APPENDIX F – SOLUTION ARCHITECTURE

1. About this Document
The document provides guidelines for MSP to design eGCA solution based on the broad Technical Architecture presented in the document. It is expected that MSP design the solution on world class standards and consider Government of India’s policy on adoption of open source software (MeitY vide F. No. 1(3)/2014-EG II).

Note: MSP is required to comply with guidelines contained in this document.

2. Design Consideration for eGCA System

<table>
<thead>
<tr>
<th></th>
<th>Ecosystem Approach</th>
<th>Usability</th>
<th>Provision for Sustainable, Scalable Solution</th>
<th>Distributed Access and Multi-channel Service Delivery</th>
<th>API Approach and Use of India Stack</th>
<th>Configurability using Business Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>It is intended to develop common platform to allow for stakeholders to seamlessly interface with DGCA via eGCA system portal or via ecosystem provided applications for various activities.</td>
<td>System user interface should be user friendly and easy to use and be 'self-serviced' by end user.</td>
<td>MSP should provide solution that would be sustainable for the next few years. The expectation is that the system should sustain at least 10 years from GO-Live. In this regards, consideration to scalability is paramount.</td>
<td>The solution proposed by MSP should allow stakeholders to interact through multiple channels/interfaces with eGCA system. For the same, an important consideration for designing is access devices and their screen capabilities (including browser variations). MSP should consider multiple channel access requirement (mobile, tablets, etc.) while designing the solution.</td>
<td>The aim of the project is to reuse the already built in API components which are available with GOI such as India Stack. Additionally, the system should be completely API driven, wherein, all functionalities should be available in the form of API.</td>
<td>All configurations including policy decisions, business parameters, rules, etc. are to be captured in central system. The system should allow addition/edition/deletion of policies/rules to ensure “single source of truth”. Hence, it is required to decouple business parameters/rules/master data from the rest of the solution architecture. There should be a central interface for managing the configurability by authorized user group.</td>
</tr>
</tbody>
</table>
## Data Driven Decision Making

All the decisions should be data driven. In this regards, it is expected that metadata sets be attached so that time taken by various functions while capturing / entering the data etc. is captured and behaviour of system is verified. This would ensure quality is measured systematically and feedback is given to improve any specific issues that are identified. The following points broadly describe objectives for eGCA, keeping in mind the large ecosystem:

1. **Drive decision making based on data analytics**: The analytics module (along with comprehensive reporting) is to be included to offer stakeholders data driven insights. Processes must be in place to drive a feedback loop overall including partner ecosystem to drive continuous improvement. Every transaction & event including that of the system administrator should go in BI to, accordingly, develop insights for decision.

2. **Empower self-improvement**: Tools, data and platform is to be created to assist stakeholders analyse their performance and operational metrics. MSP is required to capture and report continuous activities through various dashboards.
3. eGCA System Architecture Principles

The IT architecture principles defined in this section are the underlying general rules and guidelines that will drive the subsequent development, use and maintenance of architectural standards, frameworks and future state target architecture.

The overall system is required to be developed on the following architectural principles, which should form the backbone of overall system architecture:

<table>
<thead>
<tr>
<th></th>
<th>Platform Approach</th>
<th>The eGCA system is envisaged to be a faceless system with 100% API driven architecture. In this regards, Open APIs should be used to ensure openness, multi-user ecosystem, specific vendor/system independence, and ability to provide usage of various devices (mobile, tablet, etc.) that are built on top of such APIs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Performance</td>
<td>Leading technologies should be proposed for solution to ensure high performance. Design should ensure that performance of various modules (especially in case of disaster) are independent of each other to enhance the overall performance. The solution design should cater to the following:</td>
</tr>
<tr>
<td></td>
<td>Modular Approach</td>
<td>- Modular design to distribute appropriate system functions on web and app server</td>
</tr>
<tr>
<td></td>
<td>Run Time Configs</td>
<td>- Increase In-memory Operations</td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>- Reduce number of I/O operations and N/w calls using selective caching</td>
</tr>
<tr>
<td>3</td>
<td>Preference to Open Source &amp; Vendor Neutrality</td>
<td>Solution should be based on Open Source components (as per table provided in the subsequent sections). All non-open source component/service/product/framework/MSP pre-existing product or work proposed must be wrapped in vendor neutral open API. In addition, there must be at least 2 independent OEM products available using same standard to ensure system is not locked in to single vendor implementation.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>
| 4 | Scalability | The component in the architecture should be scalable to handle more user requests or input resources for various modules. Inclusion of additional application functionalities should also be catered to by upgrading the software editions with minimal effort.  
   a) The application functions to be divided logically and developed as modular solution.  
   b) The system should be able to scale horizontally & vertically.  
   c) Data Volume – Ability to support volume growth  
   d) Functionality – Ability to extend functionality of the solution without significant impact to the existing functional components and infrastructure in future |
| 5 | Application and Data Security | The application and data security services should cover (but not limited to) user profile management, authentication and authorization aspects. Security services should run across all layers and authenticate/allow access based on business rights. The system should ensure data confidentiality and data integrity.  
   The application development should be based on Secure Software Development Lifecycle, and secure coding practices (as established in the standards such as OWASP). The system should consider/include the following:  
   a) Database design must consider and incorporate data integrity requirements.  
   b) Role based access for all the stakeholders to access and use the system  
   c) Appropriate authentication mechanism adhering to industry good practice of password policies etc.  
   d) Ability to adopt other authentication mechanism such as Electronic Signature Certificates  
   e) Activities such as anti-spoofing, anti-sniffing, anti-tampering should be taken care for data in transit, as well as data at rest, from internal and external threats.  
   f) Authorization validity to be ensured for the users providing the Data to the system. Data should be accepted only from the entity authorized |
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>g)</td>
<td>Data should be visible only to the authorized entity</td>
<td></td>
</tr>
<tr>
<td>h)</td>
<td>Audit trails and Audit logging mechanism to be built to ensure user action can be established and investigated, if any. Also, the audit trails thus captured, should assist the administrator to identify/avoid data alterations etc. through unauthorized channel.</td>
<td></td>
</tr>
</tbody>
</table>
| 6 | Reliability | Due to high sensitivity, the data transfer and data management should be reliable. The system should have appropriate measures to ensure reliability of data received or accessed through the application. For the same, it would be necessary to ensure the following  
- Prevent processing of duplicate incoming files / data  
- Prevent unauthorized alteration of data uploaded  
- Ensure minimum data loss |
| 7 | Manageability & Agility | It is essential that the application architecture handles failures (such as hardware failure, network outage, software crashes, etc.) effectively. The system must be resilient to failures and have ability to restart with minimal human intervention. All layers of the system such as application, infrastructure must be managed through automation and proactive alerting. |
| 8 | Availability | The solution design and deployment architecture should ensure high availability of the eGCA system (in accordance to the SLA requirements). An important aspect of ensuring the above mentioned availability criteria would be creation of Standard Operating Procedure (SOP) for system upgrades, maintenance and other procedural needs. |
| 9 | Continuous Build & Ease of Deployment | Ease of deployment is a key consideration for eGCA. The process for releasing/deploying software MUST be repeatable and reliable. The deployment process need to be automated and secure such that components can be deployed independently without affecting the existing already deployed components. Microservices based container architecture may be followed through continuous build and continuous deployment process. |
| 10 | Reconstruction of truth | System should NOT allow database / system administrators to make any changes to data. It should ensure that the data and file (data at rest) has tamper resistance capacity. System should be able to detect any data tampering and should be able to reconstruct the truth. |
4. High Level Architecture

4.1 Architecture Overview

The technical architecture is envisaged to be modular having User Interface, Business Services, Data Storage and Infrastructure Layer (Cloud environment). The architecture is also required to have Integration Layer to integrate with external systems and third party systems. Security of eGCA application is a key layer to ensure security of data (being exchanged/managed). Further, it is required that entire eGCA application be managed through an Administration Layer. The eGCA systems architecture should consist of the following high-level components/principles:

a) Should be web enabled and mobile enabled with informational, collaborative and transactional services.
b) Store data in single location though allowing for multiple usage of same data
c) Core system (i.e. system without user interface) should be exposed via APIs
d) User identity authentication will be done through Aadhar authentication. The system should be capable of integrating with the Aadhar APIs .
e) API Layer: eGCA system should expose three distinct sets of APIs
   a. For consumption by businesses (G2B) via various application interfaces
   b. For consumption by government agencies (G2G)
   c. For aviation sector external stakeholders (G2C)

All APIs should only accessible via HTTPS protocol.

f) The service platform of the IT applications stack should contain Data processing and data mining applications, which can provide analytics backbone to eGCA solution.
g) eGCA system landscape should include web portal. The web portal should access the eCGA system service through internal APIs.

h) As the eGCA system may be accessed from remote locations with minimal connectivity, the architecture should be designed to be operated in low-bandwidth environment. The MSP should design a services-frequency-bandwidth matrix, transaction locking, offline processing (e.g. forms download) and other strategies to enable critical services to operate in low/no bandwidth environment.

i) System should provide for role based access to various stakeholders with associated role based functionalities.

j) Solution should allow for audit trail of the user’s access on the eGCA Portal

k) The functionalities and features of eGCA solution should be granular and modular enough for the administrators to enable or disable any particular functionality, at any given time, as per the requirement, without the need for a developer / code level change / custom UI change.

5. Application Stack

It is preferable to deploy open source solutions to build eGCA system, which is modular, scalable and portable across platforms. All the platforms and solutions proposed for eGCA system should be vendor neutral and DGCA can replace any platform without any constraint.

The table below provides some of the products/services that the MSP has to provide to execute the engagement. As already explained before, this table only provides an indicative requirement and it is MSP’s responsibility to bring in all required components required for developing and running the proposed eGCA system in a secure and efficient manner, meeting the overall functional, technical, performance and security considerations towards successful delivery of the system.

<table>
<thead>
<tr>
<th>#</th>
<th>Solution Component</th>
<th>Open Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Database</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Business Process Management Suite</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Web portal</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Rules Engine</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Mobile Solution Framework</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Analytical System</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>BI and Reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>APM solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Enterprise Service Bus</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Content Management Solution</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Scanning solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Bug/ Defect tracking solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>API Gateway</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Learning Management System</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Devops solution (including CI and CD)</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CRM solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>VC solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Identity Management solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Solution for Virtual Assistance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Enterprise Monitoring System</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Directory and Access Management solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Service Level Monitoring System</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Project Management Tool</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### eGCA System Requirements

#### Appendix F

<table>
<thead>
<tr>
<th>#</th>
<th>Solution Component</th>
<th>Open Source</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.</td>
<td>Automated testing</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>Load and performance testing</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td>Version management</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>Test management tool</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>Build Management</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>Operating System</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>Web Server</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td>Application Server</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td>WAF Solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td>Host Intrusion Prevention System</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>34.</td>
<td>Network Intrusion Prevention System</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td>Advanced Persistent Threat Protection solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td>DLP Solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>37.</td>
<td>Database Security Solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>38.</td>
<td>Security Testing Solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>39.</td>
<td>DDoS solution</td>
<td>COTS</td>
<td></td>
</tr>
<tr>
<td>40.</td>
<td>SSL Solution</td>
<td>COTS</td>
<td></td>
</tr>
</tbody>
</table>

---

**If COTS product is considered then the product needs to be in Gartner Leader or Challengers Quadrant.**

The Bidder will be required to provide support for updates, upgrades, security patches etc. for software licenses through the contract period. The Bidder would be required to provide enterprise level support or equivalent for software licenses, covering updates, security patches, issue resolution at software level, bug fixing etc. Bidder should inform DGCA for any updates/upgrades in the software licenses before making any upgrades on IT infrastructure provisioned on Cloud. These updates/upgrades would be tested by the application development teams on the existing application before applying and release of the same in production.
6. Security

Security is one of the utmost important aspects envisaged in the entire solution design of eGCA system. All key dimensions both from proactive and reactive security measures including authentication, authorization, access control, logging and monitoring should be an integral part of the eGCA system architecture.

![Security Framework Diagram](image)

**Figure 2: Security Framework**

The key security requirements are mentioned below:

i. Should support encryption of all data during transmission and local store of data. Local store encryption should be customizable for various levels of encryption required. Encryption does not mean password protection only in this case.

ii. Should provide inbuilt support for digital signature and PKI based on industry standards.

iii. Should provide support for simple, flexible administration via web browser and thick client.

iv. Should support TLS/SSL encryption with at least 1024 bit key.

v. Should provide integrated PKI as a foundation for numerous security features, including digital signatures and encryption; granular access control - down to the individual field level; execution control lists; local data encryption; and trust relationships in multi-organization and extranet applications.

vi. Should allow multi-level passwords to ensure that no one administrator may have full control of the important user credentials.
vii. Should provide reader control at the document/record level to ascertain only selective users may ever be able to search and access such secure and confidential content.

viii. The enterprise security policy should be applicable across various layers. In addition, the security aspect should make API gateway tamper proof with use of PKI (Public key infrastructure) using HSM card.

ix. Public key cryptography is at the heart of XML Web services security. XKMS (XML Key Management Specification) should be designed to simplify the integration of PKI and digital certificates to enable authentication, digital signature, and encryption services.

6.1 Authentication and Authorization

1. The eGCA system should comply with all requirements of security, reliability and non-repudiation as per the Government of India guidelines. For the portal, there should be a provision of logging into the system through Internet as well as Intranet. There should be provision for authentication using digital certificates as per the Government of India guidelines.

   a. Verify users credentials’ before providing access to eGCA system; access should be given based on authorization levels.
   b. Access Management should be used for restricting access to rights-protected content / sections / modules / screens / fields, etc. to authorized users only. Rights to all active users should be granted based on their hierarchy and level in the organization, designation, assigned roles and responsibilities, location etc. among other parameters.
   a. Personal ‘Dashboard’ facility should be available for all the users as first interface within the application. The type of information and content to be displayed on the personal ‘Dashboard’ should be dynamically controlled through the Access Control Module / Admin Module based on the user’s authorization level.
   b. One session should be maintained for each user. However, declared accounts may not be required with such features.
   c. Once any Users credentials are verified, he / she should be able to navigate through all the modules and functionalities of the integrated application, to which that User is authorized to access, based on Role Based Access control (RBAC).
   d. The security solution(s) must monitor all traffic to all resources of eGCA; all access attempts to the system or directly to any resource managed/access by the system, should be intercepted and analyzed for authentication/authorization.
   e. End users should not have access to the database. Only authorized administrators should have this access.
   f. Public user should be able to browse the portal with rights to view public content available on the website; remaining all types of users should enter the solution using appropriate secured authorizations.
   g. The solution should support flexible definition and modification of authentication, authorization, encryption, and data integrity assertions and requirements for each security policy
   h. The solution should support at least two factor authentication including Username password, and digital certificate.
   c. The solution should not store authentication credentials/ data on client computers after a session terminates.
   d. Solution should validate the mobile number and email address provided by each registrant by means of sending confirmation passwords via both SMS and email.
   e. Should support authentication – SMTP AUTH, POP before SMTP, File system, Database, LDAP etc.
   f. The system should have a configured directory of all authorized Users.
g. OAuth functionality should be enabled for Open API to enable access delegation through secure servers.

h. It is critical to identify the consumer of the API to validate the caller by providing a client id & secret, which needs to be embedded in each request. In addition, the end user of the app may also need to be authenticated or authorized to access the resources through a username/password.

### 6.2 User and Identity Management

i. The solution should be capable of uniquely identifying all users of the system and their activities.

ii. The solution should have the capability of providing user access rights to system and data which will be in line with the defined functional requirements.

iii. The identified solution should support both on premise and cloud implementation, or a hybrid of the two.

iv. The solution should be a directory based solution supporting LDAP.

v. The user account management component of the solution should address requesting, establishing, issuing, suspending, modifying, and closing user accounts and related privileges, with a proper approval process.

vi. The system should be able to perform regular audits and management reviews of all accounts and related privileges.

vii. Should provide Single Sign-On facility.

viii. Should support mobility.

ix. Should support privilege user management (Super user, Admin accounts).

### 6.3 Application Security

i. eGCA system must comply with the Application Security Plan and security guidelines of Government of India / DGCA as applicable

ii. OWASP Secure coding guidelines should be followed. Secure coding guidelines should include controls against SQL injection, command injection, input validation, cross site scripting, directory traversal, buffer overflows, resource exhaustion attacks etc. OWASP Top 10 standard should be mapped in the secure coding guidelines to cover all major vulnerabilities.

iii. Validation checks should be incorporated into the application to detect any corruption of information through processing errors or deliberate acts.

iv. Data output from an application should be validated to ensure that the processing of stored information is correct and appropriate to the circumstances.

v. eGCA system should have Role based access, encryption of user credentials.

vi. Application level security should be provided through leading practices and standards including the following:

   a. Prevent SQL Injection Vulnerabilities for attack on database

   b. Prevent XSS Vulnerabilities to extract user name password (Escape All Untrusted Data in HTML Contexts and Use Positive Input Validation)

   c. Secure Authentication and Session Management control functionality should be provided through a Centralize Authentication and Session Management Controls and Protect Session IDs from XSS

   d. Prevent Security Misconfiguration Vulnerabilities (Automated scanners should be used for detecting missing patches, misconfigurations, use of default accounts, unnecessary services, etc. maintain Audits for updates)
e. Prevent Insecure Cryptographic Storage Vulnerabilities (by encrypt off-site backups, ensure proper key storage and management to protect keys and passwords, using a strong algorithm)
f. Prevent Failure to Restrict URL Access Vulnerabilities (By providing authentication and authorization for each sensitive page, use role-based authentication and authorization and make authentication and authorization policies configurable)
g. Prevent Insufficient Transport Layer Protection Vulnerabilities (enable SSL for all sensitive pages, set the secure flag on all sensitive cookies and secure backend connections)
h. Prevent Id Redirects and Forwards Vulnerabilities
i. For effective prevention of SQL injection vulnerabilities, DGCA should have monitoring feature of database activity on the network and should have reporting mechanism to restrict or allow the traffic based on defined policies.

6.4 Data Encryption & Object Signing

i. All the interfaces between various applications and user should be encrypted using appropriate protocols (such as HTTPS, IPsec, and SSL etc.), algorithm and key pairs.
ii. System should support 128/256/512 bit encryption for transmission of the data over the Internet.
iii. Object signing and encryption of attachments (documents) should be compliant to published DeitY standards.
iv. Solution must be secured to both internal and external parties (such as through encryption)
v. The Network / Transport level should include Network Link Encryption (IPSEC) and encrypted HTTP session using TLS/SSL (HTTPS) and other advanced security controls such as HSTS
vi. Business data should be encrypted in the database and DBA should not be able to read or modify it.
vii. Audit controls, electronic signatures, data encryption and other methods should be used to assure the authenticity of transaction and other relevant data
viii. Following events (but not limited to) should be considered as security incidents: unsuccessful log-on, intrusion detection, malfunctioning of encryption facility, unauthorized modification of data etc.
ix. Should develop a procedure for archiving the log files and ensure security of the log files
x. Separate environment should be maintained for production, test and development to reduce the risks of unauthorized access or changes
   a. System should have the functionality to record all the administrator, user level activities including the failed attempts
   b. Should protect logging facilities and log information against tampering and unauthorized access
   c. Information security baseline document should be developed for all the infrastructure components such as database, operating system, router, switch etc. based on CERT-In technical guidelines and best practices.
   d. Provisions should be made for secure content management.
   e. Solution should ensure logs including at least the following:
i. Authentication and Authorization events – logging in, logging out, failed logins. These should include date/time, success/failure, and resources being authorized, the user requesting the authorization and IP address or location of the authentication attempt

ii. Logs for deletion of any data

iii. Logs of all administrator activity

iv. Logs of modification to data characteristics: permissions, location, field type

6.5 Securing Data at Rest

Data at rest in various Data stores should be protected by hashing (salt) confidential data in both transactional and warehouse data stores. The security for the transactional data stores must be ensured.

6.6 Data Integrity

Data in transit (from external systems or between internal systems) or data at rest must be protected from tampering. The risk, both from external and internal users (such as Database Administrators) must be protected against at all times.

To handle the risks of data being tampered by the external users and during transit, API design must ensure checksum features and digital signatures to validate the data is secured. To handle the risks of data being tampered by the internal users such as Database Administrators who have access to data, eGCA application should be designed with the below principles:

1. All the data access must be enabled only through internal API / modules.

2. Each subject area to be packages into a persistence module that exposes domain specific methods to read, insert, update or delete the records

3. Persistence module should abstract the underlying data base technologies, physical data models

4. Based on the current proposals, underlying technology used to store data could be in one of the possible ways:

   i. RDBMS store
   
   ii. File store

   a) RDBMS store:

   i. RDBMS store should be accessed using industry standard Hibernate ORM layer

   ii. Every database (related to a subject area) that is identified to have sensitive information should have one additional data column in each of the tables to track the checksum value

   iii. The columns and the order in which the columns should be sequenced should be tracked as a version number

   iv. The persistence module should compute the checksum and validate the data integrity before returning the data
vi. All the insert, update calls should also do the same logic to compute the checksum and update the new value as part of transactions

b) File store:

i. The eGCA system should store all the xml request files received from external systems as-is. Any re-use of these files by the internal systems must ensure these checks are done before using this data.

6.7 Data Confidentiality

To ensure data is secured to access only by required teams and applications, the following principles are to be adhered:

1. All the databases must be accessed by individual user accounts and user accounts cannot be shared by multiple persons.

2. All the databases/systems must be integrated with the Identity Access Management system for centralized control.

3. For reporting purposes, the data MUST be anonymized (e.g. user level id and details are to be masked - mobile number can be stored as a hash value etc.) before publishing to the BI reporting system.

4. Sensitive data stored in the main RDBMS tables must be encrypted

6.8 Digitally signed requests

eGCA system should send digitally signed JSON/XMLs as response for specific cases.

6.9 Message Protection and Integrity

The Request message XMLs/JSON should be protected using HMAC. The MAC should be used to simultaneously verify both the data integrity and the authentication of a message.
7. Software Development Lifecycle
7.1 Continuous Build

The eGCA system development should be modular and parallel development should be carried out. MSP should follow test driven development approach so that automated testing can be done from the start of development. All application modules within the same technology platform should follow a standardized build and deployment process. There should be instrumental process or tool to capture the end user behavior like the response time, browsing history etc. The application is to be developed, tested and released that adhere to:

- smaller batch sizes
- automated testing
- rollback procedures
- continuous integration
- continuous deployment
- test driven development

Following are some of the requirement to manage the development process along with virtualization of services:

a) A dedicated ‘development / customization’ environment should be setup. The MSP must provision separate development and testing environment for application development and testing. There should be sandbox environment to test the APIs. Any change, modifications in any module must follow industry standard processes.

b) Should be able to create complete, scalable environment for the design and deployment of virtual services for use across co-located and distributed development and testing teams.

c) Application source code to be maintained in source control in order to abstract related set of modules or feature to be independently included in another application.

d) Should be able to simulate delayed asynchronous responses with a Transactions per Second (TPS) metric using Performance Batch Simulation.

e) It is mandatory to create, update and maintain all relevant documentation throughout the contract duration

f) Should enable unit, functional and load testing against virtual services.

g) Should have support for end-to-end testing of virtual services with integration to proposed testing solutions by the MSP

h) Should be able to modify data, network, and performance models easily according to changes in test conditions and performance needs.

i) Should be able to create simulations of real-world application behavior.

j) Should be able to expose virtual services for parallel development and early functional testing.
k) Should be able to define and visualize topology diagrams to understand dependencies and boundaries of underlying systems at the level of remote API calls.

l) Should ensure that a bug tracking tool is maintained for proper tracking of all bugs fixes as per various tests conducted on the application.

m) Should be able to virtualize database access, including Java Database Connectivity (JDBC), and manipulate resultant virtual data services.

MSP should provide a DevOps methodology that would help to perform processes such as:

a) reduce manual hand-offs

b) provides transparency and consistency of code and artefacts from dev to prod

c) have monitoring and alerting up and down the stack, which both DGCA and the MSP can see together

### 7.2 Container Architecture

For development and deployment, the container-based architecture should be undertaken for seamless application development & deployment. Components should be developed as micro services.

![Diagram of Container Architecture](Figure 3: An illustration of Container Architecture)

a) There should be use of containers for application and data packaging and deployment

b) Container Orchestration layer needs to be considered

c) Container based architecture that has a rich API based architecture and support micro services needs to be considered

d) It should support open source and open standards like OCI, CNCF, Java, other frameworks like Node.JS, NoSQL etc.

Container platform should support a secure, enterprise-grade orchestration that provides policy-based control and automation for applications.
Container platform should have following capabilities inbuilt into the system:

1. The platform should have capability to run both stateful and stateless applications
2. The container platform should support deployment and orchestration of multiple containers formats (eg Docker etc.) for preventing any technology lock in
3. The platform should have inbuilt management and monitoring capabilities
4. The platform should have automated application build capability – from source code to a runnable container image
5. The platform should have / support integration with CI / CD tools. Integrated CI / CD tools has to be part of solution.
6. The platform should support polyglot technologies as runtime platforms for applications such as – Java, PHP, Python, Ruby, Perl, Node.js etc.
7. The platform should provide auto scaling capability for automatically running appropriate number of container instances as per load requirements.
8. The platform should provide container instance auto healing capability.
9. The platform should provide application / container version management, auto build of new application container instance in test environment basis on application code new version commit. Roll back to earlier version.
10. The platform should provide deployment strategies support such as – green / blue, canary etc. for ensuring no/minimum downtime for application updates / upgrades.
11. The platform should provide centralized logging capability (including applications logs from container instances) for audit, logs analysis & ease of management purpose.
12. The platform should provide integrated container native persistent storage capabilities.

8. Quality Assurance

Quality Check is required to be undertaken of the eGCA system (and its modules), as per Software Development Life Cycle (SDLC). In this regards, MSP is required to lay down a robust Quality Assurance program for testing of the developed application for its functionality, performance and security. The program must include an overall plan for testing and acceptance of system, in which specific methods and steps should be clearly indicated and approved by DGCA. MSP is required to incorporate all suggestions / feedback provided after the elaborate testing of the DGCA, within a pre-defined, mutually agreed timeline. MSP must undertake the following:

a. Outline the methodology that will be used for testing the system
b. Define various levels or types of testing that will be performed
c. Provide necessary checklist/documentation that will be required for testing
d. Describe technique that will be used for testing
e. Describe how the testing methodology will conform to the requirements of each of the functionalities and expected outcome
f. Indicate / demonstrate to DGCA that all applications installed in the system have been tested
8.1 Automated Testing

MSP is expected to perform automated testing with following features:

i. Should support multi-layer test scenarios with a single solution.
ii. Should support and execute testing on GUI and UI-Less (standard Web Services, non-SOAP Web Services, such as REST, etc.) Components.
iii. Should allow version control of tests and test assets providing ability to compare versions and identify changes.
iv. Should allow centralized storage and management of tests and test assets including external resources used by tests.
v. Should have an IDE environment for QA engineers, which should be configurable.
vi. Should allow system monitoring to test and validate performance issues including memory leakage, CPU overload and network overload to determine, if specific business scenarios exceed desired performance thresholds.
vii. Should provide Auto-documentation while creating automated tests.
viii. Should generate reports that can diagnose defects and can be exported to (PDF, XML, JSON, and Html) (mandatory) and doc (optional) formats.
ix. Should enable thorough validation of applications through a full complement of checkpoints such as GUI object, database, XML, XPath, JSON etc.
x. Should provide facility to parameterize tests to generate/assign test case output values automatically during runtime.

8.2 Performance and Load Testing

MSP is expected to implement performance and load testing with following features:

i. Testing workload profiles and test scenarios based on various functional requirements should be defined. Application as well as system resource utilization parameters that need to be monitored and captured for each run also needs to be defined.
ii. Should support application testing and API testing including HTTP(s), web services, mobile applications and different web 2.0 frameworks such as Ajax/Flex/HTML5.
iii. MSP should perform load testing for multiple workload profiles, multiple scenarios, and user loads to handle the envisaged users of the system.
iv. Different activities before load testing i.e. identification of work load profiles, scenarios, information capturing report formats, creation of testing scripts, infrastructure detailing and workload profile should be prepared before the start of actual load testing exercise.
v. Solution parameters needs to be tuned based on analysis of the load testing reports. The tuning process could be iterative until the issues are closed. Multiple load runs needs to be executed for users to simulate different scenarios while introducing configurable latency/jitter/packet loss etc.
vi. Should eliminate manual data manipulation and enable ease of creating data-driven tests.
vii. Should provide capability to emulate true concurrent transactions.
viii. Should identify root cause of performance issues at application or code level. Include code performance analysis to quickly pinpoint component-level bottlenecks: Slowest classes and methods, most frequently called methods, most costly (aggregate time spent for each method), response time variance etc.
ix. Should allow selection of different network bandwidth such as analog modems, ISDN, DSL, or custom bandwidth.

x. Should be able to monitor various system components e.g. Server (OS, Web, Application & Database) Monitoring, Network (between Client & Server) Delay Monitoring, Network Devices (Firewall, Switch & Router) Monitoring during the load test without having to install any data capturing agents on the monitored servers/components.

xi. Should correlate response times and system performance metrics to provide quick insights into root cause of performance issues.

xii. Reports on following parameters (but not limited to) such as transaction response time, transaction per second (Passed), user interface rendering time, transaction per second (Failed), web transaction breakdown graphs, hits per second, throughput, HTTP responses per Second, pages downloaded per second, system infrastructure performance metrics etc.

xiii. Should provide End-to-End system performance analysis based on defined SLAs. Should monitor resource utilization including memory leakage, CPU overload and network overload. Should have the ability to split end-to-end response time for Network & Server(s) and provide drill-down capability to identify and isolate bottlenecks.

8.3 Mobile Testing

i. Mobile test environment should allow to connect real mobile devices with server and access them remotely.

ii. Should be able to identify functional issues and device vitals as CPU and logs as well as fine-tune mobile applications so it can be used efficiently by remote users even with minimal network conditions as 2G

iii. Should support operating systems - Android, iOS

iv. Should be able to simulate mobile network conditions on devices while doing testing.

v. Should give an option to book and reserve devices for proper reuse.

vi. Support real-world as well as simulated and exploratory testing to report back parameters on usability, design, and defects, all on real devices.

vii. Critical defect information such as (device vital, snapshot, video, etc.) should be accurately captured and shared

viii. Capability to measure response time, understand every interaction, and track availability of key transactions inside a mobile app running on a real device.

ix. Using a combination of real devices and virtual users tool should measure and simulate the impact of load, and apply real-world network conditions for realistic assessment.

x. Get insight into how users are using the app and get actionable data (UI performance, stability, device utilization, and usability) to improve the application development prioritization under live conditions.

9. Illustrative Architecture
The logical layers identified for the proposed architecture are as follows –
9.1 **Client Layer**

This layer represents the users or stakeholders who will be using the online integrated system, either collecting information for decision support or viewing reports/dashboards or monitoring operations or submitting data or applying for specific service requests. Besides the front-end users there will be back-end users who will be acting on the service request raised, verify and approve or reject application accordingly.

The following are the key users and stakeholders as identified –

- **External Users** - These are users external to eGCA, who will use this system for specific service requests or viewing important data and information. Refer Appendix A for comprehensive list of stakeholders.
- **Internal Users** - These are users internal to DGCA/MoCA and related government recognized agencies/individuals. They would also monitor the performance of their respective operations and view interactive reports/dashboards for assessing situations and taking decisions.

All access to the eGCA core system will be either through intranet or over the internet through broadband/3G/4G connectivity or over the mobile/smart phones through cellular network. A specific user can have only one access instance from the mobile channel and one instance on the web channel of the system at a particular time. However, there may be specific cases where multiple login from single account could be allowed.

9.2 **Presentation Layer**

It encapsulates all presentation logics required to service the users that access the system. This layer should intercept all client requests, authenticates users, conducts session management, controls access to business services, constructs response and deliver response back to the client.

9.2.1 **Portal Services**

The eGCA portal is an interface to the end-users for specific services pertaining to their persona. Every user needs to pass through strong authentication before serving them their personal ‘Dashboard’ as home page. In no case, a user should be allowed multiple logins from any device(s) unless pre-defined.

Some of the specifications of the web portal are provided below:

- **a)** It should be compliant to GIGW standards
- **b)** The portal should not allow concurrent sessions for same user (except in certain cases to be specified by DGCA). The system should automatically log out a customer in case of session breakdowns (e.g., communication failure, high inactivity period - these should be parameterized)
c) The portal should support workflows

d) The portal should implement security features, such as password complexity, automatic
blocking (temporary/permanent) of user logins after given number of unsuccessful login
attempts (should be parameterized), controlled access to content stored on the portal and
logging of security incidents. It should by its own or through an integrated Identity
Management solution and should be capable of managing security rights and privileges
by individual, group and role.

e) Portal should support HTTPS protocol on Secure Socket Layer (SSL).

f) The portal should support the leading browsers including Internet Explorer, Firefox, Chrome
etc.

g) The portal should be able to expose / publish functional applications seamlessly

h) The portal should provide search engine with advanced full-text search capabilities. The
search engine should be able to search for requests within the portal.

i) Should provide support for comprehensive audit trail features such as:
   a. Daily activities log should be merged into the history log files
   b. Date, time and user-stamped transaction checklist should be on-line generated for
different transactions

j) All transaction screens should display system information

k) Daily activity reports should be provided to highlight all the transactions being processed
during the day

l) Unsuccessful attempts to log-in to the system should be recorded

m) Portal should be compatible to popular mobile devices Operating systems android, IOS.

n) Portal should be interoperable with industry standard databases

o) In addition the portal should provide the following capabilities
   I. Should be able to integrate with common office application
   II. Should authenticate users from Active Directory/LDAP, claim based
       authentication
   III. Should support virtualization
   IV. Should support customization of look and feel of the portal
   V. Should support a broad range of standards, preferably open standards. Some
      examples are DOM 1.0, HTML 5, HTTP, HTTPS, MathML, ODBC , ODF
      (IS26300), Open XML (IS29500),OpenSearch, OpenType, PDF 1.7, PDF/A, RTF,
      RSS, ATOM, SOAP, SVG, REST, UDDI, Unicode, URI/URN, W3C XML Schema,
      WCAG 2.0, WebDAV, XHTML, XML, XML Web Services, XMLDsig, XPATH, XPS,
      XSLT

p) Should integrate with instant messaging services

q) Should integrate with any other portal products through open standards such as HTML,
   JSON, XML, RSS, web services.

r) Should support encryption and compression features

s) Should support multiple roles with associated access controls.

t) Should support upload, store, organize and share documents

u) Should provide multi-channel output capabilities

v) Users should be able to upload documents in multiple formats

w) Users should be able to upload multiple files at the same time

x) Should support version control, change tracking and comments in these documents

y) Should support document linking capabilities (static, dynamic, and/or other)

z) Should support the import of content into the repository

aa) Should support document and text indexing capabilities

bb) Should support image indexing capabilities

cc) Should be able to support to store and manage documents

dd) Should support content archiving capabilities

ee) Should provide offline support for forms

ff) Upload document will be in JSON, XML, CSV and excel. Scanned document will be in
    PDF and images in JPEG, PNG.

gg) While uploading bulk data it should also tell for errors if it encounters any.
9.2.2 Mobile Services

The mobility layer encapsulates the mobile enablement framework, which deals with both rendering the web pages in mobile devices through necessary UI components as well as making available native mobile apps of individual services, developed using native mobile components and data security considerations in Android/ iOS platforms. The mobile apps will be capable of offline data capture.

9.2.2.1 Mobility features

Some of the key requirements related to Mobile application, but not limited to, are mentioned below:

i. While building the mobility solutions services should be available on low bandwidth scenario.

ii. The Mobile Application should provide an intuitive and user friendly GUI that enables users to navigate and apply actions with ease. The GUI should be responsive with very little or no delays or time lag at launch or whilst navigating through screens.

iii. It should enable ease of configuration and changes to existing GUIs, and support the introduction of new screens.

iv. It should provide on screen tips and online help to aid users while interacting with it.

v. Should make use of data available in the existing database and reduce duplicate data entry

vi. Incorporate analytics into mobile app, to track and identify users experience and actions.

vii. Apps should be easily customizable and easy to Administer data in the eGCA database

viii. Should structure overall content with proper tagging to make them screen reader friendly.

ix. Application should ensure compatibility with Android & iOS platforms.

x. Solution should develop resolution independent design structure i.e. Mobile Application should adjust itself automatically as per the screen resolution of the Mobile

xi. Mobile Apps should work flawlessly across different platforms

xii. There should be minimum use flash contents so that home page should be loaded quickly

xiii. Apps should not occupy excess client’s Mobile RAM

xiv. Should provide Role Based Access control

xv. Should come with mobile threat prevention and recovery system

xvi. Should have facility to download and upload files
9.2.3 Offline forms

A critical component with respect to data upload by various stakeholders are the Offline Forms. Some of the key requirements with respect to offline forms are mentioned below:

a. The offline Forms should be available where the user can download the form for data input. It is therefore required that the MSP should engage an efficient tool to design the UI, to achieve enhanced User experience and usability of the solution.

b. The Offline Forms should support recording information offline through Internet and future uploading. Information uploading should also be possible in bulk for multiple instances of the Offline Forms.

c. Should be able to integrate to the proposed portal, and the underlying database.

d. Should support information exchange as XML/JSON.

e. Should support offline validation of information.

f. Should be compatible with latest version of all leading browsers

9.2.4 Security Services

The security service will provide the following security control features

User Registration & User Profile Management

1. System Administrators should manage to administer the application security through a web interface to allow them to register and create user profiles for users who will access the eGCA applications.

2. User Registration service should allow administrators to perform activities like registration and login features, creation/editing/deletion of user profiles and groups, assigning roles to users, assigning users to groups etc.

3. A facility should be provided to create users under an organization and it should be in self-service mode.

9.2.5 Business Presentation (UI) Services

These are the presentation services that present the web user interface to the end users to avail various services offering of the eGCA applications. These services should make use of forms, style sheets, tiles and tag libraries and the view component of the MVC framework to render the UI page.

a) Caching Services

The Caching component should make use of state-of-the-art in memory caching to optimize portal performance. The caching services should be static, dynamic as well as distributed.

b) User Personalization & Dashboard Services

Personalization services should allow users to maintain their own specific profiles. Each user should be presented with a centralized dashboard view specific to each user profile that will display key information in a summarized manner and should allow top officials to monitor the status of specific service requests and performance as well as the KPIs for the Directorates.
c) Report UI Services

Should present user interface to the end users to provide reporting criteria, complex querying and view MIS/Custom/Analytics reports generated by the business layer MIS and Analytics reporting components.

d) Calendar Services

This custom service will provide the Calendar functionality in the eGCA application. This component will be custom built leveraging the MVC framework components. The service will display the date and time of all events scheduled in a standard monthly calendar format.

9.3 Business Layer

The business layer should manage eGCA business services wherein, all business processing for the application should be centralized in this layer. The business layer application should receive requests from the presentation tier, process the business logic based on the requests, and mediate access to the other underlying layer (data access/storage layer or integration layer) resources. Additionally this layer should handle content management services, database connection management, session management, email and SMS notifications and SMS gateway, integration with other Govt. applications and third party solutions.

The following services (but not limited to) are to be provided by the business layer:

I. Business Services & Service Request Common Components

II. Business Rule Services

The system will be rule driven and should be configurable. There will be high dependence of rule engine as there are plan to integrate technologies like Block chain etc. The system will be rule driven and should be configurable.

a) The solution should support decision model and notation (DMN) specification version 1.1 or above.
b) The solution should support data verification and consistency checks.
c) The Rules Engine must be able to submit business rules to an external repository at such time as the repository exists.
d) Support both embedded and stand-alone services.
e) The Solution should support Compute values based on input data
f) The Solution should support mechanisms and ease of use for users to edit rules.
g) The Solution should have the ability to tune individual steps in the overall decision process for maximum performance by the execution engine
h) The system should have the feature like
   I. Complex event Processing (CEP)
   II. Stateless & Stateful rules
   III. Backward & Forward chaining rules
   IV. Business Resource Planner
i) The Solution should support repository infrastructure for rule storage and versioning
j) The Solution should easily integrate with the rest of proposed Solution
k) The Solution should support seamless and easy user interaction
l) Ability to capture and maintain rules centrally.
m) Support for user roles and configurable permissions by role. The rule repository must support fine grained user/group access controls. System must provide a robust governance framework to manage business rule change.
n) Ability to test and verify combinations of rules produce the desired outcome. The system must allow the user to define key measurements against which the business logic can be verified.

III. Workflow and BPM Services

Few features (minimum requirements) that should be kept in mind while implementing Workflow Management Framework are:

a) Support process modelling based on BPMN2 notation standard

b) Facility to simulate a process before launching

c) Provide business rule engine and a management platform.

d) Users with authority should be able to modify business rules online without any need of deployment. Workflow system should also have business rule connector to talk to any 3rd party business rule engine.

e) Offer performance monitoring features and have capability to identify inefficient processes and operations and/or those with high level of error and omission.

f) Expose W3C standard web services and REST based web services so that it can communicate to any other technology layer seamlessly.

g) Have capabilities which will enable business activity monitoring and capture audit trail of all transactions as well. Web based dashboard should be made available for accessing all reports.

h) Provide dashboard view for showing multiple reports. Dashboard view and content can be customized for individuals.

i) The BPM solutions should support container-based deployment along with other components.

IV. Notification & Messaging Services

a) Email notifications to be given automatically

b) The SMS gateway must be as per prevailing TRAI/DoT norms

c) Support automated alerts that allows to set up triggers that will automatically send out reminders

d) Provide provision for International SMS

e) Resend the SMS in case of failure of the message

f) Should be instantaneous with almost no waiting time.

g) Must have common features like non-acceptance of landline nos., unacceptable mobile nos. etc.

h) Should Support for long text messages

i) The message should be sent though command line interface/API, Web Interface.
V. MIS/Custom Reporting Services

Generation of MIS/Custom Reports at various levels is one of the important service requirements of the eGCA application. Considering the various levels and types of reporting services that the monitoring component of the application will have to provide, the proposed solution should provide Visualization tool and analytical tool for decision making and data analytics.

Some of the key features of the tool(s) are as follows:

a) Report output in multiple formats such as PDF, XML, HTML, CSV, XLS, RTF, TXT, JSON
b) Print ready output
c) Multiple data sources of multiple kinds in one report
d) Mobile App interface for publishing dashboards/ reports
e) The users should be able to generate the Analytics Reports at various levels to perform predictive analysis, operational analysis, risk modelling, statistical modelling etc. for taking informed decisions
f) The reporting tool should have slicing and dicing features facilitating ad-hoc management reporting on the fly.
g) Data Visualization tool capable of interactive visualizations
h) The analytics solution should integrate with the Data Visualization tool capable of interactive visualizations.
i) The solution should have drill-down capabilities (ability to drill down to various levels of a hierarchy).
j) The solution should be able to publish the reports on the portal/mobile application and have the ability to archive reports.
k) The solution should be able to distribute reports and also have the ability to save data for later use or to a local PC/laptop or for other users to view. It should support offline viewing. It should be able to send reports electronically to other users.
l) The solution should have the ability to schedule reports.
m) The solution should provide for a browser based interface to view reports.

VI. Content Management Services

eGCA solution would also require implementation of a content and document management system.
a) Easier and quicker mechanism of creating, deploying, managing, editing, manipulating and storing content on Web pages, including but not limited to text and embedded graphics, photos, video, audio, and link as well as Automated Templates (that can be automatically applied to new and existing content), through editing tools / framework.

b) Catalogue and index content, select or assemble content at runtime, or deliver content to specific visitors in a requested way.

c) Workflow based lifecycle from content creation to review and approval, and finally publishing of the content, as per Access Control to Users and Groups

d) Comprehensive Audit Trail to capture the entire Content Life Cycle

e) Search, Archiving and versioning of the Content, for ready reference and management.

f) Ability to display content in multiple languages.

g) Act as a collaboration platform allowing content to be retrieved and worked on by one or many authorized users

h) Address both structured and semi-structured sources

i) Support Sorting and filtering of the search results and saved searches

j) Search solution which is able to address multiple sources crawling and searching like Databases, File Systems, Content/Document Repositories, Web and Web portals, Email systems. Should be able to Search highlight and collapsible summary of the search items

k) Provide rich text editor for content editing and multiple file upload functionality

l) Ability to create and administer documents, records and web content via a single user interface that simplifies the administration experience

m) Provide Administration services such as archiving and removal, configuration migration, audit trails and system reports, and backup and recovery using a web-based user interface

n) All the searches would be automatically filtered by user’s security privileges

o) The versioning feature should allow to easily track new revisions and roll back to previous versions

p) Support single-sign on

q) Provide support for web content and document management workflow

r) Ability to compress scanned images. Should provide support for annotations

s) Provide consistent look and feel to the users and standards defined for content, structure and presentation of the Web portal.

Refer Appendix D1 for additional details

VII. Analytics Reporting Services
VIII. **SLA Monitoring, Server Administration & Management Services**

   a) SLA monitoring services as a part of their solution
   
b) Provide support functionalities like audit trail, error & exception handling, logging, export & printing services to the business components of the eGCA application.
   
c) Exception, Error Handling & Logging Services

IX. **Support Services**

9.4 **Data Storage layer**

9.4.1 **Transactional Database**

This would contain the operational data. The data should be horizontally scalable. All operational database reads must be centralized with access and audit.

   i. All the applications implemented should have provision for optimizing the number of static connections to the database using connection pooling. All the applications implemented should also optimize the duration of connection to the database by using techniques like session time out.
   
i. Database should have proven scalability credentials to cater to any system load.
   
i. It should provide Unicode support.
   
i. It should support User-defined Data Types & User-defined Functions.
   
i. Database should support advanced data compression, self-healing and deployment in various cluster topology.
   
i. The database platform should support enhanced configuration and management of audits.
   
i. The database platform should support Failover Clustering and disaster recovery solutions.
   
i. It should support online indexing operations and parallel indexing operations
   
i. Database should support Schemas, Roles Based Privileges & Authentication.
   
i. The data platform should support policy-based system for managing one or more instances across enterprise
   
i. The database platform should support defining resource limits and priorities for different workloads, which enables concurrent workloads to provide consistent performance
   
i. Database security should provide different layers of database users with overall control of database security administrator, only authorized database administration users with assigned privilege should be allowed to access database.
   
i. A separate audit trail should be maintained for any direct modification, deletion and addition in database in database structure or records. User, even the database administrator should not be allowed to tamper with audit log. Database server should
support most granular column encryption to encrypt sensitive data.

xiv. The selected solution should have abilities for fault tolerance, linear scalability, mixed workload capability

xv. Database should support option of different partitioning schemes within the database to split large volumes of data into separate pieces or partitions, which can be managed independently. It should support physical columns. The partitioning should enhance the performance, manage huge volumes of data and should provide foundation for Information Life Cycle Management.

xvi. It should preferably provide options for Automated/manual performance analysis with diagnosis of the cause of performance related issues with possible resolutions.

xvii. It licenses should be unrestricted and full use licenses (read, write and modify). It should allow storing scanned images, text documents, XML, multimedia inside the tables. It should be part of the basic database distribution without any additional cost to the organization

xviii. It should support the separation of security functionality from application functionality and database administration functionality.

xix. Any proprietary OEM specific functionality of database should not be used.

xx. Should be having provision to Administer database / database clusters, Monitor performance, Maintain database, Backup and recovery, Disaster recovery management, diagnosis, performance tuning with the SQL analysis, finding the events, advisory based tuning mechanisms with the history.

xxi. Database should have option for identification and tuning of high load SQL Statements. Provide details about dynamic tuning capability of the database depending on workload requirement, system resources etc.

9.4.2 Data Analytics Store

This store would be used for receiving the structured, semi-structured and unstructured data from the different data stores and performing the data mining and analytics of the same for generating useful business insights that can be used by various stakeholders to take informed decisions.

There can be different types of Analysis. Following are some of the details

a. Statistical Analysis: The system should generate reports comprising of complex statistical dynamics and multiple parameters from historic data. These reports may be generated through the UI provided within the solution to authorized Users. It should also present patterns found in historical and transactional data to identify risks and opportunities. The proposed tool should be capable of providing viewable descriptive statistics such as mean, median, max, min etc. and also should be capable of performing/aiding advanced statistical modelling and analysis including but not limited to correlation, regression, scoring, ranking, clustering, network plot, decision trees, scenario analysis, ANOVA etc.

b. Persona Based Analysis: The proposed system should have advanced clustering and segmentation capability and should provide persona based segments for analysis based on user profile and behavior. It would provide a window to understand what the Users are doing on the system and what they are not doing, specifically analyzing each Users focus / usage area, and hence improve the services quality and the content as well as features / functionalities related to it.
c. **Predictive Analysis and Forecasting:** The system should have the capability of generating predictive statistical models. It should be capable of capturing patterns and signals in data, analyze them and provide insights into future trends in the form of graphs and charts, based on certain parameters, statistical modelling and historical data. These may be used by the Authorized Users to identify trends at certain location / period and take appropriate decisions. The tool should provide enhanced forecasting capabilities with Scenario Analysis allowing users to see impact of variable values on the forecasted trend.

d. **Fraud / Anomaly Detection:** The system should display alerts / information, based on certain pre-defined criteria, if there is any deviation from the standard trend / output. Access to such information / reports should be restricted to certain authorized Users / decision makers only. These may be used by users for decision making purposes or further investigations as required.

e. **Statistical Scoring:** The solution needs to have capability to score the partners basing on social and financial performance on a quarterly basis that combines business rules, anomaly detection and advanced analytic techniques. The solution needs to have capability to calculate the deviation for each agency and Skill counsel. The solution should have capability to calculate risk scores based on specific characteristics of the activity including geographic zones etc.

**Technical capabilities of eGCA Analytical Platform**

I. Platform standardizes on analytics and reporting technology should integrate well with the rest of the stack and at the same time meets most of the feature needs of applications built on the Platform.

II. Data storage for reporting and analytics may be separated from the core transaction data (data that is part of the live production systems). The advantage of the highly deformalized analytics data being completely separated is to ensure scalability and make least impact to production systems.

**9.5 Interface Layer**

All interfacing with external systems like payment and SMS gateway, IVR solution and internal application or department specific application will take place through this layer. External systems will not have direct access to the eGCA database, but instead will be calling the relevant business services in the business layer for availing the information. Details provided in the integration architecture section.

**9.6 Integration Architecture**

The various technologies that would be required to achieve integration between the different applications and external systems are provided below.

**9.6.1 API Management**

All services of eGCA system will be available through an API layer, so an API gateway solution is required for it.

a. API layer would not be exposed to untrusted connection

b. All data transfer to happen through APIs, File transfer mechanism is not encouraged
c. API signature authentication will be possible through the API key, application id + key and sso token mechanisms.

d. All the APIs would be stateless in nature, thus easy to load balance, even if hit through portal is very high and this requires high end processing.

e. The API Platform should be allowed to manage all your enterprise initiatives from a single solution:

f. The API platform should support existing APIs and developer preferences

g. The API Platform should provide clustering and ensure reliability, scalability and single point of administration:

h. The API Platform should provide for enterprise grade encryption

i. The API platform should provide secure access to all APIs and provide all the forms of authentication, access control and certificate/credential support:

j. Wherever required data send through the API framework should be digitally signed.

k. Data should be hazed before transmitted through APIs

l. An API design document with the specification would be shared with the stakeholders for them to start developing the interfaces. The APIs would be RESTful services with XML/JSON payload and would have the minimum information in the design document

   ▪ Purpose of the API, input and output parameter, error code, owners etc.

m. Service Provider would deploy a developer sandbox to be hosted at Data Center within 1 months of agreement signing for the stakeholders to test the APIs with dummy data.

n. There should be authorization and license key management

o. Feature to publish APIs

p. Version control and API retirement should be taken into consideration

q. The framework should have the capability for API Governance and SLA enforcement

r. Features like Audit trail, API usage, API metering should be provided.

s. API Manager and gateway should support deployment in container environment for Microservices architecture and auto scale capability.

9.6.2 SMS Integration

SMS services are envisaged to be made available as part of the solution design. SMS services can be used for OTP authentication and as well as informing the users about their registration status, payment alert, etc. Apart from outbound SMS there will be inbound SMS coming especially for remote users or users who are creating featured phone (non-smart phone). The SMS service cost will be the responsibility of MSP. If required, the service provider may integrate the solution with MSDG, and use the services available through it, or deploy SMS Gateway services developed by it for the proposed system or integrate with other SMS gateways provided by service providers, but it is a mandatory requirement that all the SMS based services (alerts and notifications) should be...
available as part of the solution. For outbound message SMS gateway can be developed. Specification related to SMS has been provided in the Notification & messaging section above.

9.6.3 Email Services
Email services are envisaged to be made available as part of the solution design to send alerts / intimations / automated messages to registered email ids, based on preferences set up / opted by individual users. An authenticated SMTP mail service (also known as a SMTP relay or smart host) is envisaged to be integrated with the solution for sending mail from the solution, and delivered to intended inbox. This should support antispam features.