different applications and data sources via a variety of adapters and interfaces. The following are critical applications used at ACSA: Airport Management Suite (On-Premises), Apex (On-Premises), Oracle E-Business Suite (On-Premises), Oracle Taleo (Cloud), Oracle Fusion (Cloud), Oracle Learn (Cloud), ServiceNow (Cloud), Microsoft SharePoint (On-Premises), Active Directory (Cloud), E-Gates, SAC, SCORE, 2D Barcode, BRS, ACSA website (Flight Information), Aircraft Docking System, Oracle ERP (Billing Data), Ground Radar (ATNS), Retail and many more.

3.1.4 File Transfer/Movement

- 3.1.4.1 The solution must be able to distribute files between systems on a automated and scheduled or manual basis, e.g., FTP directory or HTTP URL.
- 3.1.4.2 The solution must have a retry mechanism where it can send unsuccessful request.

3.1.5 Messaging

- 3.1.5.1 The solution must be able to distribute messages between endpoints. The messaging broker platform must enable publish-subscribe, message queueing or both.

3.1.6 Protocol Mapping

- 3.1.6.1 The solution must be able to map between different endpoint protocols, such as a file to a message, application connector to the data source or any combination of points.

3.1.7 Data Formats

- 3.1.7.1 The solution must be able to process different data formats such as XML, JSON, and flat files, etc. and much more.

3.1.8 Routing and Orchestration

- 3.1.8.1 The solution must be able to route data between endpoints and orchestrate the flow of information across those endpoints.

3.1.9 Data Mapping and Transformation

- 3.1.9.1 The solution must be able to map, transform, aggregate and split data between different data formats and industry standards.
- 3.1.9.2 The solution must provide standard tools and functions to convert data values from a source system's data format into a destination system's data format.

3.1.10 Data Lifecycle Management

- 3.1.10.1 The solution must manage data flow throughout its lifecycle: from creation and initial storage to obsolescence and deletion.

3.1.11 Master Data Management

- 3.1.11.1 The solution must support managing master data by removing duplicates and using rules to eliminate incorrect data from entering the system.

3.1.12 Data Standards

- 3.1.12.1 The solution must integrate different data sources (Oracle, MS SQL, Hadoop, DB2, PostgreSQL, etc.) using industry standards, e.g., data structure, data content, data value, and data communication. The solution must be data-agnostic, i.e., manage Structured, semi-structured, and unstructured data.
- 3.1.12.2 The solution must integrate into different Businesses using Open and Industry Standards enabled; ACI ACRIS and IATA AIDX compatible.

3.1.13 Data Integration

- 3.1.13.1 The solution must integrate with both On-Premises and Cloud data sources.
- 3.1.13.2 The solution must be able to schedule jobs based on defined rules or processes.
- 3.1.13.3 The solution must be able to manage streaming, real-time, event-driven, and batch data.
- 3.1.13.4 The solution must change with the evolving data sources landscape and be able to manage new data source standards introduced in the IT industry, e.g., Cassandra, HBase, Neo4j, Arango, Mongo and etc.
- 3.1.13.5 The solution must support bi-directional data synchronization between ACSA and 3rd parties.

3.1.14 API Standards

- 3.1.14.1 The solution must support multiple open and modern API standards such as JMS, MQI and Web Services (SOAP & REST) models, etc.

3.1.15 Integration

- 3.1.15.1 The solution must seamlessly integrate with Enterprise Service Bus.

3.1.16 Reusable APIs

- 3.1.16.1 The solution must be able to discover existing API integrations for reuse.
- 3.1.16.2 The solution must have the ability to reuse schemas and data models.

3.1.17 Testing

- 3.1.17.1 The solution must be able to test integrations before deploying them to a production environment, i.e., without interfering with the live production systems.

3.1.18 Version Control

- 3.1.18.1 The solution must have version control capabilities to enable roll-back when APIs are incorrectly provisioned.
- 3.1.18.2 The solution must be able to record version history.

3.1.19 API Monitoring

- 3.1.19.1 The solution must manage and monitor created APIs to ensure they work as intended. The solution must notify the relevant stakeholders if there are issues with the created APIs, e.g., the API is down due to network problems.

3.1.20 API Collaboration and Community

- 3.1.20.1 The solution must have a community forum containing an API provider blog, Developer discussion forums, and Developer issue reporting.

3.1.21 Monetization

- 3.1.21.1 The solution must assist ACSA in collecting revenue based on API usage, i.e., have the ability to bill stakeholders based on API usage.
- 3.1.21.2 The solution must be able to set up pricing rules based on usage, load and functionality.

3.1.22 Emerging Technologies

- 3.1.22.1 The solution must integrate emerging technologies as well as evolve and manage new API standards introduced in the IT industry, e.g., Swagger 2.0 and GraphQL, etc.

3.1.23 Reporting and Analytics

- 3.1.23.1 The solution must provide an integration landscape (including data source integrations and services offered), i.e., API Catalog.
- 3.1.23.2 The solution must also have a graphical representation of data structures (including data fields that have been mapped).
- 3.1.23.3 The solution must be able to provide statistics on the utilisation of APIs.

- 3.1.23.4 The solution must report on top applications using API's.
- 3.1.23.5 The solution must provide a Centralised monitoring dashboard for all integration flows in real-time.
- 3.1.23.6 The solution must report on system health, such as any issues that exist in the environment.
- 3.1.23.7 The solution must provide insights into performance and capacity.
- 3.1.23.8 The solution must allow ACSA to customise its reports.
- 3.1.23.9 The solution must report on security compliance and any other vulnerabilities that must be addressed.

3.2 Non-Functional Requirements

3.2.1.1 Hosting

- 3.2.1.1.1 The solution must be hosted in a Tier 2 level or more data center, i.e., Cloud-based.
- 3.2.1.1.2 Regulatory and compliance certificates must be provided an ISO/IEC 27001 certificate.
- 3.2.1.1.3 The platform must be a standalone service and not an embedded subset of another SaaS application.

3.2.1.2 Platform performance (Speed & Latency)

- 3.2.1.2.1 The solution must respond quickly when users are working on it, i.e., not click a section and wait for more than 5 seconds.
- 3.2.1.2.2 The solution must be able to manage volumes during peak times.
- 3.2.1.2.3 The solution must cater for bandwidth constraints and geographically dispersed systems.

3.2.1.3 Scalability

- 3.2.1.3.1 The solution must cater for future growth, e.g., adding new APIs, Connectors, i.e., no limitation to the number of APIs that can be added.
- 3.2.1.3.2 The solution must be able to expand functionality while maintaining balanced loads. It must also have the ability to serve growing demand without reducing functionality.

3.2.1.4 Usability

- 3.2.1.4.1 The solution must be easy to use with minimal user training, i.e., Web-based.

3.2.1.5 Reliability & Availability

- 3.2.1.5.1 The solution must be available 24/7 with a minimum availability of 99.8%.
- 3.2.1.5.2 The solution must cater for high availability.
- 3.2.1.5.3 The solution must be backed up daily and have offsite backup storage.
- 3.2.1.5.4 The solution must allow users to recover deleted data by requesting a restore or independently recovering the deleted data.

3.2.1.6 **Security**

- 3.2.1.6.1 The Service Provider must provide ACSA with their security best practices or controls detailing how they secure their solution.
- 3.2.1.6.2 The solution must ensure that data is transmitted in a non-readable format (encrypted) and has strong key management. The solution must provide encryption capabilities for stored data to ensure that data at rest is protected.
- 3.2.1.6.3 The Service Provider must ensure that Server-level security features are in place for the solution. They must provide information related to the following: patching, anti-virus, vulnerability scanning, intrusion detection with real-time alerts etc.
- 3.2.1.6.4 The Service Provider must ensure that Data Center security features are in place. They must provide information related to the following: Physical security measures, which include an integrated security management solution such as around-the-clock on-site security personnel, video surveillance, and monitoring—as well as industry-leading policies and practices.
- 3.2.1.6.5 The solution must also detect anomalies in functionality, user accessibility, traffic flows, and tampering.

3.2.1.7 **Authentication**

- 3.2.1.7.1 The solution must uniquely identify users and authenticate them. Administrator accounts must be segregated from standard user accounts.

3.2.1.8 **Authorization**

- 3.2.1.8.1 The solution must enable users and/or role-based permissions to be configured to control what solution features and data users can access.

3.2.1.9 **Audit Trail**

- 3.2.1.9.1 The solution must keep an audit trail of all activities performed in the solution (includes but is not limited to the following: who created, updated, and deleted (must be authorised by super users) the record, with time and date stamp.

3.2.1.10 **Assurance**

- 3.2.1.10.1 The solution must maintain data integrity and quality. The solution must be a single source of truth in terms of data and quality.

3.2.1.11 **Availability**

- 3.2.1.11.1 The solution must be secured to prevent denial of service to ACSA users. It must also provide threat protection.

3.2.1.12 **Asset Protection**

- 3.2.1.12.1 The solution must protect ACSA data from being viewed by unauthorised personnel.
- 3.2.1.12.2 The solution must limit access to suspicious visitors and monitor for traffic spikes to prevent overloads like DDoS attacks.

3.2.1.13 **Privacy and data ownership**

- 3.2.1.13.1 The solution must comply with ACSA's Information Security policies and standards (to be provided to the Service Provider once the contract agreement is awarded).
- 3.2.1.13.2 The solution must comply with POPI Act and other related laws or regulations.
- 3.2.1.13.3 All data remain the property of ACSA.
- 3.2.1.13.4 The Service Provider must detail their conformity to data / Information Security Management / ISO/IEC 27001.

3.2.1.14 **Solution Accessibility**

- 3.2.1.14.1 The solution must be accessible in one central platform, i.e., ACSA must only use one interface instead of multiple products.
- 3.2.1.14.2 The solution must be accessible via laptops, desktops, and mobile devices.

3.2.1.15 Disaster Recovery

- 3.2.1.15.1 The solution must have an alternative way to ensure business continuity in cases with an unfortunate downtime event.
- 3.2.1.15.2 The solution disaster recovery must be tested at least once annually and also be audited by an external audit company.

3.2.1.16 Local Support

- 3.2.1.16.1 The solution's first-line support solution must be based in South Africa (international support can form part of the 2nd and 3rd line support).

3.2.1.17 Environments (Development, Quality Assurance and Production)

- 3.2.1.17.1 Multiple environments, i.e., Production and non-production environments. The solution must be able to migrate a customised development environment to a quality and production environment.

3 Conceptual Design

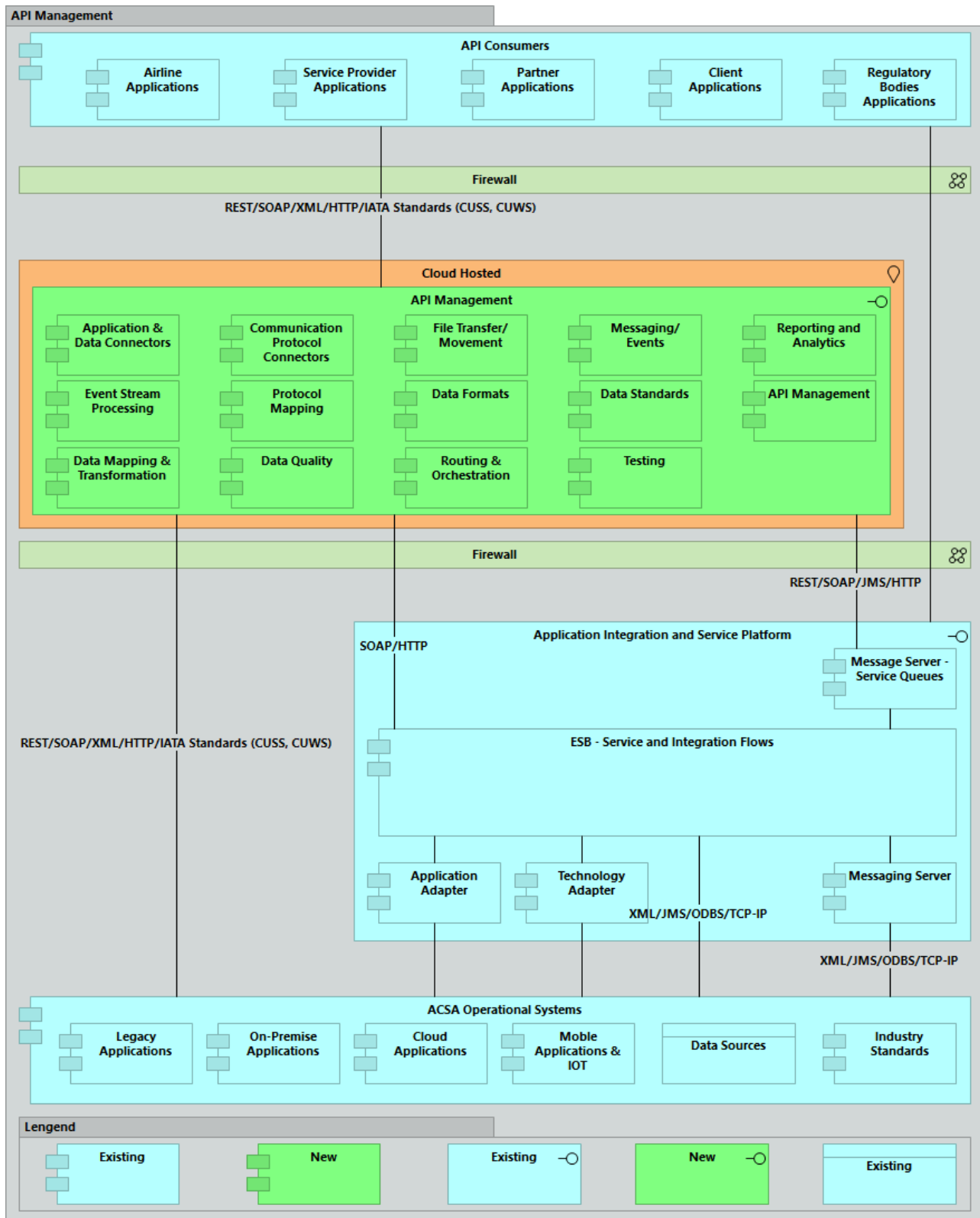


Figure 1 API Management Conceptual Design

4 Required Services from the Bidder

It must be noted that these requirements may be skewed towards current business processes. Therefore, it must be regarded as minimum requirements and that new innovative methods and processes are welcome to assist ACSA in determining the most cost-effective and efficient 'best practice' solution to meet our needs. The bidders' proposals should clearly indicate the following:

4.1 Solution implementation

- Implement the API Management solution.
- Provide skilled resources to ensure that the solution is implemented in accordance with the requested standards.

4.2 Quality assurance

- Testing: Unit, functional, performance, stress, and vulnerability testing.
- The solution must be fit for purpose.
- The solution delivery must adhere to timelines.
- The solution must be delivered according to specifications and service level agreement.

4.3 Support and Maintenance services

Refer to the maintenance and support SOW attached

4.4 Service availability

- The solution must have high availability of 99.8% uptime.

4.5 Reporting

- The Service Provider will be expected to provide the weekly progress report to an ACSA representative during implementation.
- The Service Provider must provide monthly reports to ACSA after implementation on the following:
 - Usage of the solution.
 - Solution availability and downtimes.
 - Number of calls logged and status.

4.6 Training

The bidder is expected to conduct training for the following user groups:

- Administrators, Technical, IT Help Desk and End User training across all nine (9) airport sites and corporate office.

4.7 Documentation

The Service Provider must produce the following project-related documentation during the implementation of the project:

- Project Management deliverables as per ACSA Methodology.
- Architectural design.
- Functional Specification.
- Technical Specification.
- Data Model (an abstract model that organises data elements and standardises how they relate to one another and the properties of real-world entities).
- Diagram showing solution components on the following layers (Business, Application, and Infrastructure). Component Diagram showing allocation of responsibilities within the product and its internal interfaces.
- Context Diagram showing interfaces.
- Quality Assurance Specific Documentation (Test plan, Test cases, test results for different types of solution testing (unit, functional, performance, stress, vulnerability)).
- Technology Requirements Stack Diagram and system components
- Service and Interface definitions
- Operational Manuals
- Infrastructure requirements and architecture.
- Training Manuals.
- Installation guideline document.

4.8 Solution Guidelines

- The solution must provide the functions and services required to support the business capability.
- There should be a single application to support a given business capability the solution must not re-implement a capability already available in the portfolio unless it is replacing the current one.
- The solution must be as secure as business requirements dictate.
- The solution must meet legal and conformance requirements, including those for privacy.
- The solution must provide adequate performance and responsiveness.
- The solution must scale, without redevelopment, for anticipated volume increase for the next five years, from implementation
- The solution must be reliable and easily recoverable.
- The solution must validate input data and maintain the integrity of any added, updated, or exported data.
- The solution must provide APIs which allow services to be accessed via an interface conforming to industry standards adopted by ACSA, e.g., Web Service (REST, SOAP), etc.
- The solution must avoid “hard coding” of values, i.e., any variables likely to change must be externalised to the database or parameter/rule files.
- The solution must trap errors and report them in a meaningful and persistent way.
- The solution end-user interfaces must be intuitive and standards-based to facilitate ease of adoption, reliable usage, and reduced training requirements.
- The solution must be documented to a standard that facilitates:
 - Ease of installation and configuration.
 - Ease of operation by end users.
 - Easy problem determination and resolution.
 - Impact analysis for change requests.
 - Ease of adaptation when required; and
- The solution must not expose ACSA to undue risk.