CAN 5.0

Requirements document

JUNE 10, 2019

AVANSEUS TECHNOLOGIES PVT. LTD.

Table of Contents

I. Revision history

Date	Created / Modified by	Reviewed by	Comments
04-08-2020	Abhilash	Chiranjib Bhandary	Initial draft
05-08-2020	Abhilash	Chiranjib Bhandary	Review comments incorporated

II. Release philosophy and features

The main focus of this release is on SaaS enablement. While the ease of operationalization had been the theme on the Can 4.0 release, the new release focus on improving the user experience while providing the agility of a regular click-to-install kind of software. The new features will bring in enhanced semantics that will provide more relevant insights to the user while providing the flexibility of low level programming with in the CAN environment according to the user requirement. The integration modules with existing OSS of telecom ecosystem will bring in a seamless integration and possibilities of enhanced automation which along with the NLP assistance will make CAN an ultimate solution for the communication service providers. Further, the enhanced OWASP compliance duly certified by external agency will ensure that there are no security backdoors open for hackers. The centralized knowledge management ensures knowledge sharing across CAN implementations. The CAN 5.0 also comes with a cloud native environment compatibility. The decentralized architecture enables distributed implementation compatible with leading container and orchestration software making it compatible to virtual environments like never before.

The main features of CAN 5.0 release are:

- UX/UI redesign and ergonomics
- Integrated Development Environment
- PM counter integration with prediction / Performance degrade correlation with prediction / Health index based monitoring
- Voice based interaction
- Automated generation of prediction KPI reports based on other categorization such as Cause Types, Site Profile, Zone/Location, SA/NSA and severity of alarms
- Integration with Remedy & Splunk
- Weather integration
- Identification of different domains in "Cross Domain Correlation" module
- Containerization for cloud native environments
- Security by design to enable OWASP compliance
- Knowledge repository maintenance for sharing anonymous knowledge across CAN implementations for more accurate predictions

III. Terminologies

Requirements are classified based on type & priority.

a. Requirement types

Requirement type	Definition
Business	Business requirement deals mainly with business goals and stakeholder expectations and tells us about the future state of the product and why the objective is worthwhile.
Functional	Functional requirements are much more specific and detailed compared to business requirements. They outline how a product will support business requirements and specify the steps on how the requirement will be delivered.
Non-functional	The non-functional requirement elaborates a performance characteristic of the system. These requirements fall in areas such as accessibility, documentation, efficiency, disaster recovery, security etc.,

b. Requirement priorities

Priority	Semantics
Critical	A critical requirement without which the product is not acceptable to the stakeholders
Important	A necessary but deferrable requirement which makes the product less usable but still functional
Desirable	A nice feature to have if there are resources but the product functions well without it

IV. Requirements

1. UX/UI redesign and ergonomics

Type	Business requirement
Priority	Critical

UX/UI redesign and ergonomics provides optimized UI that is more captivating and easy to use for the customer. This involves remapping of existing functionalities according to their significance, placement of semantics in such a way that maximum details are conveyed to customer every screen they browse through. This also includes adding additional filters to enable enhanced searching enabling clearer inferences and improving the relevance of CAN 5.0 in the application front.

Requirement ID	Requirement description
REQ05001	Redesign of the CAN 5.0 outlook.
REQ05002	Renaming of the functional tabs
REQ05003	Inclusion of additional filters for data output

2. Integrated Development Environment

Type	Functional requirement
Priority	Critical

Integrated development environment enables customer to use Java programing to alter or extend the functionality of CAN. It makes the software more user friendly and more relevant for its users who wishes to put their coding skills into data modelling or data preparation for CAN 5.0. This will act as an interface to customize the data input or output based on customer requirement.

Requirement ID	Requirement description
REQ05004	Syntax highlights in the Integrated development environment
REQ05005	To highlight the compilation error and warning in red and orange color respectively
REQ05006	A symbol on hovering shows the overall status of Java code
REQ05007	Enable download options to download source code
REQ05008	Enable autocomplete on dot (.) operators
REQ05009	Enable auto indentation shortcut using Ctrl + I

3. PM counter integration with prediction/ Performance degrade correlation with prediction

Type	Functional requirement
Priority	Critical

Integration of the network performance counters with the Avanseus-CAN Artificial Intelligence engine. This will enable CAN to understand the network performance counters and predict the future status of such counters that could be used as forecast for traffic and quality issues related to such networks and create a health index for the user that can monitor the health status of network nodes on real time and futuristically.

Requirement ID	Requirement description
REQ05010	Performance counter/KPI breach prediction.
REQ05011	Alarm prediction by correlating Performance counter values(Alarm superposition)
REQ05012	Matching report for KPI breach prediction.
REQ05013	Matching report for Alarm prediction.
REQ05014	KPI limit threshold configuration.
REQ05015	Health index configurations and calculations.

4. Voice based interaction

Type	Functional requirement
Priority	Critical

To enable Voice Based Interaction with the Avanseus-CAN Artificial Intelligence engine. CAN users will be able to access the semantics through voice commands initiated through microphone attached with the CAN console. CAN will provide appropriate responses to such voice commands making the customer – CAN communication an interactive one.

CAN 5.0 will have the basic version of speech interface and subsequent releases will improvise this feature.

Requirement ID	Requirement description
REQ05016	To provide answers to user’s queries in the voice form as well as it also displays the query result on the screen. The queries must be in English (US-English).
REQ05017	To enable users to raise queries related to the prediction data
REQ05018	To suggest alternatives if queries are ambiguous. If the queries are not understood, CAN will inform the same to user.

5. Automated generation of prediction KPI reports based on other categorization such as Cause Types, Site Profile, Zone/Location, SA/NSA, and Severity of alarms

Type	Functional requirement

Priority	Important

To enable Avanseus CAN to generate automated KPI reports on different fields based on customer requirement. This feature addition would enable the CAN users to schedule report generation based on different criteria reducing the manual intervention increasing the CAN agility. The users can also use the appropriate filters to understand the KPI values of the selected fields.

Requirement ID	Requirement description
REQ05019	To, automatically/on selection, generate weekly KPI reports for Cause types, Zone/Location, SA/NSA, severity of alarms on weekly basis or real time basis
REQ05020	Allow users to download the report

6. Integration with Remedy & Splunk

Type	Functional requirement
Priority	Important

To enable seamless operations of Avanseus CAN with Third Party software. Though the individual scope of integration is different for different third party software, the vision of integration remains same as to reduce the manual intervention and automate the transactions.

6.1 Integration with Remedy

Requirement ID	Requirement description
REQ05021	Generate the incident tickets
REQ05022	Pull out the ticket information
REQ05023	Allocate the tickets to the technician if the client has given the technician details

6.2 Integration with Splunk

Requirement ID	Requirement description
REQ05024	To automatically pull the alarm data at specific time daily as well as on real time.
REQ05025	To create different data search criteria and facilitate data pull

7. Weather integration

Type	Functional requirement
Priority	Important

To make Avanseus CAN Artificial Intelligence engine more susceptible to real time conditions that include environmental parameters. Weather integration will enable the CAN to understand the weather pattern and consider the related parameters while predicting the network performance, incidents or while scheduling a field visit. This will enable CAN to take more informed decisions that would make better sense to CAN users.

Requirement ID	Requirement description
REQ05026	To provide appropriate predictions and suggestions based on the weather patterns/considering the weather patterns
REQ05027	To enable dashboard view and download of weather prediction report and associated actions.

8. Identification of different domains in "Cross Domain Correlation" module

Type	Non-functional requirement
Priority	Important

To create visual differentiators for elements from different domains in the existing Cross Domain Correlation view of Avanseus CAN for Map view, Block view as well as Bit Pattern view. This will enable CAN users to understand the domain differences easily and hence the fault predictions.

Requirement ID	Requirement description
REQ05028	Tag cluster nodes across domain in the individual cluster view
REQ05029	Tag cluster nodes across domain in the block view and map view

9. Containerization for cloud native environments

Type	Non-functional requirement
Priority	Important

To enable CAN installation in cloud native environment. This feature will enable CAN to be compatible with range of container tools including Docker enabling automation of deployment, scaling and management through various orchestration applications including kubernetes.

Requirement ID	Requirement description
REQ05030	Enabling container tool compatibility

Requirement ID	Requirement description
REQ05031	Enabling orchestration tool capability
REQ05032	Enabling distributed architecture for decentralized implementation in native cloud environment

10. Security by design to enable OWASP compliance

Type	Non-functional requirement
Priority	Important

To enable CAN compliance with Web Application Security as mentioned by OWASP. This will reduce the risk of CAN deployment in customer environment where multiple applications co-exist. OWASP top 10 vulnerabilities for web application are divided as follows:

- Broken authentication
- Information Leakage
- Session management
- Role-based access control
- Parameter manipulation
- Error handling
- Insecure Transport Layer
- Injection Vulnerabilities
- Insufficient logging
- Insecure deserialization

However, the above OWASP vulnerabilities are clubbed and elaborated into more sub-requirements as shows below:

1. Information gathering

   Web applications may inadvertently disclose information that is useful to the attacker by means of verbose response headers, error messages etc. or by using common conventions, such as an admin interface being located in “/admin/”. Furthermore, some of these error messages may be cached by search engines long after the message has been remedied in the application. The first phase in security assessment is focused on collecting as much information as possible about a target application.

Requirement ID	Requirement description
REQ05033	Spiders, Robots and Crawlers
REQ05034	Search Engine
REQ05035	Identify application entry points
REQ05036	Application Discovery
REQ05037	Analysis of Error Codes

2. Configuration management

Secure web application must be deployed on secure infrastructure. In this control area, the immediately supporting infrastructure is analyzed for various misconfigurations that can be of an advantage to the attacker, for example, if application is deployed on top of a webserver, does it use file extensions (.php, .aspx, .jsp, .pl) to handle dynamic programming? If so, then possibly by uploading a file with such extensions could allow attackers to take over the web server and circumvent the application security.

Requirement ID	Requirement description
REQ05038	SSL/TLS Testing (SSL Version, Algorithms, Key length, Digital Cert. Validity)
REQ05039	DB Listener Testing
REQ05040	Infrastructure Configuration
REQ05041	Application Configuration
REQ05042	Testing for File Extensions Handling
REQ05043	Old, backup and unreferenced files
REQ05044	Infrastructure and Application Admin Interfaces
REQ05045	Testing for HTTP Methods and XST

3. Authentication

Almost every web application requires some form of user authentication (establishing identity of the user) to provide additional functionality, for example, to alter content in a content management system, administrators must authenticate themselves. Authentication mechanism are inspected in detail to examine the possibility of altering or intercepting authentication data to gain additional access to the system. For example, common usernames and passwords are checked, such as admin/admin.

Requirement ID	Requirement description
REQ05046	Credentials transport over an encrypted channel
REQ05047	Prevent user enumeration
REQ05048	Prevention of Guessable (Dictionary)
REQ05049	Testing for CAPTCHA
REQ05050	Testing for Race Conditions
REQ05051	Testing for bypassing authentication schema
REQ05052	Brute Force Testing

4. Session management

HTTP is a stateless protocol and does not have a concept of a user's session built-in. In order to avoid continuous authentication for each page of a website or service, web applications implement various mechanisms to store and validate credentials for a pre-determined timespan. These session mechanisms are subject to common risks and flaws that may lead to unauthorized access to additional functionality or can be abused to force users to unwillingly and unknowingly execute an action in the system using social engineering tricks. For example, a common error is to rely on usernames stored in a browser cookie in a way that can be easily manipulated by the attacker.

Requirement ID	Requirement description
REQ05053	Testing for Session Management Schema
REQ05054	Prevention of Cookies attributes access and manipulation
REQ05055	Prevention of Session Fixation
REQ05056	Prevention of Exposed Session Variables
REQ05057	Prevention of CSRF

5. Business logic testing

Each purpose-built web application will have a specific set of requirements and restrictions specific to the business environment it operates, for example, a junior employee may not authorize transactions over a specific sum or may not authorize transactions where he/she is the initiating party to preserve segregation of duties. To conduct business logic testing, the analyst first builds an understanding of what specific business rules and restrictions must be in place and then attempts to bypass these restrictions using a variety of tests such as form field tampering, forced browsing etc.

Requirement ID	Requirement description
REQ05058	Testing for business logic

6. Data validation testing

Web applications must accept only valid data, e.g. only valid dates, no spaces in e-mail, only plain text in comments areas. If such checks are not enforced, attackers may hijack the execution flow of the program, for example by inserting a portion of a SQL statement in a lookup query that uses user-supplied input, e.g. instead of specifying first name like “John”, attackers may input “John’ OR 1=1;--’ and possibly obtain output of all users in a directory that may be otherwise unavailable, or use this to extract data from other tables or gain a foothold in the underlying operating system. In this control area, we check if correct user input syntax is enforced and if not, what can be gained from abusing weak data validation functionality.

Requirement ID	Requirement description
REQ05059	Prevention of Reflected Cross Site Scripting
REQ05060	Prevention of Stored Cross Site Scripting
REQ05061	Prevention of DOM based Cross Site Scripting
REQ05062	Prevention of Cross Site Flashing
REQ05063	SQL Injection
REQ05064	LDAP Injection
REQ05065	ORM Injection
REQ05066	XML Injection

Requirement ID	Requirement description
REQ05067	XPath Injection
REQ05068	IMAP/SMTP Injection
REQ05069	Code Injection
REQ05070	OS Commanding
REQ05071	Buffer overflow
REQ05072	Incubated vulnerability
REQ05073	Prevention of HTTP Splitting/Smuggling

11. Knowledge repository maintenance for CAN training and modelling

Type	Non-functional requirement
Priority	Important

To create CAN knowledge repository that will act as a reference point for CAN applications during its initial configuration as well as in time of managing complex issues. Knowledge repository will store specific learnings made by the application and will get constantly updated. It will act as an external boosting input for CAN application while it makes day to day predictions.

Requirement ID	Requirement description
REQ05074	Create knowledge repository with knowledge management for CAN application
REQ05075	Design on I/O mechanism to operationalize the repository.

V. Intended Use Cases/User Requirements

User Type/Class	Use Cases
CXOs/Decision Makers	Network performance overview over voice interface, Performance Counter based network planning, Predictive Analytics based OPEX allocation
Middle Managers	Network fault management overview, performance overview, field actions management
Shift Engineers	Fault management, performance management, field assessment, ticket booking, spare management, optimization of CAN outputs through IDE, KPI assessment
Implementation/Integration engineers	Dynamic implementation in virtual environments, cloud ready implementation, access and rights control, change monitoring, data protection

VI. Intended Environment

a) Any Linux/UNIX environment
b) AWS/Microsoft Azure/GCP environment
c) Docker/Kubernetes