Solution Architecture Document
For CAE Solution

Client Name: CAE

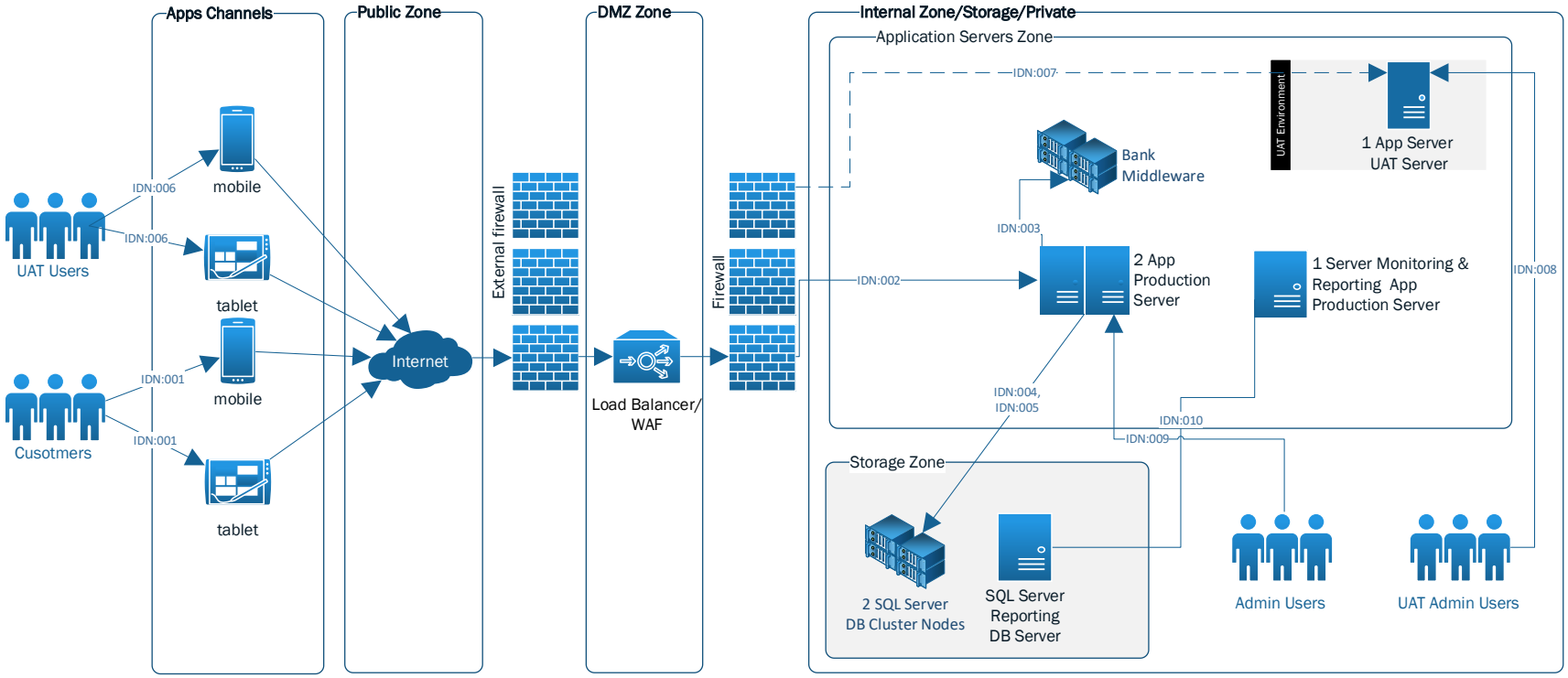
Project Name: CAE Omnichannel Solution

1-Sep-2021

TABLE OF CONTENTS

1. Network architecture

1.1 Network design diagram



1.2 Network - Connectivity details

1.2.1 Server machines

Server Name	Server Zone	Server type	Server count
Application server	Application server zone	Production server	2
Reporting and monitoring app server	Application server zone	Production server	1
SQL Server database cluster node	Storage zone	Production cluster server	2
Reporting SQL Server database	Storage zone	Production SQL server	1
Application server & SQL Server - UAT	Application server zone	UAT Application Server	1

1.2.2 Components that will be deployed on each server.

N	Server Name/Role	Components
1	2 App Servers	1. CEEP eChannel Manager 2. CEEP Touch Top Portal (ePortal5) 3. Omnichannel Business Implementation 4. CEEP Admin Portal (including business admin & CEEP Reporting)
2	2 SQL Server DB Cluster:	These Servers has two separates SQL Server instances with two separate hardware resources / disk foreach instance for performance recommendations. Common Installation servers: 1. Portal 5 DB 2. eChannel Manager. Frist SQL Server Instance: 1. Business DBs (Config and Business) Second SQL Server Instance: 2. eLogging DB. (Audit DB). 3. MS SQL Reporting Service.
3	Reporting SQL Server:	SQL Server Instance:

		1. eLogging DB. (Audit DB). 2. MS SQL Reporting Service.
4	Monitoring and Reporting App Server	1. Monitoring Windows Service (must be able to open URL's on App1 & 2) 2. Monitoring Web Portal 3. CEEP Standard Reports.

1.2.3 Connections Details

Note: All source ports are dynamic

Connection ID	Connection purpose	From system	To system	Network protocol	To port
IDN:001	Connections coming from Customer Devices through internet to the Bank Firewall HTTPs certificate is installed on Firewall	End users' devices	Firewall/WAF/Load Balancer	HTTPs	443
IDN:002	Connection from Load Balancer/ WAF	Load Balancer/ WAF	Application server - Production	HTTPs A Local HTTPs certificate is needed	443
IDN:003	Requesting business modules in core banking through application production server.	Application Production Server	Core Banking	HTTPs / Web services	Bank to advice
IDN:004	Application server will access SQL DB cluster to getting data related to application transactions.	Application Production Server	SQL Server DB. Cluster node	TCP/IP	
IDN:005	Reporting Server will read Audit Log messages from MSMQ on App Servers	Application servers – Audit Log Queues	Reporting Server - DB. Cluster node	MSMQ	MSMQ Ports

Connection ID	Connection purpose	From system	To system	Network protocol	To port
IDN:006	UAT End Users /Admins Requesting App server (UAT) HTTPs certificate is installed on Firewall	UAT End users’ devices / Admin Portal	Firewall/WAF/Load Balancer	HTTPs	443
IDN:007	Connection from Load Balancer/ WAF	Load Balancer/ WAF	UAT Application server	HTTPs A Local HTTPs certificate is needed	443
IDN:008	Connection from UAT Admin Users	UAT Admin users PC	UAT Application server	HTTPs	443
IDN:009	Connection from Prod Admin Users	Prod Admin users PC	Reporting Server	HTTPs A Local HTTPs certificate is needed	443
IDN:010	Reporting and monitoring application server will access SQL DB cluster to getting data related to reports / monitoring	Reporting and monitoring app server	SQL Reporting Server DB	TCP/IP	

2. CEEP Application Requests Flow Arch.

The System operates in Client/Server model (and not the traditional Web Server model of Thin Client/Server model).

Application running on Client devices are intelligent applications with enough logic running on the client side to handle all UI process and handling also application will download additional app HTML and JS files when needed from application server to cache.

Applications will only send requests to the servers when data is needed from the server side in the form of JSON/XML Web service requests.

- The users use the applications on their different channels (Mobile, Tablet, Desktop or others)
- Requests from the apps are sent in the form of JSON/XML messages through the load balancer to the Omni Channel Application servers where users are distributed between the two servers
- Request are processed by the eBSEG eChannel Manager which is typically running on two App Servers (App Server 1 and 2).
- eChannel Manager makes required one or more Calls to Core systems or Middleware for Business Data
- Using Data returned from Databases or core systems a JSON/XML formulated message is returned back from eChannel Manager to the Applications on the Devices

3. Physical Architecture

3.1 Production Environment

Item	Hardware Configuration (Est)	Basic software (Est)
2* Application servers	16 logical CPU 16 GB. Ram 160 GB Hard Disk	2* Windows Server 2016 – 2019 with IIS
2* SQL Database Server Cluster Nodes	16 logical CPU 16 GB. Ram 160 GB Hard Disk	2* Windows Server 2016 – 2019 2* SQL Server 2017-2019
1* SQL Reporting Database Server	16 logical CPU 16 GB. Ram	1* Windows Server 2016 – 2019 1* SQL Server 2017 -2019

	160 GB Hard Disk	
1* Reporting and Monitoring Server	8 logical CPU 8 GB. Ram 160 GB Hard Disk	Windows Server 2016 – 2019 with IIS SQL Server 2017-2019

3.2 Testing Environment (UAT)

Item	Recommended Configuration	Basic software
1 Application servers (in internal Lan)	8 logical CPU 8 GB. Ram 160 GB Hard Disk	1 Windows Server 2016 – 2019 with IIS 1 SQL Server 2017-2019

4. Nonfunctional requirement

4.1 High Availability & Load Distribution

4.1.1 Database

- eBSEG solution recommend use two Node SQL cluster System to make the Main System databases always available for the online portal access. It's recommended to configure this Cluster in (Active-Passive technique or Active – Active by distributing DBs between servers).

4.1.2 Application Servers

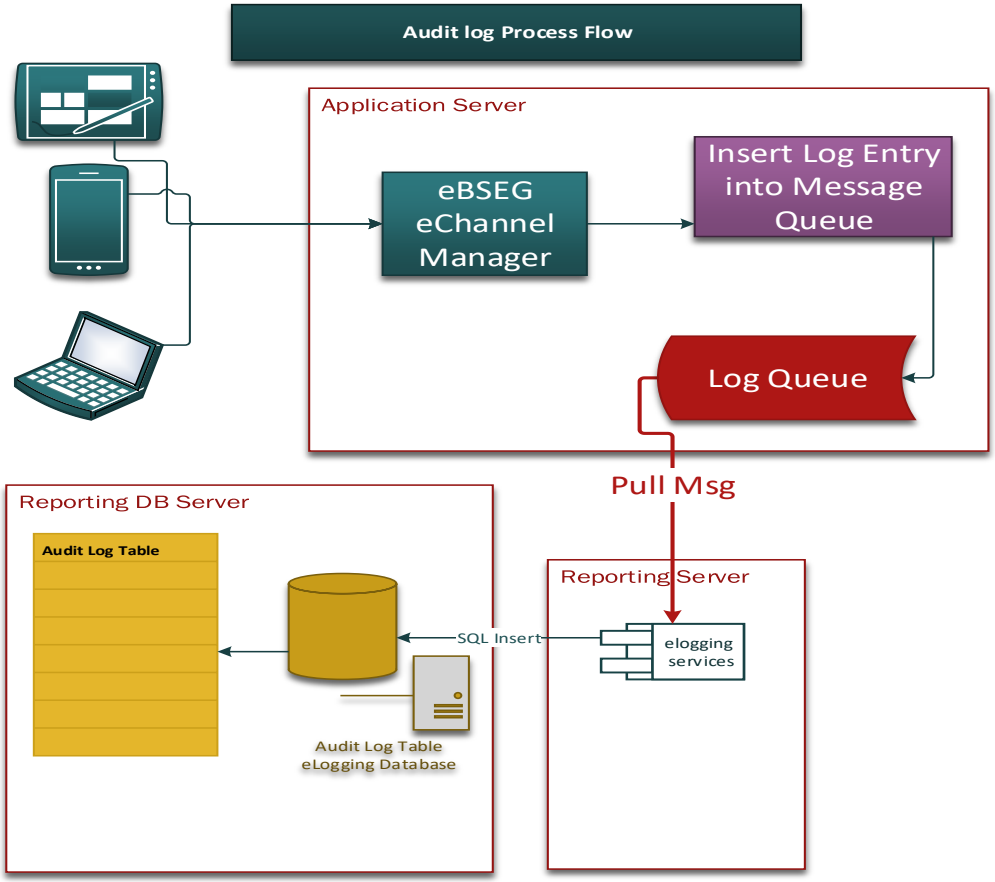
- The system is composed on a min of two application servers or more based on customer load.
- To distribute load between application servers CAE is using a load balancer component that will receive requests from the internet and distribute these requests to the available Application Server IP/Ports.
- The load balancer should be configured to use sticky session's concept to insure any user session keeps going to the same server sol that the user session is correctly maintained. This stick session should be configured based on the ASP.NET session id cookie.
- Furthermore for auditing purpose the Load Balancer should be configured to pass the real internet IP address of the requester device in X-Forwarded-For (XFF) HTTP header field (https://en.m.wikipedia.org/wiki/X-Forwarded-For) which is a common method for identifying the originating IP address of a client connecting to a web server through

an HTTP proxy or load balance. This info is later used by the App server to log that IP in the Audit log system for security reference.

4.2 DB Audit Log Architecture

4.2.1 Process Flow

1. A request is received from channels to channel Manager
2. Request is process and just before reply to channels the channel manager inserts log entry XML data into the Log Queue Located on the application server
3. An Audit Logging Windows Service running on Reporting Server is constantly checking for new messages in the Log Queue and whenever a message is found it is peaked and it info is logged into the Audit Log Table as a new row in the table
4. Then the message is deleted from the Queue (After insert into table succeeds)

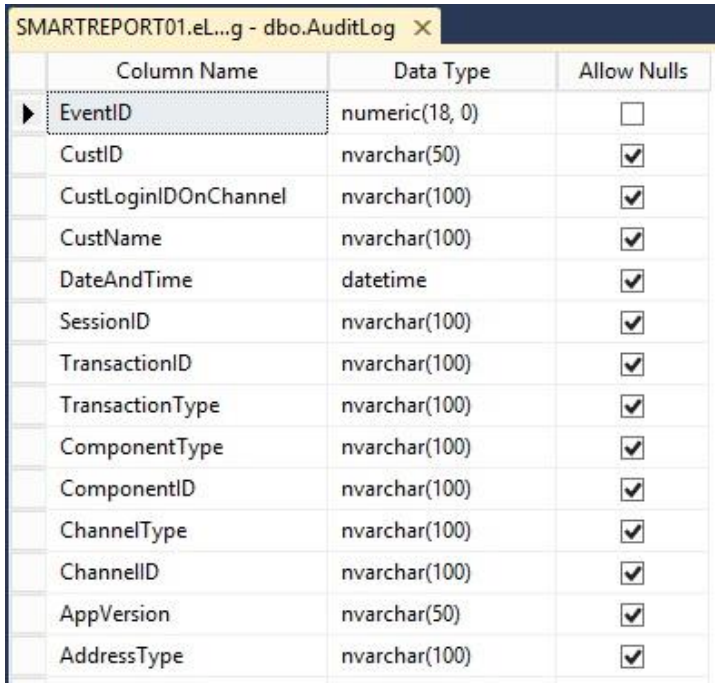


4.2.2 Logging Configuration

- Each Application Server has one instance of channel manager running
- Each Application Server has one Logging Queue
- Each Queue has one windows service instance of eLogging Service running on Reporting Server

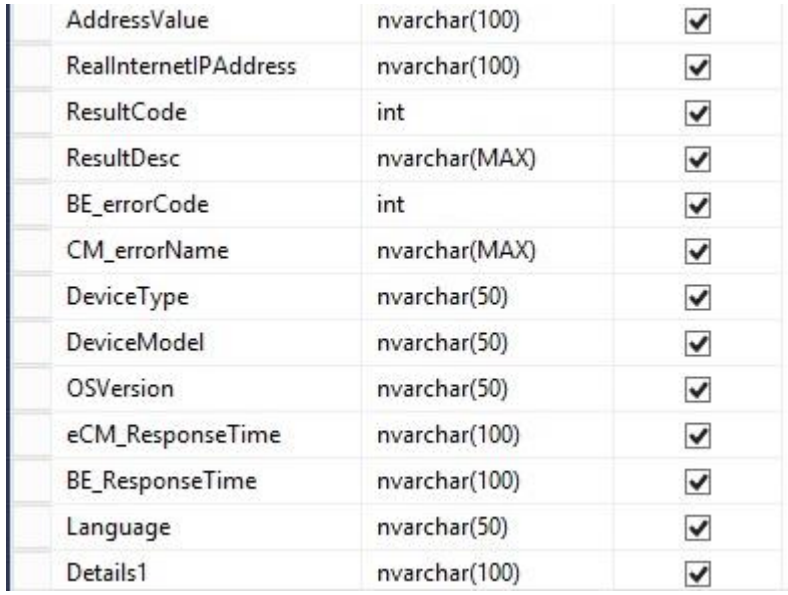
Item/Env	Prod
eLogger Service Name	eBSEG eLogger
Audit Log Table Name	AuditLog
Queue Name	eCM_Log_Queue

4.2.3 Audit Log Table Design



SMARTREPORT01.eL...g - dbo.AuditLog

Column Name	Data Type	Allow Nulls
EventID	numeric(18, 0)	☐
CustID	nvarchar(50)	☑
CustLoginIDOnChannel	nvarchar(100)	☑
CustName	nvarchar(100)	☑
DateAndTime	datetime	☑
SessionID	nvarchar(100)	☑
TransactionID	nvarchar(100)	☑
TransactionType	nvarchar(100)	☑
ComponentType	nvarchar(100)	☑
ComponentID	nvarchar(100)	☑
ChannelType	nvarchar(100)	☑
ChannelID	nvarchar(100)	☑
AppVersion	nvarchar(50)	☑
AddressType	nvarchar(100)	☑



AddressValue	nvarchar(100)	☑
RealInternetIPAddress	nvarchar(100)	☑
ResultCode	int	☑
ResultDesc	nvarchar(MAX)	☑
BE_errorCode	int	☑
CM_errorName	nvarchar(MAX)	☑
DeviceType	nvarchar(50)	☑
DeviceModel	nvarchar(50)	☑
OSVersion	nvarchar(50)	☑
eCM_ResponseTime	nvarchar(100)	☑
BE_ResponseTime	nvarchar(100)	☑
Language	nvarchar(50)	☑
Details1	nvarchar(100)	☑

Column Name	Description
CustID	Customer ID (only filled after Customer logins to the channel otherwise empty)
CustName	Customer Full name (only filled after Customer logins to the channel otherwise empty)
CustLoginIDOnChannel	Customer Login ID (username used in Login)
DateAndTime	Transaction end time
SessionID	Session ID represents a unique ID for the App on the Client Device. This is NOT related to Login Session. It represents a run of the App and is only changed when the App if fully closed/kill/or force killed. Thus Session ID will be the same before and after login.
DeviceType	Value that represents the device type running the client App; valid values are: 1. AndroidPhone 2. AndroidTablet 3. iPhone 4. iPad 5. Desktop
DeviceModel	Represent brand model of the device did the transaction.
OSVersion	Represent OS version of the device did the transaction.
TransactionType	This column is used to display the transaction type like account, transfer or credit card
Componenttype	Component used to log channel manager id as eCM
AddressType	This column is used to display device address as IP address used for call
RealInternetIPAddress	Define IP address read by load balancer and proxy server
ResultCode	Represent channel manager result code for done transaction: - If its value is zero, transaction service call to the back end done successfully. If its value is except zero, mean the transaction, call or service was failed.
BE_errorCode	Represent backend result code for done transaction: - Zero means transaction done successfully.
ResultDesc	Description mapped with backend result code in the registry. The description (application defined or object defined error) is used for cases like connection error, server recycle or restart or undefined error description in the registry.

eCM_ResponseTime	**Represent time that transaction took in channel manager level including: -** - Receive front end request - Prepare back end request - Call back end service - Receive back end response - Prepare front end response
CM_errorName	Represent error name defined in the registry.
BE_ResponseTime	**Represent time transaction took in back end level including: -** - Receiving channel manager request - Doing back end functions for this service - Sending back end response to channel manager.
Details columns	This column is used to make a configuration for any business parameters like payment amount and transactions

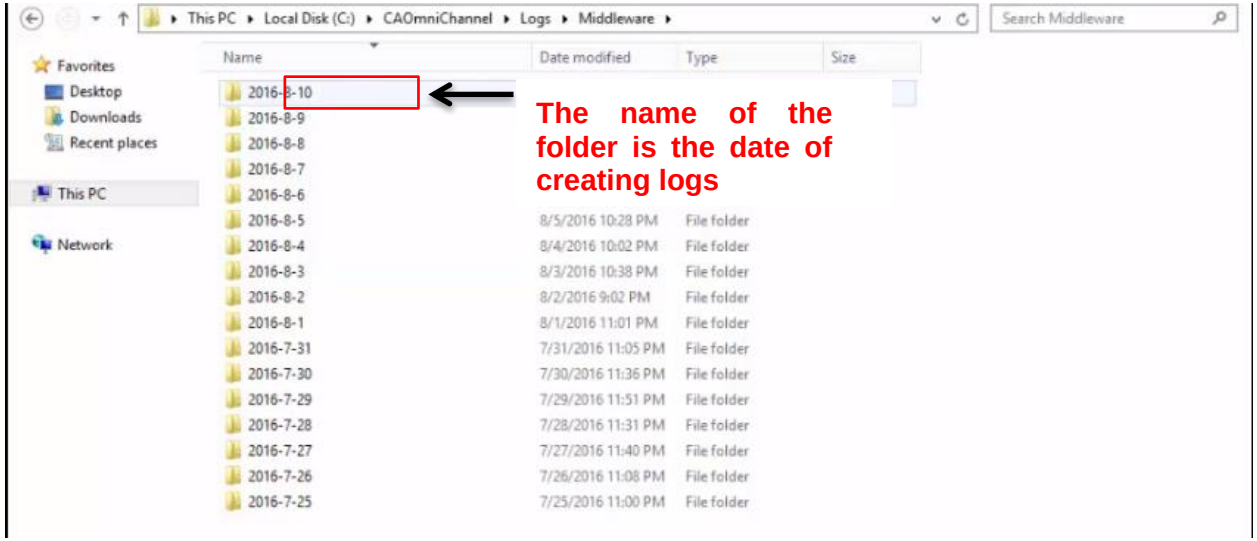
4.3 Text Trace Logs Arch

4.3.1 Log files path:

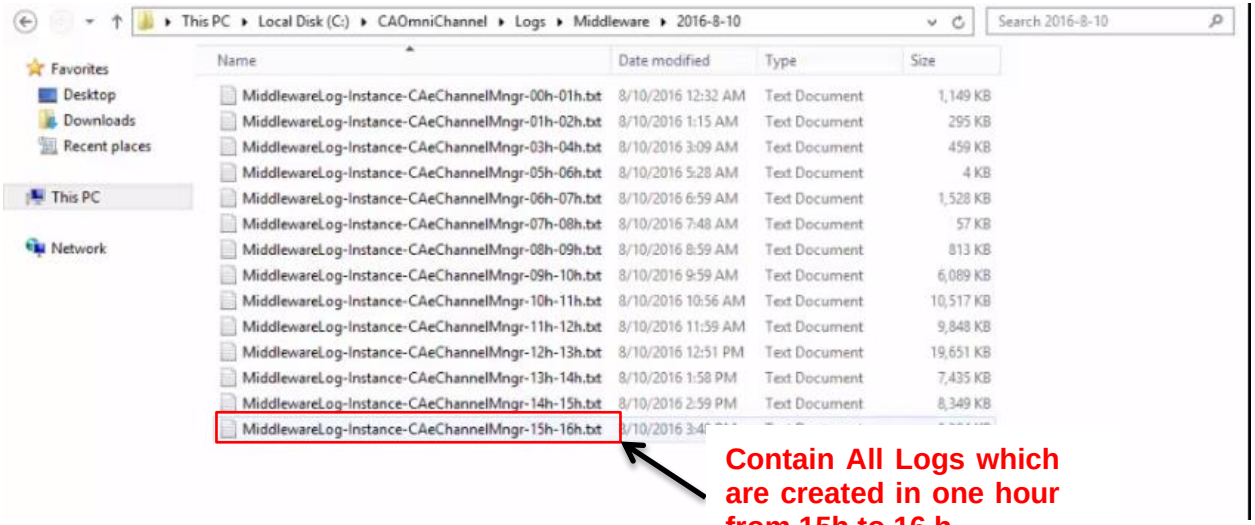
Channel Manager Text Logs are created in a configured Path on each App Server.

4.3.2 How log files are arranged in Channel manager folder?

Channel Manager logs are created by their date in new folder by this date in middleware folder, so all logs of same date will be created in the same folder to be easier while searching about an exception happened at specific date



Every folder created by specific date contain log files created in every hour to be easier while searching about an exception happened at specific hour at specific date



Every application server has its own log folder, and the log of user request is created on one of the two servers and the load balancer is responsible of distributing the user request on application server1 or on the application server2

4.3.3 Log File Structure

A log file is basically a text files composed of Lines where each line represents a log entry from the eChannel Manager
Each Line starts with a prefix part as follows for example:
-*->04/Aug/2016 12:02:40.001 (521008):

The prefix syntax is as follows:

-*->DD/MMM/YYYY HH:MM:SS.Millisecond (RequestTracerID)

The request tracer ID is a random number created for each request in the eChannel Manager only for purpose of linking the log entries together in the log file so to trace all the logs of a certain request the user should search for the tracer ID for example search for “(521008)” to get all the log entries related this request

4.3.4 Log Data Masking

To protect sensitive information in the logs a masking configuration is used where by a regular expression rules are configured on each Application server to replace sensitive tags with “*”

For example:

<FrontEndPassword>*</FrontEndPassword>

<challengeQuestion>*</challengeQuestion>

4.4 Application Version Management

The solution supports the concept of updating the mobile client app pages and scripts without the need to go through the mobile stores every time. In case of a new client app release (Mobile + Tablet) of the application the bank’ Team will want to make sure that their customers are using the latest version of the application. The system allows the admin to:

- System will notify the customer for the new version
- System can be configured to disallow app to work for a certain old version in order to force the customer to upgrade to the latest published version. This can be immediate or with a set date that is alerted to the customer beforehand.

4.5 Major Upgrade Deployment Approach

Given that the provide solution arch is based on Client/Server Technology and thus when a new Client version is created and published to iOS and Android Stores for Mobile and Tablets this creates the situation where bank customer will have an older version in the same time some customer will have a new version

To support such usage pattern and multiple version with client the recommended approach for deployment of sever side parts related to each version is as follows:

1- When Version 1.0 is released to store a Server-side URL for this V1.0 (with CEEP configuration DBs for V1.0) is created and in mobile client is coded to use V1.0 CEEP Web Service Channel Manager URL

2- Going further when releasing a new major version to stores let’s assume it is named 2.0 a new Server side env is created for CEEP with a different URL 2.0 (with CEEP configuration DBs for V2.0) that support old and new services matching the Client V2.0

Application Version Management module can be used to eventually for users to upgrade to new env. And client version and thus allow the bank to decommission older Env. Of the production server