Digital Point of Sales

SAS – Software Architecture Document

Version 2.2

Finantix

email: info@finantix.com - web: www.finantix.com

Document info	
Document owner	Finantix
Author	Marco Terzo Tomasz Ponikowski Chi Ming Isaac Au-Yeung Paolo Di Patria Daniele Galesso
Document code	ML_SAS_POS
Version	1.1.1.1.1.1.1.1.1 2.2
Creation date	October 27, 2014
Last revision date	April 18, 2016
Distribution	Confidential
Status	Draft

Change history				
Version	***Date***	***Author***	***Description of changes***	***Approved by***
0.1	27.10.14	Marco Terzo	Document creation	
0.5	25.11.14	Marco Terzo	Solution Overview update, import facility description	
0.6	25.11.14	Tomasz Ponikowski	Agents and Security	
0.7	26/11/14	Marco Terzo	customer/prospect data import / export description	
0.8	27./1/14	Tomasz Ponikowski	Agents and Security chapters updated.	
0.9	28/11/14	Marco Terzo	Resources, notifications, update of customer/prospect data import/export as per review 1.	
0.10	28/11/14	Marco Terzo	Merge of agent + audit with version 0.9. Customer & prospect services update.	
0.11	5/12/14	Marco Terzo	Restructuring of document contents with more bullet points and short sentences. Removal of export webserices. Add data batch export. Description of required PDF export service.	
0.12	9/12/14	Marco Terzo	Minor changes on notification/resources Documentation of policy data model for import	
0.13	15/12/14	Tomasz Ponikowski	Customer/prospect – webservices, export	
0.14	16/12/14	Marco Terzo	Policy data model for second review, other minor changes and clarifications	
0.15	17/12/14	Tomasz Ponikowski	Document structure changes	
0.16	18/12/14	Tomasz Ponikowski	Chapter 5.2 - statement for Japan extension Chapter 5.2.1 - Customer to prospect conversion scenarios Chapter 5.3.1.1 - point 8,10, Figure 16,17 Chaper 5.4.1 - WS endpoints URLs	
0.17	19/12/14	Tomasz Ponikowski	Updates after review: MIL review 2 / Policies on 18.12 and Customer / prospect (chapter 5) on 19/12/2014 Audit and logging chapters were missing starting from version 0.10	
0.18	29/12/14	Tomasz Ponikowski	Chapter 5.6.1 updated for review on 30.12 Chapter “Apply to MySales WS” removed as confirmed by MIL that it is no longer needed in Proposal Traceability improved - chapters in FDS linked via numbers Chapter 4.3.3 – Approval flag in WS interface added – gap after review of v0.8 on 28.11 Chapter 4.2.4, 4.2.5 – explanation on agentCode field added	
0.19	31/12/14	Tomasz Ponikowski	Traceability improved –references to FDS chapters added FDS, MIL and 2015 PHASE1 terms aligned across the document Chapter 5.5.2 – introductory information and usage patterns added	
0.20	02/01/15	Tomasz Ponikowski	Chapter 2.2 – volumes information updated Chapter 2.3.1 added – requirement for “List of accounts not used for more than 3 months” Chapter 5.2.2 – sentence about data fields superset amended, Chapter 3.4 – points 7, 8 secure protocols named explicitly	

			Chapter 4.2.3 – reference to FDS for Resources and noti console Chapter 10.3.2 10.3.3 – information about the SOAP and REST WS authentication added Chapter 11 – Audit and logging updated - Configurability, rotation and retention policies Chapter 5.2.2 - number changed to Chapter 5.2.3 Chapter 5.2.2 - now contains information about agent switching Chapter 5.3.1.1 – point 8 – Japan future KYC requirement included; Figure 17 – DELETE action added and descirbed Chapter 5.4.5 updated – PDF report transfer Chapter 5.4.6 added – PDF Printing solution Chapter 5.4.7 added – e-Consent email	
0.21	09/01/15	Tomasz Ponikowski	Figure 18 – updated ML Systems lifeline – synchronous call removed Figure 19 – added diagram for web application Chapter 5.4.5.1 – reference to e-Consent chapter in FDS added, in WS signature Action input paramteter added, parameters “Customer email” and “OTP password” changed to optional Chapters 5.5.2.1-5.5.2.4 – reference to authentication chapter 10.3.3, information that data is returned by specific agent and last update timestamp added Chapters 5.5.2.2, 5.5.2.3 – questionnaire version added in output Chapter 10.3.3 – REST WS part removed – only SOAP webservices to be used in integration points as per confirmation from MIL Chapter 5.4.7 removed – e-Consent PDF and email sending covered in 5.4.5.1 Chapter 10.3.2 – added information about the staff users which in future phases of project be authenticated via separate AD	
0.22	12/01/15	Tomasz Ponikowski	Chapter 5.3.1.1, point 8 – KYC/FNA/RPQ/Calculators excluded from customer export Chapter 5.5.3 KYC/FNA/RPQ/Calculators batch export added Figure 17 – KYC/FNA/RPQ/Calculators remarks removed from XML example as not releveant anymore Chapter 5.6 Extended customer/prospect data integration servicese – SOAP webservices, updated diagrams and flows Chapter 14.3 added – Appendix C – Infrastructure Design Considerations Chapter 2.3.1 added – Information Risk Assessment (IRA)	
0.23	20/01/15	Tomasz Ponikowski	Chapter 3 Solution overview - Office 365 exclusion removed as available in chapter 4.3.1, PDF server side generation moved to chapter Assumption 3.4 to point 12 – row 6, 7 from Outstanding Items.xls send by MIL on 19.01 Chapter 3.1 Technical overview – Tables with legends for diagrams added, figure 3 and 4 for Propose and Apply marked as OUT OF SCOPE FOR 2015 PHASE1 – row 8 from Outstanding Items.xls send by MIL on 19.01 Chapter 3.3 Synchronisation – transactionality and “already sent to ML” – 8 from Outstanding Items.xls send by MIL on 19.01 Chapter 5.4.4 – point 4 – removed as redundant – row 27 from Outstanding Items.xls send by MIL on 19.01 Chapter 10.3.1 – reference to secure token added – row 33 from Outstanding Items.xls send by MIL on 19.01 Chapter 11.2.2 – table with audit event reformatted – row 36 from from Outstanding Items.xls send by MIL on 19.01 Chapter 5.6 Extended customer/prospect data integration servicese – SOAP webservices, updated diagrams and error handling added in description of the flows, agentId removed form input as it will be a part of SOAP header Chapter 10.3.4 – wording changed to reflect the analysis to be done when analysis for next country will commence	
0.24	22/01/15	Tomasz Ponikowski	Chapter 10.3.3 – description of authentication Definitions of 2015 PHASE1 and OUT OF SCOPE added and unified accrross documentation	

			Chapter 5.3.1 – explanation about sequence number in file names added Chapter 2.3.2 – references to PHA document resolved Chapters 6,7,8,9 – matked as N/A for 2015 PHASE1	
0.25	23/01/15	Tomasz Ponikowski	Chapter 4.3.2 – references to FDS document added to avoid duplication of information – removed status flows as described in functional documentation Chapter 13 – information about deployment filled Chapter 4.3.3 – refercnes to FDS added Chapter 4.3.2, Chapter 4.3.3, Chapter 6 – reference to users Chapter 5.6.authentication for consoles added Chapter 5.6.1.2.2 – name of webservice updated to getProposalList() from retrieveIndividualProposal() Chapter 2.3.2 – L1.1 reference to PHA resolved, missing H1.2 point added Chapter 6 – added references to FDS	
0.26	26/01/15	Tomasz Ponikowski	Refernces to SAS and IMS updated and aligned with FDS 0.23.04 and IMS v0.112 - table with references udated accordingly. Chapter 5.4.4 – PDF / email / printing updated, e-Consent PDF storage Chapter 5.4 – deletion of prospect removed Chapters 5.5.1.4, 5.5.3.2 – export files transfer to MIL via SFTP Chapter 5.4.5 – PDF priniting updated Chapter 5.6.1.4 – proposal printing added Chapter 3.3 – 2 stage synchronization described Chapter 5.7.1.1 – Figure policy data model updated	
0.27	28/01/15	Tomasz Ponikowski	Chapter 5.7.1.1 (5.8.1.1) – point 13 added to explain what happens if map / table section is not available in XML import Chapter 5.7.1.1 (5.8.1.1) – fixed list fields table added – definition is kept in FDS, reference added Chapter 5.3.1.1 (5.4.1.1) point 8 – changed to reflect that KYC/RPQ/FNA/Calculators will not be a part of import / export Chapter 5.2.3 (5.2.4)– references to “Basic Custmer Synchronised Data” and “Extended Customer Sycnrhonised Data” in FDS added – list removed to avoid doubled definitions Chapter 5.7 (5.8) – XML snippet removed for policy import Chapter 5.4.4.1 (5.5.4.1) – e-Consent email sending removed Chapter 5.3 for common import processes description inserted – initial / daily import and schedule chapters moved here from import chapters Chapter 5.2.3 added – .xls with scenarios for data feeds embedded Chapter 5.4.1.2 – customer / prospect actions added	
0.28	30/01/15	Tomasz Ponikowski	Addressed comments from review of chapters 1-4, 6-11	
0.29	04/01/15	Tomasz Ponikowski	Addressed comments from MIL on version v0.28	
1.0	09/03/15	Tomasz Ponikowski	Dummy “Open points” x.1 chapters removed from document Chapter 15.1 – import / export facilities and consoles description removed Chapters 5.2, 5.3.1, 5.5.1 – references to chapter 15.1 and non-existing subchapters removed Chapter 5.5.4 added – description of FNA/RPQ/Calculators export handling Chapter 14.3.1 – updated versions of iPad and iOS Chapter 5.6.1.2.2, 5.6.1.3 – Figure 20 and Figure 21 updated – previous versions were not correct (still had REST services and wrong WS name) Chapter 2.3 Performance SLA definition for POS application added “Business scope”chapter moved to chapter 1.2 Chapter 2 renamed to “Performance indicators and non-functional requirements”	

1.0.01	18/03/15	Tomasz Ponikowski Fabio Bau	Modified lastUpdateDate with lastAccessDate as criteria to be used for the purging of the record in 5.1 Clarification about default behaviour in 6 Clarification on 10.2.1 Update the description of OWASP Compliance in 10.5 Updated Perfomance part as communicated on 2015/03/12	
1.02	23/03/15	Cristian Ciot Fabio Bau	Update data structure image in 5.1.1 Basic/extended profile Removed reference to HK in log configuration, as it is common nto HK and Japan	
1.03	08/06/15	Cristian Ciot	Chapter 5.5 Prospect date export: updated signature for getKYCData, getFNAData, getRPQData and getCalculatorsData WebServices Chapter "5.6.1.1 redirectProposal Page WS": updated signature Chapter "5.5.3 KYC/FNA/RPQ/Calculators batch export": updated naming convention for exported files Chapter "12.1.2 Audited events": removed because details are already covered in FDS "MIS Logging" chapter Chapter "12.1.1 MIS Logging": renamed to "MIS Logging CSV Export" Chapter "15.1 Appendix A – Batch Integration with Manulife systems – General information": details of folder structure added	
1.3	02/10/15	Paolo Di Patria	Chapter "5.8 Import scenarios": DELETE customer action removed due to CR030	
1.3.3	01/12/15	Albert Wang	Merged HK Apply SAS content: Chapter "5.6.1.2 getProposalListWS / getProposalWS": web service signature is updated to include proposal data fields. Chapter "9 Applications and underwriting": SAS content for HK Apply is added. Chapter "14 Deployment": server-side environment and iOS version are updated.	
1.4.0	20/01/16	Marco Terzo	(Sprint 2 HK Apply integrations) 1.1.1 Removed old Chapter 7 1.1.2 index updated 1.1.3 updated all internal references 1.1.4 Added or modified chapters: 1.1.5 9.1.1 e-Disclosure autounderwriting flow 1.1.6 9.1.2 Application submit flow 1.1.7 9.2.1.1.1 getPlansFromFNAWS (exposed by ML Integration layer) 1.1.8 9.2.1.1.2 fnaValidationWS (exposed by ML Integration layer) 1.1.9 9.2.1.1.3 oneShotWS (exposed by ML Integration layer) 1.1.10 9.2.1.1.4 submitApplicationWS (exposed by ML Integration Layer) 1.1.11 9.2.1 Notification before the application expires 1.1.12 9.2.2 Batch Integrations 1.1.13 9.2.1.2 Hong Kong ComProp Integration Flow 1.1.14 9.4.1 Data retention rules for application records 1.1.15 10 e-Disclosure – offline XML rulebook and IF01 message specification 1.1.16 11.1.1 getExchangeRateWS (exposed by ML Integration Layer) 1.1.17 11.2.1 retrievePaymentGatewayTxNo (exposed by ML Integration Layer) 1.1.18 12.2.4 Active Directory Federated Services integration	
1.4.3	29/01/2016	Marco Terzo	Updated fnaValidationWS Renamed oneShotWS to auwSubmissionWS Update flow for offline (New proposal behaviour (getProposal) and data migration (NOT CONFIRMED) Added "List of user groups [1..n] in getExtendedAgentData	

			service	
1.5	1/03/2016	Chi Ming Isaac Au-Yeung	Updated 1. Asuze Integration On/Off line (Ipad) 2. Asuze Integration (Web) 3. Comprop Integration password handling 4. EPayment Integration 5.	
1.5.1	07/03/2016	Chi ming Isaac Au-Yeung	Updated Session SSO — 4.1.6 Comprop integration — 5.6.1.1 New requirement: Pass the user id and password from the URI. Updated Scope diagram (9.1) AUW — 9.1.1.1, 10.2 (IF01 message composition) Product and Plan Mapping (New Requirement). Application Submit flow (9.1.2) Integration Points — Online p.35 Application submit flow — (9.1.2) E-Signature & PDF (9.3.1) PDF generation (9.3.2) E-Payment Gateway (Session 11 Payments) New Interfaces and integration point E-Payment gateway (11. Payments) getExchangeRate Status URL Update getPaymentStatus paymentStatus retrievePaymentGatewayTxNo auwSubmission submitApplicationWS Update interfaces getProposal Batch Process Occupation list Product, plan code, flag mapping	
2.0	16/03/2016	Chi ming Isaac AuYeung	Updated Azure flow for MIL staff Updated document upload after submission Added new batch loading for fund and product mapping Added new WS	
2.0 (D01)	29/03/2016	Paolo Di Patria	Added: Opt Out Signal (HK R1 CR032)	
2.1	15/04/2016	ChiMing AuYeung	Added information for e-disclosure.	
2.2	16/04/2016	Daniele Galesso	Added description for: 1. APP version backward compatibility 2. Device compatibility	

Review Log			
Version	***Date***	***Reviewer***	***Approval***
0.5	26/11/14	TP, Manulife Technical Team (HK+Regional)	Not approved, authentication has to reviewed to cover MIL security policies
0.7	27/11/14	MT, Manulife Technical Team (HK+Regional)	Customer data update services not approved
0.8	28/11/14	TP, Manulife Technical Team (HK+Regional)	Approved and minor updates provided in version

			0.9 – email sent on 28.11.2014 11:50 pm
0.9	01/12/14	TP, Manulife Technical Team (HK+JP+regional)	Not approved. Non—functional requirements thread opened
0.11	09/12/14	TP, Manulife Technical Team (HK+JP+Regional)	Resources/notifications mainly approved (some checks on functional side)
0.13	17/12/14	TP, Manulife Technical Team (HK+JP+Regional)	Chapter 5 – not approved
0.15	18/12/14	TP, Manulife Technical Team (H\K+JP+Regional)	Chapter 5.7 confirmed – some points to be addressed on FDS side, clarification by MIL on statuses and behaviour to be provided
0.16	19/12/14	TP, Manulife Technical Team (HK+JP+Regional)	Chapter 5 – confirmed except 5.4.5 Customer/prospect PDF report transfer and 5.5.2 KYC/FNA/RPQ/Calculators webservices
0.17	30/12/14	TP, Manulife Technical Team (HK)	MIL needs to internally clarify the way how login/keep session alive may be implemented
0.20	09/01/15	TP, Manulife Technical Team (HK+JP+Regional)	Approach for PDF confirmed, webservices confirmed, authentication for SOAP WS confirmed, changes in documentation to be included by Finantix,
0.28	02/02/2015	==THIS VERSION IS CONDITIONAL SIGNOFF ON 02/02/2015 BY Ramaraj Sivakumar ==	==THIS VERSION IS CONDITIONAL SIGNOFF ON 02/02/2015 BY Ramaraj Sivakumar ==

Issue Log			
ID	***Issue***	***Owner***	***Status***

Open Points			
ID	***Open point***	***Owner***	***Status***

Referenced document	Chapter in referenced document	Referencing chapter
FDS	2.2.3 Passcode setup	2.2.1
FDS	2.3.1 Automatic logout	2.2.1
FDS	2.1 Logon	2.2.1, 4.1.4, 10.2.1
FDS	1.5 Definitions	2.2.1, 5.1, 5.1.1
FDS	3.2.2 Synchronise	3.3, 4.1.4
FDS	25.7 Roles and Groups	4.1.3, 4.2.2, 4.2.3
FDS	25.1 Resource Console	4.2.2
FDS	25.2 Notification Console	4.2.3
FDS	16.1 Profile	5.1
FDS	24 e-Consent	5.4.4.2
FDS	21.2 PDF Reports	5.4.4.1, 5.4.5
FDS	18.4 Opt Out Signal	5.4.4.4
FDS	26. Proposals	5.4.4.1, 5.4.5
FDS	9 Getting To Know You	5.5.2, 5.5.2.1
FDS	12.1 Financial Needs Analysis	5.5.2, 5.5.2.2
FDS	12.2 Risk Profile Questionnaire	5.5.2, 5.5.2.3
FDS	11 Calculators	5.5.2, 5.5.2.4
FDS	20 Policy Portfolio	5.7.1.1, 5.7.1.2
FDS	20.1 Protection and Savings	5.7.1.1
FDS	20.2 ILAS (Investment Linked Assurance Scheme)	5.7.1.1
FDS	20.3 Wealth Management	5.7.1.1
FDS	21 All Reports	3.4
IMS	Tab BAD (Basic Agent Data)	4.1.3

IMS	Tab EAD (Get Extended Agent Data WS)	4.1.4
IMS	Tab CN-WS (Create Notification WS)	4.2.3.1
IMS	Tab SA-WS (Sales Activities WS)	4.2.4
IMS	Tab CUSI-Batch (Customer Batch Import)	5.3.1.1
IMS	Tab PROI-Batch (Prospect Batch Import)	5.3.1.1
IMS	Tab PPDF-WS (preparePDFDelivery)	5.4.4.1
IMS	Tab RPDF-WS (retrievePDFRerport)	5.4.4.1
IMS	Tab CL-BE (Prospect/Customer (Client) Batch Export)	5.5.1.1
IMS	Tab KYC-BE (KYC Batch Export / KYC WS)	5.5.2.1
IMS	Tab FNA-BE (FNA Batch Export / FNA WS)	5.5.2.2
IMS	Tab RPQ-BE (RPQ Batch Export / RPQ WS)	5.5.2.3
IMS	Tab CAL-BE (Calculators Batch Export / Calculators WS)	5.5.2.4
IMS	Tab RPP-WS (redirecrProposalPage)	5.6.1.1
IMS	Tab GP-WS (getProposal WS)	5.6.1.2
IMS	Tab RIP-WS (get Proposal List)	5.6.1.2
IMS	Tab DPR-WS (deleteProposal WS)	5.6.1.3
IMS	Tab … (Print Proposal WS)	5.6.1.4
IMS	Tab GIA-WS (Get Insured and Available amount WS)	5.6.2
IMS	Tab POLI-Batch	5.7.1.1
IMS	Tab PDI-Batch (Policy dictionary batch import)	5.7.1.2
IMS	Tab CP-WS (Create Prospect WS)	5.4.2
IMS	Tab UP-WS (Update Prospect WS)	5.4.3

Table of Contents

1. Introduction

The software architecture for a system is the structure or structures of that system, which comprise software elements, the properties externally visible of those elements, and the relationships among them.

A technical architecture consists of several aspects that cover different and independent concerns; the full analysis can proceed by covering each aspect as a single domain. In a Finantix project the typical domains include:

1. Platform ⇨ Not functional requirements, such as audit and performance.
2. Security ⇨ User authentication and authorization. Integration with the legacy systems and mapping of roles. Data Security.
3. Data Integration ⇨ The interaction with existing data management systems, such as databases, Web Services, mainframe services.
4. Presentation Tier ⇨ The interaction with the user via a mobile device application, a web interface or other.

This document provides a high level overview of the technical architecture of Finantix Components.

1.2 Purpose

The Software Architecture Specification document (**SAS**) defines the software architecture for the project**.**

- It clearly states the systems that are part of the solution and the way they interact.
- It covers the non-functional requirements.
- It does not focus on functional requirements even though some references to business concept are possible.
- It does not focus on specific mapping requirements even though some references to those integration concepts are possible. For details about data mapping, please refer to the Integration Mapping Specification document (**IMS**).
- It is comprehensive. Any technical decision that differs from the SAS content or that is not in the SAS is a **change request**.

The document includes some sections “Development Notes”. Those sections describe implementation details that might change during the development phase.

References to current Manulife solutions for each argument described in the document can be found in the Appendix.

The scope of this document is to provide a clear definition of the system for Manulife.

1.3 Business scope

Digital Point of Sales application will be used by Manulife Agents to enroll new prospects, collect prospect and customer data, propose solutions for their needs and simplify the application, payment and underwriting process.

No data is supposed to be explicitly migrated to POS solution with dedicated processes. POS deployment will simply leverage on the daily batch import (see chapter 5.2 for further details) to execute a starting massive customer and prospect migration to POS data storage. Media resources and products currently stored on Manulife systems will be manually re-created in POS application using the dedicated console.

1.4 Definition, acronyms and abbreviations

Acronyms in alphabetical order:

CBA	Component Based Approach
DDF	Device Digital Foundation
FDF	Finantix Digital Foundation
FDS	Functional Design Specification
FNA	Financial Needs Analysis
IMS	Integration Mapping Specification
MIL	Manulife (International) Limited
O365	Microsoft Office 365
PHA	POS Hosting Architecture
POS	Point of Sales
RPQ	Risk Profile Questionnaire
SAS	Software Architecture Specification
SOA	Service Oriented Architecture
2015 PHASE1	Hong Kong Meet & Refine phase

OUT OF SCOPE Items are not in scope for Finantix POS application at all or for specified phase (e.g. OUT OF SCOPE FOR 2015 PHASE1)

In the next chapters the document refers to “Prospect” as to an entity that has been involved by an agent in a selling process but that does not actually own a policy. “Customer” is instead an entity that already owns at least one policy on Manulife systems. The word “Customer/prospect” instead is used to identify either a Customer or a Prospect.

1.5 References

- Digital Point of Sales – Integration Mapping Specification
- Digital Point of Sales – Functional Design Specification
- Manulife Data Dictionary for each specific application section

2. Performance indicators and non-functional requirements

2.1 Volumes information

POS application is designed to support the following volumes that represent the current Manulife usage of similar processes, tools or applications and the projection of usage of POS application in the next 3/5 years. Volumes provided for Hong Kong and Japan will be used as average reference also for the other countries.

Different data retention regulations applied by each country have been considered to properly size the application. In particular, for the first countries that will adopt POS application, the existing regulations define this constraints:

Data Dimension	POS Application (Offline data)	FNX/MIL server
Prospect	Follow Non-function requirement - 2 months	Permanent unless it is deleted by the agent
KYC	Follow prospect data (expect to sync back to server before deletion)	Permanent unless it is deleted by the agent
FNA	Follow prospect data (expect to sync back to server before deletion)	Permanent unless it is deleted by the agent
Calculation data	Follow prospect data (expect to sync back to server before deletion)	Permanent unless it is deleted by the agent
Proposal	N/A in this phase, 6 months when FNX Q&A is implemented (It is the current practice on existing online comprop	FNX server - N/A MIL - Follow existing online comprop - 6 Months
Audit	Clean up every time after sync up to server	FNX server - 3 months MIL server - 2 months (Will backup in tape after 2 months)
MIS data	Clean up every time after sync up to server	FNX server - 3 months MIL server - 2 months (Will backup

		in tape after 2 months)

Application Entity (*)	HK		JP	
	Day 0	After 5 years	Day 0	After 5 years
Agents	6.537	9.000	40.000	300.000
Prospects	9.000.000	14.494.590	110.000	tbd
Customers	3.600.000	5.797.836	40.000	1.800.000
Products	85	tbd	8	tbd
Riders	43	tbd	tbd	tbd
Policies	8.000.000	19.906.560	200.000	250.000
Proposals	550.000	885.781	200.000	1.800.000
KYC	0	300.000	0	2.500
FNA	0	300.000	0	0
RPQ	0	300.000	0	tbd
Calculators records	0	900.000	0	tbd
Audit records (38 client side points)	0	tbd	0	tbd

- (number of records on POS system)

Batch processes (number of records)	HK	JP
Prospect Initial import	9.000.000	110.000
Customer Initial import	3.600.000	40.000
Policy Dictionary Initial import	tbd	tbd
Policy Initial import	8.000.000	200.000
Prospect Daily import (new+update+delete)	600+800+(200)	3000+1500+?
Customer Daily import (new+update+delete+convert)	5.000+28.000+(0)+(1000)	250+100+?+?
Policy Daily import (new+update+delete)	1.900+225.000+(0)	2.000+21.000+?
Policy Dictionary Daily import (new+update+delete)	(50+50+0)	tbd
Prospect Daily Export	1600	tbd
KYC Daily Export	400	tbd
FNA Daily Export	400	tbd
RPQ Daily Export	400	tbd
Calculators Daily Export	1200	tbd

Batch processes (available time window)	HK	JP
Prospect Export	1:00 AM - 4:00 AM	1:00 AM - 4:00 AM
KYC Export	1:00 AM - 4:00 AM	tbd
FNA Export	1:00 AM - 4:00 AM	tbd
RPQ Export	1:00 AM - 4:00 AM	tbd
Calculators Export	1:00 AM - 4:00 AM	tbd

Prospect Import	5:00 AM - 8:00 AM	5:00 AM - 8:00 AM
Policy Dictionary Import	5:00 AM - 8:00 AM	tbd
Policy Import	5:00 AM - 8:00 AM	tbd
Customer Import	5:00 AM - 8:00 AM	tbd
Occupation	On-Demand	TBD
Fund Configuration	On-Demand	TBD
Rulebook	On-Demand	TBD
Fund Plan Mapping	On-Demand	TBD
Product, plan, flag mapping	On-Demand	TBD

Web Services (# of calls during peak hour)	HK	JP
retrievePDFReport WS	300	300
KYC WS	300	300
FNA WS	300	300
RPQ WS	300	300
Calculators WS	300	300
getProposal-WS		
AUWS-WS		
BAD2-WS		
EAD-WS		
SUBA-WS		
getExchangeRateWS		
paymentStatus-WS		

Other volume information	HK	JP
Peak period for WS call	10:00-11:00 and 14:00-15:00	10:00-11:00 and 14:00-15:00
Peak period for login (and synch)	10:00-11:00 and 14:00-15:00	10:00-11:00 and 14:00-15:00

Other volume information	HK		JP	
	Average	Maximum	Average	Maximum
# Customers/Prospects per agent	500	tbd	20	tbd
# Policies per customer	1	10	3	tbd
# Proposals per customer	1	10	5	tbd
# Resources downloaded by agent on device	50	50	30	30
Size of resource (media)	7MB	n.a.	500KB	n.a.
# Prospects and customers synchronized by an agent	tbd	tbd	tbd	tbd

2.2 Non-functional requirements

2.2.1 Information Risk Assessment (IRA) safeguards

Ref.	Information Security Risk	Safeguards	Finantix reference
L1	Unauthorized modification of information	L1.1 Data file stored in the system cannot be accessed other than system administrator, which is a restricted privilege account. L1.2 Access permission is restricted on need basis. L1.3 SSL.	L1.1: Document Name - Access Control Process V5.0, Section 4 and Section 8 Document Name - ISMS Policy V1 0, Section 8 L1.2 – chapter 4.1.3 L1.3 – chapter 3.4, point 5, 6, 8
L2	Deletion of valid information (electronic data)	L2.1 Daily backup arrangement is required by Finantix to ensure the availability of data files. L2.2 Backup tapes are kept in off-site and are protected by password L2.3 Only authorized users can modify data in system	L2.1 Backup details are not part of SAS document and will be covered by hosting solution L2.2 Backup solution are not part of SAS document and will be covered by hosting solution L2.3 – chapter 10.2
L4	Unauthorized access to client's data by third parties other than the authorized advisors	L4.1 Client data will be encrypted with https transfer for security safety, L4.2 Client data stored in the iPAD will be protected by endorsed Mobile Application Management (MAM) solution L4.3 User identification is unique and its access right is reviewed on an annual basis. L4.4 User access is authenticated by strong password control: v 8 characters with alpha and numeric combination v password is required to change at first logon and at most every 45 days v session idle time of 15 minutes is implemented. If the session is idle for 14 mins, a warning will be issued, and advisor can click to confirm continue session, if not, the session will be expired at 15th min. v password is encrypted and stored in the authentication server v account will be locked after 3 bad attempts v password generation = 8 v Audit trail report which should logged down the invalid login activities (both online & offline) for each user, is required to generate and reviewed by Finantix on daily basis	L4.1 – chapter 3.4 L4.2 – Symantec Mobility Suite would provide client data protection L4.3 – OUT OF SCOPE of Finantix POS application L4.4 – OUT OF SCOPE of Finantix POS application, except audit trail – please refer to chapter 11.1 and "session idle time" – please refer to FDS, chapter 2.3.1 Automatic logout, report

L5	Unauthorized modification of information - Data may be hacked during transmission	L5. 1) All data communicate between mobile application and MIL servers are protected by SSL 2) New token will be generated for each transmission involve API	1) see L1.3 above 2) all the connections are over HTTPS, no public APIs involved
L8	Using password cracking tool	L8.1 SSL L8.2 Access to password authentication server is restricted to system administrator and approved vendor support.	L8.1 See L1.3 above L8.2 OUT OF SCOPE of Finantix POS application
L9	Guessing the password of a privileged / typical user	L9.1 Intrusion / detection tool is required in Finantix solution	Intrusion detection tool would be part of the solution. Corero IPS devices would fulfil the requirement.
L10	Exploiting vulnerabilities to gain access or to bring down the system	L10.1 IT infrastructure team developed a regular vulnerability scanning process. L10.2 Problems will be identified and resolved on a timely basis	L10.1 Please refer Sec 10.6 of the document ISMS Policy V1 0 L10.2 Please refer Sec 12, 16.2 of the document ISMS Policy V1 0. Note that the requirement would be fulfilled by Xchanging and Finantix.
L11	Guessing user identity.	L11.1 Success user authentication is required to access system resources. By knowing user identity itself could not be harmful to the information.	chapter 10.2
L15	Unauthorized access and modification of source program	L15.1 Source program is stored in a designated folder, which is protected by folder password and is restricted to librarian only. L15.2 All changes and modification of source program are controlled by a program change process, which is executed by IT custodian. Users cannot make change to the program directly.	L15.1 Source code is managed by SCM (Software Configuration Management) servers where only authorised users can access L15.2 Changes in source code are part of Finantix S-SDLC, which takes care of validating the modifications, via SAST software (SonarCube) and revisions by developer managers.
L17	Unauthorized application for user account maintenance.	L17.1 Appropriate approval authority is checked before processing user accounts.	OUT OF SCOPE for Finantix POS application
T1	Capture a session between the servers to listen passively or with a view to intercept, modify or delete information.	T1.1 Servers are physically secured in Xchanging data centre to prevent from unauthorized installation of sniffing tools. T1.2 Servers are physically secured in Finantix data centre to prevent from unauthorized installation of sniffing tools.	T1.1 Please refer Sec 10.5, 10.6 of the document ISMS Policy V1 0. T1.2 Servers are hosted by Xchanging, please refer to T1.1
T7	Unauthorized printing	Only authorized email use these method	Outside Finantix POS application. In case if printing will be realised via email to print solution, MIL to provide authorised email account to handle.

T8	Unauthorized printout retrieval	Printers locate in restricted office premise with authentication	Outside Finantix POS application
T9	Data loss during data transfer between Manulife and Finantix	T9.1 For the data sync between Manulife and Finantix, the data feed will be send from Manulife to Finantix by sFTP under a dedicated channel. T9.2 For the web service between Manulife and Finantix, The web services is either REST or SOAP over https and IBM DataPower in a dedicated channel.	T9.1 Outside Finantix POS application. Finantix expects files in the file system before the import process starts - there is no check implemented on Finantix side. Please refer to chapter 14.1.1 for details of Finantix import facilities. T9.2 chapter 3.4, point 66
S1	Hardware or communication overload or failure	S1.1 Review system performance and usage report by Finantix S1.2 Discuss and resolve the performance problems or anticipated issues with IT teams. S1.3 Backup and recovery plan will be executed when required.	S1.1 Xchanging would provide the system performance reports (CPU, Memory, IO and Network Reports). S1.2 Xchanging would share the system reports and possible bottlenecks with Finantix. Xchanging, Finantix would discuss about the potential issues that could arise in the future and how proactively fix those issues. S1.3 Please refer Business Continuity Plan v1 0
S2	Unstable internet connection	S2.1 Sufficient capacity planning, internet bandwidth and technical infrastructure are well addressed to business requirement during design and implementation. On-going monitoring and annual review on capacity planning to ensure the production performance	Initial architecture is designed to support the business and techinical requirements in terms of networking, servers and storage. As part of the capacity planning, constant monitoring on KPI will provide a proactive way to adapt the infrastructure to usage peaks. Rest assured Manulife will promptly communicate new business or technical requirements (e.g. higher number of users, new countries) as Change Request
D1	Tampering with log files	D1.1 Electronic copy cannot be modified. D1.2 Log files are generated automatically and are sent to reviewer in a sealed envelope directly.	D1.1 – Please refer Sec 10.4.2 and 10.4.3 of the document ISMS Policy V1 0. D1.2 OUT OF SCOPE of Finantix POS application; Application log files are generated automatically and possibly stored on ML log servers via secure channel (e.g. SFTP)
D2	Inadequate monitoring of logs	D2.1 A management process over log review is established and implemented to ensure logs are reviewed, investigations are performed on a timely basis.	D 2.1 - Please refer Sec 10.4.2 and 10.4.3 of the document ISMS Policy V1 0. Critical log files would be analyzed frequently.

D3	Inadequate analysis of access to detect patterns of access	D3.1 Audit trail of privileged account activities and logon violations are generated and reviewed. Criteria of review is defined, any suspicious event will be followed up by team managers.	D 3.1 - Please refer Sec 10.4.2 and 10.4.3 of the document ISMS Policy V1 0. Critical log files would be analyzed frequently.
G1	Using information for fraudulent activities	Manulife has developed a set of employee conducts/agent code of conduct that must be abided by all staff/agent. All staff are required to sign and acknowledge the conduct on an annual basis. All agents are required to sign and acknowledge the conduct at the moment of being contracted. Various measures including training and regular reminder messages to agents have been established to inform the agents of and to reinforce them the proper code of conduct and business ethics. Vendor needs to sign on the Non-disclosure Agreement regarding how to handle Manulife's information	OUT OF SCOPE of Finantix POS application
G2	Falsify electronic transaction or message.	Manulife has developed a set of employee conducts /agent code of conduct that must be abided by all staff /agent. All staff are required to sign and acknowledge the conduct on an annual basis. All agents are required to sign and acknowledge the conduct at the moment of being contracted. Various measures including training and regular reminder messages to agents have been established to inform the agents of and to reinforce them the proper code of conduct and business ethics. Outbound message to external parties are independently reviewed before delivery.	OUT OF SCOPE of Finantix POS application
G3	Unethical usage of system resources.	Manulife has developed a set of employee conducts /agent code of conduct that must be abided by all staff /agent All staff are required to sign and acknowledge the conduct on an annual basis. All agents are required to sign and acknowledge the conduct at the moment of being contracted. Various measures including training and regular reminder messages to agents have been established to inform the agents of and to reinforce them the proper code of	OUT OF SCOPE of Finantix POS application

		conduct and business ethics. Report on system usage and activity pattern is reviewed by managers on a request basis.	
G5	Stolen devices (eg. iPad, iPhone)	G5.1 Data stored in iPad, iPhone virtual memory (sandbox) is protected at file level encryption provided by iOS security framework. Even if users lost their mobile device, others will not be able to hack the data. G5.2 Five consecutive logon failures (follow existing practice in Manutouch G5.3 Re-logon is required for lost connection for 15 mins. G5.4 Reminder is prompted to require set up of device passcode during installation of App. The passcode is an application control for switching between the customer view (i.e information can be shown for customer/prospect) and agent view (e.g. information should be read by agent only) G5.5 The Application requires AWS login ID and password to access. Control is built and restricted to authorised distributors only. G5.6 The offline data will be remote wipe thru the MAM solution G5.7 User is required to perform online synchronization at least every 3 days (72 hours). Otherwise, user will not able to access any components or data on offline app. G5.9 Apart from the security features offered by MAM solution, there are addition requirements as control to mitigate the unauthorized usage of data a) a customer/prospect profile (offline data) has been downloaded but inactive for 2 months, the data should be automatically removed`. User can re-download the profile anytime on demand basis. Inactive means no access from any of POS components, except search result. b) Generate a report on daily basis with a list of account and devices, which have been registered before but no longer sign-in for more than 3 months. De-registration will be triggered and the app and offline data will be removed and swiped accordingly	G5.1 chapter 10.4 G5.2 FDS, chapter 2.1 Logon FDS, chapter 2.3.1 Automatic logout G5.3 FDS, chapter 2.3.1 Automatic logout G5.4 FDS, chapter 2.2.3 Passcode setup G5.5 OUT OF SCOPE of Finantix POS application G 5.2 – Symantec mobile security suite (MAM) would detect and alert helpdesk in the case of consecutive logon failures. G 5.6 - Symantec mobile security suite (MAM) would remotely wipe the POS application data when the mobile device lost/stolen. G5.9 a) For inactive customer/prospect profile data removing please refer to FDS, chapter 1.5 Definitions (definition of “Downloaded (in the context of Customer)”); b) requirement is to be covered by service catalogue provided by Finantix.
G7	Inadequate backup and recovery plans	G7.1 Backup and recovery plans are reviewed and revised at least on an annual basis. G7.2 Problems encountered in the daily	G7.1 – Backup and Recovery plans would be tested every year, the results would be reviewed and revised if

		backup procedures are required to escalate to MIL. Regular update to MIL is required until it is resolved.	required. G7.2 Finantix will manage in case of daily backup job failures and will keep MIL updated until it is resolved
H1	Alter device configuration	H1.1: Jailbreak detection is required. Jailbreak version of iPad, iPhone is prohibited H1.2 An on-going hardening process and security review is performed to ensure the security. H1.3 Security patch will be updated according to the Manulife IT hardening standard and procedures	H1.1 Symantec mobile security suite (MAM) would identify Jailbroken and rooted devices H1.2 Finantix will ensure that hardening process implemented by Xchanging will be compliant with MIL policies H1.3 Symantec mobile security suite (MAM) would force the app upgrades, if required. When a new version of a wrapped app is published, end-users can be forced to upgrade by having the wrapped app immediately take the end-user to the app store client upon launch

2.3 Performance SLA definition for POS application

This section is for reference. The detail of the SLA requirements will be in the official SLA documents.

Assumptions:

A.	This document refers to Meet & Refine Hong Kong phase 1 (release 1 and 2)
B.	The application performance, data sync, for the online operations is defined considering an average of no more than 30 concurrent requests satisfied in the response time. The provided response time values for online operations have been calculated considering an average distribution of this 30 concurrent requests against the below scenarios as per following table: G1. - 2 G2. - 3 H. - 1 I. - 11 J. - 13 *** The number of concurrent user is subject to change based on the stat from the systems time-by-time and aligned with the non-functional requirements.
C.	The total number of users is limited to the current Hong Kong declared volume, that is, about 7000 users.
D.	Proposal & Apply functionalities as well as the extension to other countries and the projection on the next years will imply increased amount of data to be analysed in respective design phases.
E.	The standard use case scenario considered for the following definition is: - a user installs the application on her/his iPad; - after install, the user executes the first login and the first time synhronization; - every day, the user in the morning synchronize her/his device with POS server using a good quality connection (at home or in the office) and then move out to meet clients; - during the day the user works mostly off-line with minimal online interactions to print some PDF reports;

	- sometime during the working hours the user downloads few new resources or customers data; - at the end of the day the user synchronizes again her/his device with POS server using a good quality connection (at home or in the office).
F.	The estimated available bandwith considered in the following performance definition is 100Mbps.
G.	First synchronization average response time for the blocking stage has been defined considering an average of: - 3000 basic customer data records to synch, - 300 resource basic data (download only from server to device), - 20 notifications, - 20 products (download only from server to device, limited to M&R scope), - all configurations and translations, The total size of these data is considered to be about 15MB (download only). After the first synch, each user will download approximately 50 resources (3MB each on average) for a total of 150MB.
H.	Worst case scenario average response time for a full synchronization after a re-installation (e.g. for a user that already worked with POS and changed her/his device) has been defined considering same data as above plus 200 customers full data downloaded in background, each of which with: - 10 notifications, - 5 forms data, - 20 records of previous reports, - 5 policies, - 20 meeting notes. The total size of these background data is considered to be about 90MB (download only).

I.	Daily synchronization average response time for the blocking stage has been defined considering an average of: - 100 basic customer data records to synch, - 5 resources basic data to download, - 2 products to download (contents limited to M&R scope), - 2 notifications to download, - 10 schedule events and meeting notes to synch, - rarely few configurations and translations to download. The total size of these data is considered to be about 400KB in download and 100KB in upload.
J.	Daily synchronization average response time for the background stage has been defined considering an average of: - 20 customers full data to synch (each of which with: 5 notifications, 3 forms data, 5 records of previous reports, 2 policies, 20 meeting notes), - 5 resources (1MB each on average), - 400 audit records (upload only from device to server). The total size of these data is considered to be about 24MB (15MB download, 9MB upload).
K.	Synchronization and import/export average throuput have been defined considering the volumes agreed on the POS Software Architecture Specification document version 0.28 (e.g. about 230.000 policies updated every day).
L.	The following average response times represent the standard response time achievable on 85% of the cases. Some umpredictable factors might increase the response time in a small amount of cases (e.g. not predictable connection speed reduction or other operations running in background on the device).
M.	Device-Server communications and Web browser-server communications have been evaluated in three scenarios: 1. A standard WIFi connection over a good connection local provider (30Mbps download, 10Mbps upload) fully available to the application. 2. A standard 4G connection (20Mbps download, 7Mbps upload) fully available to the application. 3. A standard 3G connection (7Mbps download, 0.6Mbps upload) fully available to the application.
N.	On operations that require a call to Manulife services, the declared average response time excludes the actual call to the external Manulife system. The total perceived response time might be increased depending on the actual response time of the called Manulife service (e.g. prospects synchronization with internal call to Manulife systems to have the new prospect id).

Offline application pages and actions on device

User Transaction or Action	Average response time less or equal to (sec)	Specific Assumptions
Login Authentication	< 0.5	
Open Agent Dashboard	1-3	Minimum and maximum total time to load the contents of all panels depending on the panel contents. The agent dashboard is loaded immediately with empty panels that will be filled one after the other starting from the first one visible
UI Language Switching (average page content)	0.5-2	depends on the actual page shown by the application
Open Customer List	0.5	
Customer Search	1	
Field Level Customer Search(Auto-suggestion list)	1	
Pin/bookmark a Customer	< 0.5	
Open About Me	0.5	
Open Meeting Notes	1	
Open Sales-Aids Library (resources)	0.5	
Open eConsent	1	
Open KYC	1	
Open FNA, RPQ	2	for first form page opening, succeeding pages are <0.5sec
Open Solution Library	1	
Open Simplified Calculators	1	

Creation of New Prospect	1	
Open Holistic Customer View for a Customer	1-3	minimum and maximum total time to load the contents of all panels depending on the panel contents
Toggle Wealth Management Portfolio Chart	0.5	
Open Any Content in Resource Library	< 0.5	
Open Any Product in Solution Library	0.5	
Generation of FNA Summary Page	1	
Calculation of RPQ Result	< 0.5	
Reflective Chart on each Simplified Calculators Upon User Change Any of The Values	< 0.5	
Switching from Customer mode to Agent Mode (after correct 4 digit PIN inputted)	<0.5-2	It depends on the number of panels that have to be updated after the previous operation
Move from Page to Page (applicable for components with multiple pages)	< 0.5	
Open of Notification Message	0.5	
Create or Edit of Calendar Event	0.5	
Embedding on-Glass Signature	< 0.5	
Financial Report Generation (on-Screen)	2	

Online application pages and actions on device

User Transaction or Action	Average response time less or equal to (sec) - WiFi	Average response time less or equal to (sec) - 3G	Specific Assumptions
Login Authentication	2	3	
Download Customer Holistic Profile	3	8	single customer data download (full set)
Open Holistic Customer View for a Customer with data preview for not already downloaded customers	3	5	Total time to load the contents of all panels by invoking dedicated onlne services. The agent dashboard is in any case loaded immediately with empty panels that will be filled one after the other starting from the first one visible
Financial Report Generation (in PDF format)	5	15	based on an average 1MB long report
Email or e-Delivery Submission	2	1	without considering the mail server response time
First time synchronization and data download (Assumption G first part)	20	70	
First time synchronization resource manual download (Assumption G second part)	100	600	After the initial download of the clients and resources metadata, the user will activate manually the download of the resources of interest
Worst case scenario re-installation synchronization (Assumption H)	210	1000	After the initial download of the clients and resources metadata, the user will activate manually the download of the clients extended data and resources of interest
Daily synchronization (blocking stage, Assumption I)	5	30	

Daily synchronization (background stage, Assumption J)	25	120	total time, considering the audit (slowest upload) and the client details executed in background
Download single resource	3	8	Average resource considered is 3MB. The response time depends directly on the resource size and available connection

Integration points - online

Point of Integration (POI)	Protocol / Data Format	Synopsis	Service Caller	Service Provider	Average response time less or equal to (sec)
Basic Agent Data2	SOAP/XML	Gets basic attributes from LDAP (id and groups)	Finantix POS	ML	1
Azure AD	SAML	SSO	Finantix POS	ML-Azure	??
Get Extended Agent Data WS	SOAP / XML	Get additional details for agent (name and licenses information)	Finantix POS	ML	1
Create Prospect WS	SOAP / XML	This service will create/update the prospect on Manulife systems return an ID for the new prospect.	Finantix POS	ML	2
Update Prospect WS	SOAP / XML	This service will create/update the prospect on Manulife systems return an ID for the new prospect.	Finantix POS	ML	2
preparePDFDelivery	SOAP / XML	Send to ML request to send e-consent email or PDF to the client	Finantix POS	ML	1
retrievePDFReport	SOAP / XML	Generates PDF report requested by ML	ML	Finantix POS	5
getProposalList	SOAP / XML	(Comprop) Retrieve all the proposals which belongs to the client	Finantix POS	ML	2
redirectProposalPage	SOAP / XML	(Comprop) Return URL for opening proposal page in Comprop	Finantix POS	ML	1

getProposal	SOAP / XML	(Comprop) Retrieve the physical file of the proposal (PDF)	Finantix POS	ML	2
deleteProposal	SOAP / XML	(Comprop) Delete the proposal	Finantix POS	ML	1
printProposal	SOAP / XML	(Comprop) Print request for the proposal	Finantix POS	ML	1
KYC Export	SOAP / XML	Export KYC information for prospect/customer collected by specified agent	ML	Finantix POS	4
FNA Export	SOAP / XML	Export the latest FNA questionnaire question id's and answers	ML	Finantix POS	4
RPQ Export	SOAP / XML	Export the latest RPQ questionnaire questions and answers together with the risk profile scoring collected by specified agent	ML	Finantix POS	4
Calculators Export	SOAP / XML	Export the latest Calculators input/output collected by specified agent	ML	Finantix POS	4
getExchangeRateWS	SOAP / XML	Get Exchange Rate	Finantix POS	ML	
auwSubmissionWS	SOAP / XML	Auto underwriting submission	Finantix POS	ML	
submitApplicationWS	SOAP / XML	Application Submission 2, 3, and 4	Finantix POS	ML	
PaymentStatus_WS	SOAP / XML	Provide payment status from e-payment to ML	Finantix POS	ML	
retrievePaymentGatewayTxNo WS	SOAP / XML	Reference number for e-payment gatement	Finantix POS	ML	

Integration points – batch

Point of Integration (POI)	Protocol / Data Format	Synopsis	Service Provider	Average throuput (record/sec)

Prospect Batch Import	Batch Import / XML	Prospect batch import. Include only 'Base Data'	ML	100
Customer Batch Import	Batch Import / XML	Customer batch import. Include only 'Base Data'	ML	100
Prospect/Customer Batch Export	Batch Export / XML	Export contains Client Data (no KYC,FNA,RPQ)	Finantix POS	200
Policy batch import	Batch Import / XML	Daily batch import for policies	ML	100
Policy dictionary batch import	Batch Import / XML	Daily import for labels and translations used in policy component	ML	250
KYC Batch Export	Batch Export / XML	Export KYC information for prospect/customer collected by specified agent	Finantix POS	150
FNA Batch Export	Batch Export / XML	Export the latest FNA questionnaire question id's and answers	Finantix POS	140
RPQ Batch Export	Batch Export / XML	Export the latest RPQ questionnaire questions and answers together with the risk profile scoring collected by specified agent	Finantix POS	140
Calculators Batch Export	Batch Export / XML	Export the latest Calculators input/output collected by specified agent	Finantix POS	150
Occupational List	Batch Import / XML	Import occupation list to FNX	Finantix POS	
Product and Plan mapping	Batch Import / XML	Import product, plan mapping with flag information	Finantix POS	
Fund Configuration	XML	Import fund information	Finantix POS	
RuleBook	XML	Rule Book	Finantix POS	

3. Solution overview

The Manulife regional Point of Sales (POS) solution is designed with a Component Base Approach (CBA) in order to support Manulife distribution channels and to deliver a customer centric holistic solution for the sales process.

One of the main purpose of this solution is to provide Manulife agents with a unique application available both on traditional desktop browsers and on mobile tablet devices

On mobile devices, POS application can be used also in “offline” mode whenever the network connection is not available or too slow.

The following architecture design refers in particular to Apple tablet devices (iPad), but the possible extension to every kind of tablet device (Android and Windows) have been considered during the definition of POS architecture.

When necessary, each process and integration described here will explicit reference to the “mobile” solution or to the “web” solution (web solution is OUT OF SCOPE FOR 2015 PHASE1).

3.1 Technical overview

POS application in mainly split in four parts that will be covered by the following architecture specification:

1. Agent dashboard (with notifications, resources and application settings)
2. Meet and Refine (with customer/prospect search and data collection)
3. Proposal
4. Apply

Each part has its technical peculiarity leveraging on some common elements provided by Finantix Digital Foundation, such as import facilities, auditing and logging.

The following pictures describe at high level the Finantix Components involved on each part of MIL POS application and the integration points of those components.

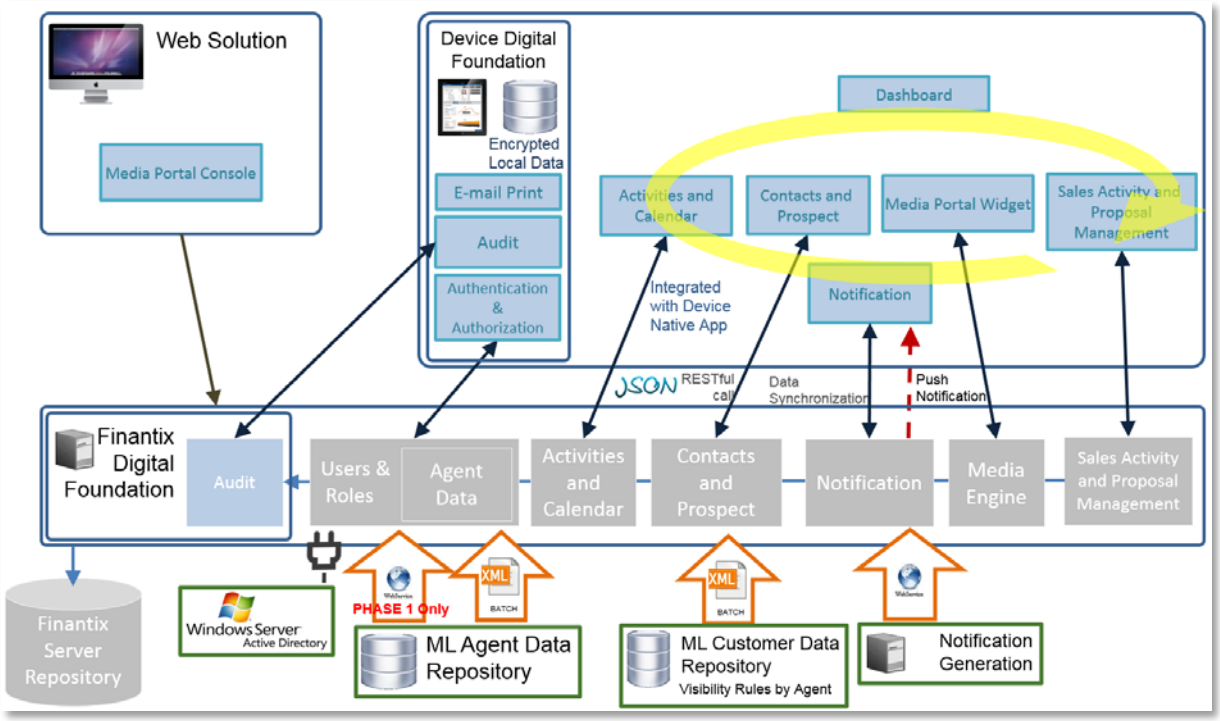


Figure 1 – Agent Dashboard and customer data components and integrations

Integration point	Description
Windows Server Active Directory	*Identity and authentication provision for Finantix POS system (LDAPS protocol). Please refer to chapter 4.1.4*
ML Agent Data Repository (inbound)	*SOAP XML webservice exposed by MIL to get agent related data which do not come from Active Directory. In later phases of project XML batch import will be used. Please refer to chapters 4.1.4, 0*
ML Customer Data Repository	*MIL systems hosting customer / prospect data. Interfaced by XML batch import integration. Please refer to chapter 5.2*
Notification Generation	*Webservice integration specifically requested by Japan (OUT OF SCOPE FOR 2015 PHASE1), please refer to chapter 4.2.3.1*

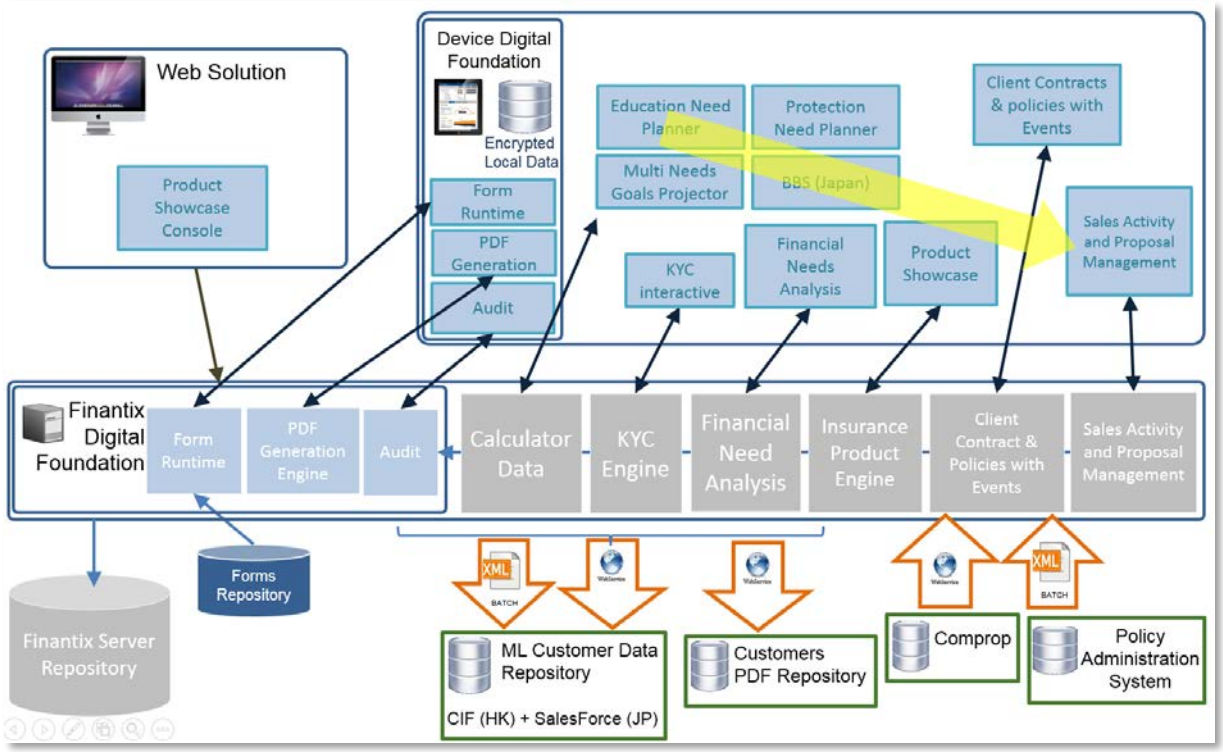


Figure 2 – Meet and Refine components and integrations

Integration point	Description
ML Customer Data Repository (outbound)	*SOAP XML webservices exposed by MIL to create/update/delete prospect (please refer to chapter 5.4.2), XML export of customer / prospect data from Finantix POS (please refer to chapters 5.5*
Customers PDF Repository (outbound)	*Finantix POS PDF reports transfer mechanism to MIL (please refer to chapter 5.4.4)*
Comprop (inbound)	*SOAP XML webservices integration for "Proposals" (see FDS, chapter 27), please refer to chapter 5.6.1*
Policy Administration System (inbound)	*XML batch import for policy information (please refer to chapter 5.7*

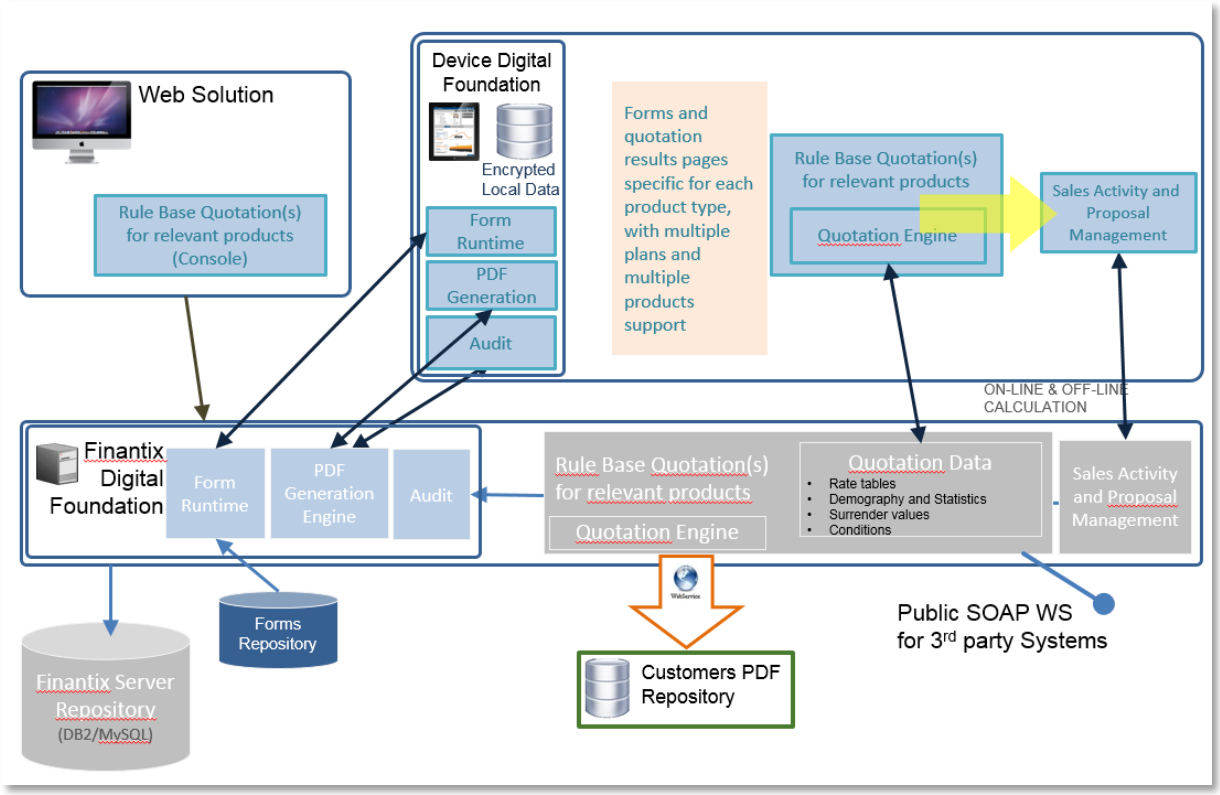


Figure 3 – Proposal components and integrations (OUT OF SCOPE FOR 2015 PHASE1)

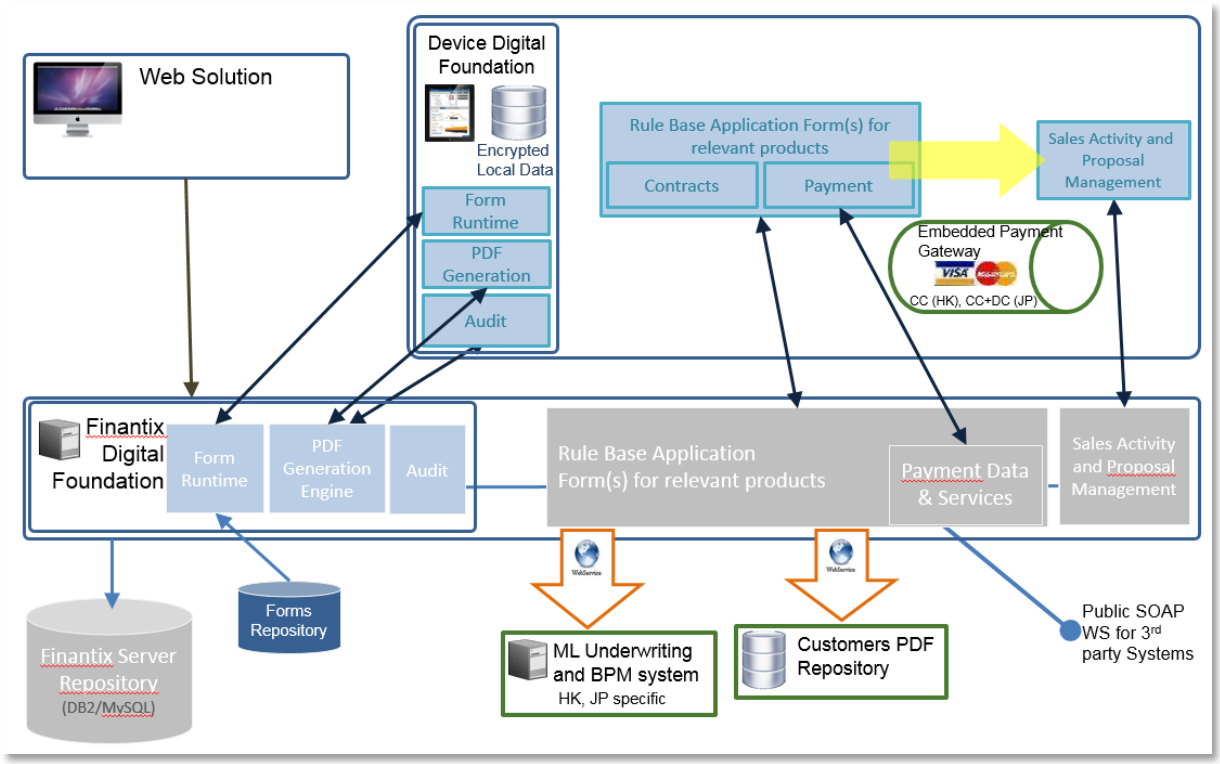


Figure 4 – Apply components and integrations (OUT OF SCOPE FOR 2015 PHASE 1)

3.2 Finantix Digital Foundation

Finantix Digital Foundation (FDF) is a set of tools and APIs that represent the common core application on top of which every Finantix Business Component is built. It's the FDF that manages the synchronization between the POS application installed on a tablet device and the POS server in order to allow the offline usage of the application. Moreover FDF provides common audit and logging facilities and the core authentication and authorization mechanisms.

The following picture describes the element and layers inside each business component.

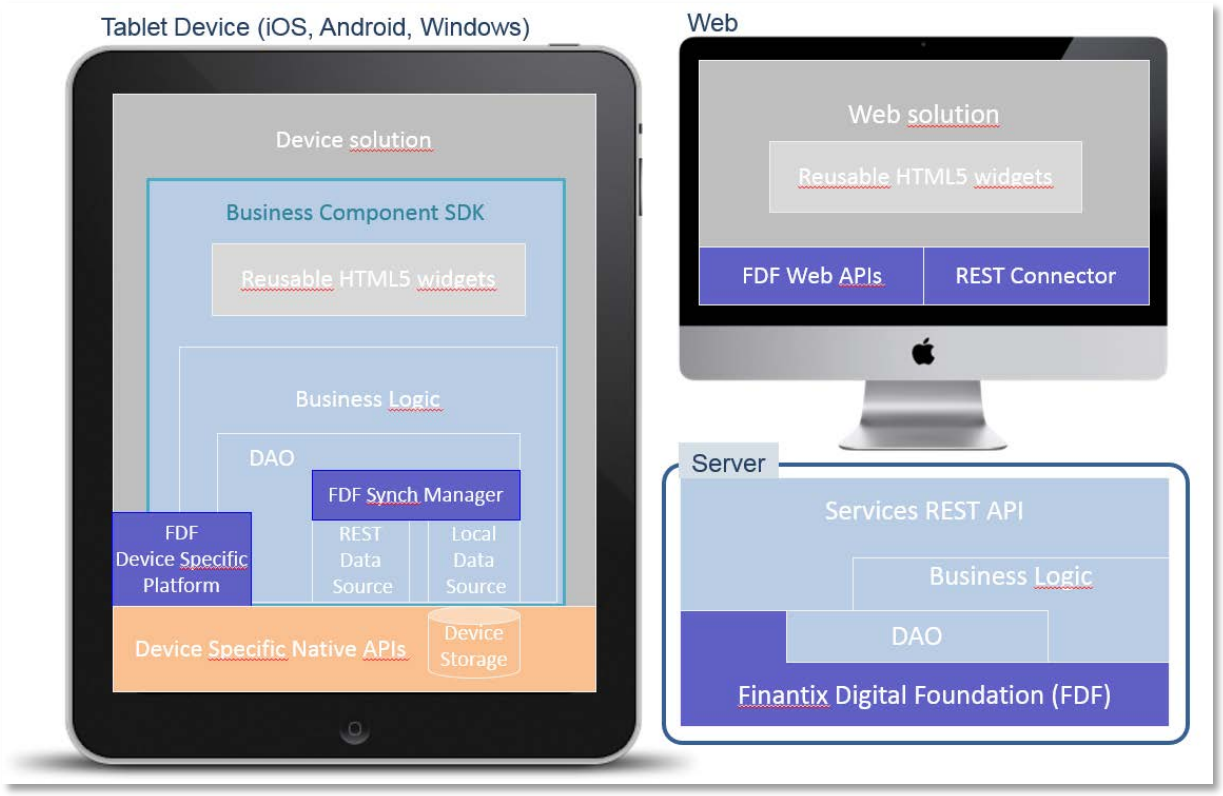


Figure 5 – Apply components and integrations

Finantix Digital Foundation allows also an administrator to publish a different set of translations shared between the device and web solutions. Those translations can be later updated by the device application on the next synchronization.

3.3 Device data and configuration synchronization

This chapter specifically describes mechanisms of Finantix POS application synchronization with Finantix server side (FDF).

To allow a full offline usage of the application, all data necessary to all components to properly operate have to be stored inside the device local storage.

FDF provides an OOTB synchronization mechanism that can be triggered automatically after the login or manually by the user:

- When the synchronization is started, all data entities defined in the data model and configured for remote synchronization are separately synchronized with the server.
- The data model is designed accordingly to this behaviour and to Manulife requirements to define a proper data consistency on the device.
- The device application mainly works “Off-line”: the online status is necessary only for initial and periodic authentication, data synchronization and PDF generation.
- Some features of the POS application, in particular on proposal & apply phase, will be available only when on-line.
- The synchronization server **keeps track of the version of each server-side entity**, in terms of a revision number which is incremented each time an entity is updated.
- The synchronization device app, on its side, monitors customer/prospect-side activity, keeping track of which new entities have been created and which existing entities have been updated or removed since the last synchronization.

Each entity in the device database is in one sync status among those in this list:

- **CREATED**: The entity did not exist and has been created on the device
- **UPDATED**: The entity existed and has been modified on the device
- **REMOVED**: The entity existed and has been removed on the device
- **NONE**: The entity existed and has neither been updated nor removed on the device

The following diagram shows the state machine showing transitions between sync status of an entity (for simplicity, the status transitions in case of synchronization conflicts have been omitted):

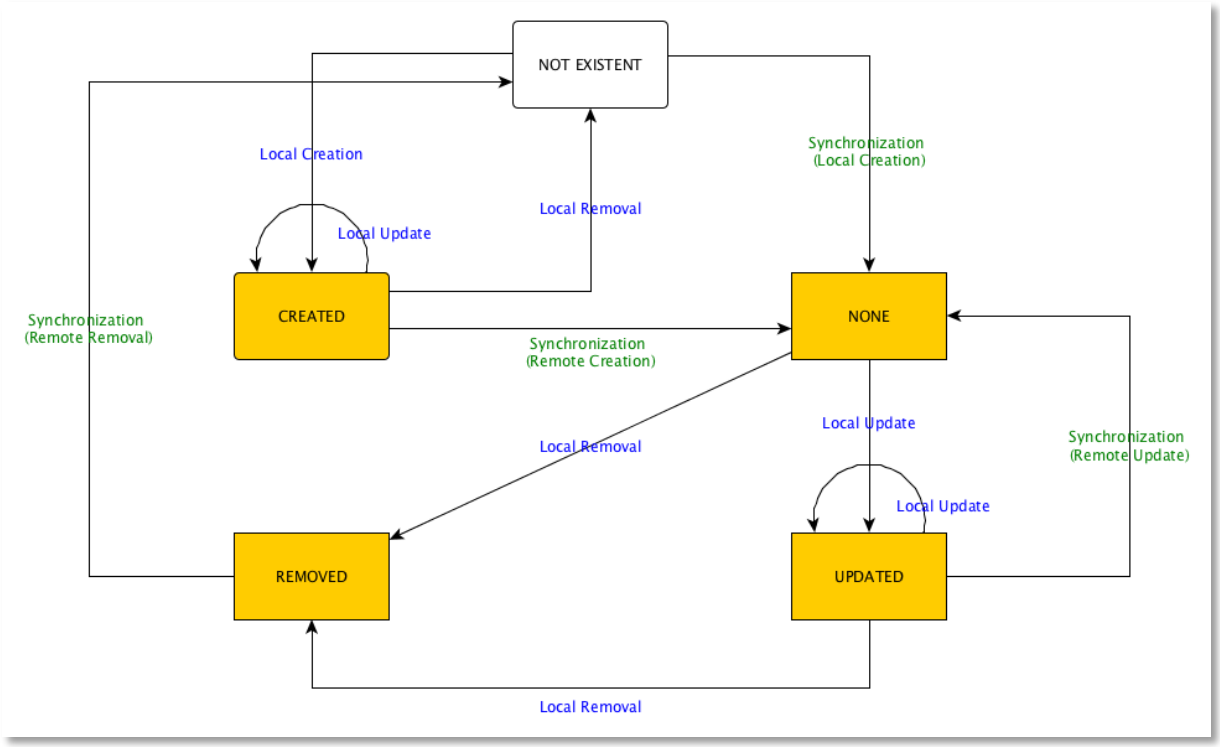


Figure 6 – Synchronization status

At a high level, the entity synchronization process works this way:

1. The **device prepares** a **sync request** containing a description of what happened to customer/prospect-side data since the last synchronization, and sends it to the server.

2. The server processes the sync request, computes a sync plan containing a description of what the device must do in order to get its data synchronized with server data and sends it to the device.
3. The device app processes the sync plan, computes the sync strategy, the ordered list sync tasks at the ends of which local storage and remote storage will be in sync, and executes it.
4. When an entity is removed on both sides, or updated on both sides or removed on one side and updated on the other side there is a conflict which must be somehow managed.

The configurable **conflict resolution strategies** that the algorithm may use are:

- **DEVICE WINS**: This strategy must be set if you want to let the synchronization algorithm automatically resolve conflicts and the device storage is considered more reliable than the server storage.
- **SERVER WINS**: This strategy must be set if you want to let the synchronization algorithm automatically resolve conflicts and the server storage is considered more reliable than the server storage.
- **LATEST WINS**: This strategy must be set if you want to let the synchronization algorithm automatically resolve conflicts and the last updated data, either on the server storage or in device storage is considered the more reliable. PAY ATTENTION that this strategy indirectly relies on device date for the comparison (update timestamp is set on entities in the device storage) and users might change the date to gain priority over the server.
- **FAIL**: This strategy can be set if the conflict is managed programmatically.

For Finantix POS application LATEST WINS strategy will be used.

The synchronization algorithm is not transactional, however, even if an error occurs during the synchronization (e.g. network failure), just repeating the procedure will complete the synchronization safely. – transactional here means that in case if process has been broken changes are not rolledback at the full synchronisation process level, but only to the level of “transactional” blocks.- such approach is performance optimal as it didn’t start full process again in case if some failure occurred.

To obtain the maximum data consistency, the synchronization is executed in a pre-defined order with some entity blocks that are executed as “transactional” blocks – what means that in scope of the data entities from below groups transactionality is maintained and in case if failure occurs when particular record is updated change is rolled-back for this record – below blocks are independent from each other.

- Basic Customer Synchronized Data
- Application configuration and default values including translations
- Products data
- Resource meta data
- Notifications data
- Calendar data
- Extended Customer Synchronized Data for downloaded Customers
- Audit Data

To properly handle the data synchronization requirement necessary for the offline usage and the data providing to Manulife systems requirement, each entity that has to be synchronized and sent to MIL has a special "already sent to Manulife" flag managed by POS server. Flag is specifically dedicated to the records which needs the update in MIL servers via exposed webservices integration points – in case if update in MIL system failed due to the issues to connectivity the flag will not be set and record will remain to be synchronised at the next synchronisation event.

Synchronisation process for MIL will be separated in two stages – blocking and background. Data which is synchronised in each of the stages is specified in FDS, chapter 3.2.2.

Blocking stage will not allow the user to do any operation in POS application, till the time it is completed.

In 2nd, background synchronisation stage, records:

1) Which are supposed to be synchronised as per synchronisation plan cannot be accessed by agent if synchronisation hasn't started yet or is in progress
2) For which synchronisation has been completed or which were not in synchronisation plan (synchronisation not needed for those records) can be accessed by agent

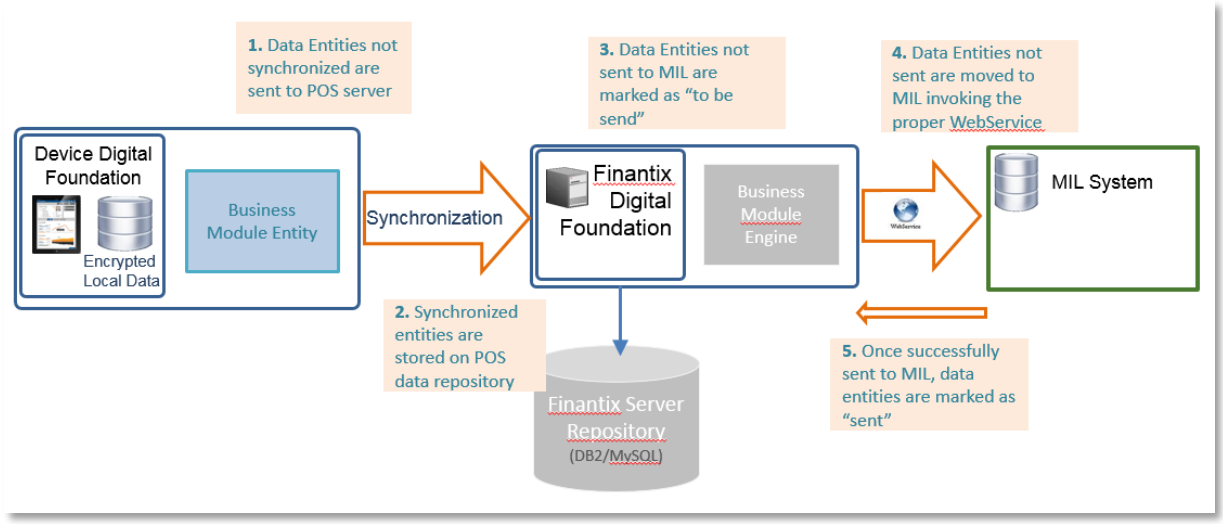


Figure 7 – Synchronization process overview

3.4 Assumptions

1. On iOS devices, apps installation and update requires explicit actions from the user, so that an automatic application update is not part of the solution.
2. Some minor application changes will be designed as "application configuration data changes" and will be managed automatically by the POS application (e.g. calculators default value parameters changes is considered a minor change and can be done automatically).
3. All webservice and batch integrations described in the following chapters will be based on Unicode UTF-8 encoding.
4. Webservice and batch integrations described in the following chapters will consider common interface except country specific integration requirements.
5. All communication strategies described in this document (between POS device/web application and POS server, POS device/web application and Manulife systems, POS server and Manulife systems) will be executed in a secure way (secure protocols: HTTPS, LDAPS, SFTP).
6. All webservice exposed by Finantix are considered to be SOAP XML over HTTP/S.
7. All exposed / consumed webservices will leverage on application specific authentication via SOAP header attributes.- see chapter 10.2.3
8. Finantix POS will use LDAP simple authentication to Active Directory as connection will be in Manulife network infrastructure within LDAP over SSL (LDAPS) secure channel (for 2015 PHASE1)
9. For conflicts, "LATEST WINS" approach will be used as default strategy.
10. The synchronization between POS application on device and POS server should not be available during the batch import processes (see chapter 5.2 for batch import processes details)

11. On the next chapters, physical PDF files generation, from a predefined fixed form layout, is available only when the device works in “online” mode. Generation of PDF files on the offline solution is OUT OF SCOPE for Finantix POS application, only preview functionality is available (please refer to FDS, chapter 21 All Reports)

4. Agents

Different types of agents (e.g. Manulife agents vs 3rd party agency agents, agents with special roles or authorizations, sub-country specific logics) are distinguished in POS application using the "group" concept.

4.1 Agent and roles

For purpose of this document name agent will be used as equivalent of the POS solution user.

4.1.1 Agent login Id

Currently in MIL Agent Id may not be unique across all countries. For 2015 PHASE1 existing Manutouch system login will be used.. As uniqueness is assumed for regional solution, Manulife confirmed naming of the agent login id in form agent@domain (e.g. agent@manulife.com.hk) which will be an unique when integration with ADFS will be implemented (see chapter 10.2.4)

4.1.2 Automatic enrolment and update of agent

As there is no initial pre-population of agent data in POS system, mechanism of automatic enrolment of agent account needs to be introduced.

4.1.2.1 Mobile application

When agent is accessing mobile POS application for the first time, he needs to be connected to the Internet, to let the POS security component check if he is authorised to access the application. In case if access is granted, new account in POS is created with stored secure token for off-line login purposes (expiration of the token off-line access is country specific configuration).

- A device will be assigned to an agent.
- An agent will be assigned to one device.

If it is subsequent access (means user account already exists in POS) :

- when agent is authenticated and is in role to access POS, agent information and roles will be updated in POS
- when agent is authenticated, but not in role to access POS, agent information and roles will be updated in POS and offline access token will be invalidated

For details of authentication/authorisation process please refer to chapter 10.2

4.1.2.2 Web application (OUT OF SCOPE FOR 2015 PHASE1)

For web application there is no need to store token for off-line access. Agent details will be stored in FDF user repository when user accesses application for the first time or updated in when on subsequent accesses.

4.1.3 Agent roles

Agent roles will be mapped from the group membership defined in Active Directory and retrieved by POS application during login sequence (please refer to chapter 10.2.1).

There will be a set of roles available – one for general access to POS application and the others defined by functional specification – for roles description and functionalities assigned please refer to FDS, chapter 25.7 User Roles and Groups

For group names please refer to IMS document, tab “BAD2” and “EAD”.

4.1.4 Agent repository and agent data retrieval – TEMPORARY SOLUTION FOR 2015 PHASE1

As a central agent repository Manulife has chosen Microsoft Active Directory.

It will be interfaced by POS solution via LDAP protocol for agent authentication and basic data set limited to the group membership for agent (as described in chapter 4.1.3) – for mapping details please refer to IMS document, tab “BAD”.

Extended agent information (please refer to chapter 13.1 About Me of FDS document for functional specification and tab “EAD-WS” of IMS document for mapping details) will be retrieved via web service call:

- For mobile application (with Internet connectivity)
  - it will be called when synchronization process will be triggered automatically on agent login
  - it will be called when synchronization process will be triggered manually via “sync” option (please refer to FDS, chapter 3.2.2 Synchronise)
- in web application it will be called after login sequence be completed (OUT OF SCOPE FOR 2015 PHASE1)

Extended agent data to be retrieved via MIL Agent Data WS call (for authentication details please refer to chapter 10.2.3)

Service Name	getExtendedAgentData
Output	*agentCode – used by data feeds to link agent to customer / prospect / policy data* *Channel* 4.1.5 List of user groups *Extended agent data fields – refer to IMS document (e.g. full names, license numbers etc.)*

Agent account management is delegated to Manulife Active Directory – in case if account is locked, Active Directory is not explicitly passing information about the authentication failure reason and indicates only invalid credentials – in such case invalid credentials error will be reported (for full list of login errors reported please refer to the chapter 2.1 Logon in FDS document).

Manulife to ensure that Active Directory server and WS to retrieve additional agent data (getExtendedAgentData) will meet the performance requirement coming from the volumes information related to authentication / authorization (please refer to chapter 14.3 for sizing of the connection of MIL facilities with POS hosting data center).

Finantix will take care of proper sizing of POS environments to ensure that Finantix Digital Foundation will handle number of calls.

Below figure illustrates the infrastructure assumed for design of the solution. Finantix assumes availability of Active Directory services exposed via LDAP protocol over the secure link between MIL and POS hosting data centre (please refer to chapter 14.3.5).

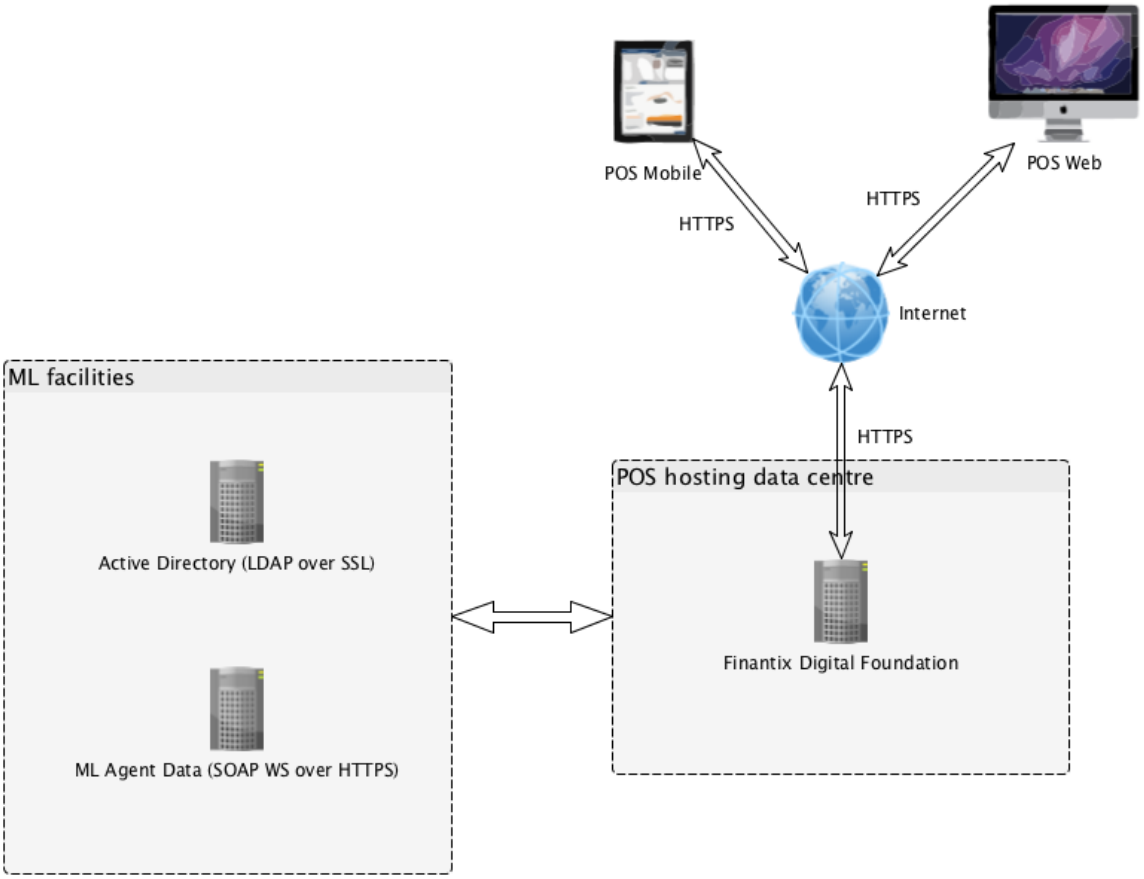
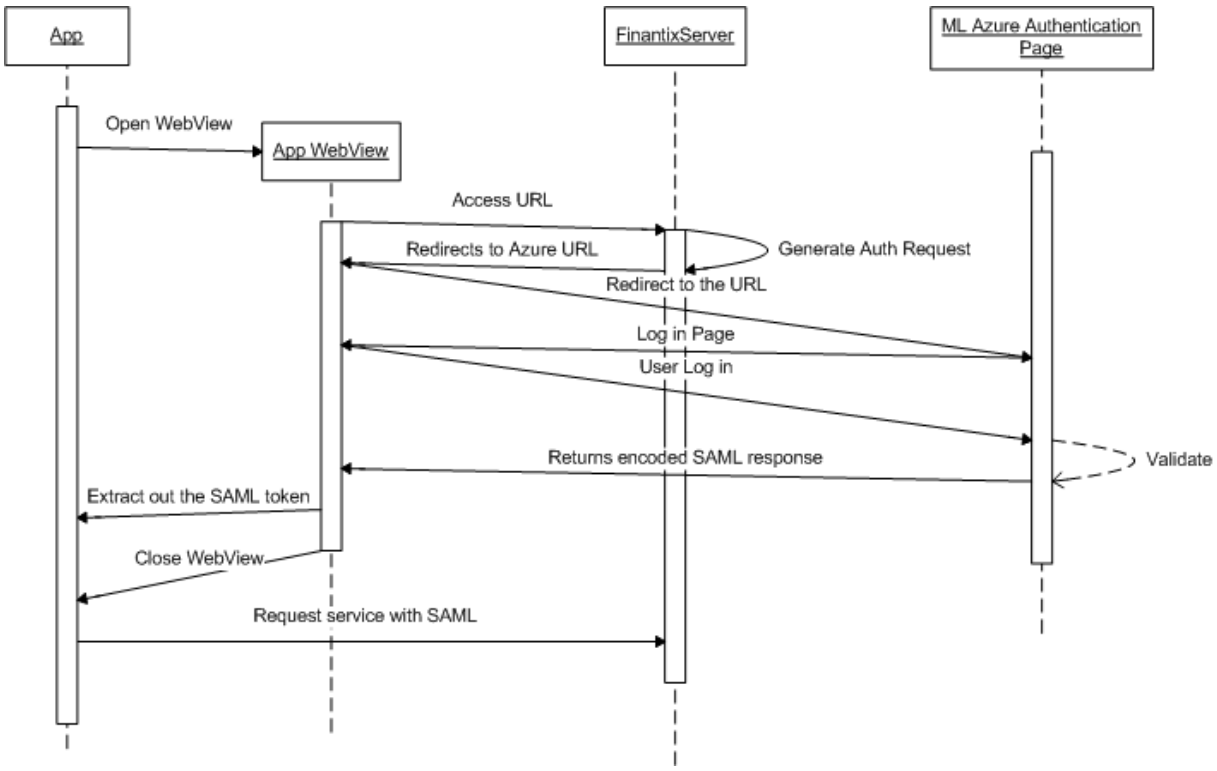


Figure 8 – Finantix connectivity to MIL Active Directory and MIL Agent Data WS.

4.1.6 Azure and E-POS integration (TBC from Finantix)



- E-POS will be integrated with Azure. The authorization process as following.
  - Online
    - Once the user log out from the application, the SAML token should be removed/invalidated.
  - Offline
    - There is no offline password need for the APP. (Confirmed By ML) That is, once the user logged into the iPad that assigned to (s)he, the agent will be able to use EPos without any authentication process.

4.1.7 Agent repository and agent data import / retrieval – OUT OF SCOPE FOR 2015 PHASE1

Specific requirement for Japan agent hierarchy structure to be analysed when Japan analysis phase will commence. Initial concept is to use Finantix User & Management component and batch import for agent data. Solution to be adapted as final regional solution when next country will be rolled-out.

4.2 Agent dashboard

4.2.1 Calendar integration - OUT OF SCOPE FOR 2015 PHASE1

POS schedule component will be integrated with mobile device calendar and Office365 . Requriement will be analysed when analysis of next country requirements will commence.. Any integration with device calendar and external data feeds from Office 365 is OUT OF SCOPE FOR 2015 PHASE1.

4.2.1.1 Mobile application

Integration of schedule component with device calendar will be considered under below agreed assumptions:

1) Manulife will integrate all the potential sources of calendar information with Office365 for POS schedule component usage purposes (e.g. salesforce, Google Calendar, Manutouch etc.).
2) Office365 will be configured as a calendar account in device OS (as part of first time device setup for agent),
3) Manulife may expose few calendars from Office365 – there will be one main agent calendar and auxiliary ones which will contain predefined events to be visualized in schedule component view in read-only mode (e.g. for public holidays) – set of calendars to be agreed,
4) Only main agent calendar will be synchronized in bi-directional mode (changes in POS, device calendar and through POS web interface will be reflected accordingly)

4.2.1.2 Web application

Office365 does not support CalDAV protocol for calendar synchronization. Schedule component available in POS web interface will use the Office365 API to get the information about the events in online mode. Regarding component content, the same approach will be taken as for Mobile application – only one, main agent calendar will provide bi-directional synchronization feature, the others will only be source of data to visualize in schedule component calendar view.

4.2.1.3 Schedule component calendar integration

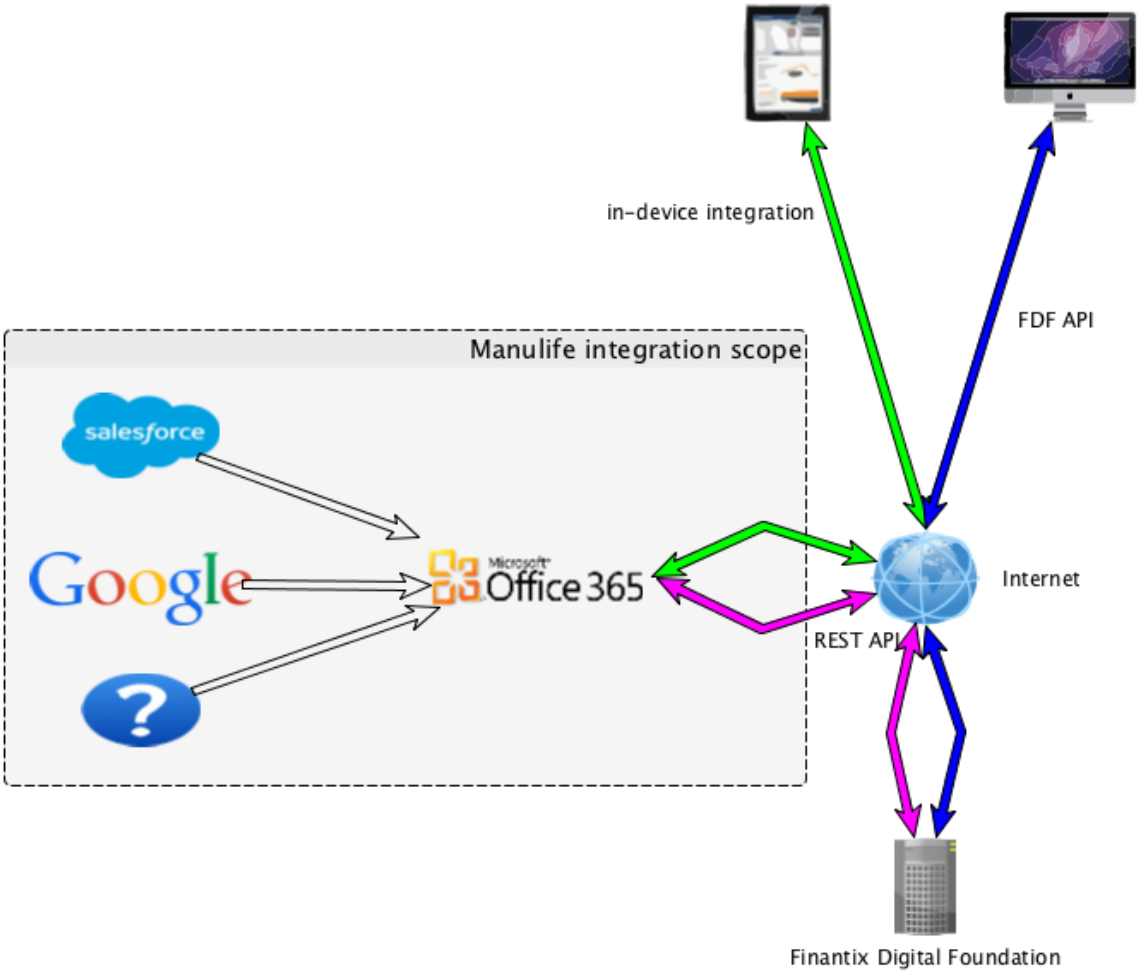


Figure 9 – Overview of calendar integration for POS solution

Description:

- Mobile application

POS application will not be using FDF as an intermediate party in calendar synchronization process as synchronization is performed via Office365 account configured at mobile device calendar level. Mobile POS application connects to device calendar and builds POS calendar view basing on its content

- Web application

Web application calendar widget in browser will synchronize information on-line through FDF server calls to Office365 API (blue path shows interaction between web application and FDF, purple – interaction between FDF and Office365). There will be background SSO to Office365 (10.2.4) Calendar information will not be stored in Finantix server side.

4.2.2 Resources – Finantix Media Engine

Media are managed as resources by the Media Engine component on top of Finantix Digital Foundation.

Each resource can have multiple visibility rules depending on users groups, i.e. different agents can see different resources depending on their groups (feature of Finantix console – please refer to FDS, chapter 25.1 Resource Console) . Roles for access to the console and the functionalities are

described in FDS, chapter 25.7 Roles and User Groups). User management for the resource console is provided as per chapter 10.2.2

This is a high level data model diagram about resources. The diagram highlight also how notifications can be managed as a subtype of resource on POS solution (see next paragraph for details).

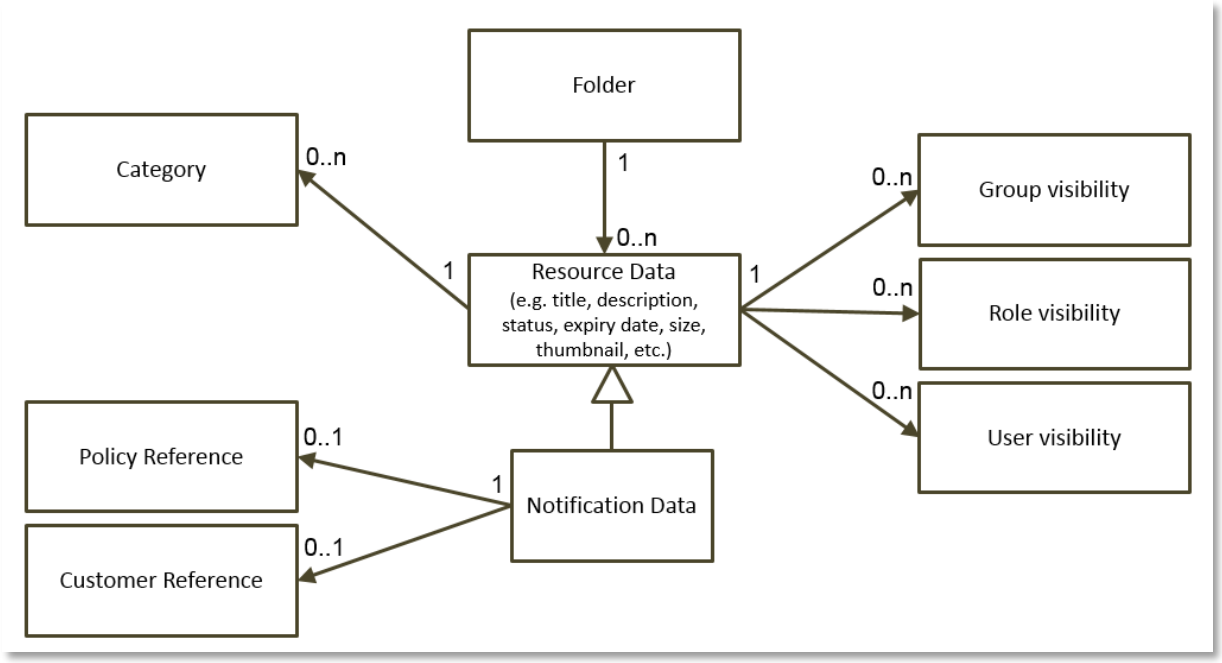


Figure 10 – Resource and notification high level data model

1) Resources and media are synchronized automatically between POS server and POS device application only for the common data part strictly necessary to search: name, short description, size, type, thumbnail, expiry date.
2) Visibility and status information are used to filter the resources “synchronizable” for the user during the synchronization process.
3) The actual file represented by the resource object has to be explicitly downloaded by the user.
4) Once downloaded, a resource is not kept in synch with the server: the user has to manually update it in case of new versions.
5) There is no functional limit in the number of resources that a user can download in her/his device, however a technical limitation depends on the device itself (available storage, computation performance).
6) The expiry date associated to a resource is used by the device application and by the server to automatically change the resource status (from “published” to “expired”), but the resource is not actually removed from the POS server data storage.. Expired resources must be purged manually from the system – for 2015 PHASE1. Solution to remove expired resources from the server will be introduced when next country be rolled-out. In mobile device expired resources will be deleted automatically.
7) Resource categorization can be used to associate the resources to particular areas of the POS application.
8) Categories can be defined in a hierarchical way so that each country will be a category root under which the actual category tree starts (with product families or other categorization on the second level and subcategories on the third level if necessary).

9) “Folders” are used to group resource for a third level of filtering: a folder can contain resources belonging to multiple categories and associated with multiple users’ visibility conditions. There will be three main folders used to separate “product” resources from “general” resources and also “media” resources (product+general) from “notification” resources.

Any migration from the existing resources available in Manulife systems to the POS solution is OUT OF SCOPE for Finantix POS application. Before go live, all media resources have to be uploaded and configured properly in POS application using the resource publishing console.

4.2.3 Notifications

Notifications will be managed from the dedicated console (please see FDS, chapter 25.2 Notification Console) . Access to the console and the functionalities mapped are described in FDS, chapter 25.7 Roles and User Groups).

1) Notifications are considered a subtype of resources given that they have the same constraints of visibility and the same multimedia content.
2) Notifications can be created by an administrator user using the dedicated console (as extension of the Media console). Notifications can also be created with a dedicated webservice described below.
3) When a notification is created, the corresponding data is stored in POS server.
4) A notification must be “approved” by an administrator.
5) Notification data content must be synchronized by the user device application before being available inside the notifications panel of POS application.
6) Only when approved, the notification is actually published to make it available for the next device application synchronization.
7) After the approval, the notification message is sent to Apple Push Notification service (APNs) so that an alert message is directly sent to the users’ devices (also Android push notification system will be supported).
8) The user must manually open POS application and the notification panel (after a proper synchronization) to get the notification details.
9) If a notification links a particular customer or prospect data, when clicked, the customer/prospect profile is opened only if the customer/prospect data has already been downloaded into the device. If not, an error message is prompted to the user.
10) iOS 8 and later, the maximum size allowed for a notification payload is 2 kilobytes and Apple Push Notification service refuses any notification that exceeds this limit. Prior to iOS 8 and in OS X, the maximum payload size was 256 bytes.
11) The number of characters visible may change from device to device. To ensure the maximum compatibility, the size limit for the notification title used as push message should be 256 bytes.
12) A notification must have an expiry date. The maximum allowed expiry period will be defined in a global configuration with a meaningful value (e.g. two months) in order to limit the number of pending notifications.
13) Expired notifications will be purged automatically from POS system to avoid storage memory leakage.

4.2.3.1 Notification creation from external systems – OUT OF SCOPE FOR 2015 PHASE1

1) To create a notification, Manulife has to invoke the following webservice provided by POS. When invoked, the service stores the notification data inside POS notification component.
2) Notifications created using the exposed webservice are automatically set as "published" only if approval flag is provided as "true" otherwise those are created in "draft" status.
3) A data validation check will be executed on the notification title and expiry date. If the size exceeds the message payload limit, the title will be truncated. If the expiry date is longer than the maximum expiry date period configured globally, the expiry date is set to the maximum limit.
4) No data consistency checks will be executed on the linked customer/prospect identifier or on the linked policy identifier.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab CN-WS):

Service Name	createNotification
Input	Country identifier Title Description Expiry date Approval flag (OPTIONAL) Groups (OPTIONAL) Attached image or other media (OPTIONAL) Customer or prospect unique identifier (OPTIONAL) Policy unique identifier
Output	Confirm successful operation

4.2.4 Sales Activities (Japan specific) – OUT OF SCOPE FOR 2015 PHASE1

1) For users belonging to "Japan" country, POS device/web application displays a new component on agent dashboard called "Sales Activities".
2) The content of this panel is provided by Manulife (Japan) system by exposing a dedicated webservice. The service will be called by POS server when the user executes the device data synchronization (or when the user refresh the dashboard page on the web solution).
3) The service must return the list of Sales activity to show in the exact order in which they should appear on the panel. No pagination strategy is designed for this data retrieve, that is, all records that have to be shown must be returned by the service.
4) Each Sales Activity returned by the service will contain the list of fields required by the such as: title, short description, start date, due date, reference customer/prospect identifier (if necessary), reference policy identifier (if necessary) – requirement to be confirmed when Japan analysis will commence.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab SA-WS):

Service Name	getSalesActivities
Input	Country identifier Agent identifier
Output	List of Sales Activity entities

5. Customers and prospects

Customers and prospects are managed almost in the same way by POS solution:

1) They both represent an entity to which MIL could sell or has sold a policy.
2) They are both related to agents with a strict data segregation rule.
3) They can both used by POS application to collect needs and information necessary to the sale process.
4) Prospects can be created by the POS solution as well as by other Manulife applications.
5) Customers can be created only by Manulife backend system after a proper enrolment of a prospect to a customer throughout a possible de-duplication process when necessary.
6) Prospects created by POS solution have to be provided to MIL backend systems for further processing.
7) Prospects created by other MIL systems and customers created by MIL enrolment procedure have to be imported into POS solution to make them available to agents.
8) The synchronization strategy between POS device application and POS server has been described in the solution overview.
9) During the synchronization process all customer/prospect basic data will be synchronized for all customers/prospects that belong to the agent doing the synchronization. It means that about ten data fields (see further on this chapter) will be always synchronized (if necessary) from server to device and vice-versa.
10) The extended data set of a customer or prospect is downloaded on demand by the user.
11) Data of already downloaded customers/prospects is synchronized automatically if necessary.
12) There is no functional limitation to the number of customers/prospects that an agent can download on her/his device. The technical limitation depends on the single device storage (available device memory storage for this application) and computation performance (faster devices will have acceptable search/retrieve performance also with a big number of entities available on the device, while old/slower devices might face performance problems with a too high customers/prospects number).

5.1 Customer and prospect information record contents

Prospects will be created by Finantix POS solution and associated to the Agent that created them.

For 2015 PHASE1 only the agent that created the prospect can search and display it. A direct link between Prospect and Agent will exist in the data model (using agent's Finantix unique identifier). This approach may be extended for later phases of project to cover all prospect data sharing requirements by Manulife Japan.

When an application for the prospect is accepted by Manulife, a new policy is created in Manulife systems, the prospect become a customer and the new customer record is provided to POS on the next data feed. In order to let POS server to match properly the prospect data stored on POS storage and transform it to a customer, a proper communication protocol has to take place – for scenarios supported please refer to chapter 5.1

1. when prospect is created from Finantix POS application, dedicated webservice exposed by MIL system is called (please refer to "createProspect" webservice in chapter 5.4.2) The invoked service returns to POS application the (country) unique Manulife Prospect Id of newly created prospect record.
2. Next customer data feed from Manulife to Finantix, will provide the original MIL prospect identifier previously returned by webservice to POS application to correctly link the data.
3. In case if multiple prospect records has been merged into single customer records MIL will provide the list of original Manulife Prospect id's.

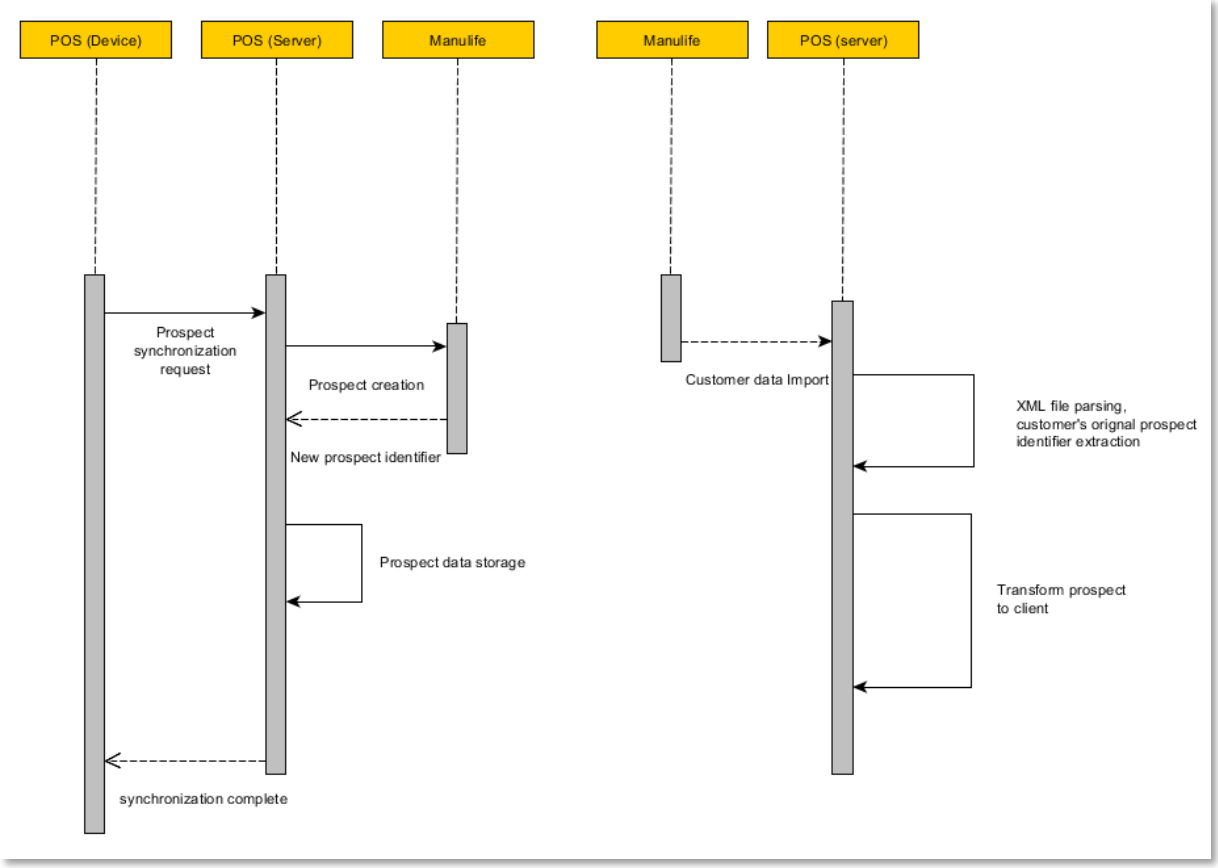


Figure 11 – Prospect to customer transformation – Calls sequence

Another common attribute of a customer/prospect is the “lastAccessDate” managed by the device in order to enable the automatic purging of the offline customer/prospect data (please refer to FDS, definition “Downloaded (in the context of Customer)”, chapter 1.5 Definitions) if it has not been accessed in a given number of days (internal regulation) – access will be assumed as opening of customer/prospect profile screen (please refer to FDS, chapter 16.1 Profile)

5.1.1 Basic/extended profile

Customer/prospect data is organized, so that a basic data set available for both prospect and customer can be easily synchronized between POS device app and server to allow the search feature On top of this data set, the application defines some extended data sets, with some differences for customers and prospects specific of each part of POS application. The FDS document specifies in details Basic and Extended data sets (please refer to FDS, Basic Customer Synchronised Data and Extended Customer Synchronised Data in chapter 1.5 Definitions).

In order to preserve the unique regional solution, in POS application will use a superset of customer/prospect extended data set which will be available to all countries (2015 PHASE1 set is limited to HK and will be extended in when subsequent country requirements will be collected). Assumption is that fields not necessary in a country can be left empty on the integration points defined in this document.

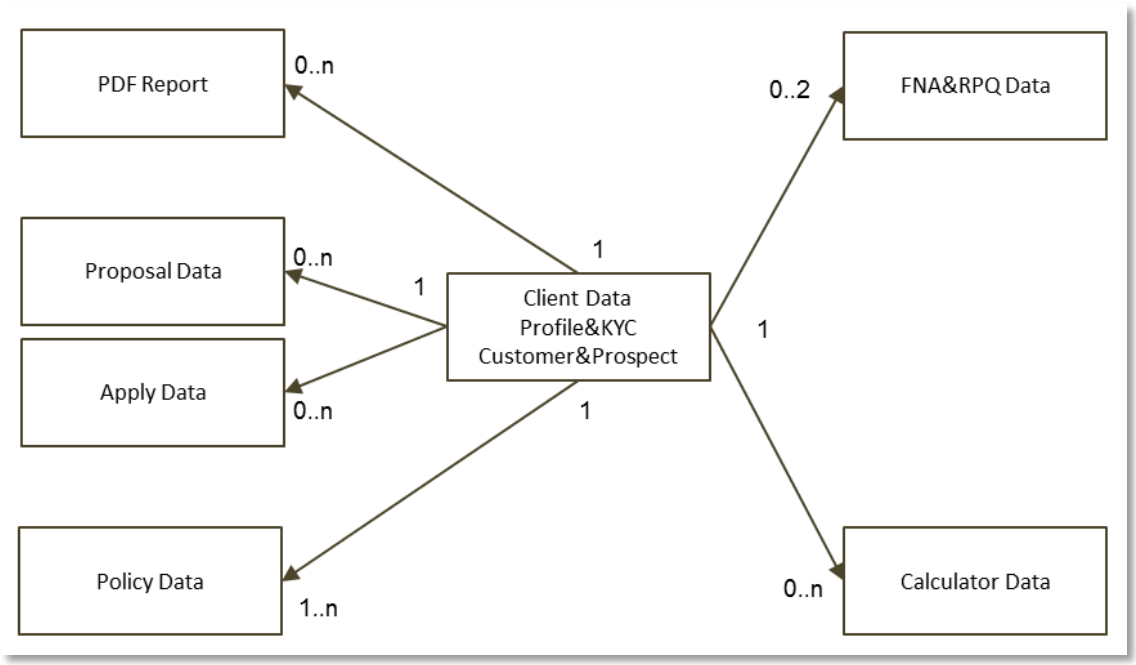


Figure 12 –High level customer/prospect data model

Customer/prospect data, automatically synchronised by POS application are defined in FDS, chapter 1.5 Definitions as “Basic Customer Synchronized Data”

Customer/prospect data, synchronised by POS application via “download” function are defined in FDS, chapter 1.5 Definitions as “Extended Customer Synchronised Data”

5.1.2 Individual and corporate profile – it is not a part of scope of 2015 PHASE1

Customers and prospects can be individuals or corporates. Please refer to the FDS document for a detailed list of fields and the distinction between individual and corporates. The following SAS documents will not make any distinction between the two.

5.2 Batch import processes in Finantix POS application

Information collected in this chapter applies to all the processes described in chapters 5.3, 5.7

For batch import processes in Finantix POS application, standard Finantix import facility will be used. An overview of Finantix standard import facility is described in the Appendix A.

All basic and extended data sets belonging to the same customer or prospect provided by Manulife to POS solution must be contained in a single XML file (except for policies).

The single record stored as XML entry in the file can be read, processed and stored into POS data storage in a unique transaction as explained on previous introduction.

The three types of files that POS solution has to process are:

- Prospects
- Customers
- Customers' policies

Any other customer/prospect data not contained in the above files is provided separately from the main customer/prospect data set (e.g. proposals list) and is loaded on demand by consuming a dedicated webservice provided by Manulife backend (see chapter 5.6.1) – solution for 2015 PHASE1 only.

More than one file can be submitted by each country for each entity to load. The files naming convention extends the one defined in the appendix as standard Finantix import naming convention. In particular, to enable the multi-country import, each file must start with a prefix identifying the country:

<country code>_<type prefix>_<yyyyMMdd>_<HHmmssSSS>.xml

e.g HK_PROSPECT_20150531_010004624.xml

Information about country for which records are imported will be provided as an attribute of root element of XML file to be imported.

Below points apply to the import processes:

1) For time windows defined for import process please refer to chapter 2.1. The exact time window necessary to import all entities will be estimated when exact volumes and data size will be provided.
2) All data to be imported must be provided in time by Manulife systems on a per country base (each country will follow a dedicated import process).

3) If the import files are not available in time at the beginning of the time window, the import process does not take place and only “day -2” data will be available to users for the synchronization.

4) Some countries might decide to start the import later, but in this case the import procedure might take place during the morning when users already started to use the POS application installed on their devices. In this case, to ensure the maximum data consistency and server performance, the users will not be allowed to synchronize their devices during the entire import process.

5) For parallelisation and transactionality reasons Finantix application is splitting XML files to be imported into “chunks” of records (e.g. 100) which are treated as a single bulk database transaction which may be rolled back in case if failure happens (e.g. connection timed out)

6) If there are errors at the record level (to be included in “chunk”) those records will be marked failed but transaction will be executed

7) Assuming the sequence of import: policy, then customer, in case if any of the import processes for the same customer record has failed / hasn’t happened for some reason, agent would see (assumed that policy / customer data where available in the system before import was executed):

   - If policy record import has failed, but customer record import succeeded – agent will see updated customer profile and previous policy data
   - If policy record has been imported but customer record import failed – agent will see previous customer profile and updated policy data

8) Manulife will provide the daily “delta” for customers and prospects, that is, only the customers and prospects actually created or modified during the day before the import.

5.2.1 Import process schedule, volumes and performance requirements

5.2.1.1 Initial import

There will be an initial import executed once only to migrate all existing MIL customer/prospect/policies to Finantix POS solution. This import might exceed defined time window as of the initial volume of customers/prospects/policies data to migrate. Succeeding imports will be executed on a daily base with the “delta” of updated/created/delete customers/prospects/policies only.

The exact time window required for the initial import will be defined during the UAT tests with exact Manulife volumes for all countries.

5.2.1.2 Delta import

The “delta” import will be executed daily. Please refer to chapter 2.1 for import volumes and time windows agreed.

The exact time window required for the delta import will be defined during the UAT tests with exact Manulife volumes for all countries.

5.3 Customer and prospect data import

This paragraph is focused on the basic customer/prospect profile data and on the extended customer/prospect profile data import. For details about policies data, please refer to next paragraph (“Policy information data import”).

5.3.1 Customer and prospect batch import (XML)

Import process:

1) The import process for customer/prospect data starts every day at a predefined time that can be configured by the application administrator for each country.
2) Each country will have a dedicated import process that proceeds independently from the other countries,
3) Manulife has to be sure that all files to be imported are available on the shared folder configured as upload folder before the process is scheduled to start.
4) If the files are not available when the import process starts, that import step will be skipped and no data will be imported until next scheduled import (or until a manually triggered import).
5) The import process defines all steps necessary to properly fill the POS data storage with the latest data coming from Manulife systems.
6) The process will execute the import steps in this sequence (considering the scenario of not blocking the synchronization if the available time window is exceeded):
    a) Prospects: in this way new prospects or updated prospects are immediately available to agents for their daily appointments.
    b) Customers’ policy data: Policies are related to customers with a weak link to the customer identifier. Policy data will be imported into POS storage and “parked” there waiting for the corresponding customer if not yet existing on POS system. If an agent synchronizes her/his device at this stage or do a search at this stage, the correct policy data will be returned for existing customers, and new customers will not be returned at all because not imported yet.
    c) Customers: updated customers’ data and new customers are imported as last step of the import process to guarantee the maximum possible data consistency in case of a synchronization during the import execution.

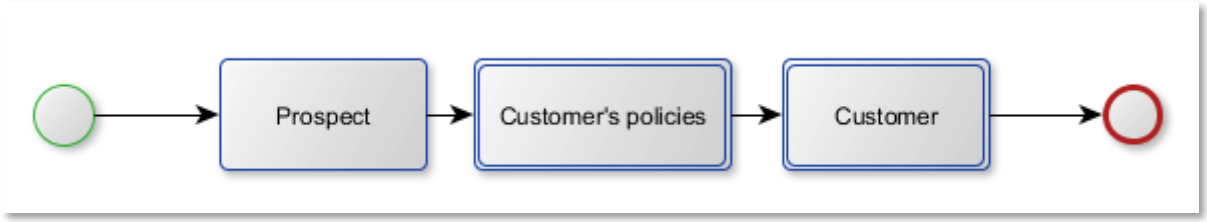


Figure 13 – Customers/prospects import process steps

5.3.1.1 XML file structure and data mapping

For high-level illustration of below points please refer to the Figure 14 (for prospect), Figure 15 (for customer) depicting conceptual, sample XML file structure – all the names of fields referred in below chapter are reflected in the figures – for actual structure, data mapping and field names please refer to IMS document (tabs PROI-Batch and CUSI-Batch).

1) For both prospects and customers, the XML file has to be structured so that there is a single node for each customer/prospect and this node is the first level after the file root. The main attribute of the node is the unique identifier of prospect / customer record (mlProspectId, mlCustomerId)
2) If import files have to be split to proceed for a parallel import of chunks, the splitting will be done on the first node level that is the customer/prospect.
3) Each prospect node must contain the unique identifier (referenceAgentCode) of the agent who "owns" that prospect, so that POS solution can properly associate the prospect to the agent.
4) Customers → agents association is derived from the policies that the customer subscribed (for policy data import see chapter 5.7)
5) Under the first level attributes, the customer node structure must contain the profile common data node (commonData) and then the list of nodes with data belonging to a specific agent (customerProspecting for both prospect/customer).
6) The commonData node contains common information such as the customer/prospect name, phone number, age, etc. (please refer to IMS, Tab CUSTI-Batch, PROI-Batch)
7) Customer/prospect prospecting data nodes:
   - For prospect data export - will not contain FNA, RPQ, KYC and Calculators– there are separate export processes for those data sets (please refer to chapter 5.5.3)
   - For prospect/customer data import – for 2015 PHASE1 will not contain FNA, RPQ, KYC and Calculators data – for Japan requirement of KYC data import, separate KYC import process will be evaluated when analysis will commence – initial approach would be, to follow interface and data mapping of already defined KYC export process (please refer to chapter 5.5.3)
   - For customer import
     - will be used for conversion from prospect to customer (see chapter 5.8.5for sample conversion)
     - will be used for customer prospecting data deletion (customer will not be visible to agent identified by referenceAgentNode and data associated with that agent will be deleted) – see chapter 5.8.9 for sample deletion.
8) Information about country will be provided as an attribute of root element of prospect/customer import file

5.3.1.2 Customer/prospect actions

Actions will be provided via attribute at the customer / prospect record level in XML import file to let POS Finantix system determine the operation to be performed.

5.3.1.2.1 Prospect record actions

1) DELETE – will delete prospect record from Finantix POS system including all the data associated with that record like KYC, RPQ, FNA, calculators, etc. ; for DELETE action the only parameter which is mandatory is mlProspectId

2) UPDATE – will create (if doesn't exist) or update existing prospect record in Finantix POS system

5.3.1.2.2 Customer record actions

1) "DELETE" – will indicate that customer prospecting data for agents specified in customer prospecting data nodes (via referenceAgentCode) needs to be deleted from POS application and customer needs to be made unavailable for search for this agent – the only parameters needed for deletion are mlCustomerId and referenceAgentCode in customer prospecting data sections.
2) "UPDATE" – will create (if doesn't exist) or update existing customer data record in Finantix POS system
    a. As of HKCR021, a new attribute 'status' controls the visibility of Customers imported via data feed. Possible values for the Customer 'status' attribute are defined in IMS document. Statuses are split between active and inactive. Customers in an active status are searchable and sychronized. Those in an inactive status are deleted from the device, kept on server and only synchronized once they eventually return to an active status. Refer to IMS for list of statuses.
3) "CONVERT" – will convert prospect to customer, customer prospecting data section will contain Manulife Prospect Id of prospects to be converted to the customer (identified by Manulife Customer Id)

```
<prospectImportExport country="HK">
<records>
    <prospect action="UPDATE|DELETE">
     <mlProspectId>MLHK635111111</mlProspectId>
    <commonData>
        <lastName>Gladstone</lastName>
        <firstName>John</firstName>
        <birthDate>1974-02-19+00:00</birthDate>
     <referenceAgentcode>111222333</referenceAgentcode>
    </commonData>
    ...
    <prospectProspectingData>

        <maritalStatusId refData="SINGLE"/>
        ...
    </prospectProspectingData>
     </prospect>
     <prospect>
        ...
     </prospect>
    ...
 </records>
</prospectImportExport>
```

Figure 14 – Prospect XML file contents sample (for full specification please check IMS document, chapter PROI-Batch)

```
<CustomerImportExport country=”HK”>
 <records>
       <customer action=”UPDATE|CONVERT|DELETE”>


      <mlCustomerId>MLHK635111111</mlCustomerId>
      <commonData>
            <lastName>Gladstone</lastName>
            <firstName>John</firstName>
            <birthDate>1974-02-19+00:00</birthDate>
            ...
      </commonData>
       ...
       <customerProspectingData>
            <referenceAgentcode>111222333</referenceAgentcode>
            <mlProspectID>MLHK3500000</mlProspectID>
            <maritalStatusId refData="SINGLE"/>
            ...
       </customerProspectingData>
       <customerProspectingData>
            <referenceAgentcode>111222334</referenceAgentcode>
            <mlProspectID>MLHK3500001</MLProspectID>
            <maritalStatusId refData="MARRIED"/>
            ...
       </customerProspectingData>
        <customer>
            ...
        </customer>
 </records>
</CustomerImportExport>
```

Figure 15 – Customer XML file contents sample (for full specification please check IMS document, chapter CUSI-Batch)

Please, refer to the IMS document, tab “PROI-batch” (prospect) and “CUSI-batch” (customer) for details about actual data mapping.

5.4 Update of data entities in Manulife systems

In order to provide to Manulife all data created or updated using POS solution, POS server application during the device synchronization needs to call services exposed by Manulife. This paragraph lists these services.

Manulife has to guarantee a reasonable response time for every service call in order to achieve the requested overall maximum response time for the synchronization operation.

5.4.1 WS endpoints URLs

Fully qualified URL's for accessing services exposed by Manulife to be consumed by POS application will conform below format

https://api.asia-manulife.com/<scope>/<businesscontext>/<applicationcontext>/servicename

where:

scope – scope of organization division, country, e.g. "hk"

business context – business domain that owns the service, e.g. "glh"

application context – a specific application name. e.g. "cws" – customer website

The "Service Name" used in SAS document in chapters defining webservices will refer to <service name> URL part. Manulife need to provide full endpoint URLs to be used for all the countries.

Assumption is that endpoints exposed by ManuLife will be visible to the Finantix hosting provider and can be reached using secure communication channel which will be setup as a part of hosting solution and no additional configuration and infrastructure setup will be needed.

5.4.2 Prospect creation

When a new prospect is created in the POS application, this service exposed by Manulife systems is called in order to retrieve the new Manulife prospect identifier (unique by country):

1) The input provided to Manulife are the minimal set of data to properly create a prospect record on Manulife systems and allow the proper identification and data deduplication (details defined in IMS document in tab "CP-WS"),
2) The service returns synchronously to the prospect creation logic the new identifier stored by POS server together with the other prospect information.
3) The creation logic can be triggered directly by the web application (OUT OF SCOPE FOR 2015 PHASE1) or indirectly by the device-service synchronization logic (in this second case, the actual prospect creation on MIL systems is triggered only when the user synchronizes his/her device).
4) The returned identifier will be used by the reconciliation process executed during the customer data import.
5) The actual response time of this service exposed by Manulife will impact the overall prospect creation page response time for the web solution and the data synchronization performance for the device solution.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab "CP-WS"):

Service Name	createProspect
Input	Finantix unique identifier Prospect profile data
Output	Manulife prospect identifier

5.4.3 Prospect profile update

When the basic prospect data is updated in the POS application, this service exposed by Manulife systems is called in order to update accordingly the same data on Manulife systems.

1) The input provided to Manulife are the minimal set of data to properly update a prospect record on Manulife systems and allow the proper identification and data deduplication (details defined in IMS document in tab “UP-WS”)
2) The service returns synchronously to the prospect creation logic the prospect identifier that will be updated by POS server.
3) The update logic can be triggered directly by the web application (OUT OF SCOPE FOR 2015 PHASE1) or indirectly by the device-service synchronization logic (in this second case, the actual prospect update on MIL systems is triggered only when the user synchronizes his/her device).
4) The actual response time of this service exposed by Manulife will impact the prospect update page response time for the web solution and the data synchronization performance for the device solution.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “UP-WS”):

Service Name	updateProspect
Input	Manulife prospect identifier Finantix prospect identifier Prospect profile data
Output	Confirm of successful operation

5.4.4 Customer/prospect PDF report transfer to MIL (e-Delivery) and printing solution

5.4.4.1 e-Delivery / printing solution for 2015 PHASE1

This chapter covers e-Delivery / printing of PDF reports generated by Finantix POS application – for 2015 PHASE1 there is only one PDF report to be printed / delivered to customer website from Finantix POS application (please refer to FDS, chapter 21.2 PDF Reports).

Printing of proposals PDF is handled outside Finantix POS via integration point with MIL Comprop system (see FDS chapter 26 for details).

Process of PDF print / e-Delivery is leveraging on two webservices:

1) preparePDFDelivery (exposed by MIL) – will be used to pass information about the PDF to be printed / delivered to customer:
   a) for e-Delivery Finantix POS will pass customer email, action as EDELIVERY, OTP (password entered by customer) and PDF report Id later to be used by MIL to retrieve PDF,
   b) for “All Reports” PDF printing Finantix POS will pass action as MOBILE_PRINT and PDF report Id later to be used by MIL to retrieve PDF

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “PPDF-WS”):

Service Name	preparePDFDelivery()
Input	(OPTIONAL) Customer email (OPTIONAL) OTP Action (e.g. EDELIVERY, MOBILE_PRINT) (OPTIONAL) PDF report Id (to be used in MIL callback to retrievePDFReport)
Output	Success/Failure

2) retrievePDFReport (exposed by Finanitx POS) – used by MIL to retrieve PDF report for printing / e-Delivery

For volumes of calls and webservices calls peak hours please refer to chapter 2.1.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “RPDF-WS”):

Service Name	retrievePDFReport()
Input	Finantix POS PDF report Id
Output	PDF file generated / Failure

Process for mobile application is depicted in diagram below (it doesn’t cover the MIL side e-Delivery / mobile printing):

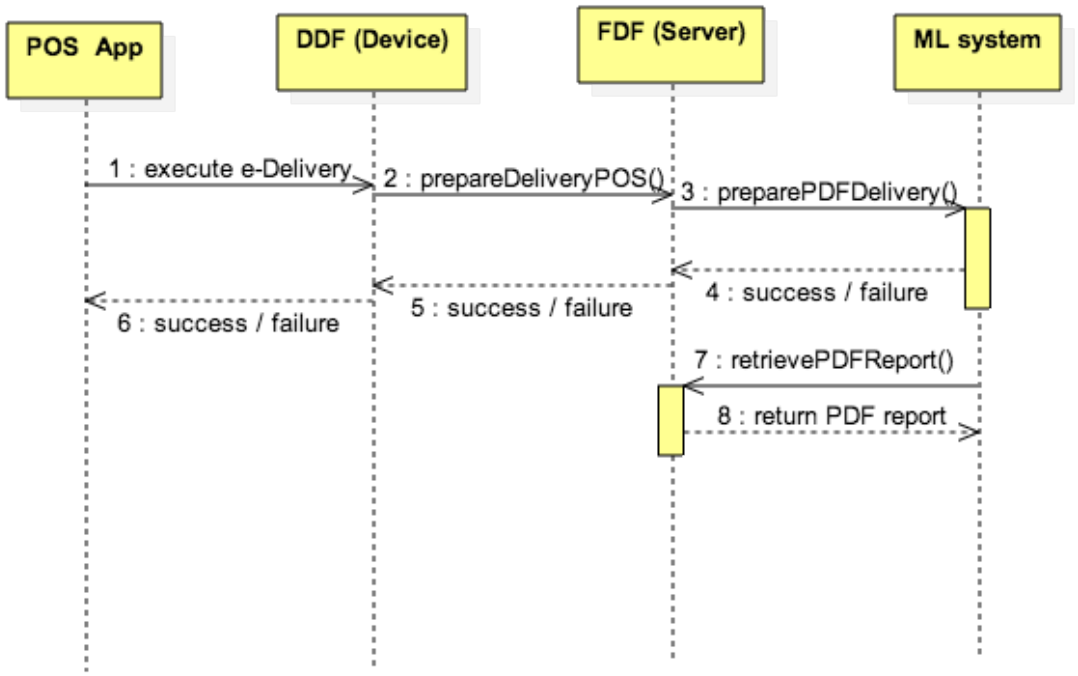


Figure 16 – e-Delivery / mobile printing process steps (mobile application)

Description:

1. E-delivery or mobile printing process has been executed from device (application has access to the Internet),
2. device (DDF) calls Finantix POS server side (FDF) to trigger preparation of e-Delivery in MIL system
3. Finantix POS server side (FDF) calls MIL webservice (see point 1) above for scenarios)
4. Propagation of success / failure response to Finantix POS server side (FDF) – if failure, error will be logged in server side log
5. Propagation of success / failure response to device (DDF)
6. In case of failure received error message will be displayed in device
7. MIL calls Finantix POS service to retrieve PDF report file
8. Finantix POS service returns PDF file or reports error if PDF retrieval failed

Process for web application (OUT OF SCOPE FOR 2015 PHASE1) is depicted in diagram below (it doesn't cover the MIL e-Delivery part):

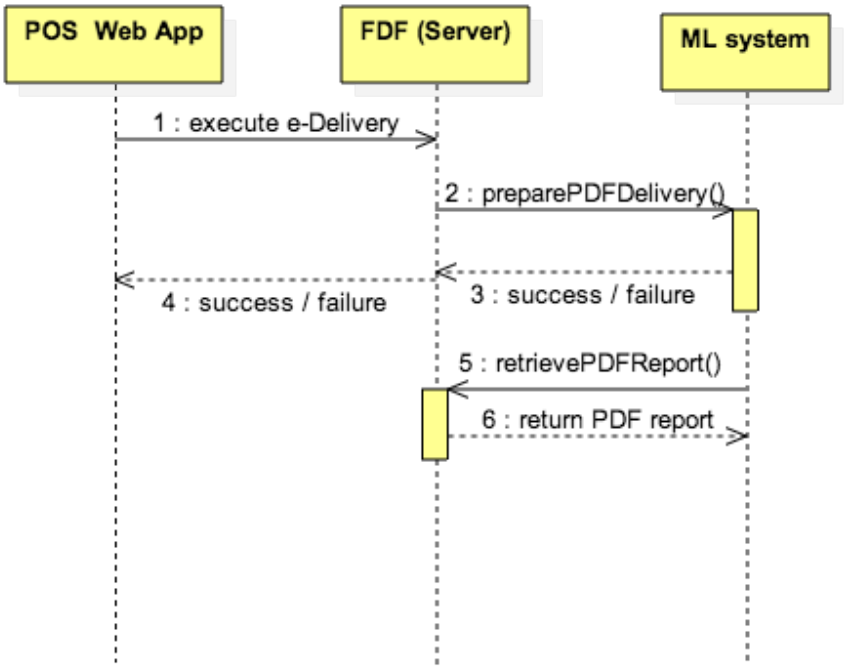


Figure 17 – e-Delivery / mobile printing process steps (web application in browser)

Description:

1. E-delivery or mobile printing process has been executed from device (it doesn't cover the MIL side e-Delivery / mobile printing),
2. Finantix POS server side (FDF) calls MIL webservice (see point 1) above for scenarios)
3. Propagation of success / failure response to Finantix POS server side (FDF) – if failure, error will be logged in server side log
4. In case of failure, received error message will be displayed in browser by POS web application
5. MIL calls Finantix POS service to retrieve PDF report file,

6. Finantix POS service returns PDF file or reports error if PDF retrieval failed

5.4.4.2 E-Consent PDF storage and transfer to MIL

When e-Consent (please refer to FDS, chapter 24 e-Consent) will be signed Finantix POS will generate PDF and store it in its server file system under the name of form:

econsent_[agent_code]_[prospect_id]_[date]_[time].pdf

e.g.: econsent_300693_4588727_280414-11_36_04.pdf

eConsent PDF files will be fetched by MIL on daily basis via SFTP server exposed by Finantix.

e-Consent PDF files will be kept in Finantix POS server file system for 7 days. Files older than 7 days will be purged.

5.4.4.3 End-to-end PDF documents e-Delivery service – OUT OF SCOPE FOR 2015 PHASE1

MIL wants Finantix to be a provider of regional end-to-end PDF documents e-Delivery service. Analysis will be executed when next country analysis phase will commence.

5.4.4.4 Opt Out Signal for prospects

When the agent accesses the Opt Out panel of POS APP (see FDS – Opt Out signal), and the APP is in online mode, the Finantix backend will be asked for an update of the email Opt Out 'signal' via a REST service call. This call will be processed by the Finantix server calling a dedicated WS, the Check Unsubscribe Email WS, provided by ML's backend systems.

1) The REST service call, as well as the WS service call, are synchronous. This means that the Opt Out panel will wait for REST service call's result.
2) When APP is in offline mode, no REST service call will take place, and the last available Opt Out signal will be displayed.
3) The actual response time of this service exposed by Manulife will impact the Opt Out page response time, when APP is in online mode.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab "CUEM-WS"):

Service Name	checkUnsubscribeEmail
Input	Email(s) to check (list of)
Output	List of email(s) checked, along with their Opt Out 'signal'

5.4.5 PDF printing solution

PDF documents can be printed in two ways:

1) via mobile device registered printer (e.g. AirPrint technology for iOS) – standard Finantix printing method – for “All Reports” (see FDS, chapter PDF Reports 21.2) and “Proposals” (see FDS, chapter 26)
2) via dedicated mobile printing solution available in MIL via SOAP webservices:
    a. for “All Reports” PDF approach described above in chapter 5.4.4.1
    b. for “Proposals” dedicated service exposed by MIL will be called by Finantix POS (please refer to chapter 5.6.1.4) – INTERIM SOLUTION FOR 2015 PHASE1

5.5 Prospect data export

1) POS application will export daily to Manulife only the extended data set of prospects (basic prospect data is sent with already defined web service).
2) The exact time window necessary to export all entities will be estimated when exact volumes and data size will be provided (please refer to chapter 2.1)
3) All exported data will be provided to Manulife on a per country base (each country will follow a dedicated export process).

5.5.1 Prospect batch export (XML)

All basic and extended data sets belonging to the same prospect exported to Manulife by POS solution will be contained in a single XML file.

More than one file can be exported for each country for each entity to export. Export files naming convention will follow below format:

<country code>_<type prefix>_<yyyyMMdd>_<HHmmssSSS>.xml

e.g. HK_PROSPECT_20150531_010004624.xml

The prefix can be used by Manulife to correctly associate the entities inside the file with the correct country given that the customer/prospect identifier is not unique across countries. Besides that country will be specified as an attribute of root element of export files.

Export process:

Each country will have a dedicated export process that proceeds independently from the other countries.

5.5.1.1 XML file structure and data mapping

XML file has the structure already described in chapter 5.3.1.1

Please, refer to the IMS document, tab “CL-BE” for details about actual data mapping.

5.5.1.2 Export process schedule, volumes and performance requirements

5.5.1.3 Delta export

1) The “delta” export will be executed daily.
2) The export process for customer/prospect data starts every day at a predefined time configured by POS application administrator – please refer to chapter 2.1
3) The expected number of records exported every day depends on volume of usage of POS application, not identified – for approximate volumes please refer to chapter 2.1
4) Exported files will be kept in Finantix POS server file system for 7 days. Files older than 7 days will be purged.

5.5.2 KYC/FNA/RPQ/Calculators WS

Finantix POS will expose SOAP webservices to get the KYC (see FDS, chapter 9 Getting To Know You), FNA (see FDS, chapter 12.1 Financial Needs Analysis), RPQ (see FDS, chapter 12.2 Risk Profile Questionnaire) and Calculators (see FDS, chapter 11 Calculators).

For webservices authentication details and agent identity propagation please refer to chapter 10.2.3

Webservices will be used by below MIL systems:

a) Comprop – all webservices will be used
b) Manutouch – only getKYCData webservice will be used

For volumes of calls and webservices calls peak hours please refer to chapter 2.1.

5.5.2.1 getKYCData WS

Finantix will expose webservice for Manulife to fetch the KYC information (see FDS, chapter 9 Getting To Know You) collected in POS application. Webservice will return KYC information for prospect/customer collected by specified agent (see chapter 10.2.3 for details how agent identity is handled).

This is high level service signature (for details of the webservice interface please refer to IMS document, tab KYC-BE):

Service Name	getKYCData
Input	Country identifier Manulife customer/prospect Identifier Agent Code
Output	General information Financial Information Your life stage Last updated DateTime

5.5.2.2 getFNAData WS

Finantix will expose webservice for Manulife to fetch the FNA information (see FDS, chapter 12.1 Financial Needs Analysis) collected in POS application. Webservice will return the latest FNA questionnaire question id's and answers collected by specified agent (see chapter 10.2.3 for details how agent identity is handled).

This is high level service signature (for details of the webservice interface please refer to IMS document, tab FNA-BE):

Service Name	getFNAData
Input	Country identifier Manulife customer/prospect Identifier Agent Code
Output	Personal information FNA questionnaire – list of question identifiers and answers FNA questionnaire version Last updated DateTime

5.5.2.3 getRPQData WS

Finantix will expose webservice for Manulife to fetch the RPQ information (see FDS, chapter 12.2 Risk Profile Questionnaire) collected in POS application. Webservice will return the latest RPQ questionnaire questions and answers together with the risk profile scoring collected by specified agent (see chapter 10.2.3 for details how agent identity is handled).

This is high level service signature (for details of the webservice interface please refer to IMS document, tab RPQ-BE):

Service Name	**getRPQData**
Input	Country identifier Manulife customer/prospect Identifier Agent Code
Output	RPQ questionnaire – list of question identifiers and answers Risk Profile score RPQ questionnaire version Last updated DateTime

5.5.2.4 getCalculatorsData WS

Finantix will expose webservice for Manulife to fetch the Calculators input/output from POS application (see FDS, chapter 11 Calculators). Webservice will return the latest Calculators input/output collected by specified agent (see chapter 10.2.3 for details how agent identity is handled).

This is high level service signature (for details of the webservice interface please refer to IMS document, tab CAL-BE):

Service Name	getCalculatorsData
Input	Country identifier Manulife customer/prospect Identifier

	Agent Code
Output	List of input/output structures of the calculators Last updated DateTime

5.5.3 KYC/FNA/RPQ/Calculators batch export

Besides webservices described in chapter 5.5.2 there will also available export process for the same data (from data mapping point of view the same XML structure as for webservices will be used).

More than one file can be exported for each country for each entity to export. The files naming convention extends the one defined in the appendix as standard Finantix export naming convention. In particular, to enable the multi-country export, each file starts with a prefix identifying the country:

<country code>_<type prefix>_<yyyyMMdd>_<HHmmssSSS>.xml

e.g.:

HK_KYC_20150531_010004756.xml

HK_FNA_20150603_010024289.xml

HK_RPQ_20150603_010026086.xml

HK_CALCULATOR_20150603_010022644.xml

Information about country for which records are exported will be provided as an attribute of root element of exported XML file.

5.5.3.1 Delta export

1) The “delta” export will be executed daily.
2) The export process for KYC/FNA/RPQ/Calculators data starts every day at a predefined time configured by POS application administrator – please refer to chapter 2.1
3) The expected number of records exported every day depends on volume of usage of POS application, not identified yet by the functional requirement.
4) Exported files files will be kept in Finantix POS server file system for 7 days. Files older than 7 days will be purged.

5.5.4 Handling of the labels for FNA / RPQ / Calculators WS and export

For WS and export of FNA / RPQ / Calculators besides the information about the question / field ids labels displayed in the UI needs to be provided for MIL.

For both WS and export, Finantix will represent FNA / RPQ / Calculators data in XML format using <key, value> pairs:

a) key refers to the label the agent can see in the screen,
b) value corresponds to the input (case for FNA / RPQ) or input / output (for Calculators)
c) in case if there is a multiple-choice field answered only selected answer will be provided (full set of answers is defined in IMS document for all the relevant fields)

Example entry of XML file (final specification is as per IMS document):

```
<mapValues>
    <key>targetSavingPeriod</key>
    <value>28</value>
</mapValues>
<mapValues>
    <key>savingAndInvestmentamount</key>
    <value>80000</value>
</mapValues>
```

In addition to such mechanism Finantix will provide the dictionary file which will contain all the labels referred by the keys – file will not automatically be exported from the POS system, but provided as a part of Finantix solution delivery.

Dictionary file will be available for all the languages in country scope, e.g. if HK will have Chinese and English, there will be two dictionary files available – one with labels in English, and second with labels in Chinese.

File names will follow below pattern:

key_translations_XX_YY_ZZ.txt

where: XX – fna / rpq / calc; YY – country code; YY – language code, e.g. for English labels for Hong Kong there will be three files:

key_translations_fna_hk_en.txt

key_translations_rpq_hk_en.txt

key_translatons_calc_hk_en.txt

File will contain entries in format “key”=”label”, e.g:

“targerSavingPeriod” = “Saving Period yr”;

“savingsAndInvestmentamount” = “Savings and Investments (excluding property)”

MIL would use the dictionary files together with exported XML / webservices response to get the labels for country in particular language.

In case if any change will be introduced in FNA / RPQ / Caculators which affects the fields and labels as a part of POS solution delivery, Finantix will provide updated dictionary files containing all the key-label pairs including the change.

5.6 Extended customer/prospect data integration services

Some customer/prospect data, such as the list of proposals defined on Comprop Manulife system, will not be retrieved by POS solution using the batch import strategy. This paragraph lists the webservices that POS solution will consume to retrieve other customer/prospect related data and interact with MIL systems (country specific).

Please, refer to the Integration Mapping Specification document for details about actual data mapping.

5.6.1 Customer/prospect's proposals (Hong Kong specific Comprop integration)

Approach described in this paragraph applies to 2015 PHASE1 of the project and is specific for the Hong Kong country. Information for proposals is fetched via SOAP XML webservices exposed by MIL. Authentication approach used for webservice calls is as per chapter 10.2.3. This functionality will be available only when the device is connected to the Internet.

5.6.1.1 redirectProposalPage WS

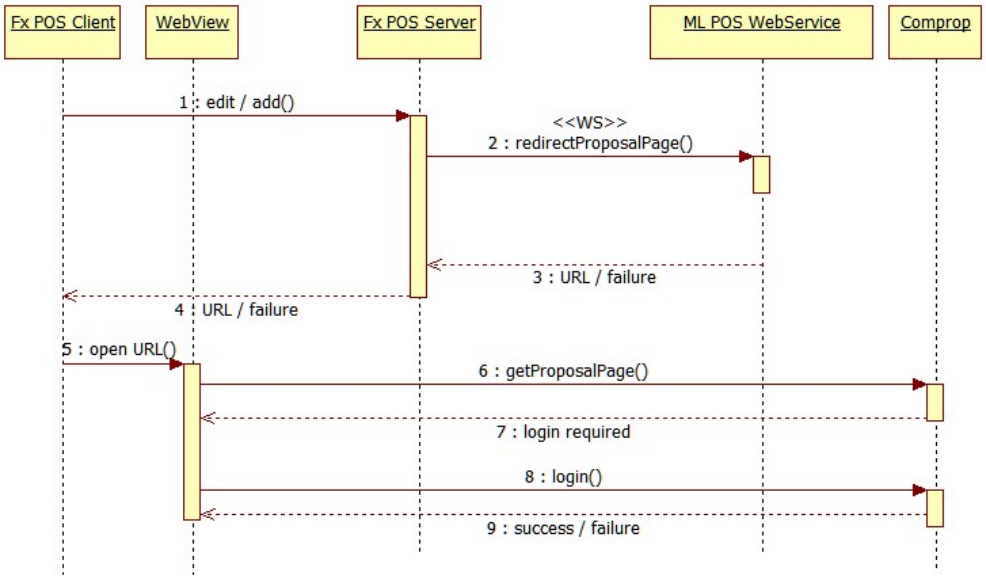


Figure 18 Open Comprop application – sequence diagram

Description:

1. Agent clicks "Edit" or Add (+) button in "Proposals" page (see FDS, chapter 26) – in case if (+) clicked, proposal number is not available and not passed as input; in case if "Edit" is clicked, proposal number is passed as an input
2. Finantix POS server application calls web service exposed by Manulife that return the URL to open Comprop login page. Parameters passed are Manulife unique prospect/customer identifier and agent ID
3. Web service return the URL to call
4. URL is returned back to the device
5. If call is successful device application open a WebView with the returned URL, otherwise error is logged in the Fx POS Server, then information propagated to Fx POS Client
6. If call is successful Proposal page specific for active customer/propect is requested to Comprop application, otherwise error message is displayed to the agent (see FDS, chapter 26)
7. User name and password to access Comprop application are required to the agent

8. User type username and password
9. If login is successfully completed, Comprop page is opened

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “RPP-WS”):

Service Name	redirectProposalPage
Input	Action (NEW/EDIT) (OPTIONAL) Proposal Number Manulife customer/prospect Identifier Client Type
Output	Login URL Logout URL

5.6.1.2 getProposalListWS / getProposalWS

5.6.1.2.1 User is landing in “Proposals” page from Finantix POS (see chapter 26 of FDS document for possible entry points):

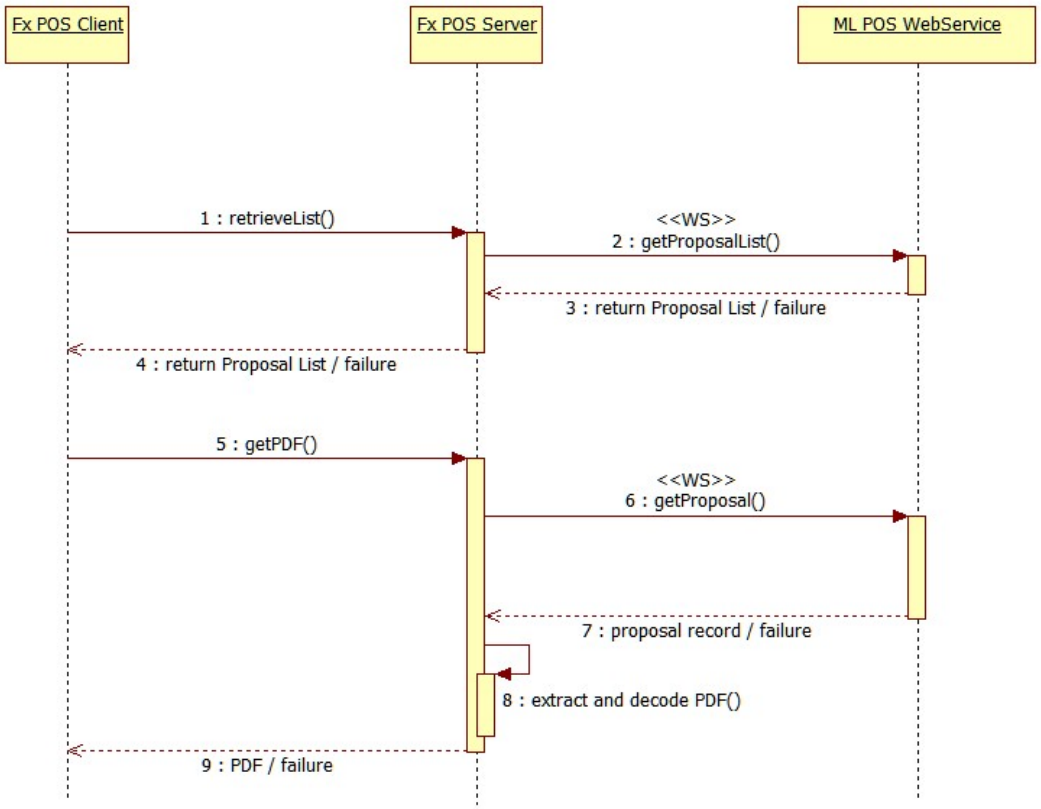


Figure 19 Open Proposal List (user is landing into “Proposals” page from Finantix POS) – Sequence diagram

Description:

1. Fx POS request to open page with Proposal list to FX POS Server (refer to chapter 26 of FDS document)
2. Fx POS Server application retrieve all the list off proposal with a Web service call .
3. If success, the list of all proposal for that client and agent are returned, otherwise error is logged in Fx POS Server and information propagated to Fx POS Client
4. If success, data is passed to the device, otherwise error is displayed to the agent (see FDS, chapter 26)
5. When page is display, first row will be selected and PDF to be displayed is requested to the server
6. All required Proposal data is fetched via web service
7. PDF data (encoded) and other info is returned by Manulife webservice
8. From returned xml, pdf is extracted and decoded
9. PDF is returned to device application

5.6.1.2.2 User has completed his activity in Comprop application and is moved to “Proposals” page in Finantix POS.

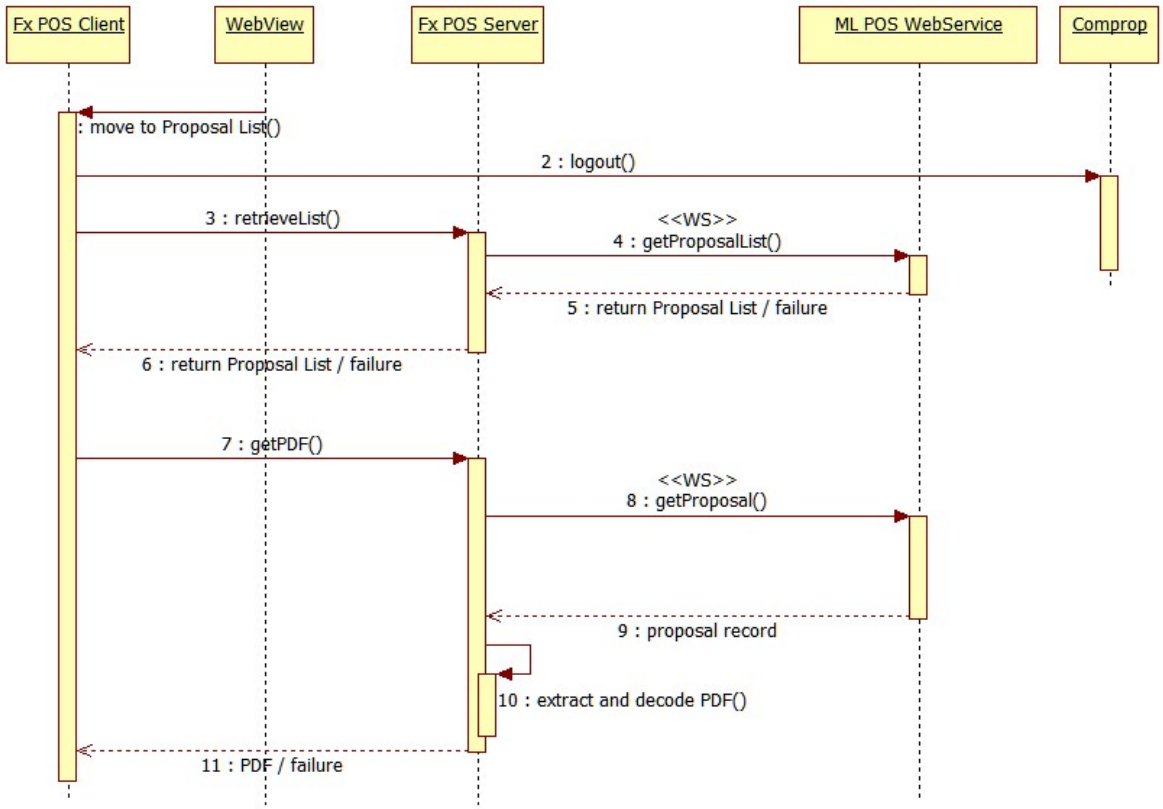


Figure 20 Open Proposal List (user has completed his activity in Comprop application and is moved to “Proposals” page in Finantix POS) – Sequence diagram

Description:

1. Agent closes WebView that contains Comprop application or click on move button
2. Device Fx POS application redirects to logout URL from Comprop (to be provided during mapping sessions) and closes WebView
3. Fx POS request to open page with Proposal list to FX POS Server

4. Fx POS Server application retrieve all the list off proposal with a Web service call .
5. If success, the list of all proposal for that client and agent are returned, otherwise error is logged in Fx POS Server, then information propagated to Fx POS Client
6. If call was successful, data is passed to the device, otherwise error message is displayed to the agent (see FDS, chapter 26)
7. When page is display, first row will be selected and PDF to be displayed is requested to the server
8. All required Proposal data is fetched via web service
9. If success, proposal record PDF data (encoded) and other info is returned by Manulife webservice, otherwise error is logged in Fx POS Server and information propagated to Fx POS Client
10. If success, from returned xml, pdf is extracted and decoded, otherwise error message is displayed to the agent (see FDS, chapter 26)
11. PDF is returned to device application

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “RIP-WS”):

Service Name	getProposalList
Input	Client Type MIL customer/ prospect unique identifier
Output	List of proposals (with Proposal Number)

5.6.1.2.3 New proposal behaviour and data migration (NOT CONFIRMED if for Day 1)

As new requirement for “day 1” scope (**delivery plan not confirmed yet),** only proposals created and edited from ePOS will be available in ePOS. It will not be possible anymore to access proposals crated from ePOs by other external system in order to edit them. Given that, the flows described above will not be valid anymore. Instead, proposal list and extended proposal data set (not only PDF) belonging to the given customer and Agent will be stored inside ePOS and synchronized with the device for offline usage.

“getProposal” service below will be extended in order to save both on device and on server all proposal data. “getProposalList” service will be dismissed.

A proposal porting procedure is necessary before “day 1” go live in order to move all existing proposals in comprop that belong to ePOS to ePOS server if those existing proposal might need to be displayed in ePOS. This procedure should be executed only once, but it is designed so that in case of any error during the execution it can be restarted in order to process only the remaining records.

Porting procedure:

1. Retrieve the full list of customers/prospects after the most recent batch import.
2. For each customer/prospect retrieved above, retrieve the full list of proposals by invoking the “getProposalList” webservice above.
3. For each proposal retrieved above, get the most recent proposal data by invoking the “getProposal” webservice below.
4. Store the retrieved data in ePOS server data storage.

This is high level service signature for “getProposal” (for details of the webservice interface please refer to IMS document, tab “GP-WS”):

Service Name	getProposal
Input	Client Type MIL customer / prospect unique identifier Proposal Number
Output	Basic Response Information [1..1] Error Information [1..N] PDF [1..1] **Basic information [1..1]** **Agency Information [1..1]** **Application Information [1..1]** **Prospect / Customer Information [1..N]** **Product Information [1..N]**

5.6.1.3 deleteProposalWS

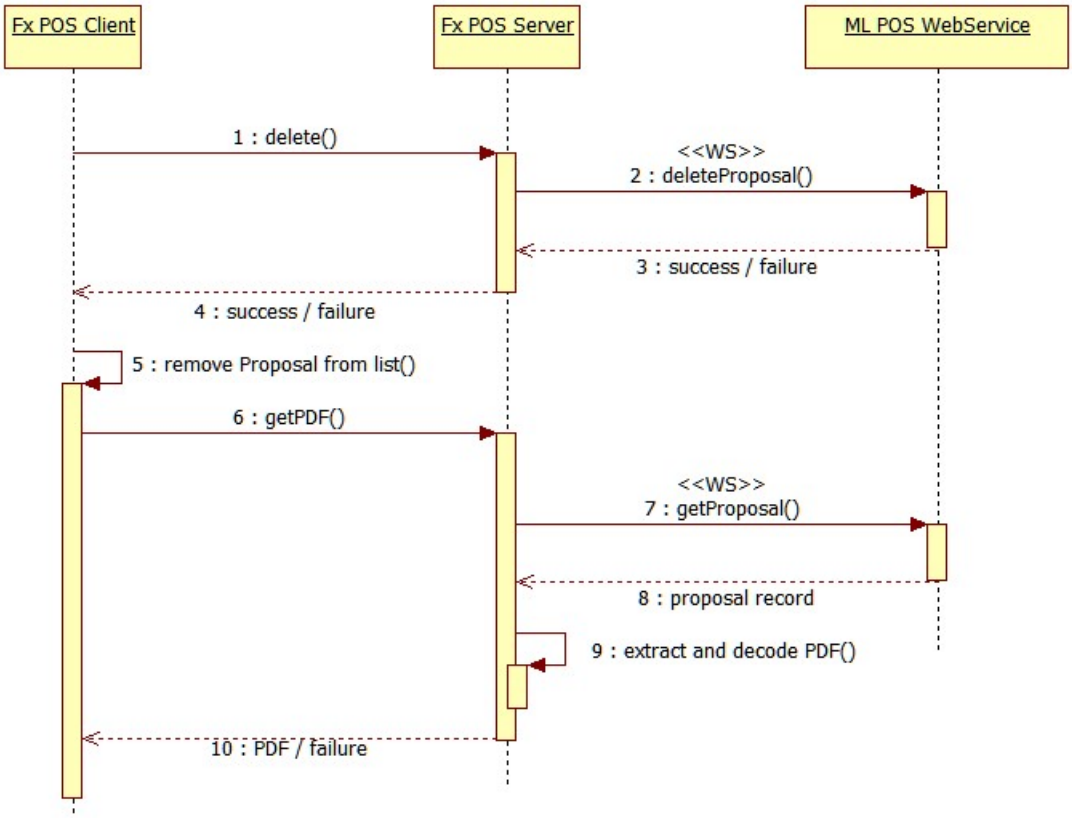


Figure 21 Delete Proposal – Sequence diagram

Description:

1. Agent clicks on Delete button
2. FX POS Server calls “Delete Proposal WS” to delete the proposal
3. Deletion result is returned to Fx POS Server
4. Result is propagated to Fx POS Client, in case of failure error is logged in Fx POS Server
5. If success, proposal from the list is locally removed, otherwise no change in list content and error displayed to the agent (see FDS, chapter 26)
6. Another proposal is active on client side and a new call to retrieve PDF is triggered (flow the same as depicted in Figure 23 above),

This is high level service signature (for details of the webservice interface please refer to IMS document, tab “DPR-WS”):

Service Name	deleteProposal
Input	Client Type MIL customer/ prospect unique identifier Proposal Number
Output	Result

5.6.1.4 printProposal WS

When printing of proposal is requested via Manulife printing solution, WS printProposal will be called by Finantix POS application. In case if webservice call will return an error, information will be logged in Finantix POS server side and error displayed to the user in Finantix POS application (see FDS, chapter 26).

This is high level service signature (for details of the webservice interface please refer to IMS document, tab PPR-WS).

Service Name	printProposal
Input	Client Type MIL customer/ prospect unique identifier Proposal Number
Output	Result

5.6.2 Customer’s insured amount and available amount (Japan specific) – OUT OF SCOPE FOR 2015 PHASE1

This service is used to retrieve the total insured amount of a customer that has to be shown inside the customer profile panel as per Japan specific customization.

This is high level service signature (for details of the webservice interface please refer to IMS document, tab GIA-WS):

Service Name	getInsuredAndAvailableAmount
Input	Country identifier Customer/prospect unique identifier
Output	Insured amount Available amount

5.7 Policy information data import

5.7.1 Policy batch import (XML)

1) Customers' policies will be imported on a daily base using the Finantix standard import tools as depicted by previous paragraphs.
2) Given that policies might refer to very old products no more available on POS system, every information necessary to display policy data must be provided by Manulife to POS application server, that is, for example, also labels and translations of related products and riders.
3) Manulife will provide the daily "delta" for policies, that is, only the policies actually created, deleted or modified during the day before the import.

5.7.1.1 XML file structure and data mapping

1) Customer's policies XML file has to be structured so that there is a single node for each customer and this node is the first level after the file root.
2) The main attribute of the node is the unique identifier (by country) of the customer.
3) Inside the first level node, all policies belonging to that customer have to be stored. In this way, if the policy file have to be split to proceed for a parallel import of chunks, the splitting will be done on the first node level that is the customer and so the data consistency (correct list of policies belonging to the customer) is preserved.
4) Each policy node must contain the unique identifier of the agent related to that policy, so that POS solution can properly associate the customer to the agent.
5) The three types of policies to show are Protection and Savings (see FDS, chapter 20.1 Protection and Savings), ILAS (see chapter 20.2 ILAS (Investment Linked Assurance Scheme), Wealth Management (20.3 Wealth Management), so the data model will define these three types each of which with its peculiarity.
6) The purpose of having policy data on POS application (for customers) is only to properly display them in an aggregate view that depends on the type above.
7) To properly display the aggregated charts on "Wealth" component under the Holistic Customer Profile, aggregated data must be provided as content of the customer node.
8) Policy data depends on the particular product referred and moreover on the historical structure of that product at the moment in which the policy was approved.
9) To support the dynamic structure of policies, the policy data model is designed to be flexible enough to contain any data coming from Manulife backend systems, but structured enough to support the three families described above and a proper

identification of the policy and visualization of each data set inside the requested area of the policy detail page (and report).

10) Each policy model element will be mapped to the area of the page named in the same way as described in the FDS document, chapter 16 Holistic Customer Portfolio (English name will be used)

11) A specific map is dedicated only to the “cards” data visualization. If necessary, data has to be duplicated inside this special map and the main detail map

12) In case if any of the maps / tables is not provided by MIL in policy record, import will not discard the record, but UI section will not be displayed

This is a high level data model of the policies that shows the “fixed” parts and the “dynamic” parts (please refer to FDS, chapter 20 Policy Portfolio, Figure 4 Map Areas)

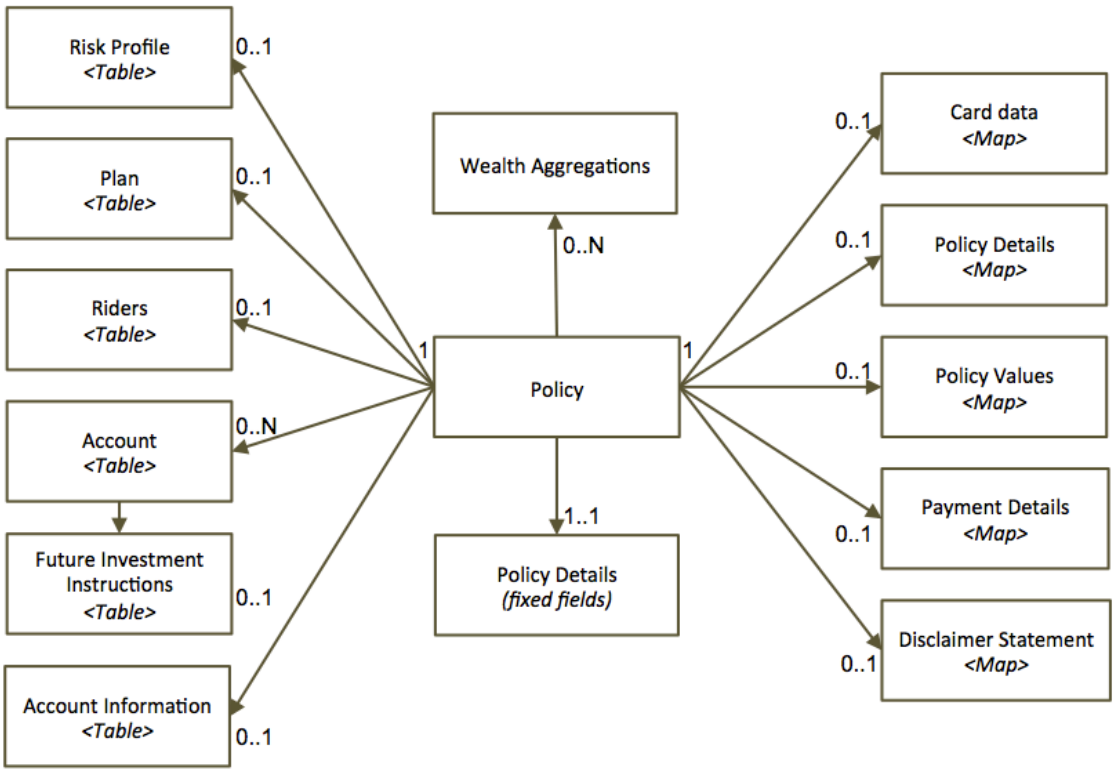


Figure 22 – Policy data model

13) To support the functional design of the policy details and panel pages (for example, the policy date has to be used to sort the policies, the insured name has to be shown as card header in the policies panel, etc.) the minimal set of fixed fields is required on the “Policy Detail” object – please refer to FDS, chapters 20.1 Protection and Savings, 20.2 ILAS (Investment Linked Assurance Scheme), 20.3 Wealth Management for the list of fixed fields relevant for all of the policy types.

14) Complex data structures like Japan Universal Life policy will need a specific page rendering not provided yet by HeathWallace. In order to support this complex rendering, a more complex “map of maps” will be supported as an extension to existing data model and will not affect other countries (OUT OF SCOPE FOR 2015 PHASE1)

Please, refer to the IMS document, tab “POLI-Batch” for details about actual fields and data mapping.

5.7.1.2 Dictionary for field labels and enumeration values translation

Please refer FDS, chapter 20 Policy Portfolio for details.

1) Manulife has to provide the labels of the "dynamic" parts of the policy data model to POS application (labels of the fields provided inside the maps and labels of the column headers).
2) Labels are provided separately from the main policy import feed in a dedicated "dictionary" file in order to have a:
   - much more compact import file for policies;
   - much less data base space consumed by policies;
   - highly reduced synchronization time (much less strings to send to the device);
   - better control of the labels provided and of the coherence of those labels;
3) Enumeration translations can be managed as labels and so stored inside the dictionary.
4) The “key” information of the map entries inside the policy import file (see XML sample below) is used to uniquely identify the label information inside the dictionary file.
5) A dedicate import for the dictionary is activated together with the policy daily import if a new dictionary file is provided by MIL.
6) The dictionary is updated as a “delta” by Manulife whenever a new label or enumeration value is introduced or a label/enumeration is changed (in any translation).
7) Field values which will not be a part of dictionary file will be displayed as provided in import file and will not have different translation when language settings be changed by POS user

Please, refer to the IMS document, tab “PDI-Batch” for details about actual data mapping.

5.7.1.3 Policy actions

Insert / update or deletion of policy is driven by action attribute. There are two actions defined for policy record: “UPDATE” and “DELETE”.

1. “UPDATE” – creates or updates the record

   When record is not available in POS application it will be inserted with full data set provided by Manulife in XML import file. In case of updates of policy data, to save import time and synchronization time, only the minimal set of information have to be sent from Manulife to POS application server inside the batch import XML. If only one policy of a customer has been changed, only that policy data has to be sent inside the XML file, not the full list of policies of the customer;

2. “DELETE” – deletes policy record

   To delete policy record in POS application, minimal set of data to be provided in the XML file is policy number and action specified as “DELETE”.

5.8 Import scenarios

5.8.1 New prospect creation (prospect created in MIL system)

Prospect is created in MIL system and not available in Finantix POS.

1. Prospect import
    a. MIL provides action UPDATE
    b. MIL provides Manulife Prospect Id (prospecting data)
    c. MIL provides reference agent code (prospecting data)
    d. MIL provides data for prospect record
    e. Finantix POS creates new prospect record with provided Manulife Prospect Id
2. Policy import

    N/A
3. Customer import

    N/A

5.8.2 Update prospect

Exisitng prospect record is to be updated in Finantix POS system.

1. Prospect batch
    a. MIL provides action UPDATE
    b. MIL provides Manulife Prospect Id
    c. MIL provides reference agent code (prospecting data)
    d. MIL provides data for prospect record
    e. Finantix POS updates prospect record with Manulife Prospect Id
2. Policy batch

    N/A
3. Customer batch

    N/A

5.8.3 Delete prospect

Prospect has been deleted in MIL system and should be deleted in Finantix POS.

1. Prospect batch
    a. MIL provides action DELETE
    b. MIL provides Manulife Prospect Id
    c. Finantix POS deletes the prospect record with Manulife Prospect Id and all the data associated
2. Policy batch

    N/A

3. Customer batch

   N/A

5.8.4 Update customer

1. Prospect batch

   N/A

2. Policy batch

   N/A

3. Customer batch

   a. MIL provides action UPDATE

   b. MIL provides Manulife Customer Id

   c. MIL provides reference agent code (prospecting data)

   d. MIL provides data for customer record

   e. Finantix POS updates the record with Manulife Customer Id

5.8.5 New policy (new customer from prospect)

Prospect bought policy from agent and become customer in MIL system. He needs to be converted to customer in Finatnix POS and all the prospecting data remains visible for the agent.

1. Prospect batch

   N/A

2. Policy batch

   a. MIL provides action UPDATE

   b. MIL provides Manulife Customer Id

   c. MIL provides reference agent code

   d. MIL provides policy number

   e. MIL provides policy data

   f. Finantix POS creates policy for customer with Manulife Customer Id, linked to agent with provided agent reference agent code

3. Customer batch

   a. MIL provides action CONVERT

   b. MIL provides Manulife Customer Id

   c. MIL provides Manulife Prospect Id (prospecting data)

   d. MIL provides reference agent code

   e. Finantix POS converts Prospect to Customer

5.8.6 New policy (new customer not from prospect)

Existing customer record from MIL system to be created in Finantix POS (no conversion of prospect in Finantix POS).

1. Prospect batch

N/A

2. Policy batch
    a. MIL provides action UPDATE
    b. MIL provides Manulife Customer Id
    c. MIL provides policy number
    d. MIL provides reference agent code
    e. MIL provides policy data
    f. Finantix POS creates policy record for customer with Manulife Customer Id, linked to agent with provided agent reference code
3. Customer batch
    a. MIL provides action UPDATE
    b. MIL provides Manulife Customer Id
    c. MIL provides reference agent code (prospecting data)
    d. Finantix POS creates customer record with Manulife Customer Id

5.8.7 New policy (existing customer)

New policy has been sold by agent to existing customer outside Finatix POS application.

1. Prospect batch

    N/A
2. Policy batch
    a. MIL provides action UPDATE
    b. MIL provides Manulife Customer Id
    c. MIL provides policy number
    d. MIL provides reference agent code
    e. Finantix POS creates policy record for customer with Manulife Customer Id, linked to agent with provided agent reference code
3. Customer batch

    N/A

5.8.8 Agent leaves and policy becomes orphaned

When agent leaves, policy becomes orphaned for certain period of time. Later it will be reassigned to another agent. All the prospect records and all the customer prospecting data assigned to the agent must be deleted.

1. Prospect batch
    a. MIL provides action DELETE
    b. MIL provides Manulife Prospect Id
    a. Finantix POS deletes record with Manulife Prospect Id and all its prospecting data
2. Policy batch

a. MIL provides action UPDATE
b. MIL provides Manulife Customer Id
c. MIL provides policy number
d. MIL provides empty reference agent code
e. Finantix POS drops existing link between policy and agent

3. Customer batch
   a. ~~MIL provides action DELETE~~ (as per HKCR030 - action not available due to Backend limitations)
   b. MIL provides Manulife Customer Id
   c. MIL provides reference agent code
   d. Finantix POS deletes all the customer prospecting data for agent (customer record is not deleted)

5.8.9 Agent leaves and policy re-assigned to new agent

When agent leaves, policy is directly reassigned to new agent. All the prospect records and all the customer prospecting data assigned to the original agent must be deleted.

1. Prospect batch
   a. MIL provides action DELETE
   b. MIL provides Manulife Prospect Id
   c. Finanitx POS deletes record with Manulife Prospect Id
2. Policy batch
   a. MIL provides action UPDATE
   b. MIL provides Manulife Customer Id
   c. MIL provides policy number
   d. MIL provides reference agent code of new agent
   e. Finantix POS updates policy record, changing servicing agent to the new agent
3. Customer batch
   a. ~~MIL provides action DELETE~~ (as per HKCR030 - action not available due to Backend limitations)
   b. MIL provides Manulife Customer Id
   c. MIL provides original agent reference code (prospecting data)
   d. Finantix POS deletes all the customer prospecting data for original agent (customer record is not deleted)

5.8.10 Policy re-assigned to new agent – scenario 1

Existing policy is reassigned to new agent and as a result original agent will not have policies for the customer. This customer prospecting data for original agent must be deleted.

1. Prospect batch

   N/A
2. Policy batch
   a. MIL provides action UPDATE
   b. MIL provides Manulife Customer Id
   c. MIL provides policy number
   d. MIL provides agent reference code of new agent
   e. Finantix POS updates policy record, changing servicing agent to the new agent
3. Customer batch
   a. ~~MIL provides action DELETE~~ (as per HKCR030 - action not available due to Backend limitations)
   b. MIL provides Manulife CustomerId
   c. MIL provides original agent reference code (prospecting data)
   d. Finantix POS deletes all the customer prospecting data for original agent

5.8.11 Policy re-assigned to new agent – scenario 2

Existing policy is reassigned to new agent but original agent still has policies for the customer.

1. Prospect batch

   N/A
2. Policy batch
   a. MIL provides action UPDATE
   b. MIL provides Manulife Customer Id
   c. MIL provides policy number
   d. MIL provides agent reference code of new agent
   e. Finantix POS updates policy record, changing servicing agent to the new agent
3. Customer batch

   N/A

5.8.12 Update policy

1. Prospect batch

   N/A
2. Policy batch
   a. MIL provides action UPDATE
   b. MIL provides Manulife Customer Id
   c. MIL provides policy number
   d. MIL provides reference agent code
   e. MIL provides new policy data

f. Finantix POS updates policy record for customer with Manulife Customer Id and linked to the agent

3. Customer batch

   N/A

5.8.13 Policy becomes inactive – scenario 1

Policy became inactive in MIL system and should not be available for agent in POS application. As the result agent will not have policies for the customer. This customer data for agent must be deleted.

1. Prospect batch

   N/A

2. Policy batch

   a. MIL provides action DELETE

   b. MIL provides Manulife Customer Id

   c. MIL provides policy number

   d. MIL provides reference agent code

   e. Finantix POS deletes policy record for customer with Manulife Customer Id and linked to the agent

3. Customer batch

   ~~a. MIL provides action DELETE~~ (as per HKCR030 - action not available due to Backend limitations)

   b. MIL provides Manulfe Customer Id

   c. MIL provides reference agent code

   d. Finantix POS deletes all customer prospecting data for agent

5.8.14 Policy becomes inactive – scenario 2

Policy becomes inactive in MIL system and should not be available for agent in POS application, but agent still has active policies for the customer.

1. Prospect batch

   N/A

2. Policy batch

   a. MIL provides action DELETE

   b. MIL provides Manulife Customer Id

   c. MIKL provides reference agent code

   d. Finantix POS application updates policy record for customer with Manulife Customer Id and links to the agent with provided reference agent code

3. Customer batch

   N/A

5.8.15 Conversion of two prospect records into single customer record

The same person buys policy from two agents – it is prospect 1 for agent 1 and prospect 2 for agent 2. MIL system merges two prospect records into single customer record.

1. Prospect batch

   N/A

2. Policy batch
   a. MIL provides two policy records one for agent 1 and one for agent 2 both assigned to the same customer

3. Customer batch
   a. MIL provides action CONVERT
   b. MIL provides Manulife Customer Id
   c. MIL provides two prospecting data sets, one with Manulife Prospect Id of prospect 1 and reference agent code 1 and second with Manulife Prospect Id of prospect 2 and reference agent code 2
   d. Finantix POS converts agent 1 prospect and agent 2 prospect to the single customer record and ensures that all the agent 1 and agent 2 customer prospecting data will be available for them under customer record (including data which is not part of batch import, like KYC, RPQ, FNA, Calculators) – agent 1 will be able to see only customer prospecting data of prospect 1, agent 2 will be able to see only customer prospecting data of prospect 2

6. Configuration of defaults for calculators and POS application input fields

1) Default values of input fields and calculator's input and parameters are stored inside POS DataBase
2) Default values of input fields and calulator's input and parameters can be configured within configuration file and uploaded to POS DB.
3) These data don't change frequently (once two per year) and so can be stored on dedicated property files inside the server on a predefined shared folder.
4) Each component and calculator has its dedicated property files with configuration data.
5) Each file is also country-specific, so the file name structure is:

   <country code>_<component name>.xml

6) After upload of changed property file refresh on POS server (property files reload) would need to be triggered to get the changes effective.
7) If the file data load or content parsing fails, , values currently stored in DB are kept and an error message will be logged. In case of first upload, with no existing configuration in DB a set of predefined "hardcoded" default values will be set and an error message will be logged (as defined in FDS in respective "Data Fields" tables)

8) Default values configured using the property files are sent to the device application using the standard entity synchronization mechanism.

9) Format of the file content will be a set of e <key>=<value> entries, where <key> would be used to uniquely identify the field in application UI and value will be the default value field will take when initialised in POS application.

7. Applications and underwriting

7.1 Scope

Apply is aimed to deliver solution for ML insurance agents to finalize sales of the products selected in the proposal stage. Solution (referred as “Apply” in this document) will be a tablet-based application integrated with several external systems (please see the below figure for integration points).

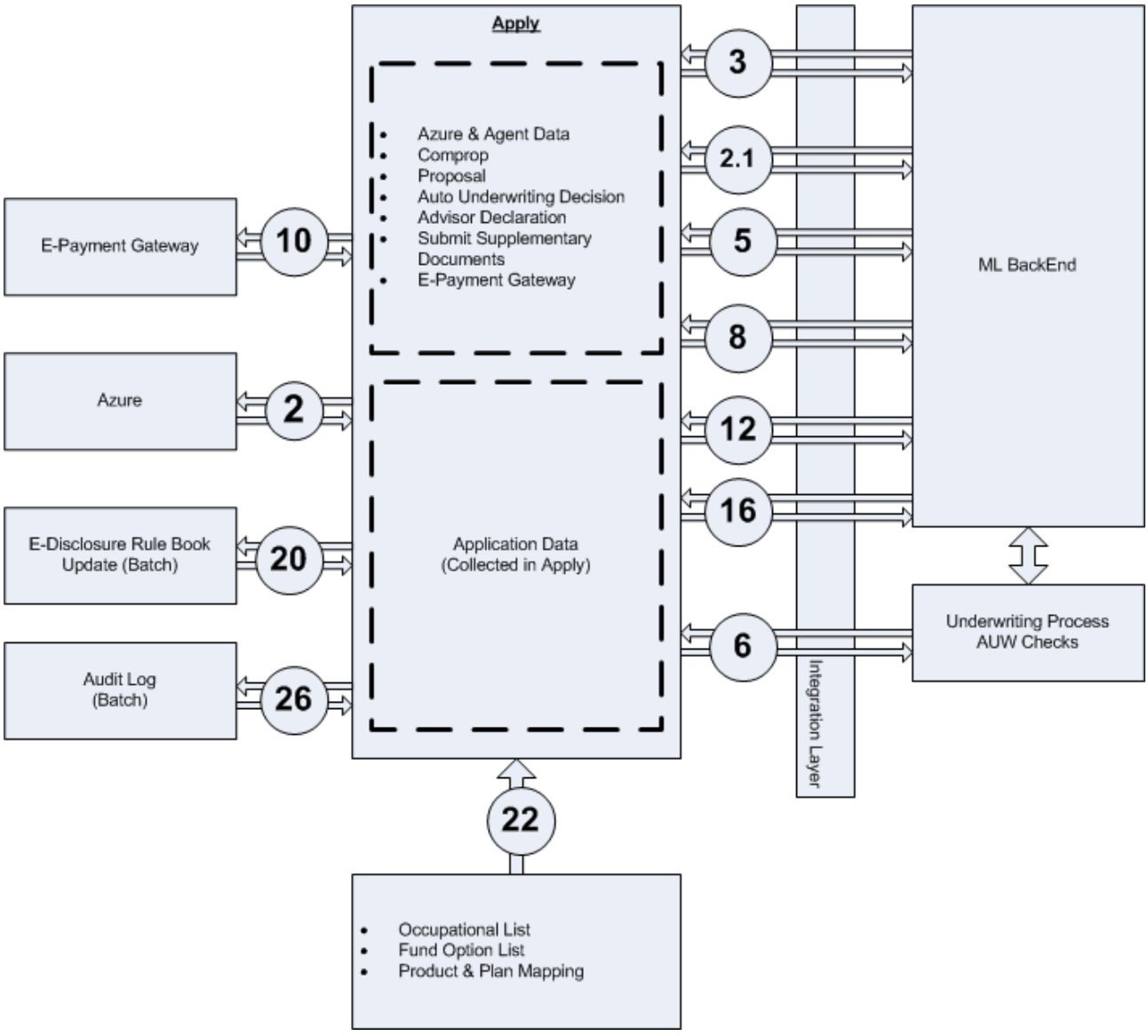


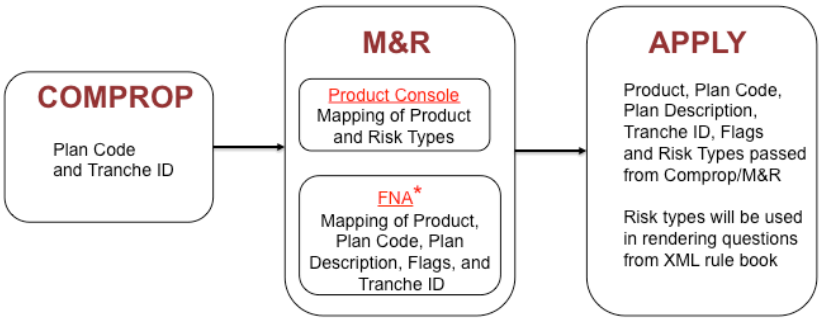
Figure 23: Integration points of Apply system

Note that the figure above reflects overall POS Apply solution, however only the following integration points are covered in SAS:

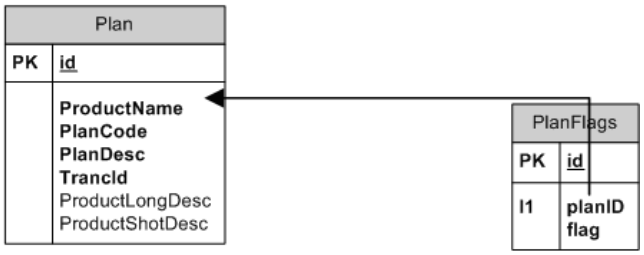
	Integration point	Scope	Reference in document
2	Authentication (Azure)	Regional	10.2.4
2.1	Basic Agent Data 2 (BAD2)	Hong Kong	10.2.1
2.1	Get Extended Agent Data WS (EAD-WS)	Hong Kong	10.2.2
5	Proposal (getProposalDetail)	Hong Kong	5.6.1.2
5	Underwriting Decision request / Underwriting Decision response (UWA-WS)	Hong Kong	7.5.1.1.2
10	E-Payment Gateway	Hong Kong	9.29.2
10	Payment (retrievePaymentGatewayTxNo, UTN-WS)	Hong Kong	9.3.4
10	Payment (GetExchangeRate, GER-WS)	Hong Kong	9.2.1
10	Payment Status URL	Hong Kong	9.3.1
10	Payment Status (payment_status_WS)	Hong Kong	9.3.3
10	GetPaymentStatus	Hong Kong	9.3.2
8	Application Submit (SubA-WS)	Hong Kong	7.5.1.1.3
12	Advisor Declaration (SubA-WS)	Hong Kong	7.5.1.1.3
16	Submit Supplementary Documents (SubA-WS)	Hong Kong	7.5.1.1.3

	Batch	Scope	Reference in document
20	eDisclosure Rule Book update	Regional	8.1.2
22	Fund Configuration	Hong Kong	7.5.2.3
22	Occupational List (OCC-Batch)	Hong Kong	7.5.2.1
22	Product and Plan mapping	Hong Kong	1.1
26	Auditing And Logging	Hong Kong	11

7.2 Product, Plan, Risk Type, and Flag mapping



- Plan Code and Tranche ID: Provided by the getProposal WS.
- Risk Types for Products: Provided by Product Console.
- Product, plan, risk type, and flags:
    - Provided by new look up tables.
    - Loading process (Ref to: 7.5.2.1)
    - The loading will be happen during the maintenance windows (1:00am to 8:00am). During the loading, the system is not expected any user log into the system.

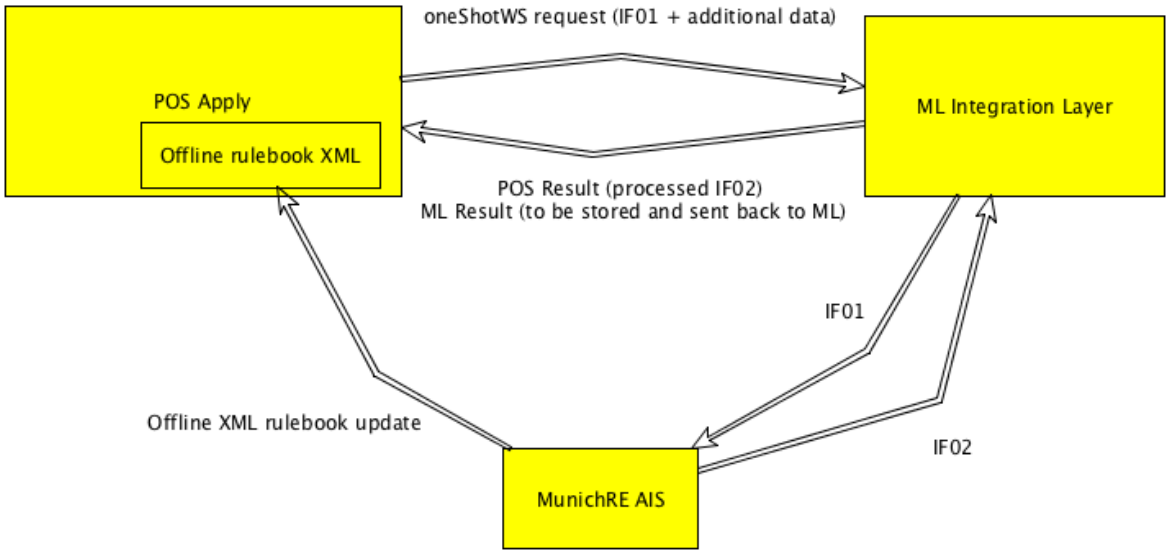


7.3 e-Disclosure auto-underwriting flow

MunichRE generate a country specific rulebook and send it over to the POS system. Based on the content, Epos then will getter the information for the application. Once it done, EPos will call the AUWS-WS (Ref to: 7.5.1.1.2) provided by ML integration layer along with the application data. As a result, the Web service response the policy number, validation status, result, and etc.

****** AUWS-WS is replaced for oneShotWS.***

Figure 24: e-Disclosure flow



7.4 Application submission flow

Application submits happens as the very final step of application process in POS Apply after the payment made (Ref to Chapter: Payments)

Besides the pure data transmission also binary files like PDF or any accompanying materials needs to be send to ML.

Application submission is involved the following 3 processes.

Submission 2: Application Data such as AML, product information, insure, case data, and etc.

Submission 3: Advisor Declaration such as policy number, agent code, and etc

Submission 4: Supplementary Documents.

Web Service: submitApplicationWS (ref to: 7.5.1.1.3)

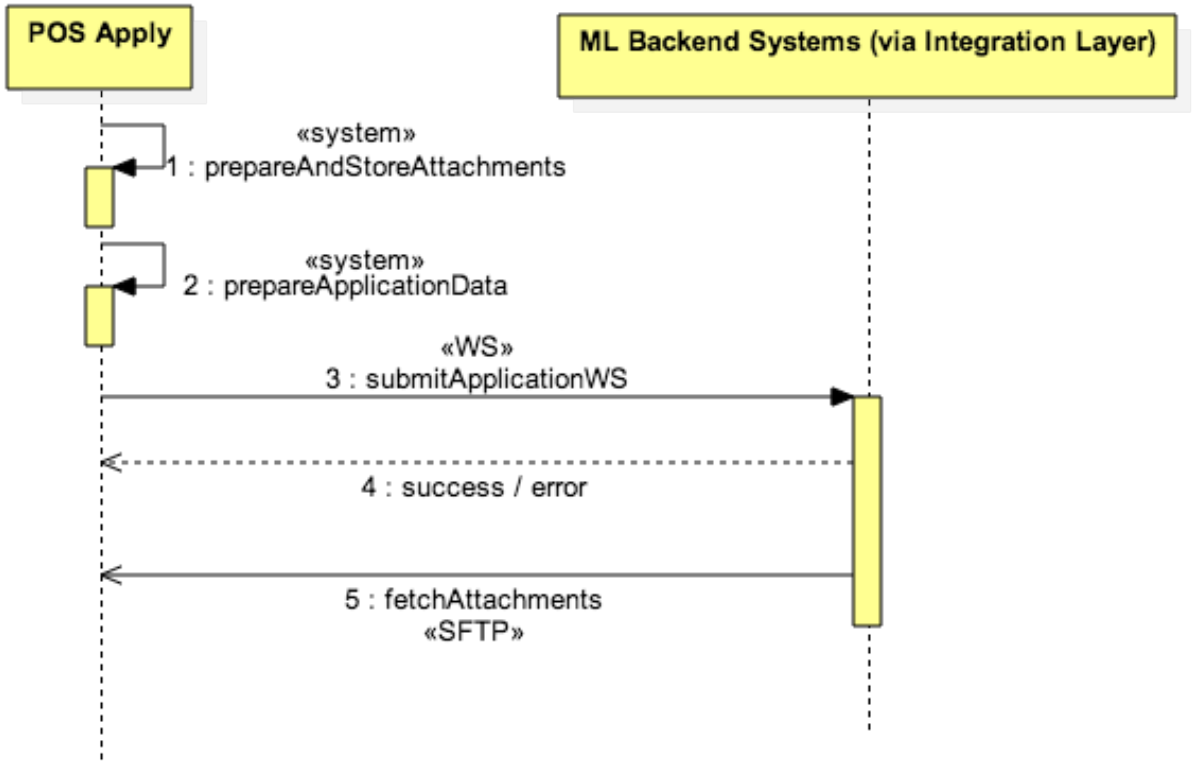


Figure 25:Submit application flow

Description:

1. POS system prepares attachments (e.g. PDF files) and stores those in file system accessible by POS Apply and ML Integration Layer
2. POS system prepares application data for WS request
3. POS system calls submitApplicationWS to propagate the application data

4. Error / success information is propagated to POS Apply.
5. ML Backend Systems fetch the attachments from file system location via SFTP (configuration of SFTP and housekeeping is in ML scope).

7.5 Integration

This chapter covers the topics related to all the integration points for Apply system.

Detailed, data field level specification of all the online and batch integrations will be covered by IMS document.

7.5.1 Online Integrations

7.5.1.1 Web Services

7.5.1.1.1 getProposal (for details of the webservice interface please refer to IMS document, tab "GP-WS")

getProposal SOAP XML webservice will be exposed by ML integration system for getting the proposal data from ComProp, after the proposal created, to the POS Apply:

High-level service signature is presented in the table below.

Service Name	getProposal
Input	proposal Number ProspectID / customerID
Output	Proposal PDF Basic Response Information [1..1] Prospect Information [1..n] Application Information Coverage/Rider Information [1..n]

7.5.1.1.2 auwSubmissionWS (exposed by ML Integration layer)

Web service which will be called to get the underwriting response from MunichRE AUW engine. Finantix will not call the AUW directly, but through ML integration layer.

Apply will prepare IF01 message to be embedded in the request call – "AUW Case Data" entity in table below (see chapter 8.2 for IF01 message structure details).

High-level service signature is presented in the table below.

Service Name	auwSubmissionWS
Input	AML Information nbRecord fnalist Product Information [1..n] Insured Information [1..1] beneficiaryInfo[1..1]

	fundRecord nbQuestionnaire AUW Case Data [1..1]
Output	policyNumber validationStatus posResult [1..n] action message paymentIndicator

POS Result entity is the part of response, which will contain ready-to-use in POS Apply UI underwriting result – ML will provide this after processing response from MunichRE (IF02 message).

ML Result entity is not to be used by POS Apply, but should be stored in POS till the time of application submission via submitApplicationWS.

7.5.1.1.3 submitApplicationWS (exposed by ML Integration Layer)

POS Apply will use this service to send application data (submission 2), Advisor Declaration (submission 3), and supplementary Documents (submission 4) in forms of attachments to the ML backend systems. Attachments will not be sending as part of webservice call, but stored in the file system location accessible by both POS Apply and ML target system. List of the stored files will be passed in the webservice request. For details of the flow please refer to (1.1). AUW Case Data (IF01 message) will be stored in POS DB when auwSubmissionWS be executed without error response (please refer to chapter 7.8.1).

High-level service signature is presented in the table below:

Service Name	submitApplicationWS
Input	AML Information nbRecord finalist Application Data [1..1] ML Result from auwSubmissionWS [1..1] List of file attachments [1..n]
Output	Error Information [1..n]

Attachments will be stored in location:

```
/storage/AppAttachments/<country>/<agent>/EPOS/<ApplicationNumber>
```

7.5.1.2 Hong Kong ComProp Integration Flow

Proposal data from ComProp proposal system will be transferred to Apply and available on device for offline access. Application creation will be based on proposal stored on device.

NEW / UPDATE mode – creation of new or update of existing application record with data pulled from ComProp

In this scenario getProposal WS (see chapter 5.6.1.2) call is needed to propagate proposal information from ComProp to Apply system, whenever a proposal is created or updated in ComProp.

Description:

1. In POS M&R, user creates/updates proposal in ComProp WebView and exit WebView once proposal is saved. POS M&R device app calls Finantix POS server to get updated proposal list.
2. Finantix POS server calls ComProp web service (existing getProposalList WS) to get updated proposal list.
3. Propagation of success / failure response to Finantix POS server.
4. Propagation of success / failure response to POS device app, and device app stores proposal list on device for offline use. Device app uses last updated timestamp to determine the recently updated proposal.
5. When entering single proposal details, POS device app calls Finantix POS server to get the recently updated proposal data fields and PDF.
6. Finantix POS server calls ComProp web service (extended getProposal WS) to get proposal data fields and PDF.
7. Propagation of success / failure response to Finantix POS server.
8. Propagation of success / failure response to POS device app, and device app stores proposal data for offline use.
9. POS device app stores proposal data fields in device database as application “plan”object (data structure replicated from “sendPlanData” webservice) if a new Application process is started by the user from the selected proposal.
10. Application “plan” data is synchronized back to Finantix POS server on next device synchronization (as part of customer extended data set).

7.5.2 Batch Integrations

Some master data sets provided by ML needs to be uploaded and available in Apply system. This chapter provides high-level specification of all the batch integration points, defines purpose of the particular batch processes, frequency and number of records to be imported.

All the batch files provided by Manulife will contain Unicode characters encoded in UTF-8 standard.

All the batch processes will be executed in insert / update mode with initial upload and only delta records provided with specified frequency unless specified differently.

Batch processes will be scheduled and executed at the specified time in dedicated time window (refer to the sub-chapters for details) unless specified differently (manually triggered batch integrations).

Batch processes (available time window)	Time window
Funds	TBC
Product, Plan, and Flag mapping	TBC
Occupational List	TBC

Files must be provided in the POS server file system by the time of batch process execution. Directory structure required by Finantix import facility is to be created under the agreed location (suggested to be the same for all the batch import processes per country:

/storage/Batch main folder>/upload/

<Batch main folder> is defined in the sub-chapters for all the defined batch processes.

File naming convention proposed for XML files is:

<country code>_<typprefix>_<yyyyMMdd>_<HHmmssSSS>.xml

e.g. HK_FUNDS_20160109_110054111.xml

File naming convention proposed for CSV files is:

<country code>_<type prefix>_<yyyyMMdd>_<HHmmssSSS>.csv

e.g. HK_FUNDS_20160109_110054111.csv

<type prefix> parameter is defined in the sub-chapters for all the defined batch processes.

7.5.2.1 Product, Plan, and Flag mapping

The batch processes will be executed in delete all / insert mode. ML will provide complete set of the data to be imported by batch process. All the records will be deleted before new batch import execution.

Method	CSV batch
Frequency	TBC
Initial upload (records)	TBC
Delta (records)	Not applicable. Full data set will always be included in the feed.
Process schedule (time within time window)	
Process <Batch main folder>	ProductPlanFlagImport
File name <type prefix>	ProductPlanFlag

7.5.2.2 Occupational List

ML will request and provide the occupation data file to upload the occupation list to the system. The data will be uploaded onto the system during the maintenance window. The process will be always full load at day 1.

Method	XML batch
Frequency	TBC
Initial upload (records)	700
Delta (records)	Not applicable. Full data set will always be included in the feed.
Process schedule (time within time window)	
Process <Batch main folder>	OccupationalListImport
File name <type prefix>	OccupationalList

7.5.2.3 Funds

Data set contains the information about funds for ILAS products, such as fund code, risk level, start/end date, etc.

Note that this import process will always include full data set.

Method	XML batch
Frequency	7-8 times per year. ML HK IS team will import the feed manually.
Initial upload (records)	Roughly 1000
Delta (records)	Not applicable. Full data set will always be included in the feed.
Process schedule (time within time window)	Not applicable.
Process <Batch main folder>	fundsXMLImport
File name <type prefix>	FUNDS

7.6 e-Signature and PDF Generation

7.7 e-Signature & PDF Digital Signature

To ensure integrity and prevent tampering, one PDF files will be created and that will merge all the PDF files generated for the application. Then, it will be digitally signed with digital certificate. ML will provide the digital certificate and Finantix will choose the library to sign the PDF files. All the PDF files and the merged file will transfer to the ML backend system.

7.7.1.1 Supplemental Documents (Pictures)

- The file name will based on the type as following:
  - Type.{insured|owner}.seq.png (TBC for the detail)
  - The captured picture will be converted to PDF format
  - All the PDFs will be merge with their type such as multiple pictures that related to the passport will be merged to one pdf file.

7.7.1.2 Fixed-format Forms

ML will provide PDF files for fixed-format forms as templates, and data collected from Apply will be filled into PDF templates as overlay objects.

7.7.1.3 Non-fixed-format Forms

Non-fixed format forms will be dynamically generated.

For example, Application Summary PDF will consist of a number of sections, which in turn will consist of a number of questions and answers. The questions and answers may come from different UI sections/forms (please refer to example in Appendix of “POS Asia Division Business Requirements Document - Apply Hong Kong”).

ML will provide details for each non-fixed format form:

- Data fields to include in the PDF
- e-Signatures to include in the PDF

7.7.1.4 CompProp generated PDF

1. ComProp will generate the pdf with all the information needed and the "hidden labels" for the following fields
   1. For ILAS
      - policy number
      - signature
      - signature Date (System Date)
   2. For Non-ILAS
      - initial
      - policy number
      - signature
      - signature Date (System Date)
2. Finantix Engine will filled the policy number and signature date on the PDF
3. The client will

1. For ILAS
   - Sign on the Signature field
2. For Non-ILAS
   - sign initial on the Initial field
   - sign on the Signature field

7.7.2 Coversheet Barcode

Barcode on application cover sheet will be rendered using a font provided by ML.

7.8 Data retention and application data cleanup process

7.8.1 Data retention rules for application records

Application submitted state	Retention rules
Not submitted	Application record will be purged after two years from creation date (730 days in Day 1 release).
Submitted	Submitted application records will not be purged from Apply system

7.8.2 Purging mechanism (scheduled process)

For application records which are supposed to be purged according to the rules described in chapter 7.8.1 daily scheduled process will be implemented.

7.9 Notification before the application expires (Check if this is a requirnment in Apply…)

POS server will run scheduled job on daily basis to check applications against the expiry conditions. The job execution time will be 8AM HKT.

Push notifications will be sent out for each matching application via APNs as per existing Meet and Refine solution. Application ID(Finantix internal ID) will be included in the notification metadata.

8. e-Disclosure – offline XML rulebook and IF01 message specification

Chapter contains description of off-line rulebook XML file used to render base and reflexive questions configured in MunichRE AIS Rule Designer. Questionnaire rendered after answering by user, will be serialized to the XML format and sent to via ML integration layer to the MunichRE AIS autounderwriting engine.

e-Disclosure related chapters do not cover UI elements design – for details of UI elements please refer to FSD document.

8.1 Off-line XML rulebook

8.1.1 Generic considerations

Rulebook XML file contains Unicode characters encoded in UTF-8 standard.

Each rulebook instance and it's locale is identified by "ruleset" root element in the rulebook XML file:

<ruleset id="68211903-1514-77AC-4034-A5BD65C83CD4" locale="ja">

Value of attribute "id" of ruleset element will be used to link the set of answers given by user with corresponding rulebook. Such mechanism is in place due to the fact that rulebook may be updated over the time (please refer to chapter 8.1.2).

Value of attribute "locale" defines locale name for rulebook - it may be that some country would use rulebook in different languages, then "locale" attribute may be used to present proper language version of e-Disclosure questionnaire to the user.

The ruleset element contains three child elements:

- baseQuestions – contains the base questions in the rulebook – please see the chapter 8.1.3.1)
- categories – contains the rule categories defined in the rulebook (for example medical, occupational, avocational) - it lists the categories and their associated rules, e.g.:

```
<categories>
 <category id="AVOCATIONAL">
      <alias ruleId="R3002">Arnis</alias>
      <alias ruleId="R3003">Boxing</alias>
      <alias ruleId="R3020">Kung Fu</alias>
      <alias ruleId="R3027">Take Kwon do</alias>
 </category>
</categories>
```

-

- rules – the rules element contains the properties for the rules in the rulebook. It contains the reflexive questions and options available to the client. The rules element contains the child

element rule - with the ID attribute denoting the rule code in the rulebook. The reflexiveQuestion element is a child of the rule element.

In the application eSignature section of Hong Kong solution, the questions asked to the customer must be shown, together with the customer's answers, both in English and in Chinese language. To achieve that, all questions must be in synch on multiple language RuleBook xml files, well identified by their unique ids. A post-processing of the published RuleBook files is necessary after they have been imported in order to pair different language questions and enable them to be displayed later together.

8.1.2 Off-line rulebook update

Off-line rulebook will be stored on POS Apply server side in the below file system location:

- /storage/offlineRulebook/<countrycode>/**upload**
- /storage/offlineRulebook/<countrycode>/**backup**

When new e-Disclosure process will be executed latest version of rulebook will be used to generate e-Disclosure questionnaire.

Previous versions of rulebook XML will be stored in the same location and timestamp will be used to distinguish the latest one to be used by process – housekeeping will be in scope of ML.

Below file naming convention will be used:

<country code>_<locale>_RULEBOOK_ yyyyMMdd>_<HHmmssSSS>.xml

e.g. HK_HK_RULEBOOK_20151225_110054111.xml

Locale will allow maintaining different language versions of rulebook for regional solution in cases where more than a language is available.

Rulebook XML files belonging to different languages must be exported and published always together to POS Apply server through the import described above.

8.1.3 Definition of base and reflexive questions

The are two main types of question which may be configured in XML rulebook:

- base questions – top level questions in the most cases to be answered with Yes / No answer leading potentially to the reflexive questions,
- reflexive questions – associated to the base questions via rules definition and rendered as per previous given answer basis – reflexive questions will be rendered in the same UI logical page* as the base question from which those were derived

Generic rendering rules:

- as a base rule, rendering is performed as per order of elements appearance in rulebook XML. (there is an exception for Heading elements modified with special filter – please refer to example **Error! Reference source not found.** and chapter **Error! Reference source not found.**)
- single base questions not included in sections will be rendered in separate UI logical page,
- sections containing multiple base questions will be rendered in separate UI logical page – one section will be rendered in on UI logical page
- sub-questions linked to conditional parent base questions will be rendered in the same UI logical page

* “UI logical page” is single page within the UI which may exceed physical display height

UI elements needs to be designed for all the types of base and reflexive questions to let the rendering engine present the e-Disclosure questionnaire correctly.

Types of the questions will be defined in subsequent chapters.

Order of base questions rendering is as per order they appear in rulebook XML.

Reflexive questions are rendered as per rule selected in base questions and answers given to subsequent reflexive questions.

Following chapters cover details of all handled types of questions.

Example:

UI logical Page 1 – contains base question not enclosed in section

Base question 1 (standard question linked to rule deriving reflexive questions)

|----Reflexive question 1

|----Reflexive question 2

|----Reflexive question 3

UI logical Page 2 – contains section consisting of two base questions

Section 1

Base question 1 (standard question linked to rule deriving reflexive questions)

|----Reflexive question 1

|----Reflexive question 2

|----Reflexive question 3

Base question 2 (standard question linked to rule deriving reflexive questions)

|----Reflexive question 1

|----Reflexive question 2

|----Reflexive question 3

End of Section 1

8.1.3.1 Base question definition and types

The base question types which will be handled by Apply system basing on offline rulebook XML file are:

- Standard – question which will be asked by application and linked to the rules basing on the answer given,
- Conditional – question which is linked to other base question(s) which will be presented to the user basing on the answer given,
- Heading – rulebook element used to give the descriptive introduction to the base question / group of base questions (section),

- Section – rulebook element used to group the questions

Each base question needs to have defined answer type which is used to trigger the further rules derived by answering the base question. Answer type may be defined as per below list:

- Trigger On Yes* – when base question is answered with Yes, the linked rule (for reflexive questions) or linked base question will be triggered
- Trigger On No* - when base question is answered with No, the linked rule (for reflexive questions) or linked base question will be triggered,

  *Questions with “Trigger On No” / “Trigger On Yes” answer type should be rendered with mechanism which allows only to give Yes / No type of answer (e.g. “Yes” and “No” buttons).
- Enter Details – the linked rule (for reflexive questions) or linked base question will be triggered when user will enter details

Questions are listed in the XML file in the same order they have been defined by the user on the rulebook editor.

In case of Standard question when rule is triggered basing on the answer type parameter, rules may be linked as per one of the patterns listed below:

- Search – the base question is linked to all the rules in specified category (category is a list of the rules defined outside the base question and therefore may be used by many base questions) – when the Search type is triggered as per base question settings, rendering engine should display UI element which allows to select the values from the category set and provide the autocomplete search feature (e.g. combo-box with autocomplete search feature)
- Specified Rule - the base question is linked to the single rule – for Specified Rule type of question there is no specific UI rendering needed when rule is triggered as per base question settings,
- Pick List Items – the base question is linked to the pre-defined list of rules (definition of the list is strictly linked with particular base question definition and is not available for other base questions) – when the Search type is triggered as per base question settings, rendering engine should display UI element which allows to select the values from the rule list defined at the level of base question (e.g. combo-box or selection list)
- None – the base question is not linked to the rule – usage limited to Conditional or Heading type of base questions

8.1.3.1.1 Example – Two Base Question enclosed in a section

```
<section id="Section 1">
 <baseQuestion id="BQQ_1">
      <category>SPECIFIED</category>
      <answerType>TRIGGER_YES</answerType>
      <searchType>SPECIFIED</searchType>
      <prompt>Please select “Yes” to trigger Rule 1</prompt>
      <rule ruleId="R1">Rule 1</rule>
 </baseQuestion>

 <baseQuestion id="BQQ_2">
      <category>SPECIFIED</category>
      <answerType>TRIGGER_YES</answerType>
      <searchType>SPECIFIED</searchType>
      <prompt>Please select “Yes” to trigger Rule 2</prompt>
```

```
        <rule ruleId="R2">Rule 2</rule>
    </baseQuestion>
</section>
```

UI behavior:

Base questions will be rendered in the same UI logical page.

8.1.3.1.2 Example - Base Question of type Search, where rule is triggered on “Yes” selection and category contains three rules to select

```
<baseQuestion id="BQQ_1">
 <category>CAT1</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>SEARCH</searchType>
 <prompt>Do you want to select rule from category ?</prompt>
</baseQuestion>

<categories>
 <category id="CAT1">
      <alias ruleId="R1">Rule 1</alias>
      <alias ruleId="R2">Rule 2</alias>
      <alias tuleId=”R3”>Rule 3</alias>
 </category>
</categories>
```

UI behavior:

1. prompt of base question BQQ_1 is displayed and user can answer Yes or No:
   Do you want to select rule from category ? <Yes>/<No>
2. User select answer “Yes”
3. As question is configured as TRIGGER_YES, UI element which allows to select value from the list defined as per CAT1 category is displayed:

   Rule 1

   Rule 2

   Rule 3
4. User can select ONE value form the category with support of autocomplete feature.
5. If user select the rule, then reflexive questions according to the selected rule will be displayed.

8.1.3.1.3 Example - Base Question of type Specified Rule, where rule is triggered on “Yes” selection

```
<baseQuestion id="BQQ_2">
 <category>SPECIFIED</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>SPECIFIED</searchType>
 <prompt>Please select: “Yes” to trigger Rule 4, “No” to move to
the next base question</prompt>
 <rule ruleId="R4">Rule 4</rule>
</baseQuestion>
```

*<category> is used only for categorization of the question – can be ignored by rendering engine

UI behavior:

1. <prompt> of base question BQQ_2 is displayed and user can answer Yes or No:

Please select: “Yes” to trigger Rule 4, “No” to move to the next base question <Yes>/<No>

2. User select answer “Yes”,
3. As question is configured as TRIGGER_YES, reflexive questions defined by Rule 4 will be displayed.

8.1.3.1.4 Example - Base Question of type Pick List Items, where rule is triggered on “Yes” selection

```
<baseQuestion id="BQQ_3">
 <category>MEDICALCHECK</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>PICK_LIST</searchType>
 <prompt>Do you want to select rule from the pick list ?</prompt>
 <rule ruleId="R5">Rule 5</rule>
 <rule ruleId="R6">Rule 6</rule>
 <rule ruleId="R7">Rule 7</rule>
</baseQuestion>
```

*<category> is used for categorization of the question – may be ignored or handled by the rendering engine with hardcoded logic

UI behavior:

1. prompt of base question BQQ_3 is displayed and user can answer Yes or No:
2. Do you want to select rule from the pick list ?</<Yes>/<No>
3. User select answer “Yes”, UI element is displayed which allows to select value from the list defined as per rule elements in base question definition:

   Rule 5

   Rule 6

   Rule 7
4. User can select ONE rule from the list.
5. Reflexive questions associated to the selected rule will be displayed.

8.1.3.1.5 Example - Base Question of type None, where rule is triggered on “Yes” selection

```
<baseQuestion id="BQQ_4">
 <category>MEDICALCHECK</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>NONE</searchType>
 <prompt>Do you want to answer linked base question ?</prompt>
 <conditionalParent>true</conditionalParent>
</baseQuestion>

<baseQuestion id="BQQ_4_1">
 <category>MEDICALCHECK</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>PICK_LIST</searchType>
 <prompt>Do you want to select rule from the pick list ?</prompt>
 <filter name="session/CONDITIONAL_BASE_QUESTION">
`     <filterValue>BQQ_4</filterValue>
 </filter>
 <rule ruleId="R8">Rule 8</rule>
 <rule ruleId="R9">Rule 9</rule>
 <rule ruleId="R10">Rule 10</rule>
</baseQuestion>
```

*<category> is used for categorization of the question – may be ignored or handled by the rendering engine with hardcoded logic

Remarks:

1. when type is NONE, question is marked as conditional parent (element <condtionalParent> contains value true),
2. question BQQ_4_1 which is linked to the parent base question BQQ_4 contains <filter> element defined with attribute name set to “session/CONDITIONAL_BASE_QUESITON” and inner <filterValue> element with value set to BQQ_4 (for details about filtering and <filter> elements please refer to chapter 8.1.4) – it is used to determine that question BQQ_4_1 will be rendered ONLY when question BQQ_4 has been answered as per trigger type (Yes in presented case).

   Generic rule is that rendering engine in case if it will encounter NONE type base question with element <conditionalParent> set to “true” should find the linked base question by scanning filter elements of all base questions to look for filter with attribute name equal to “session/CONDITIONAL_BASE_QUESITON” and <filterValue> set to identifier of NONE type base question

   In case if question is of type NONE and doesn’t have <conditionalParent> element with value “true” flow should move to the next base question regardless the answer selected.

UI behavior:

1. prompt of base question BQQ_3 is displayed and user can answer Yes or No:
   Do you want to answer linked base question ?</<Yes>/<No>

2. User select answer “Yes”, base question “BQQ_4_1” is rendered (in presented case it is Pick List Items type)

8.1.3.2 Reflexive question definition and types

Reflexive questions are rendered as per rules defined in rulebook XML file and derived from the answers given to base questions. Subsequent reflexive questions can be rendered as an outcome of the answers of preceding reflexive questions (so called: drilldown questions).

The reflexive question types that will be handled by Apply system basing on offline rulebook XML file are:

- Boolean – question type which can take boolean value as an answer (Yes/No, true/false) and drive other reflexive question according to the selected value – UI element design should allow to select only from the set of two values
- Option – question type which allows to select the answer form the list and drive the other reflexive questions according to the selected value (it is a number which is an index of the selected value) – UI element design should allow to select value from pre-defined list (e.g. combo-box)
- Range – question type which allows the user to type a numeric value and classify the value under the range – UI element design should not disclose defined ranges to the user and allow only typing data in (e.g. numeric input field)
- Free Text – question type which allows the user to type the answer as a free text - UI element design should allow to enter free text and provide mechanism to confirm when data has been entered

8.1.3.2.1 Example - Reflexive Question of type Boolean

```
<rule id="R1">
 <reflexiveQuestion id="RQ1">
      <type>BOOLEAN</type>
      <prompt>Answer Yes for index 0 path, No for index 1
path</prompt>
      <values>
           <value index="0">Yes</value>
           <value index="1">No</value>
      </values>
      <choices>
           <reflexiveQuestion index="0" id="RQ10">
                <type>BOOLEAN</type>
                <prompt>Index 0 path question - no further reflexive
question asked</prompt>
                <values>
                     <value index="0">Yes</value>
                     <value index="1">No</value>
                     </values>
                     <choices>
                     </choices>
           </reflexiveQuestion>
           <reflexiveQuestion index="1" id="RQ11">
                <type>BOOLEAN</type>
                <prompt>Index 1 path question - no further reflexive
```

```
question asked</prompt>
                <values>
                     <value index="0">Yes</value>
                     <value index="1">No</value>
                     </values>
                     <choices>
                     </choices>
           </reflexiveQuestion>
      </choices>
 </reflexiveQuestion>
</rule>
```

UI behaviour:

1. when rule R1 be triggered as an output of base question, reflexive question RQ1 will be rendered as a Boolean type – prompt of that question will be displayed:

   Answer Yes for index 0 path, No for index 1 path

2. when user will select Yes/No he will be presented following reflexive question assigned to the index:

   - Selected Yes, Boolean question RQ10 will be rendered with prompt:

     Index 0 path question - no further reflexive questions asked

   - Selected Yes, Boolean question RQ11 will be rendered with prompt:

     Index 1 path question - no further reflexive questions asked

8.1.3.2.2 Example - Reflexive Question of type Option

```
<rule id="R2">
 <reflexiveQuestion id="RQ2">
      <type>OPTION</type>
      <prompt>Please select option</prompt>
      <values>
           <value index="0">Option - index 0</value>
           <value index="1">Option - index 1</value>
           <value index="1">Option - index 2</value>
      </values>
      <choices>
           <reflexiveQuestion index="0" id="RQ20">
           ...
           </reflexiveQuestion>
           <reflexiveQuestion index="1" id="RQ21">
           ...
           </reflexiveQuestion>
           <reflexiveQuestion index="2" id="RQ22">
           ...
```

```
            </reflexiveQuestion>
        </choices>
    </reflexiveQuestion>
</rule>
```

UI behavior:

1. when rule R2 be triggered as an output of base question, reflexive question RQ2 will be rendered as a Option type – prompt of that question will be displayed:
   Please select option

2. User select answer "Yes", UI element is displayed which allows to select value from the list defined as per rule elements in base question definition:
   Option – index 0
   Option – index 1
   Option – index 2

3. User can select ONE option from the list.
4. Reflexive question, associated to the index of option selected will be rendered:
   "Option – index 0" –> question RQ20
   "Option – index 1" –> question RQ21
   "Option – index 2" –> question RQ22

8.1.3.2.3 Example - Reflexive Question of type Range

```
<rule id="R2">
 <reflexiveQuestion id="RQ3">
      <type>RANGE</type>
      <prompt>Please type the number</prompt>
      <values>
           <value index="0">0 </value>
           <value index="1">50</value>
           <value index="2">100</value>
      </values>
      <choices>
           <reflexiveQuestion index="0" id="RQ30">
           ...
           </reflexiveQuestion>
           <reflexiveQuestion index="1" id="RQ31">
           ...
           </reflexiveQuestion>
           <reflexiveQuestion index="2" id="RQ32">
           ...
           </reflexiveQuestion>
      </choices>
```

```
  </reflexiveQuestion>
 </rule>
```

UI behavior:

1. when rule R3 be triggered as an output of base question, reflexive question RQ3 will be rendered as a Range type – prompt of that question will be displayed:
   Please type the number

2. User types the number and below rules will be applied to derive the range to which the number (n) belongs:
   n >= 0 and n < 50 -> index 0 will be assigned
   n >=50 and n < 100 -> index 1 will be assigned
   n >=100 -> index 2 will be assigned

3. Reflexive question, associated to the index calculated will be displayed:
   index 0 –> question RQ30
   index 1 –> question RQ31
   index 2 –> question RQ32

8.1.3.2.4 Example - Reflexive Question of type Free Text and “Auto Select”

```
<rule id="R4">
 <reflexiveQuestion index="0" id="RQ4">
 <type>FREE_TEXT</type>
 <prompt>Please enter details and confirm</prompt>
      <values>
           <value index="0">Auto Select</value>
      </values>
      <choices>
      <reflexiveQuestion index="0" id="RQ41">
           <type>FREE_TEXT</type>
           <prompt>Please enter details - no further reflexive
question be asked </prompt>
           <values>
                <value index="0">Auto Select</value>
           </values>
           <choices>
           </choices>
      </reflexiveQuestion>
 </reflexiveQuestion>
</rule>
```

UI behavior:

1. when rule R4 be triggered as an output of base question, reflexive question RQ4 will be rendered as a Free Text type question and prompt will be displayed:
   Please enter details and confirm
2. User enters details and confirms that he finished
3. As the question has only one value defined with index 0 this value is used as an outcome to derive further reflexive question which in this case will be RQ41
4. Question RQ41 will be rendered as Free Text type question and prompt will be displayed:
   Please enter details - no further questions be asked

5. User enters details and confirms that he finished – flow ends and no further reflexive questions be asked

8.1.4 Filtering of the base questions and headings

Offline rulebook XML provides a way to dynamically present e-Disclosure questionnaire to the user showing only relevant questions and headings – for purpose of this document name “standard filter” will be used for filters with such behavior.

List of the “standard filters” (sourced from the application fields) is presented in below table,.

Mapping of the filter values is defined in IMS document in tab “auwSubmission WS” (table Case Data - Entity type "life" - List of attributes).

Name of filter	Filter value	Filter description
life/APPLICATION_TYPE	FULL_APP FULL_APP_TRAD SIO_PA SIO	No logic to be implemented, always E
life/RISK_TYPES	Based on the base Question will have different values	
life/DI_TYPE	DI PIB_HKD30K_ABOVE	
life/PRC_RESIDENT_FLAG	Y	Y mapped to true
life/AGE_BAND	16_17 OVER_65 10_15 18_65	
life/GENDER	M, F	M for male, F for female – please refer to IMS tab auwSubmission WS for details of mapping
life/MEDICAL_INDEMNITY_BENEFIT_FLAG	Y	Y mapped to true

Filtering mechanism will also be used to cause specific, non-standard behavior of rendering engine (e.g. Pick List and Search, headings, hiding of dummy questions which must be included in rulebook but hidden for user) – for purpose of this document name “special filters” will be used.

Special filters defined for e-Disclosure process:

Name of filter	Filter value	Filter description
case/POS_FORM_FILTER	PL_AND_SEARCH	Used in base question of type Pick List to specify rendering as Pick List And Search– please see chapter **Error! Reference source not found.**
case/POS_FORM_FILTER	UNDER_BASE_QUESTION	Used in heading to specify that particular heading needs to be rendered under

		prompt of following base question – please see chapter **Error! Reference source not found.**
case/HIDE_EXPORT_SEARCH_BQ	N	Used to hide the “dummy” base question needed by MunichRE AUW engine – not to be asked to the customer. If value is provided as “N” question needs to be hidden in UI.
session/CONDITIONAL_BASE_QUESTION	Ref to **Error! Reference source not found.**	Technical use

Filter within Base Question / Heading is defined in <filter> XML element with attribute name and one or more inner filterValue elements:

```
<baseQuestion id=”BQQ_A”>
        ...
  <filter name=”FILTER1”>
        <filterValue>VALUE_1</filterValue>
        <filterValue>VALUE_2</filterValue>
        <filterValue>VALUE_3</filterValue>
  </filter>
  <filter name="FILTER2">
        <filterValue>VALUE_1</filterValue>
  </filter>
        ...
</baseQuestion>
```

Above definition means that base question will be shown only when:

FILTER1 parameter value will be

VALUE_1 or VALUE_2 or VALUE_3

and

FILTER2 parameter value will be

VALUE_1

Example:

Qustion will be shown only when:

GENDER = FEMALE and

MEDICAL_INDEMNITY_BENEFIT_FLAG = Y and

RISK_TYPES = DI_90 and

APPLICATION_TYPE = FULL_APP

```
<baseQuestion id="BQQ_FILTERS">
        <category>SPECIFIED</category>
        <answerType>TRIGGER_YES</answerType>
        <searchType>SPECIFIED</searchType>
        <prompt>Shown only when parameter names match values defined in the filters </prompt>
    <filter name="life/MEDICAL_INDEMNITY_BENEFIT_FLAG">
      <filterValue>Y</filterValue>
    </filter>
        <filter name="life/GENDER">
                <filterValue>FEMALE</filterValue>
        </filter>
        <filter name="life/RISK_TYPES">
      <filterValue>CANCER</filterValue>
    </filter>
        <filter name="life/APPLICATION_TYPE”>
                <filterValue>FULL_APP</filterValue>
        </filter>
        <rule ruleId="RULE_A">Rule</rule>
</baseQuestion>
```

8.1.5 Reflexive questions - pre-selection of the values from the source

Rulebook allows to preselect values for reflexive questions from the application record or from the disclosure given by customer (question already answered by customer). Definition of reflexive question will contain element <source> specifying the value to which answer should be serialized to IF01. Reflexive question with the <source> element will not be displayed to the user.

Case data	Reflexive question type	Description
life/AGE	RANGE	Basing on the numeric value, range index should be preselected – for mapping of filter please refer to IMS tab auwSubmision WS
life/AGE_IN_MONTH		
life/BMI	RANGE	Basing on the numeric value, range index should be preselected – for mapping of filter please refer to IMS tab auwSubmision WS
life/SMOKER_STATUS	OPTION	Basing on the code index should be preselected
life/GENDER	OPTION	Basing on the code index should be preselected
life/ID_DOC_TYPE	OPTION	<value index="0" code="PERMANENT_HKID">Permanent HKID</value> <value index="1" code="NON_PERMANENT_HKID">Non-permanent HKID</value> <value index="2" code="PRC_ID">PRC ID</value> <value index="3" code="PASSPORT">Passport</value> <value index="4" code="OTHER_ID">Other ID</value>
life/MMC_APPLY_FLA	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/MPC_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/MEC_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/MANUMASTER_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS

life/MANUGUARD_DOMESTIC_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/MANUGUARD_FOREIGN_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/MANUSHINE_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/CANCER_DOMESTIC_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/CANCER_FOREIGN_APPLY_FLAG	BOOLEAN	Value true mapped to index “0”, false to index “1”, for mapping details please refer to IMS tab auwSubmission WS
life/ALCOHOL_QUANTITY	RANGE	Basing on the numeric value, range index should be preselected – for mapping of filter please refer to IMS tab auwSubmision WS
life/DAILY_CIGARETTE_QUANTITY	OPTION	<value index="0" code="UNDER_5_CPD">5 cigarettes or below</value> <value index="1" code="UNDER_30_CPD">Up to 30 cigarettes</value> <value index="2" code="OVER_30_CPD">More than 30 cigarettes</value> <value index="3" code="NEVER_SMOKE">Never smoke</value>

8.1.5.1 BOOLEAN type reflexive question with <source> element

```
<reflexiveQuestion index="1" id="R_1">
<type>BOOLEAN</type>
<prompt>Prompt will not be displayed</prompt>
<source>life/HBP_DISCLOSURE_FLAG</source>
    <values>
        <value index="0">Yes</value>
        <value index="1">No</value>
    </values>
<choices>
    <reflexiveQuestion index="0" id="R_1_0">
        ...
    </reflexiveQuestion>
    <reflexiveQuestion index="1" id="R_1_1">
        ...
```

```
                </reflexiveQuestion>
            </choices>
        </reflexiveQuestion>
```

UI behavior:

1. reflexive question R_1 has been triggered due to the previous choices
2. as reflexive question R_1 contains source element it will not be rendered
3. engine will check the life/HBP_DISCLOSURE_FLG case data value (according to the mapping)
4. according to the value of life/HBP_DISCLOSURE_FLG it will trigger reflexive question R_1_0 in case if life/HBP_DISCLOSURE_FLG will be evaluated to “true” or R_1_1 in case if life/HBP_DISCLOSURE_FLG will be evaluated to “false”
5. question R_1_0 or R_1_1 will be rendered accordingly

8.1.5.2 OPTION type reflexive question with <source> element

```
            <reflexiveQuestion index="1" id="R_1">
            <type>RANGE</type>
            <prompt>Prompt will not be displayed</prompt>
            <source>life/SMOKER_STATUS</source>
                <values>
                    <value index="0" code="NON_SMOKER">Non smoker</value>
                    <value index="1" code="SMOKER">Smoker</value>
                </values>
            <choices>
                <reflexiveQuestion index="0" id="R_1_0">
                    ...
                </reflexiveQuestion>
                <reflexiveQuestion index="1" id="R_1_1">
                    ...
                </reflexiveQuestion>
            </choices>
        </reflexiveQuestion>
```

UI behavior:

1. reflexive question R_1 has been triggered due to the previous choices
2. as reflexive question R_1 contains source element it will not be rendered
3. engine will check the life/SMOKER_STATUS case data value (according to the mapping)
4. according to the value of life/SMOKER_STATUS it will trigger reflexive question R_1_0 in case if life/SMOKER_STATUS will be evaluated to NON_SMOKER or R_1_1 in case if life/SMOKER_STATUS will be evaluated to SMOKER
5. question R_1_0 or R_1_1 will be rendered accordingly

8.1.5.3 RANGE type reflexive question with <source> element

```
<reflexiveQuestion index="1" id="R_1">
    <type>RANGE</type>
    <prompt>Prompt will not be displayed</prompt>
    <source>life/AGE</source>
    <values>
        <value index="0">0</value>
        <value index="1">60</value>
    </values>
    <choices>
        <reflexiveQuestion index="0" id="R_1_0">
            ...
        </reflexiveQuestion>
        <reflexiveQuestion index="1" id="R_1_1">
            ...
        </reflexiveQuestion>
    </choices>
</reflexiveQuestion>
```

UI behavior:

1. reflexive question R_1 has been triggered due to the previous choices
2. as reflexive question R_1 contains source element it will not be rendered
3. engine will check the life/AGE case data value (according to the mapping)
4. according to the value of life/AGE it will trigger reflexive question R_1_0 (if life/AGE > 0 and life/AGE <= 60) or R_1_1 (if life/AGE >=60).
5. question R_1_0 or R_1_1 will be rendered accordily

8.1.6 Questions rendering flow.

In below diagram BQ and RQ for Base Question and Reflexive Question accordingly.

In a diagram special feature for Search and Pick List Items type questions is depicted - user should have possibility to decide if he wants to answer question multiple times e.g. to disclose more than one medical conditions he may have.

IF01 response generation is not a part of diagram – it is limited only to questions rendering algorithm.

Diagram does not illustrate filtering of questions and does not include grouping concept of base questions – assumed filtering is applied.

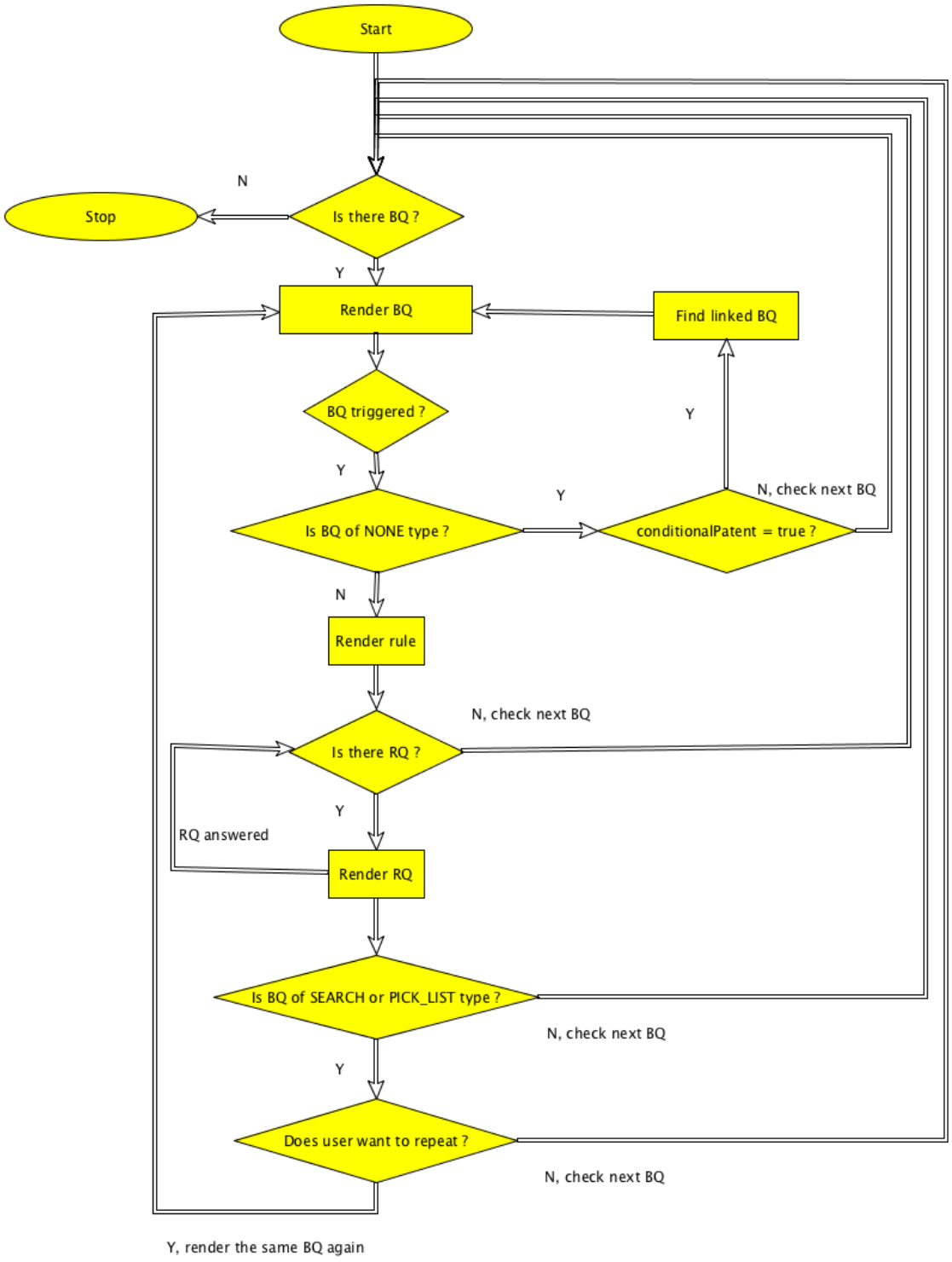


Figure 26: Questions rendering flow

8.2 IF01 message composition – used as a part of auwSubmissionWS call

As a result of e-Disclosure questions answered request to Munich RE AUW engine needs to be prepared and call to oneShotWS exposed by ML integration layer needs to be executed.

High level structure of IF01 message:

- case properties – e.g. ruleset id – used to identify rulebook against which e-Disclosure was answered,
- case data – data related to the particular underwriting case – combined from proposal system, POS and ML backend systems (some data would be supplemented by ML integration layer) – in particular there will be one attribute in case data – RISK_TYPE, which would need special treatment as it could only be mapped via master data provided by ML through plan code provided in PA Web (please refer to chapter … for details of master data import) – detailed mapping of the case data will be covered by IMS document
- underwriting data – data generated by POS system as a result of answers given by user to e-Disclosure questionnaire, this section contains two sub-sections:
    - base question – list of the answers given to base questions
    - disclosures – list of conditions disclosed together with information about base questions linked

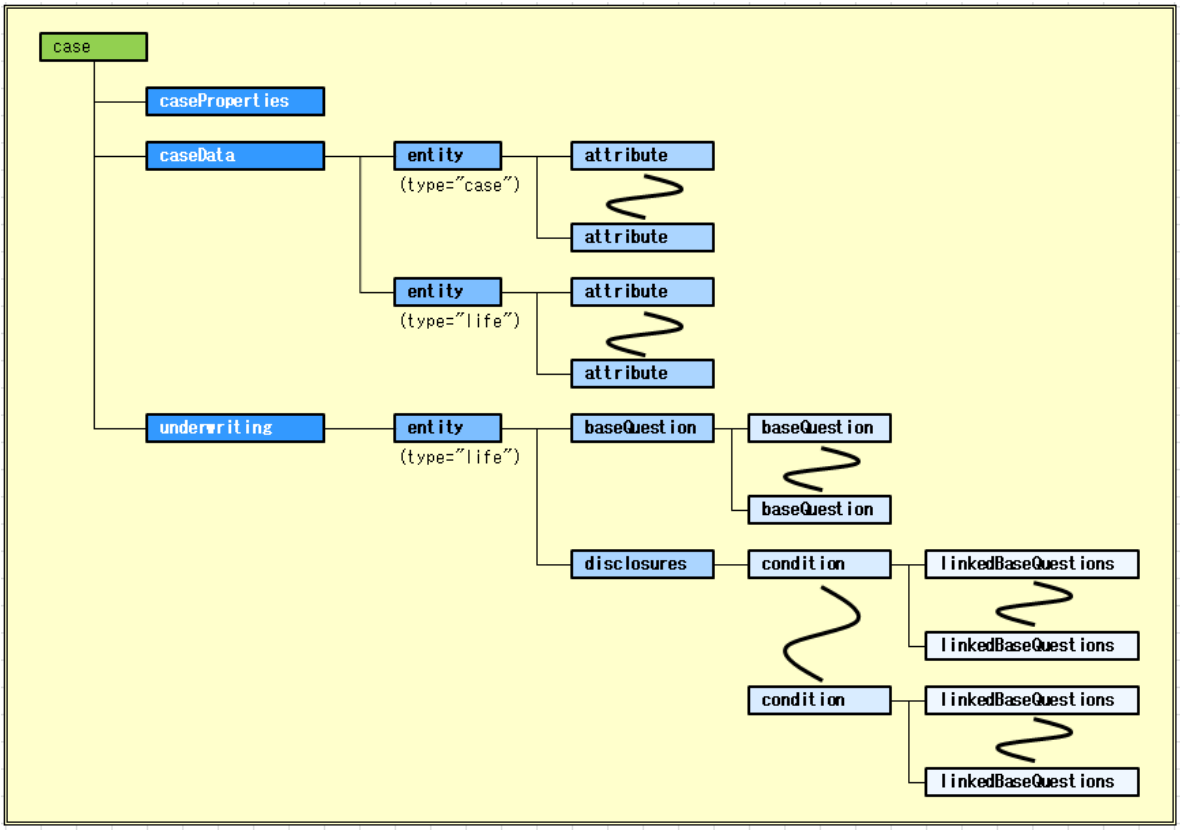


Figure 27: Structure of IF01 message - for more details please refer to IF01SAMPLE

Attributes of case properties and case data sections are not dependent from the questions asked and can be mapped against the POS data dictionary – mapping will be defined in IMS document.

This chapter will be focused on underwriting part of IF01 request.

General rule is that underwriting section contains ONLY base questions, which were asked in e-Disclosure according to the filtering rules applied – those are listed in <baseQuestions> section. In case if there were reflexive questions asked as well, section <disclosures> will list disclosed <conditions> together with a link to corresponding base questions.

<disclosure> section may not be included in IF01 message in case if no reflexive questions were asked (e.g. due to the fact that all base question were answered against the trigger answer type).

Pay attention that the “occupation” information that is collected as base question disclosure in Hong Kong Apply solution, must be copied also as part of case data information in order to satisfy the regional requirement for business analytics.

8.2.1 Example – all base questions answered “No”

In the below example there is no <diclosures> section included as all the base questions were answered “No” and in rulebook those were configured as “Trigger On Yes”.

Example is an extract of underwriting section from Case2AllBaseQuestionsNo.xml.

```
<underwriting>
     <entity type="life" name="1">
          <baseQuestions>
               <baseQuestion >
                    <code>BQQ_1</cFollowode>
                    <answer>no</answer>
               </baseQuestion>
               <baseQuestion >
                    <code>BQQ_2</code>
                    <answer>no</answer>
               </baseQuestion>
               <baseQuestion >
                    <code>BQQ_5</code>
                    <answer>no</answer>
               </baseQuestion>
               <baseQuestion >
                    <code>BQQ_7</code>
                    <answer>no</answer>
               </baseQuestion>
               <baseQuestion >
                    <code>BQ_GROUPING1</code>
                    <answer>no</answer>
               </baseQuestion>
               <baseQuestion >
                    <code>BQ_GROUPING2</code>
                    <answer>no</answer>
               </baseQuestion>
          </baseQuestions>
     </entity>
</underwriting>
```

8.2.2 Example – base question of Search type triggers the empty rule without rendering reflexive questions

In below example user is presented question BQQ_1 of Search type. He answers “Yes” and rule selection UI is triggered. User selects entry assigned to the rule R3008 which is a final rule without any reflexive questions defined.

As a result in the IF01 section <disclosures> which contains one condition disclosed is included. Condition contains <alias> element with value set to value of <alias> element of category with ruleid R3008.

<linkedBaseQuestions> element inside <condition> element shows the link to base question BQQ_1 which triggered the rule R3008 for selection.

Example created basing on Case3SearchQuestionAnsweredYesAndAliasLinked.xml example.

```
Rulebook snippet:
<!–category definition -->
 <category id="DISEASE">
      <alias ruleId="R3008">R3008-alias</alias>
      <alias ruleId="R3013">R3013-alias</alias>
 </category>

<!–rule definition -->
<rule id="R3008">
</rule>

<!–base question definition -->
 <baseQuestion id="BQQ_1">
      <category>DISEASE</category>
      <answerType>TRIGGER_YES</answerType>
      <searchType>SEARCH</searchType>
      <prompt>Search question prompt</prompt>
      …
 </baseQuestion>

IF01 genreated:
 <underwriting>
      <entity type="life" name="1">
           <baseQuestions>
                <baseQuestion >
                     <code>BQQ_1</code>
                     <answer>yes</answer>
                </baseQuestion>
                ...
           </baseQuestions>
           <disclosures>
                <condition>
                     <alias> R3008-alias </alias>
```

```
                    <linkedBaseQuestions>
                        <baseQuestionRef>BQQ_1</baseQuestionRef>
                    </linkedBaseQuestions>
                </condition>
            </disclosures>
        </entity>
    </underwriting>
```

8.2.3 Example – base question of type Specified with reflexive drill-down questions of type OPTION, RANGE and BOOLEAN

In below example user is presented question BQQ_1 of Specified type.

1. User answers “Yes” to base question BQQ_1 and rule R1 is triggered.
2. As per result of user action first reflexive question R1_1 is rendered – this is question of type Option and it contains three possible values to be selected:

   Option 0

   Option 1

   Option 2
3. User selects Option 0 (it has index=”0” in definition) and according to that selection reflexive question R_1_0 is rendered (choice made per index=”0” value),
4. Question contains numeric input field to specify value – user enters number 60 (50 <= 60 < 100, so it falls under range with index=”1”)
5. As per index=”1” question R_1_0_1 of type Boolean is rendered
6. User selects answer “Yes” (index = “0”) and according to that selection reflexive question R1_1_0_1_0
7. As the base question BQQ_1 is of type PICK_LIST user is given an option to answer it again – see chapter 8.1.6
8. User answers “Yes” to base question BQQ_1
9. As per result of user action reflexive question R2_1 is rendered – this is question of type Option and it contains three possible values to be selected:

   Option 0

   Option 1

   Option 2
10. User selects Option 2 (it has index = “2” in definition) and rule ends as no choices defined.

```
Rulebook snippet:
<baseQuestion id="BQQ_1">
 <category>CATEGORY</category>
 <answerType>TRIGGER_YES</answerType>
 <searchType>PICK_LIST</searchType>
 <prompt>Please select: “Yes” to trigger Rule 1, “No” to move to the
next base question</prompt>
 <rule ruleId="R1">Rule 1</rule>
```

```
 <rule ruleId="R2">Rule 2</rule>
</baseQuestion>

<rule id="R1">
 <reflexiveQuestion id="R1_1">
      <type>OPTION</type>
      <prompt>Please select option</prompt>
      <values>
           <value index="0">Option 0</value>
           <value index="1">Option 1</value>
           <value index="1">Option 2</value>
      </values>
      <choices>
           <reflexiveQuestion index="0" id="R1_1_0">
                <type>OPTION</type>
                <prompt>Please input the number</prompt>
                <values>
                     <value index="0">0</value>
                     <value index="1">50</value>
                     <value index="2">100</value>
                </values>
                <choices>
                     <reflexiveQuestion index="1" id="R1_1_0_1">
                          <type>BOOLEAN</type>
                          <prompt>Do you like the number entered
?</prompt>
                          <values>
                               <value index="0">Yes</value> <-- false
                               <value index="1">No</value>  <-- true
                          </values>
                          <choices>
                               <reflexiveQuestion index="0"
id="R1_1_0_1_0">
                                    <type>FREE_TEXT</type>
                                    <prompt>Please alaborate why do you
like the entered?</prompt>
                                    <values>
                                         <value index="0">Auto
Select</value>
                                    </values>
                                    <choices>
                                         <!—rule ends here -->
                                    </choices>
                               </reflexiveQuestion>
                          </choices>
                     </reflexiveQuestion>
                </choices>
```

```
            </reflexiveQuestion>
      </choices>
 </reflexiveQuestion>
</rule>

<rule id="R2">
 <reflexiveQuestion id="R2_1">
      <type>OPTION</type>
      <prompt>Please select option</prompt>
      <values>
            <value index="0">Option 0</value>
            <value index="1">Option 1</value>
            <value index="2">Option 2</value>
      </values>
      <choices>
            <!-- rule ends here -->
      </choices>
</rule>

IF01 generated:
<underwriting>
 <entity type="life" name="1">
      <baseQuestions>
            <code>BQQ_1</code>
            <answer>yes</answer>
      </baseQuestions>
      <disclosures>
   <condition>
                                                <alias>Rule 1</alias>

<linkedBaseQuestions>

<baseQuestionRef>BQQ_1</baseQuestionRef>

</linkedBaseQuestions>

<decisionPath>

<questionStep>

<reflexiveQuestion>
            <!—answer to question R1_1_0 -->
            <answerValue>0</answerValue>

</reflexiveQuestion>

</questionStep>
```

```
<questionStep>

<reflexiveQuestion>
               <!—answer to question R1_1_0_1 -->
               <answerValue>1</answerValue>
 </reflexiveQuestion>

</questionStep>

<questionStep>
<reflexiveQuestion>
<answerValue>false</answerValue>
</reflexiveQuestion>

</questionStep>
           </decisionPath>
</condition>
<condition>
    <alias>Rule 2</alias>

<linkedBaseQuestions>

<baseQuestionRef>BQ_1</baseQuestionRef>

</linkedBaseQuestions>

<decisionPath>

<questionStep>

<reflexiveQuestion>

<answerValue>2</answerValue>

</reflexiveQuestion>

</questionStep>
                    <questionStep>
                         <reflexiveQuestion>
                              <answerValue>TEXT TYPED BY
USER</answerValue>
                         </reflexiveQuestion>
                    </questionStep>

</decisionPath>
                                        </condition>
                              </disclosures>
            </entity>
</underwriting>
```

8.3 Supplementary documents

This appendix collects the supplementary documentation provided by ML for SAS document creation purposes – it may be referred by the reader in case for additional information.

Reference	Document name	Description
AISOFFLINE	AIS 4.11 Offline Risk Assessment.pdf	AIS offline risk assessment documentation
RULEBOOK	Offline.MANULIFE_JP.v3.xml	Sample XML rulebook
IF01SAMPLE	AIS_IF01_XML Sample(J) - v1.0_20151126.doc	Document describing sample IF01 XML
IF01XMLSAMPLES	Case2AllBaseQuestionsNo.xml Case3SearchQuestionAnsweredYesAndAliasLinked.xml Case4PickListBaseQuestionAnsweredAndTwoConditions Linked.xml	Sample XML files with IF01 message
TAS	Technical Architecture Specification	Regional POS Solution Architecture document owned by ML

9. Payments

9.1 Common integration approach

The system is supporting electronic activity through E-Payment gateway (Wildcard, provided by ML). The E-Payment service is happing inside the E-Payment that E-Pos will redirect the user to the screen with some required information. Once it completed, E-Payment gateway will provide the status to the system and the status will be redirected to ML. The following diagram will show the integration between the Epos app, Epos server, ML integration layer, and e-payment gateway. If the user selected pay by e-payment, the user cannot go to the next step till the e-payment successfully completed.

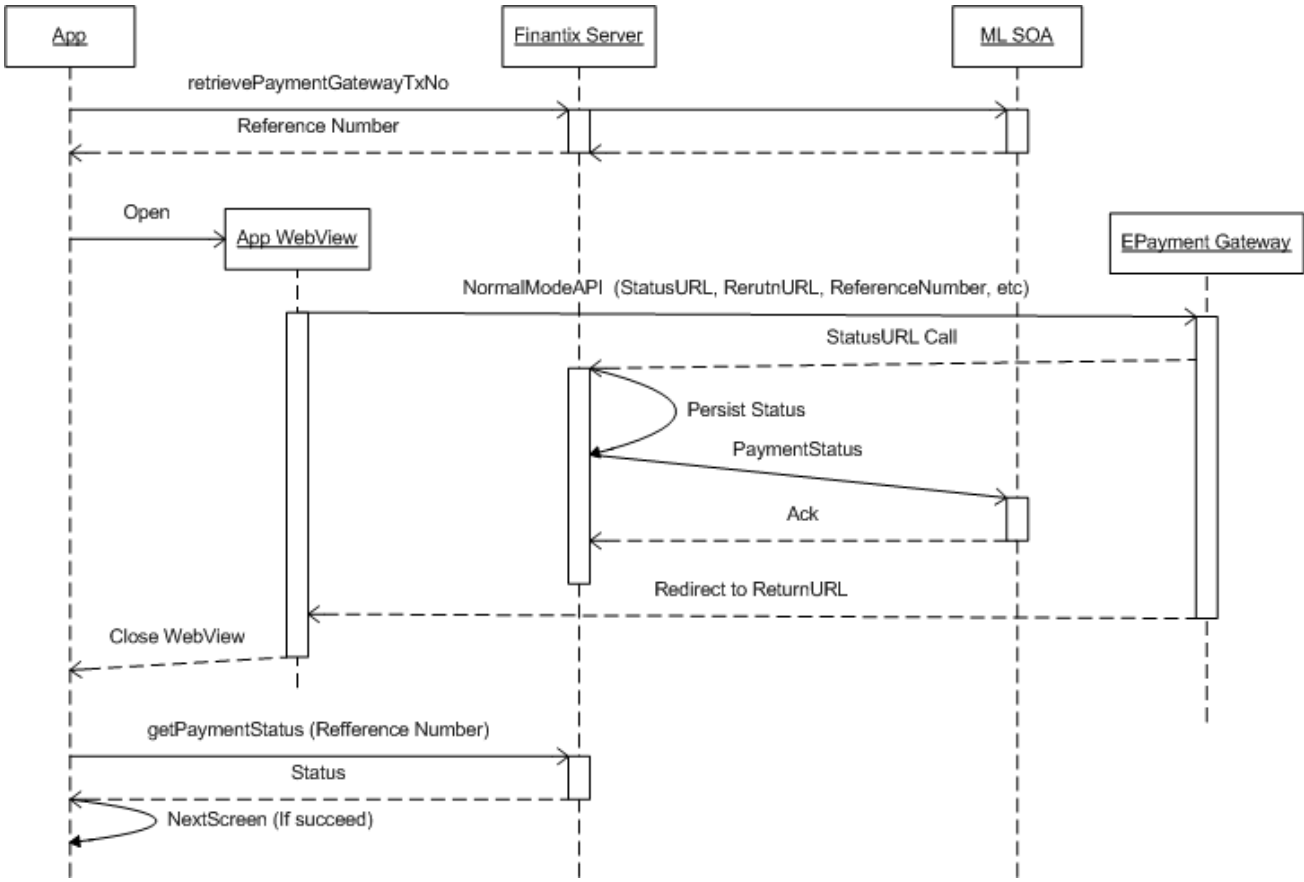
9.2 E-payment Gateway Processes.

- FNX will first call the retrievePaymentGatementTxNo to get the reference number for the transaction.
- Normal Mode API will be used to call the e-payment gateway. For the completed reference, please refer to the E-payment specification, Standard-easyPay2-PaymentPage-Specification-r3.5.pdf.

  Example: https://<EP-Payment Page URL>?mid=20150922005&ref=test-20121228-110&amt=10.00&cur=SGD&rcard=64&returnurl=http://www.ml.com&statusurl= http://www.ml.com.sg&version=2&validity=2012-12-30-11:22:33&signature=706e16062f19170d75457f4e15d18551

- Once the transaction completed, Wildcard will call the service, statusURL, from FNX to provide the status information as described in 7.1.1 from the e-payment specification.
- EPOS will persist the information in the database
- EPOS then call the paymentStatus WS to update ML system with the status.
- Wildcard will call the returnURL to trigger the app to and close the WebView
- The App will call getPaymentStatus service to get the status.
- If the TM_STATUS is equal to “YES”, the APP will redirect the user to go to the next card; otherwise, will display the error message, TM_ErrorMsg, to the user. Based on the error message, the agent then can design rather need to re-attempt the e-payment process.

*** EPOS system will not handle REVERSAL/VOID



9.2.1 getExchangeRateWS (exposed by ML Integration Layer)

*** Refer to the IMS for detail field level mapping.

POS Application will need to convert premiums to different currencies as per user choice.

Apply needs the full set of exchange rates type mappings for the following payment method in scope: EPS, TeleTransfer, UnionPay, All-in-Pay and for each type the full set of exchange rates.

Given that multiple conversions might take place during a payment process and that the conversion rates are updated by Manulife on a daily base at most, to maximize the app performance (reducing the number of service calls) the device will update the conversion rates only every a configurable time period. When a conversion is necessary on ePOS mobile solution, the app will evaluate if the last update time of the exchange rates available in the device is greater than (current time – rates expiry period). If not and if the app is on “online” mode, it will invoke a specific internal service (from ePOS to Finantix ePOS server only) to synch the latest exchange rates.

Exchange rates update might happen in different and unpredictable moments by Manulife (depending on payment mode too). Anyway, Finantix ePOS server will call Manulife server to obtain the whole set of changed exchange rates only when the “expiry” period is passed. The suggested period is of 12 hours.

Only necessary currencies will be provided by Manulife upon request.

High-level service signature is presented in the table below:

Service Name	getExchangeRateWS

Input	sourceCurrency (Optional) destinationCurrency(optional) paymentType
Output	Conversion rates [1..n] destinationCurrency sourceCurrency ExchangeRate Error Information [1..n]

9.3 Hong Kong country-specific Gateways

9.3.1 Status URL Update

Upon receiving authorization status from the financial institution, EasyPay2 will send the payment status to merchant's server via the statusurl.

This is a back-end server-to-server operation that is not visible to the customer for security reasons.

The following parameters will be sent to the merchant's statusurl through HTTP POST

Note:

1) Please ensure that the retrieval of the response parameters use name-value handling as the list of response parameters may increase in future and the position of the parameters are not standard

2) Merchant's server must match the transaction response details against the original transaction. This includes verifying the amount

Merchant's server must be able to handle multiple post back from EasyPay2 in the event that a customer re-attempts the same transaction more than once

3) The recommendation is for the merchant's server to use the first status update from EasyPay2. Once the transaction is marked as Successful / Failed, subsequent updates from EasyPay2 should be ignored / flagged for follow up. Upon receiving authorization status from the financial institution, EasyPay2 will send the payment status to merchant's server via the statusurl.

This is a back-end server-to-server operation that is not visible to the customer for security reasons.

The following parameters will be sent to the merchant's statusurl through HTTP POST

Note:

1) Please ensure that the retrieval of the response parameters use name-value handling as the list of response parameters may increase in future and the position of the parameters are not standard

2) Merchant's server must match the transaction response details against the original transaction. This includes verifying the amount

Merchant's server must be able to handle multiple post back from EasyPay2 in the event that a customer re-attempts the same transaction more than once

3) The recommendation is for the merchant's server to use the first status update from EasyPay2. Once the transaction is marked as Successful / Failed, subsequent updates from EasyPay2 should be ignored / flagged for follow up

9.3.2 getPaymentStatus_WS (Exposed by Finantix)

High-level service signature is presented in the table below:

Service Name	getPaymentStatus
Input	ReferenceNumber
Output	Status Message

9.3.3 PaymentStatus_WS (Exposed by ML integration Layer)

*** Refer to the IMS for detail

High-level service signature is presented in the table below:

Service Name	PaymentStatus_WS
Input	policyNumber TM_MCode TM_RefNo TM_Currency TM_DebitAmt TM_PaymentType TM_Status TM_Error TM_ErrorMsg TM_ApprovalCode TM_BankRespCode TM_TrnType TM_SubTrnType TM_CCLast4Digit TM_ExpiryDate TM_CCNum TM_RecurrentId TM_SubSequentMCode TM_UserField1 TM_UserField2 TM_UserField3 TM_UserField4 TM_UserField5 TM_IPP_FirstPayment TM_IPP_LastPayment TM_IPP_MonthlyPayment TM_IPP_TransTenure TM_IPP_TotalInterest TM_IPP_DownPayment TM_IPP_MonthlyInterest TM_TokenId TM_Version TM_Signature TM_ECI TM_CAVV TM_XID

	TM_Original_RefNo TM_OriginalPayType
Output	Ack

9.3.4 retrievePaymentGatewayTxNo (exposed by ML Integration Layer)

*** Refer to the IMS for detail

*** Payment Transaction Number is unique for the ePos system.

POS Application will call this service to retrieve a unique transaction reference number generated by Manulife for payments.

High-level service signature is presented in the table below:

Service Name	retrievePaymentGatewayTxNo
Input	Policy Number
Output	Payment Transaction Number response details [1..1] Error Information [1..n]

10. Security

This section of the document covers the detailed security requirements for Manulife POS.

Approach described in this chapter applies to 2015 PHASE1 of project. Next country release approach is to adopt the Microsoft Active Directory Federated Services (ADFS) which leverages SAML based authentication and authorisation exchanging. Finantix will evaluate integration with ADFS when Manulife provides roadmap and implementation specification together with non-functional requirements.

For the Singapore POS application, a special localised Singapore only customisation will be provided. It will includes 2FA (two-factor-authentication) and support for HSM and UAS solution provided by i-Sprint. It is a heavily customisation of OOTB. Finantix solution requirement will be evaluated on a change control basis and analysis has be executed – This is OUT OF SCOPE FOR 2015 PHASE1.

10.1 Security environment

Finantix architecture deploys standard Java Enterprise Servlets that manage the communication with the Web browser or Device through the HTTP protocol. Such Servlets leverage the application server capabilities and delegates to it the management of the HTTP transport layer.

In order to secure the communication between customer/prospect and application server, the HTTPS protocol can be implemented by leveraging the encryption facilities of the JavaEE application server or by deploying the preferred security component, such as an HTTP server or SSL accelerator.

As per Manulife requirement all the inter-connections (including internal) must be secured with secure protocol (regarding the use, e.g. HTTPS, LDAPS, etc.).

10.2 Agent and staff users authentication and authorization

2015 PHASE1 of the project will leverage only on AD stand-alone authentication with no SSO services to external system – 14.2.2.1

Future approach is to get the Active Directory Federated Services (ADFS) and leveraging on SAML based authentication and authorization exchanging for integration with MIL systems (details in chapter 10.2.4) – 14.2.2.2.

10.2.1 Agent authentication and authorisation flows

Mobile application access (Agent) Online – mobile device has connection to the Internet

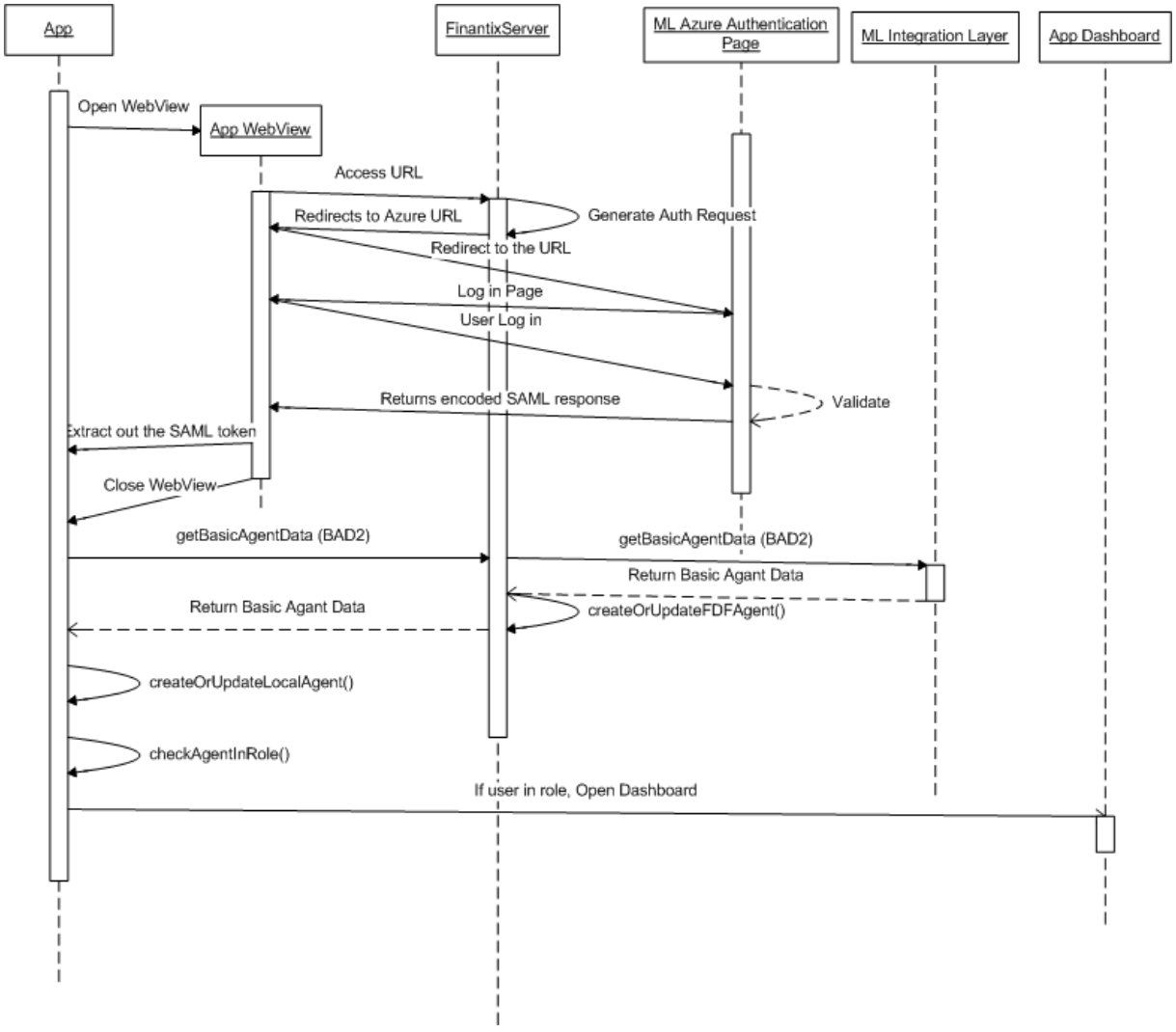


Figure 28 – Authentication and authorization flow for POS in online mode in mobile device

Description:

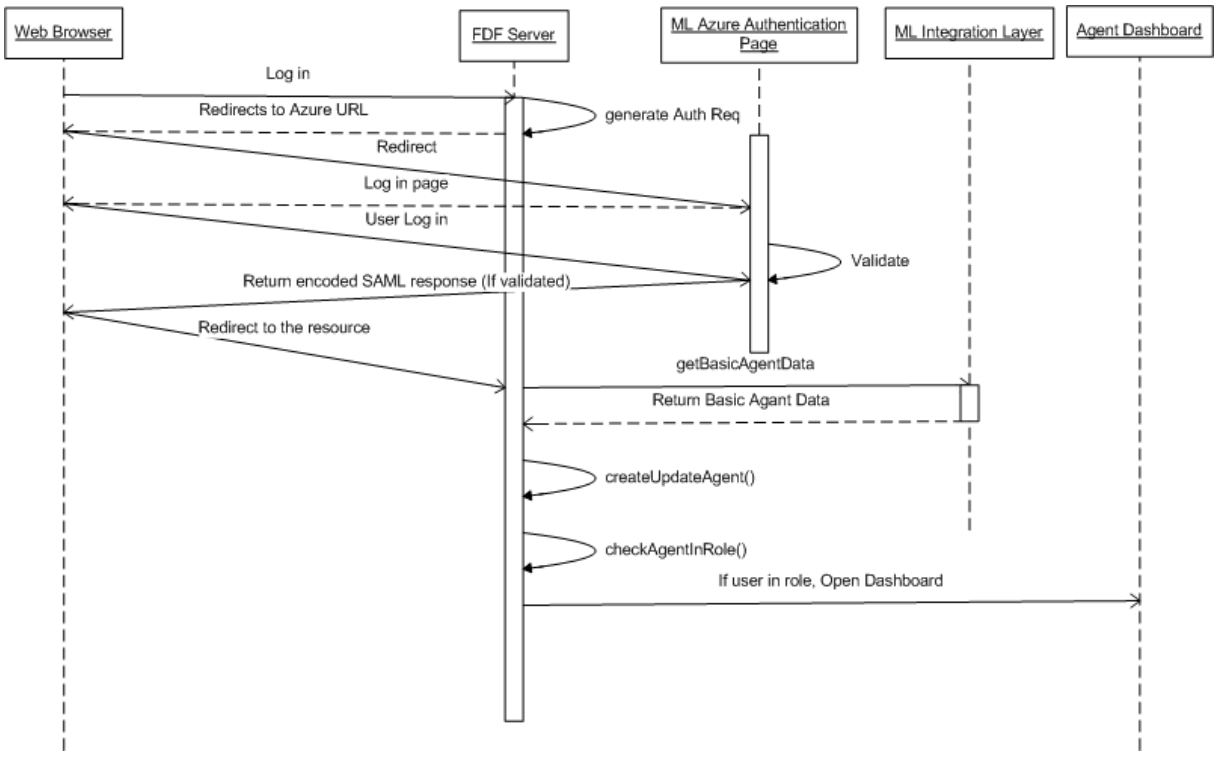
1. *Agent opens the POS application.*
2. The application server will redirect to Azure authentication page (Office365) to properly authenticate the user session.
3. Authentication is done by the user inside Azure authentication page (Office365)
4. The session authentication token is returned to the device and stored for next calls
5. The application server redirects to requested resource.
6. Get the BasicAgentData (BAD2) through the ML WS.
7. *In case if in FDF user registry, there is no agent account yet – it is created, otherwise its data is updated.*
8. *Agent data is returned to DDF*

9. *In case if in DDF user registry, there is no agent account yet – it is created, otherwise its data is updated.*
10. *Check if agent is in role to use POS application is performed.*
11. *If agent is in role to use POS application its offline secure access token is created / updated with new timestamp, otherwise his token is revoked (see chapter 10.4.1, point 3). Agent roles are store in the device for offline functionalities, to perform validation in case of offline access.*
12. *If agent is not in role redirection to login screen is performed (messages to be displayed are defined in chapter 2.1 Logon of FDS document).*
13. *If agent is in role his access is authorized and Agent Dashboard screen is displayed.*

- Offline – mobile device does not have access to the Internet
    - Starting Sprint 3, there is no auththentication is needed for the offline. That is, after the user log into the iPad, the user is able to go to E-POS directly.

10.2.2 MIL staff users authentication and authorisation

- MIL staff will use the “Admin console” using the browser.
- AD: Azure.



10.2.3 SOAP Webservices authentication approach (Finantix and MIL)

Manulife is introducing unified approach for interfaces definition and authentication in their organization.

SOAP webservices exposed/consumed by Finantix will be available over secure HTTPS protocol. Authentication will be performed as SOAP header based (WSE WS-security with application specific customToken and SSOToken).

Webservice user credentials will be passed as systemId (user) and password. SystemId and password is to be provided by Finantix/MIL for services they expose.

SSOToken will not be used before next country rollout after 2015 PHASE1.

For detailed information regarding the SOAP header contents please refer to particular interfaces specified in IMS document.

Element	Description
customToken	Will contain Agent login Id – see chapter 4.1.1 – it is aligned with current MIL implementations
SSOToken	SSO token will be used when integration with ADFS will be implemented (see chapter 10.2.4) – optional parameter in 2015 PHASE1, if provided will be ignored
UsernameToken	As per WSE WS-security extension will contain credentials for authentication (systemId and password)

10.2.4 Active Directory Federated Services integration

New MIL organisation identity is based on Active Directory Federated Services (ADFS) – 14.2.2.2.

The picture below explains the possible integration flows with ADFS using SAML (for the web app and iOS device app).

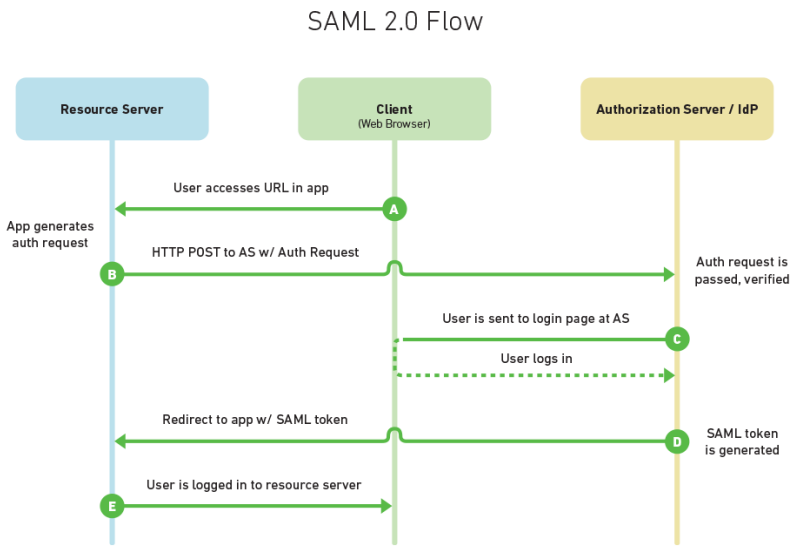


Figure 29 – Authentication and authorisation flow for POS web app (ADFS)

With the introduction of Azure, the “tenant” (country) information will be retrieved from the user login as the last token of the login name.

Given that the login page for online authentication is managed directly by Microsoft, the current approach of saving inside the device the password as hash string for offline authentication cannot be used again..

Offline will not required password access (Refer to the new CR submitted from ML)

10.3 Additional checks after user authentication

After a successful user authentication, two additional checks are performed to verify:

1) Device model ID support
2) Application version backward compatibility

If these verifications are successful, the interaction between POS App and the Application Server proceeds with the remaining login steps.

Otherwise the login procedure is blocked and a proper error message is displayed.

10.3.1 Device model ID verification

The list of iPad versions authorised to run ePOS solution is defined in a property file stored in a configurable path. The file contains all the iPad versions that can synchronize with that specific server instance (whitelist).

The device models listed in the file are the Model IDs (e.g. “iPad4,2”, “iPad2,1”). Device model will be retrieved relying upon available iOS API calls.

Every time the user logs in ePOS application (Online mode), the check verifies if the device model is listed in the whitelist stored in the server. In case of login in Offline mode, the check is done on latest check result stored locally: If it is a successful result the agent can log in otherwise it is blocked.

Manulife has the ownership of maintaining the device codes whitelist, by removing codes of device models no longer supported and adding new models once certified for the ePOS solution.

10.3.2 Application version backward compatibility

The list of App versions authorised to log in ePOS solution is defined in a property file stored in a configurable path. The file contains all the App versions that can synchronize with that specific server instance (whitelist).

Every time the user login into ePOS application (Online mode), the check verifies if the App version currently used is listed in the whitelist stored in the server. In case of login in Offline mode, the check is done on latest check result stored locally: If it was successful result the agent can login otherwise it’s blocked.

Manulife has the ownership of maintaining the whitelist of compatible App versions whenever a new ePOS solution release is deployed, both App and Server.

10.4 Data level security requirements

10.4.1 Mobile device

The following list summarizes the device-level security elements of a Finantix solution:

1) Communication over secure protocols (e.g. HTTPS),
2) Mobile application performs authentication verifying user credentials against the remote server, by simulating the browser behavior via a HTTPS call. Therefore the

user is granted access to the mobile application only if and only if the Server accepts the credentials.

3) Offline Login support. Finantix solution will store hashed version (SHA-256 with salt) of the credentials in the local storage (if and only if authentication with the server succeeded). Hashed credentials are subject to a configurable expiry time (grace period) and will be used for Offline authentication

4) Local database encryption (Native device encryption AES-256 – handled by mobile device) – no specific requirements issued by Manulife for data level encryption

5) Data security requirements for particular data entities are specified in FDS document in chapter 1.5 Definitions.

10.4.2 Web Browser

Finantix architecture deploys a standard J2EE Servlet that manages the communication with the Web browser through the HTTP protocol. It leverages on the J2EE application server capabilities and delegates to it the management of the HTTP transport layer.

Communication browser-application server will be performed over HTTPS protocol.

For list of supported browsers please refer to chapter 12.1

10.4.3 Server side data repository encryption

Finantix POS will not provide server side data encryption as a part of POS solution. Manulife can leverage on encryption capabilities of database server, operating systems or other 3rd party products that provide additional security at data level.

Any specific encryption needs for particular entities or particular fields would be addressed through change control.

10.5 OWASP compliance

The OWASP Top Ten represents a broad consensus about what the most critical web application security flaws are. A methodical application of MDA together with a well-engineered architecture strengthens defences against security flaws at design level.

The model-driven development of Finantix allows developers to focus on the functional specifications. Important OWASP-related information are automatically extracted from the model and applied by the run-time architecture, which is aware of model information. This reduces the level of awareness of the functional developers who are not required to engineer technical solutions for all functionalities in order to respect OWASP guidelines since such solutions are provided out of the box by the architecture of Finantix framework.

Besides built-in security features of Finantix framework, software development lifecycle adopted by Finantix incorporates steps ensuring the quality of code and security:

1. Development phase:
    a. Consistent coding style across organisation
    b. Static code analysis – SonarQube with security rules enabled
    c. Code peer reviews
2. Testing phase:

a. OWASP Top Ten security checks
b. Penetration tests at least before major product release (may be executed internally or potentially by independent third-parties)

Finantix standard IDE and continuous integration environment leverages on SonarQube as aggregator dashboard of static code analysis and coding conventions enforcement. Analysis results are directly available for development teams for their action.

11. Auditing and Logging

11.1 Auditing

This chapter covers the topics related to auditing of technical and business operations. It defines the format and describes content of audit records created by FDF and DDF and also specifies the events which are audited.

Finantix audit framework is providing general purpose audit facade service which can be used for "sending out" audit events. It can be used as an API by application code to programmatically send out audit events as required (e.g. when business operations audit needs to be put in place for some special auditing requirements).

Audit service can be plugged with possibly multiple audit event processors that can decide to process an event or not based on the event source value (see the audit record specification).

There is a default audit processor accepting all audit events that logs events using a dedicated logger.

By default the audit records are stored in file system at FDF side (both from server and mobile side after synchronisation). In mobile device audit records are stored in device DB.

11.1.1 MIS Logging CSV Export

MIS Logging is exported to Manulife in CSV format in a dedicated folder. CSV file will be zipped and naming convention for exported files will be:

mis-audit_<server node identifier>_<yyyyMMdd>.csv.zip

e.g. “mis-audit_10.235.66.14_LibSrv2_2015-06-03.csv.zip”

Every day there will be one exported zip file for each server node of the cluster. CSV file will contain event id and input/output parameters as described in the FDS.

For details of the export interface please refer to IMS document, tab MIS-BE.

11.1.2 Audit records purging

11.1.2.1 Mobile application

In mobile application records will be purged after successful synchronization with FDF.

11.1.2.2 Server side (both from mobile and web application)

Mobile application audit records synchronized from mobile device will be kept in server file system for 3 months (90 days) - records older than that will be purged from file system.

Web application audit records will be kept in server file system for 3 months (90 days) - records older than Y will be purged from file system.

Audit record files will be fetched by MIL on daily basis via SFTP server exposed by Finantix.

11.2 Logging

11.2.1 Finantix Digital Foundation

This section describes the logging mechanisms available in Finantix framework. This section also shows the available loggers.

Logging framework follows the practice to name loggers from the class name of the class where it is used, e.g. com.finantix.somecomponent.AClass and logging settings are inherits setting of logs in hierarchy, e.g. “com.finantix” inherits settings for “com”.

As Finantix framework architecture is modular, each module has it's root logger. If module name starts with “com.thedigitalstack”, the main root logger for all the loggers defined for the module will inherit from “com.thedigitalstack”.

Finantix uses SLF4J (Simple Logging Facade 4 Java) abstraction layer over “logback” logging framework for logging purposes.

11.2.1.1 Log configuration

Log format configuration used by Finantix framework is of the below form:

TIMESTAMP, LEVEL, THREAD, FRAMEWORK-COMPONENT, MESSAGE

Example of log in DEBUG mode:

```
2014-11-18 10:36:06,024 DEBUG | http-bio-8080-exec-6 |
c.t.restful.internal.service.RESTRequestDispatcher      | Request 'POST - /rest/user/users'
served by [contrib=com.thedigitalstack.mdl.user.core_1.0.0.qualifier [202], ver=0]
2014-11-18 10:36:06,026 DEBUG | http-bio-8080-exec-6 |
c.t.restful.internal.service.RESTRequestDispatcher      | Serving transactional request
2014-11-18 10:36:06,026 DEBUG | http-bio-8080-exec-6 |
c.t.model.internal.persistence.PersistenceManager       | Beginning logical unit of work
2014-11-18 10:36:06,073 DEBUG | http-bio-8080-exec-6 |
c.t.model.internal.persistence.PersistenceManager       | Committed logical unit of work
2014-11-18 10:36:06,074 DEBUG | http-bio-8080-exec-6 | RESTDispatcher
| [contrib=com.thedigitalstack.mdl.user.core_1.0.0.qualifier [202], ver=0] responded with
status: '200' for request 'POST - /rest/user/users'
```

MIL has specific requirements related to logging configurability:

1) application log files needs to be separate for countries – OUT OF SCOPE FOR 2015 PHASE1 as only HK log will be available – feature will be provided next country rollout after 2015 PHASE1
2) rotation and retention policies configurable by country – OUT OF SCOPE FOR 2015 PHASE1 – feature will be provided next country rollout after 2015 PHASE1

2015 PHASE1 requirements for HK:

a) log rotation will be file size based and will occur when log will reach 10MB of size,
b) archived log files will be retained for period of 3 months (90 days), files older tgan that will be purged from file system
c) Audit record files will be fetched by MIL on daily basis via SFTP server exposed by Finantix.

11.2.1.2 List of the root loggers

Module	Root logger	Description
Analytics	c.t.analytics	Provides an interactive dashboard for monitoring the application usage. The graphical dashboard is based on content extracted from incoming HTTP request and tracks information such as average response time, HTTP codes, geo localization, most invoked services etc.
Authentication	c.t.authentication	The authentication component provides multitenant authentication support. May use its own database table or just views on top of a functional module that provides a user registry. For testing purposes can be configured to authenticate users without requiring a database.
Core	c.t.core	Represents the core of the platform. Provides among other thisgs: - identity management (principal and tenant) - caches - dependency injection support - cluster support (cluster level semaphores and distributed execution of code) - adapter manager - contribution registry The above typically have dedicated subloggers, e.g. **c.t.core.cache** or **c.t.core.cluster**
Configuration	c.t.configuration	Provides API for convenient management of configuration artifacts
Jackrabbit	c.t.jcr.jackrabbit c.t.jcr.jackrabbit.configuration	Platform integration with Apache Jackrabbit, a fully conforming implementation of the Content Repository for Java Technology API (JCR, specified in JSR 170 and 283). — Needed only for CMS support (e.g. content management, document management)
Scheduler	c.t.scheduler	Scheduler component
Model	c.t.model	Provides JPA-based persistence, JAXB-based XML/JSON marshalling, Per-tenant extensible models and metadata support, as well as base classes for DAO services, Model Synchronization API (synch logger is c.t.model.service.sync) Some subloggers: - c.t.model.persistence - c.t.model.service - c.t.model.metadata - c.t.model.binding (XML/JSON)
Model Hieachy	c.t.model.hierarchy	Provides a convenient API for JPA-based hierarchical tree storage
Security	c.t.security	Responsible for making available to each application thread the current user and the current tenant. It also declares an extension point used for multi-factor authorization mechanisms (e.g. used to implement license control, OTP etc)
Sharing	c.t.sharing	Provides API for implementing 'shareable'entities, which are entities whose access can be controlled.
Rest	c.t.restful	**com.thedigitalstack.restful** provides platform support for RESTful services.

Vaadin	c.t.core.vaadin c.t.core.vaadin7	Reusable Widgets and utilities for developing Web apps on top of the vaadin framework. Two bundles exists, one for vaadin version **6** and one for vaadin version **7**.

11.2.2 Device Digital Foundation

At the DDF side in mobile device Finantix is not performing any extensive logging except residual debugging information logging purely for development purposes, which is not available in production application environment. For troubleshooting purposes Finantix will use the audit mechanisms described in chapter 11.1 together with server side logs described in chapter 11.2.1 – this information will be sufficient for troubleshooting purposes.

In case of application crash iOS application crash logs will be intercepted by MAM solution – not covered by SAS document.

12. User Interface and Localization

12.1 Browser certification requirements

Because of the several and important animated transitions requested by the POS solution for agents, the web application can be used with the following web browsers that support HTML5 in a proper way (the + means that later known versions are supported):

Internet Explorer 10+

Mozilla Firefox 24+

Google Chrome 24+

Safari 7.1+

Administration consoles have less animations/transitions requirements so they can be accessed using the following browsers:

Google Chrome 23+

Internet Explorer 8+

Mozilla Firefox 17+

Safari 6+

13. Deployment

13.1 Hosting solution

Hosting solution is not covered by SAS document.

13.2 Mobile application distribution

For 2015 PHASE1 mobile application will be build from common code base and will contain only HK specific logic. Regional mobile application deployment model will be designed and agreed when regional solution review is performed after 2015 PHASE1 design completion.

Mobile application distribution will be managed by MAM solution - not covered by SAS document.

13.3 Platform

Chapter contains information about the hardware platform requirements and software platform certified by Finantix for ML regional solution.

13.3.1 Hardware

The hardware infrastructure consists of below servers:

- Application server
- Web server
- Database server

13.3.1.1 Application Server Hardware – Production – TBC

13.3.1.2 Web Server Hardware – Production - TBC

13.3.1.3 Database Server Hardware – Production - TBC

13.3.1.4 Tablet devices

Operating System	iOS 9.1
Device	iPad 3 and 4, iPad mini not included

13.3.1.5 Workstation – Web browser

N/A for 2015 PHASE1

13.3.2 Software

13.3.2.1 Application Server

Name	IBM WebSphere Express Edition
Profile	Liberty Profile
Version	8.5.5.5
OS	Windows 2012 R2 64bit

13.3.2.2 Web Server

Name	IBM HTTP Server
Version	8.5.5.5
OS	Windows 2012 R2 64bit

13.3.2.3 Database Server

Name	IBM DB2 Workgroup Edition
Version	10.5
OS	AIX 7.1

14. Appendixes

14.1 Appendix A – Batch Integration with Manulife systems – General information

14.1.1 Batch import integrations – service managed by Finantix

Batch integration is achieved by uploading data files to the dedicated location on the application server. SFTP server will be exposed by Finantix.

This is folder structure where MIL will upload files for import processes:

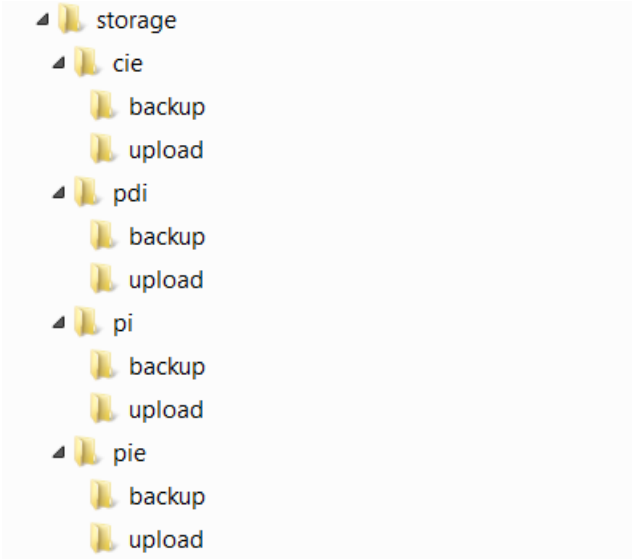


Figure 30 Import Folder structure

Each batch process requires separate folder:

- *cie*: customer import
- *pdi:* policy dictionary import
- *pi*: policy import
- *pie*: prospect import

Each import process has an *upload* folder that will store files transferred by Manulife and a *backup* folder that will store imported files.

14.1.2 Batch export integrations – service managed by Finantix

****** ML requested to have a HTTPS open for the download purpose. From the technical point of view, it should be fine, but need to check and confirm with ML security team if any concern.***

Batch integration is achieved by exporting data files to the dedicated location on the application server. SFTP server will be exposed by Finantix.

This is folder structure where MIL will read files for export processes:

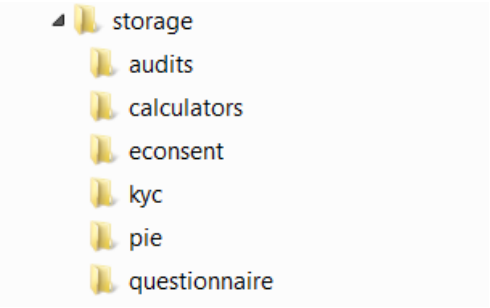


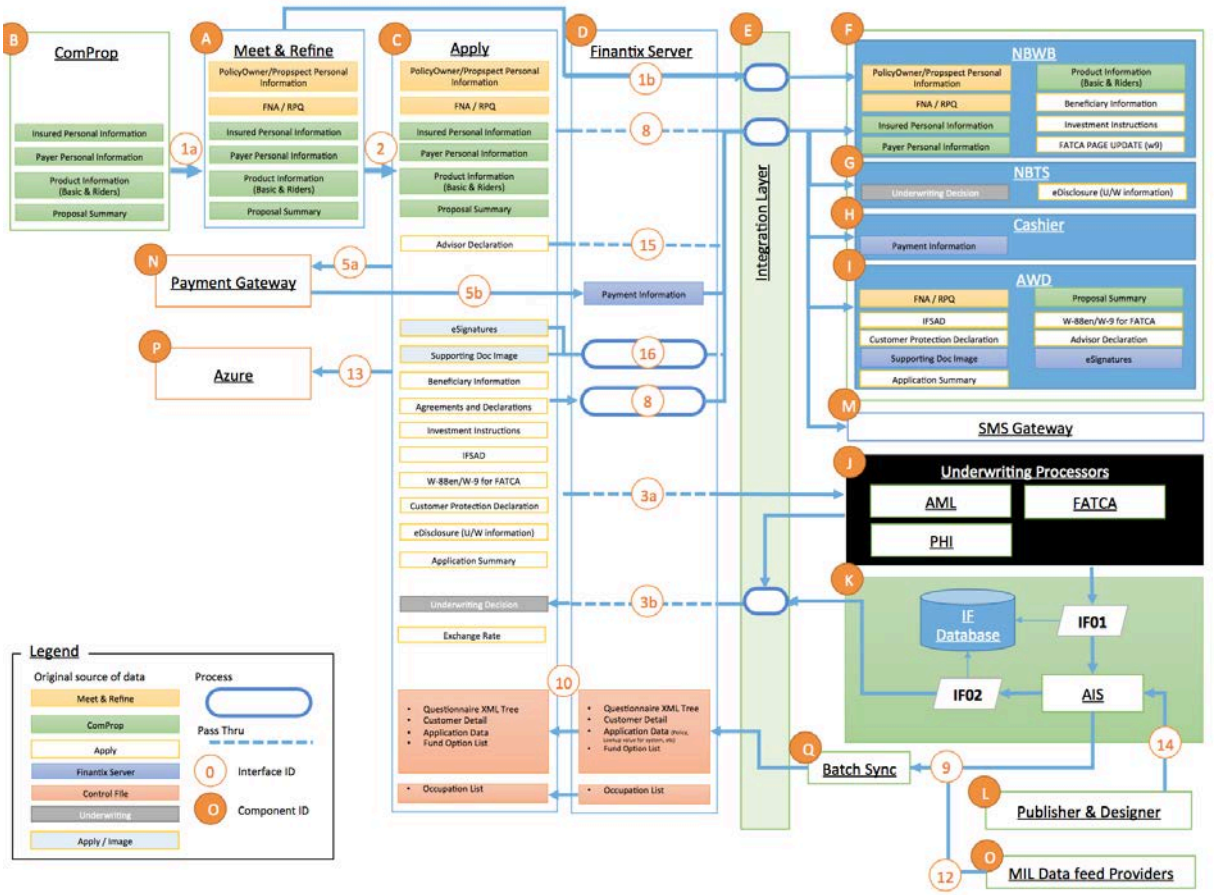
Figure 31 Export Folder structure

Each batch process requires separate folder:

- *audits*: MIS logging zip files
- *calculators:* calculator xml files
- *econsent*: e-consent PDF files
- *kyc*: kyc xml files
- *pie:* prospect xml files
- *questionnaire:* RPQ and FNA xml files

14.2 Appendix B – Manulife existing solutions

14.2.1 Integration Points and the related ML backend systems



14.2.2 Agent repository

14.2.2.1 2015 PHASE1 (standalone AD over LDAP)

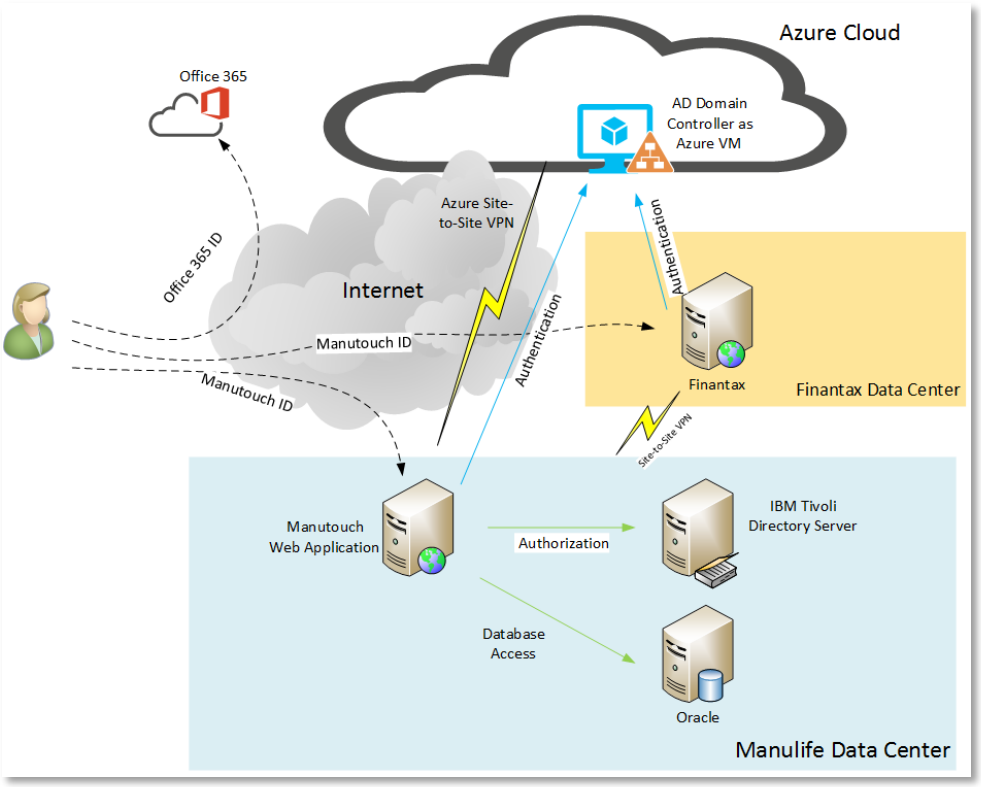


Figure 32 – Authentication integrations – 2015 PHASE1

14.2.2.2 Future phases of the POS project (ADFS) – for next country rollout after 2015 PHASE1

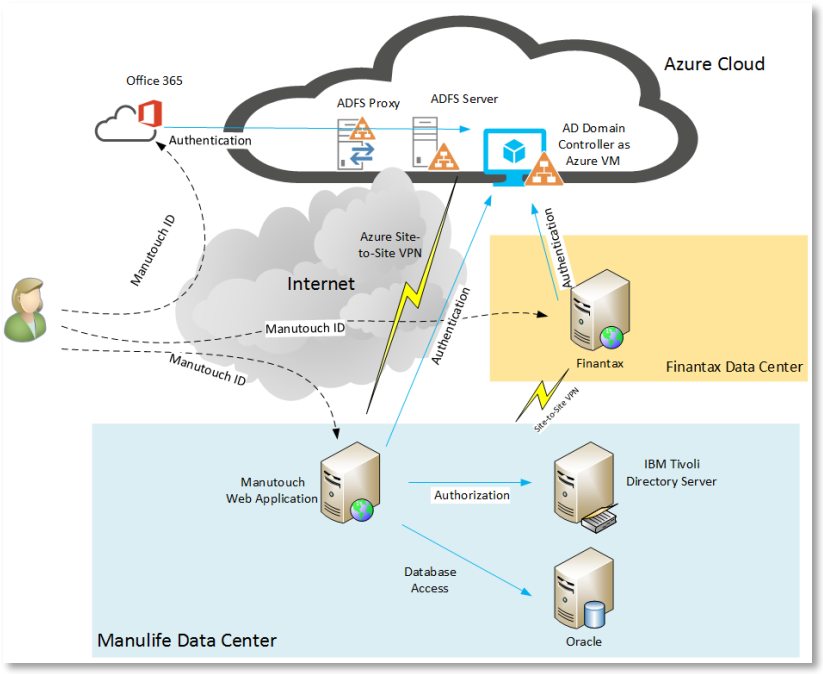


Figure 33 – Authentication integrations – regional solution

14.2.3 Customer and prospect information systems

A prospect can be seen and update only by the Agent that created it. If the same person is in contact with two different agents, she/he is represented by two different prospects on Manulife backend system.

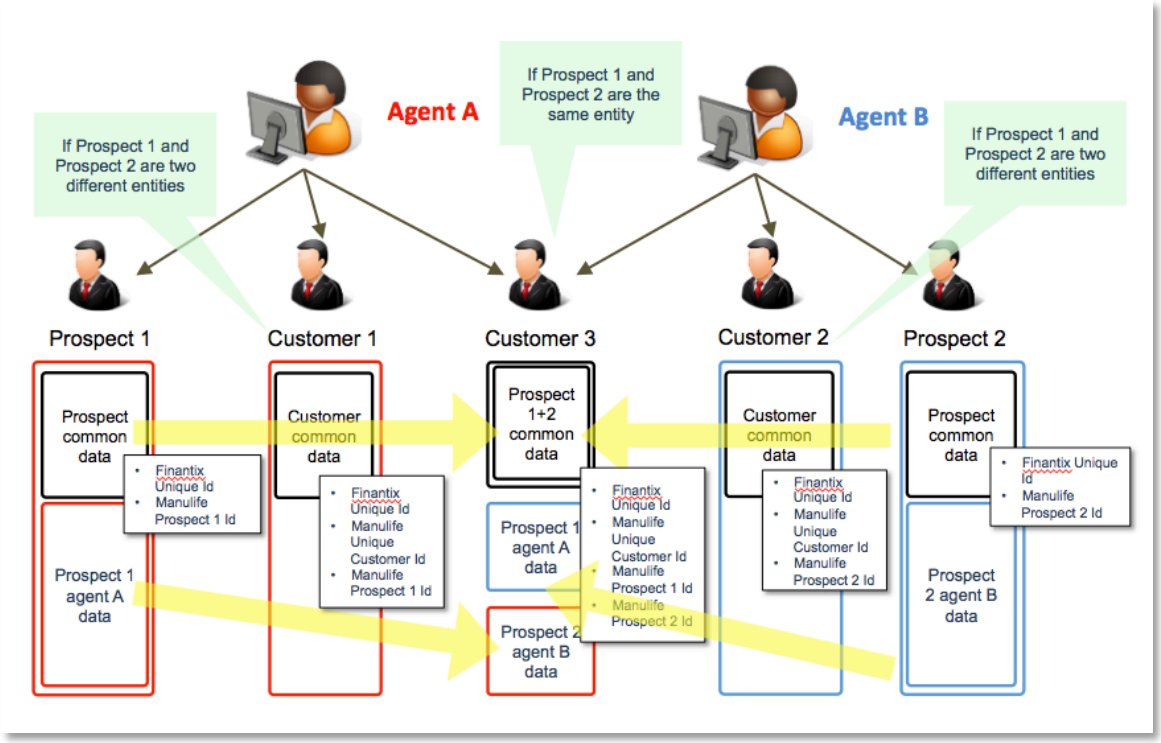


Figure 34 – Prospect→Customer data deduplication

In particular for Hong Kong, agents must not share between each other the prospect data as well as the customer data not considered “shareable” to all customers (see picture above).

Prospect creation can be performed by ManuTouch as well as by other country specific systems. Prospects are “enrolled” as customers by Manulife backend system that take care of

A prospect is transformed to a “Customer/prospect” only if she/he applies for policy and the policy is confirmed. At that point, Manulife backend system “de-duplicates” the prospect with existing customers (if any) by merging the common data provided by the different agents. The prospect is then removed and only the new customer remains (with no references to the original prospect).

14.2.3.1 Available services and interfaces

No services are available for retrieving/uploading Customer/prospect or Prospect data as it is requested by POS application requirements.

A set of SOAP-XML services are available to retrieve the list of proposals associated to a given Customer/prospect and apply for a proposal. They are accessible by providing an authentication token that is usually retrieved by “ManuTouch” application during the authentication phase.

14.2.4 Policy data storage

14.2.4.1 Policy data structure

14.2.4.2 Link of the policy to customer and product

14.2.5 Existing quotation engines

14.2.6 Underwriting system

14.2.6.1 Application underwriting flow

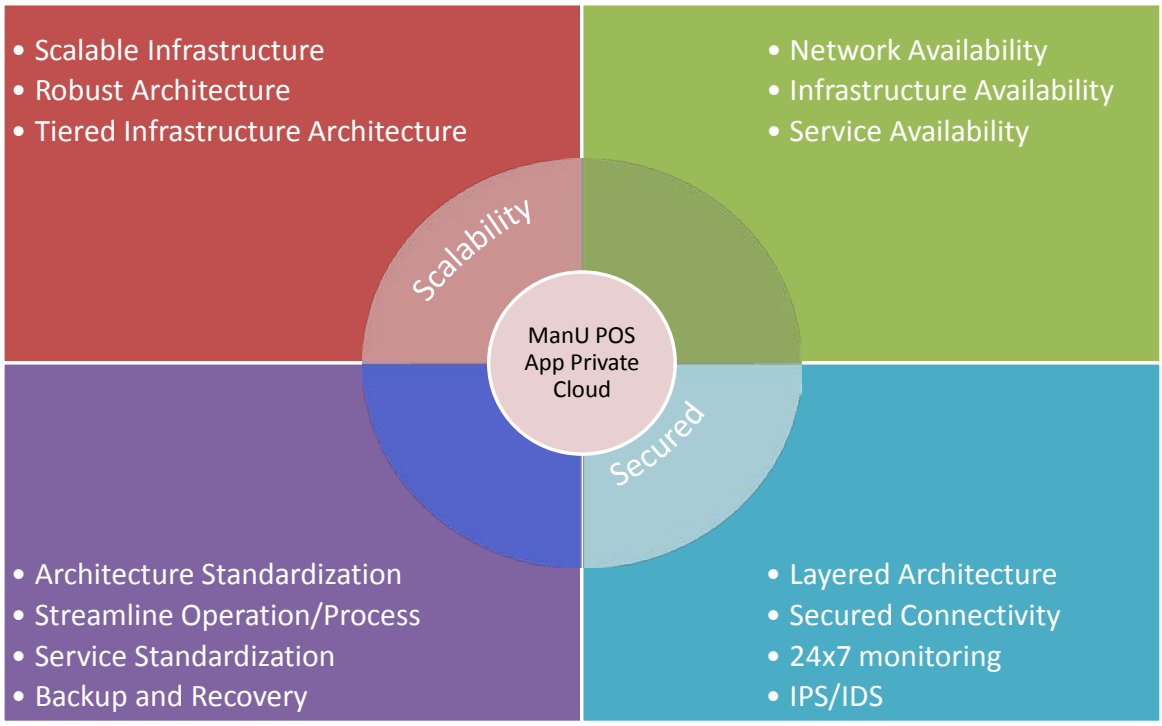
14.2.6.2 Available interfaces and application data structures

14.3 Appendix C – Infrastructure Design Considerations

The following assumptions are made for the current Infrastructure:

- 100 Mbps internet bandwidth is provided to access the production environment
- Network devices (such as switches and patch panels) shall be shared with other applications as well.
- Server devices will be dedicated to Manulife hosting platform.
- The current design is based on active-passive option for primary and DR site.
- DR shall be activated when hardware and software services at PDC are unavailable.
- Secure tunnel would be used for site-to-site VPN.
- An adequate backup of data should be lodged off-site (safety vault storage).
- DR shall be 50% of the production capacity

Based on the four pillars of design principles (i.e. Scalability, Availability, Standardized, Simplified), the underlining supporting features of these pillars of principles are illustrated in the diagram below.



14.3.1.1 Availability

Availability design principal ensures the availability of services by not just looking at the availability of the up-running applications or services, but also at the underneath infrastructure and networking layers.

- Network Availability

Network is the key component making the services/applications available for access by business users or other system(s), redundancy of networks, network equipment, appliances, connectivity would be taken into consider to ensure availability of services.

- Infrastructure Availability

The infrastructure Availability is one of the key components for the hosting of applications/services, these covers the physical and/or virtual servers, and storage architecture to support the availability of services hosted in the Production Data Center. Requirements on processing powers, build-in components such as NICs, HBAs, HDDs, required external storage designs, redundant power supplies, and others would be identified by each services during the design phase. Some of the key design considerations are as follows:

  - Firewalls will be configured as Active-Passive failover configuration
  - Routers will be configured as Active-Passive failover configuration
  - Core Switches will be configured as Active-Active failover configuration
  - DMZ switches also configured with redundant failover configuration
  - Load balancers to distribute the requests based on round-robin algorithm
  - Virtualized servers with High availability feature enable server failovers with minimal downtime
  - Multiple Storage/Disk arrays with storage virtualization (RAID)

- Service Availability

The availability of the services/applications itself is critical when looking at service availability criteria. Services to the business users would not be affected by planned and unplanned outages during operation hours. Server Virtualization with HA features, redundant hardware components and etc., would help to achieve the required service availability. Application/System design/implementation shall support the HA requirements to meet the service availability requirements.

14.3.1.2 Scalability

The Scalability design principal is important to ensure the scalability of infrastructure and services deployed on top of it. This ensures the seamless deployment to support the growth of users and business requirements. Proposed server infrastructure could be scalable in both horizontally and vertically. User Requests (Load) can be distributed to web servers based on the type of requests or server load. Application servers would not be load balanced or clustered as the application is not session-dependent. Webserver plugin would be used for the application server clustering. Separate Application server clustering would not be required. Number of servers and Hardware components (CPU, Core, Memory, Network, Disk and etc.) could be increased for each layer, based on the performance/utilization requirements.

14.3.1.3 Standardization

Standardization design principal is important to streamline and simplify the operation and monitoring process, and support and maintenance of the Data Center(s). Proposed Implementation Operations and Support services teams would provide standardized services in the following key areas:

- Architecture Standardization
- Streamline Operation/Process
- Service Standardization
- Backup and Recovery

14.3.1.4 Secured Environment

Please refer Section 14.3.4 for the details.

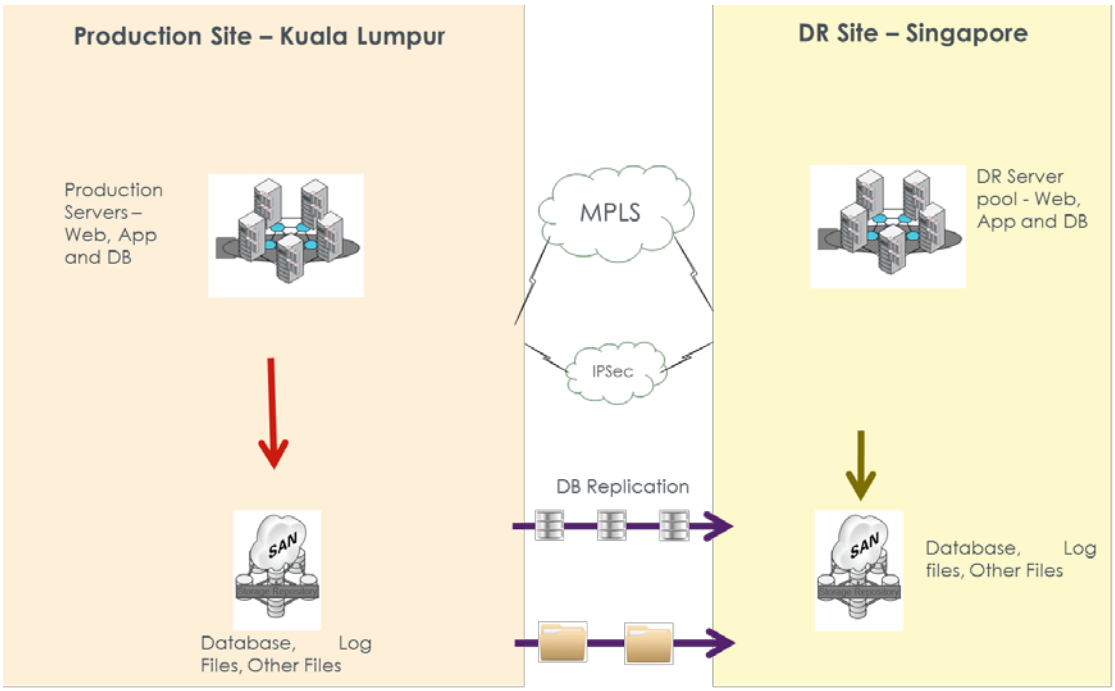
14.3.2 Backup and Recovery

Proposed solution is based on RTO and RPO requirements of 2 and 4 hours. Backup window would be from 1 AM to 5 AM Japan Time. Snapshots of the virtual machines would be done hourly, without impacting the performance. Backup retention would be 7 days since the data would be available in multiple locations (DR DB, ManuLife DB).

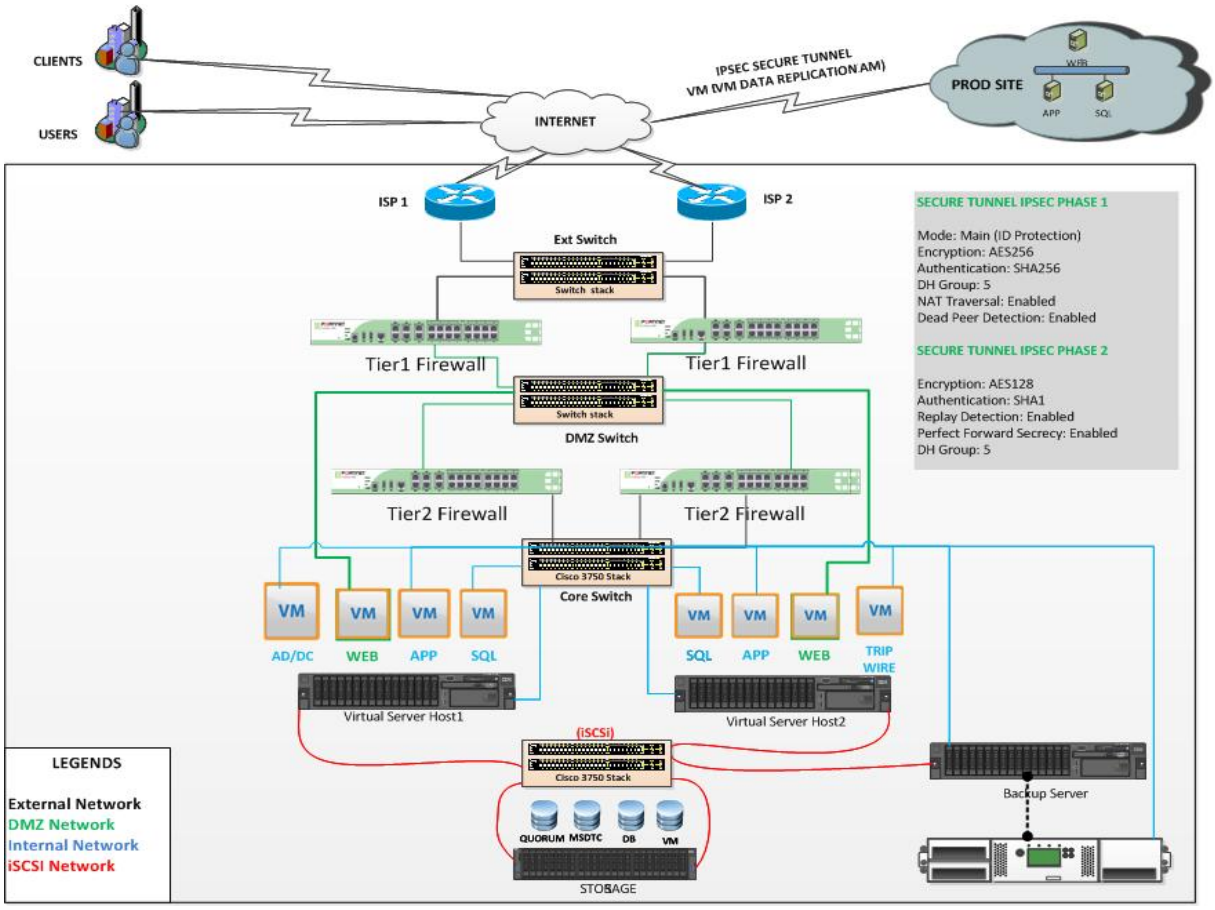
14.3.3 DR Environment

Data from the primary site would be transferred to the DR environment using Log shipping. Log shipping is a method where transaction logs are automatically backed up from a primary DB2 server and made accessible to a standby server. Other DR servers like Web and App servers (except DB servers) would be in cold state (servers would be switched off). In the case of DR activation, DR servers/services would be brought up and database would have all the committed transactions.

Refer the below diagram for the logical architecture for Production – DR replication. Note that MPLS link between Production and DR site (as shown in the diagram) is optional and decision can be made later during the detailed design stage.



Refer the below diagram for the Physical architecture of DR infrastructure.

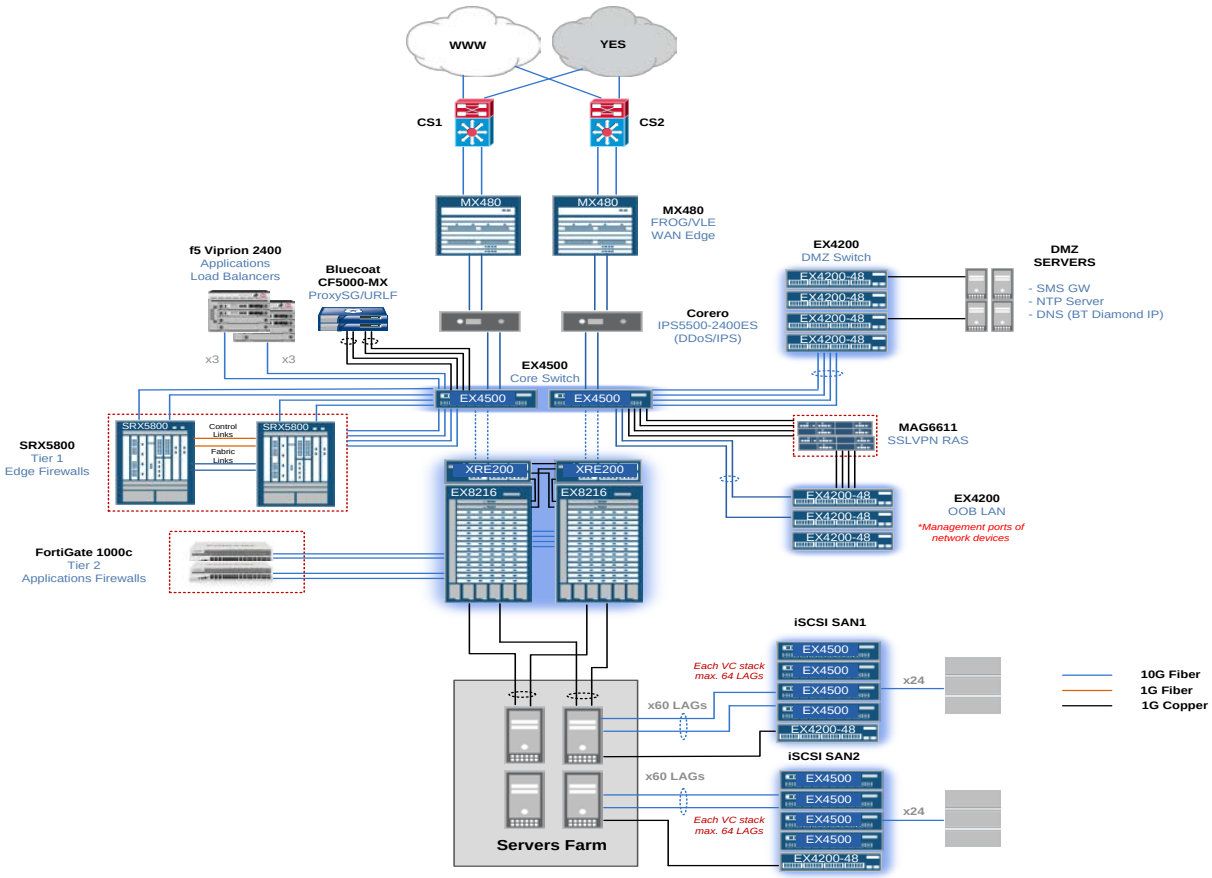


14.3.4 Security Design Considerations

14.3.4.1 Availability

- No single point of failures in all the network components such as routers, switches, firewalls, network, WAN Optimization appliances, MPLS links, etc.
- Firewalls will be configured as Active-Passive failover configuration
- Routers will be configured as Active-Passive failover configuration
- Core Switches will be configured as Active-Hot Standby failover configuration
- DMZ and iSCSI switches also have redundant failover configuration

Refer the below diagram for the Physical architecture of Production infrastructure



14.3.4.2 Secured Access

- Dual layer of Firewall pairs will used for the internet segment
- Flexible, highly scalable and resilient security design following industry best practices to encompass main security goals: confidentiality, integrity and availability
- Stringent Encryption and deep inspection concept: encrypted traffic such as IPSec or SSL is decrypted at the first Tier and can thus be fully inspected by the Firewall.

14.3.4.3 Traffic Inspection & Optimization

- IPS/IDS inspection
- Firewall will also be enabled with Intrusion prevention (IPS), Gateway level AntiVirus/Malware protection
- On the Internet link firewall will be terminating the IPSec/SSL VPN. Users connecting through the Remote access will be given access based on the policy.
- Core Switch ACLs will be configured on the Core switch, for additional layer of firewalling required between the Web, Application and Database tiers server farms, as per ManuLife's requirement.

14.3.4.4 Systems Monitoring

- Systems and Network Monitoring will performed with best of the breed Systems and Network Monitoring systems
- Console server were used to access the Network devices in the case of device failures, even from remote NOCs.
- Log monitoring and archiving will performed by the Syslog server
- Managed Security Services will be available optionally, to have 24x7 monitoring of the network/security if required.
- All the Systems were protected with Symantec End-point protection against Virus / Malware

14.3.5 Logical Architecture