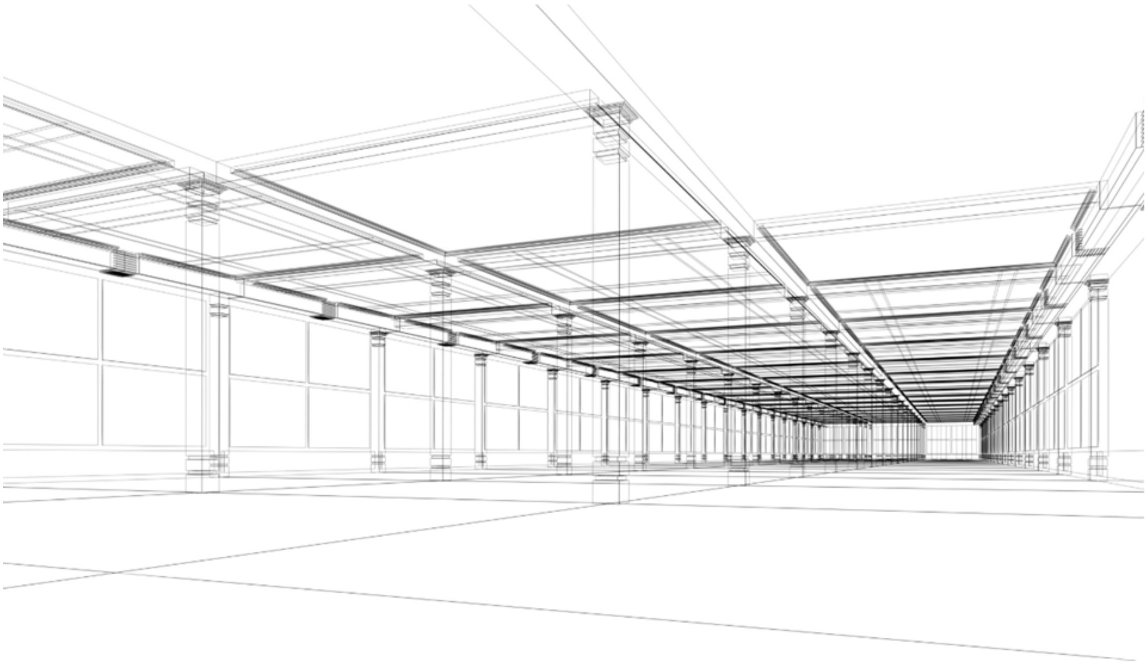
Architectural Thinking Workshop

Course Exercise Model Answer - Solution Architecture Definition Complete



Version: 7

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose of this document

This document contains the model answers for the exercises offered as part of the Architectural Thinking Workshop training course. The course uses a single case study where you as a participant play the role of Solution Architect during a proposal that has already started.

This particular document is the complete Solution Architecture Definition document and includes model answer for all the exercises.

More importantly, this document demonstrates a full example of a Solution Architecture Definition and can be used by students as a “best of breed” example in their future work.

1.2 Definitions

Table 1. List of Terms and Acronyms.

Acronyms	Meaning
AD	Architectural Decision
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
AWS	Amazon Web Services
CDN	Content Delivery Network
CSCC	Client Standard Customer Council
COTS	Commercial Off-The-Shelf
DB	Database
DNS	Domain Name Server
DoS	Department of Statistics
DR	Disaster Recovery
ECN or eCN	Electronic Census Number
ECP	Electronic Census Processing
ECS	Electronic Census System
FW	Firewall
GUI	Graphical User Interface
HA	High Availability
HTML	HyperText Markup Language
IaaS	Infrastructure As A Service
IFP	Intelligent Forms Processing
IPS	Intrusion Protection System
LN	Logical Node

Acronyms	Meaning
LOM	Logical Operational Model
MVC	Model / View / Controller (pattern)
PaaS	Platform As A Service
PL	Physical Location
PN	Physical Node
POM	Physical Operational Model
PoP	Point Of Presence
OS	Operating System
QoS	Quality of Service
SaaS	Software As A Service
SDLC	Systems Development Lifecycle
TCO	Total Cost of Ownership
UI	User Interface
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
WAS	WebSphere Application Server
XML	eXtensible Markup Language

1.3 References and Related Documents

Table 2. References and related documents.

No	ID	Document Title
[1]	AT_CS00	Case Study Background
[2]	AT_CS01	Requirements Specification

2 Overview of requirements

2.1 Summary of functional requirements

The government of the Republic of Bolumbia, a prosperous country with a population of over 23 million people, has a Department of Statistics (DoS) which collects and analyses information about various aspects of the country including its population, society, health and the economy to name the key areas. The department has been running a Population and Housing Census for over 100 years and in recent history the Census has run every 5 years, the next census being three years from now. The government is looking for an IT partner to deliver the complete Electronic Census System (ECS) solution including implementation, hosting and support.

The following diagram is the System Context for ECS showing ECS as a *black box*, the external actors that interact with ECS and what they do. Note that the use case model provides more detail about the nature of the interaction between the actors and ECS.

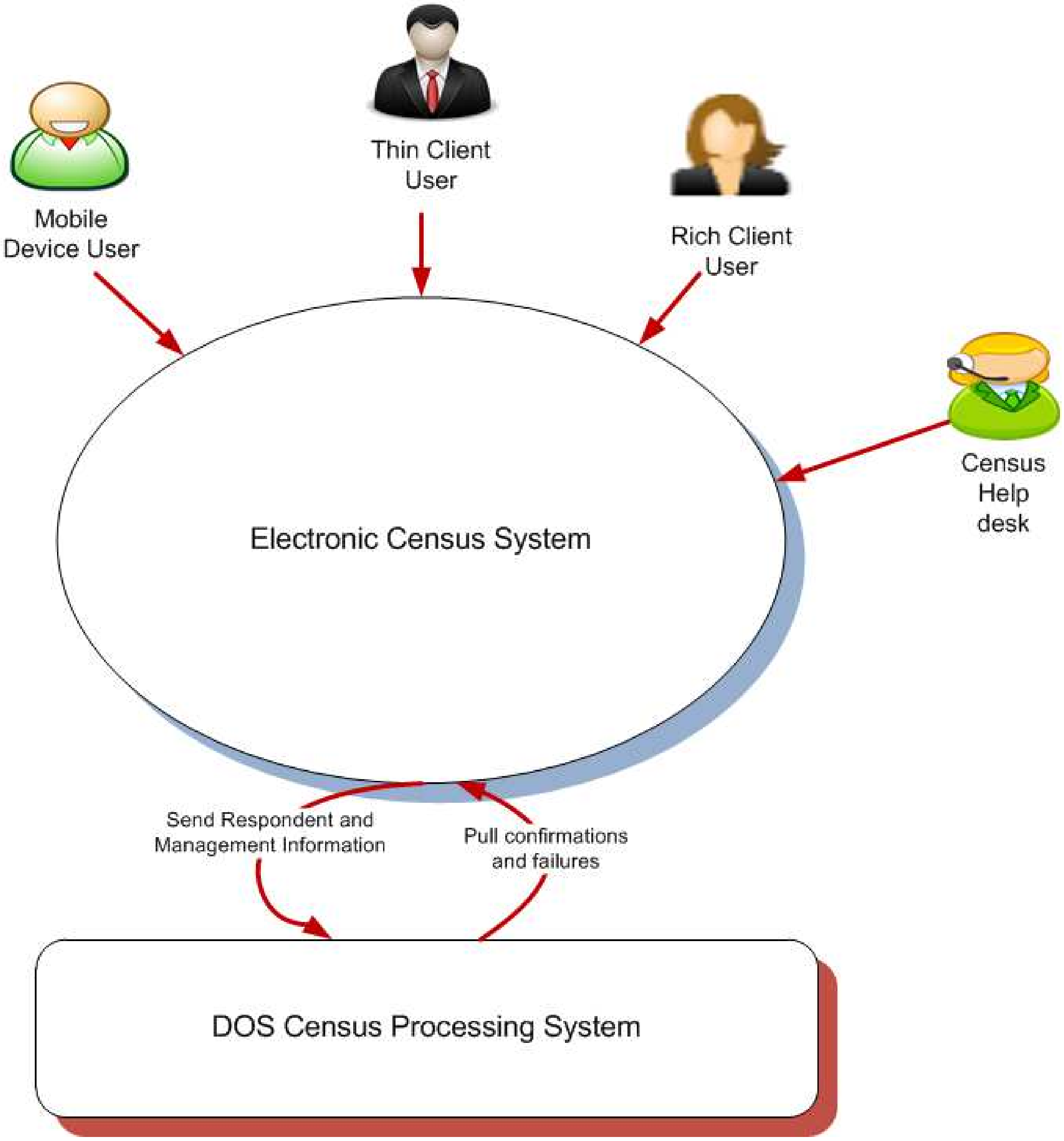

Mobile Device User
Thin Client User
Rich Client User
Census Help desk
Electronic Census System
Send Respondent and Management Information
Pull confirmations and failures
DOS Census Processing System

Figure 1 - System Context

The following diagram and associated table provide an overview of the system functionality required.

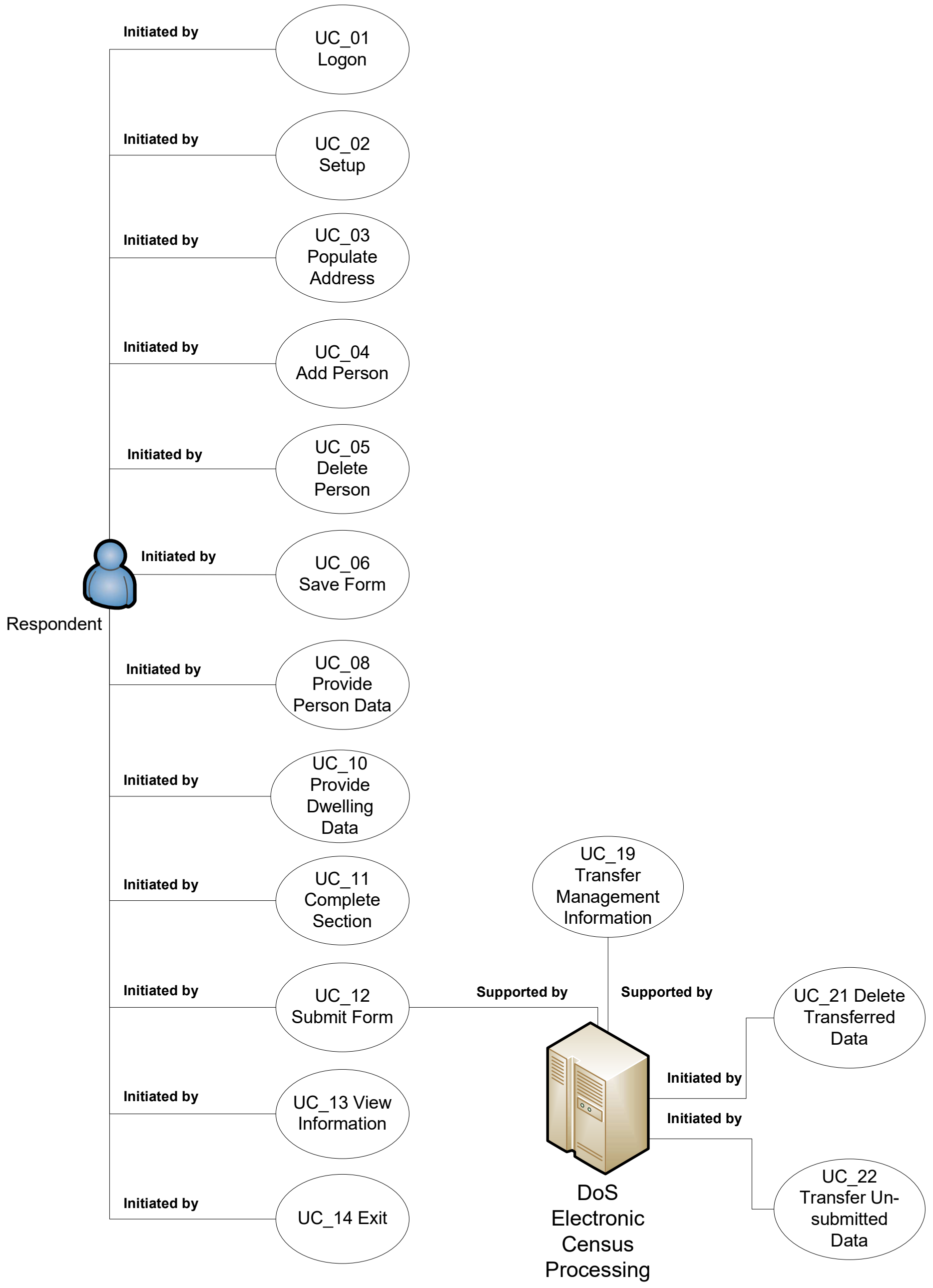


Figure 2 - Use Case Model Diagram

Table 3 - Use Case Overview

User Case ID	*Use case name*	*Description*
UC_01	Logon	The Respondent logs onto the System using a Census Form Number (CFN) and Electronic Census Number (ECN) which constitute logon credentials. Following successful logon, the Respondent is presented with the ECS Census Form.
UC_02	Setup	The Census Form can contain a list of up to 10 people residing in the dwelling on Census night. Setup is used to capture the number of persons present on Census night to configure the person section.
UC_03	Populate Address	Addresses must be completed for all dwellings.

User Case ID	*Use case name*	*Description*
UC_04	Add Person	The Respondent must be able to add, and modify a person at any time prior to submission. For those persons added, where the Respondent indicates that they are to be included in the Census, a new person record is also created for them.
UC_05	Delete Person	The Respondent must be able to delete a person at any time prior to submission.
UC_06	Save Form	A Save of Census data may be requested at the completion of a page, section, on exit or at submission whilst logged onto the System with a current session. The data is stored for later retrieval by the Respondent.
UC_08	Provide Person Data	The Respondent provides answers to the questions contained in the person section.
UC_10	Provide Dwelling Data	The Respondent provides answers to the questions contained in the dwelling section.
UC_11	Complete Section	The Respondent may request completion of a Census section once mandatory questions have been answered. The completion process entails an implicit save initiated by the System. When all sections have a status of complete, the Respondent may request submission of their data to the DoS.
UC_12	Submit Form	The Respondent may submit their Census data only once to the DoS, after completing all sections. After successful submission, the System issues the Respondent with a receipt number. Any subsequent access to the System will provide the Respondent with their receipt number. Respondent is prevented from making changes or resubmitting their Census form.
UC_13	View Information	The Respondent may at any point request a viewing of extra information. This includes links such as Copyright, Conditions of Use, Privacy and Security and Contextual Help.
UC_14	Exit	The Respondent may exit the System at any time with or without saving their Census data. If the Respondent exits without first saving any changes, the System provides the opportunity to save before exiting. The Respondent may decline this opportunity, which results in unsaved data being discarded.
UC_19	Transfer Management Information	The System transfers management information data to the DoS.
UC_21	Delete Transferred Data	The System receives confirmation from the DoS of Respondent Census data that can be deleted, identified by ECN. For each ECN received, the System deletes the corresponding Respondent data.
UC_22	Transfer Un-submitted Data	The System transfers un-submitted Respondent data to the DoS on request.

2.2 Summary of non-functional requirements

Table 4 – NFR Summary

ID	*Category*	*Requirement statement*

ID	***Category***	***Requirement statement***
1	Volumetric	Capacity to support up to 9.5 Million households during the census enumeration period.
2	Volumetric	Peak system load on the first Census night after 5pm
3	Volumetric	Population of Bolumbia is approximately 23 Million people so that many person records need to be accounted for.
4	Performance	The transaction processing time (under ECS control) should be under 3 seconds on average
5	Performance	The target overall response time should be less than 10 seconds on average
6	Hours of Operation	Respondents must be able to logon, complete and submit census responses 24 hours a day during every day of the enumeration period.
7	Availability	The ECS must be available 98% of the time during the hours of operation.
8	Security	Audit trails must be provided as part of the security arrangements for the solution.
9	Accessibility	Support for Chrome, Firefox and Internet Explorer.
10	Accessibility	Support for Android, iOS7 or later and Microsoft Surface.

2.3 Out of scope statements

The ECS proposal does not include:

- System(s) that analyse Census information collected.
- Non-private dwellings: These responses will be paper based for this first release of ECS.

3 Architecture Overview

This section provides the Architecture Overview work product. It contains the solution architecture that facilitates understanding of the detailed solution elements and their mutual relationships described in the remainder of the document. The latter sections provide progressively more detailed views of the solution.

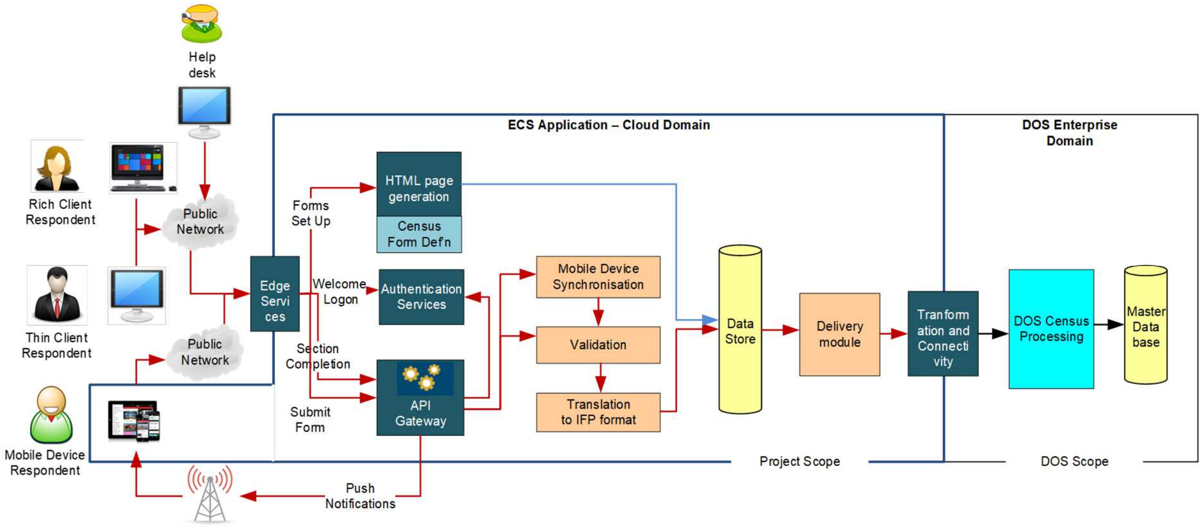


Figure 3 - Architecture Overview Diagram

The following major architectural components of the ECS solution are illustrated in Figure 3 above.

a) Thin client user interface - must meet the accessibility requirements including operation with JavaScript disabled and support for screen readers. ECS application running on the server generates all HTML pages displayed to the respondent and performs all field validation, business rule and mandatory question checks, and subsequent processing. The browser performs the pure presentation layer function, and thus every individual page transition requires interaction with the server.

b) Rich client user interface – given expected high load during peak hours usage (the first day of census period after business hours), thin client user interface is likely to generate substantial performance load on the server. A more efficient approach is to develop a rich ECS client separate from the thin ECS client. The rich AJAX client application runs within the browser and generates all of the HTML pages displayed to the respondent based on definition data, locally. The respondent interacts with the displayed pages and the client performs all field and business rule validations, collecting the respondent and management information within the browser memory. The client application posts the collected respondent and management information to the server at each form completion point and when the respondent submits their data, or creates feedback or a technical help request.

c) Mobile device application – must be available for Android, iOS and Windows mobile devices. The functional of the mobile device application will be like that offered by the Rich client browser application. Additionally, the solution will support "offline" mode of operation whereby census can be filled in by the mobile device user while the device is not connected to any network. The data is temporarily stored on the device and replicated back to the servers as soon as connection to a network is established. The data is removed from the mobile device once it is replicated to the server.

In summary the ECS solution will support; HTML-only client for approximately 2% of web users without Javascript, AJAX client that will provide a richer and more responsive client for the 98% of web users who have Javascript enabled and a mobile device application that allows users to submit their census forms via their mobile devices.

d) Edge Services - provide network capability to deliver content through the Internet (DNS, CDN, firewall, load balancer).

e) API Gateway – Invokes APIs (exposed by Microservices), routes and connects requests from the front-end applications.

f) Authentication – ECS is secured using a combination of Census Form Number (CFN), and ECS Number (ECN). The Authentication module implements the required ECN authentication algorithms and security rules related to number of logins per ECN, IP Lockout and other security features. Validation and authentication of the user-provided CFN and ECN is performed with algorithms provided by the DoS but implemented as part of the ESC.

g) Census Form Definition – all question information (question type, question text, help text, message content), in fact all text information displayed within the census form is defined within XML definition file(s). Thus the ECS application is a relatively generic mass scale internet based questionnaire solution that is configured to fulfil the specific DoS requirements. The configuration data has the following elements:

- Census Page Definition – is an XML file that specifies the overall information for the Navigation panel, and the set of pages that make up the census, and various overall parameters such as the core application button labels (Next, Previous etc.). The census form is composed of multiple sections with questions; each section requires one or more pages to respond to the questions.

- Census Form Definition – is an XML file that specifies each of the questions belonging to a section. This includes the question response type (Text, Date, Address and Selection), the question text, alternate question text, help text, labels, field sizes, field validation rules, question dependency rules, business rules, and all other presentation and functional aspects of each question on the form.

h) HTML page generation – page generation after the Welcome and Login page is performed on the server and rendered on the client side.

i) Validation – this component performs all field (data type/length) validation, mandatory question checking, and business rule error and warning checks. Server side validation is performed on all incoming data submitted by every HTML interface page. The server side validation component is also used to validate all completed form data received from the client at the time of submission. Field and page level validation is performed upon each page submission (from the thin client), while the section and form level validation (cross-page validation) is performed at the time of form submission (from any client).

j) Translation – this component translates the respondent data from the input format to the IFP format required for delivery to DoS. The IFP labels required for the answer data are specified in the Form Data definitions. Translation occurs after validation and prior to the data being stored to the database. Translation from IFP back to the client format is required when a respondent has saved their data on exit and subsequently logs back onto the system.

k) Data Store – the temporary (store and forward) data store contains the respondent's data which is subsequently sent to the DoS' Electronic Census Processing system using asynchronous ‘store & forward’ pattern.

l) Delivery module – this module runs periodically and performs three distinct flows:

- Extracts the respondents’ data from the database and transfers it to DoS.
- Retrieves acknowledgment files from DoS and deletes census records that are positively acknowledged by DoS.
- Retrieves re-submission requests from DoS where respondent’s data has not been successfully processed by DoS. These respondent records will be included in the first sub-sequent extraction and delivery to DoS.

m) Transformation and Connectivity - enables and connects securely between modules running in the cloud and applications running in the DoS enterprise data centre.

n) DoS Census Processing system – this component receives data files, separates out the different data types (collector notifications, technical help requests, respondent data and feedback) processing each as appropriate, and generates acknowledgement and resend request files if required.

4 Functional Viewpoint: Component Model

4.1 Static View

This section documents the static functional view for ECS. The static functional view describes the software components of the system, their responsibilities, relationships and the way they collaborate to implement the required functionality. Figure below shows the component relationship diagram (using UML notation) for ECS showing all functional and key technical components and their dependency relationships.

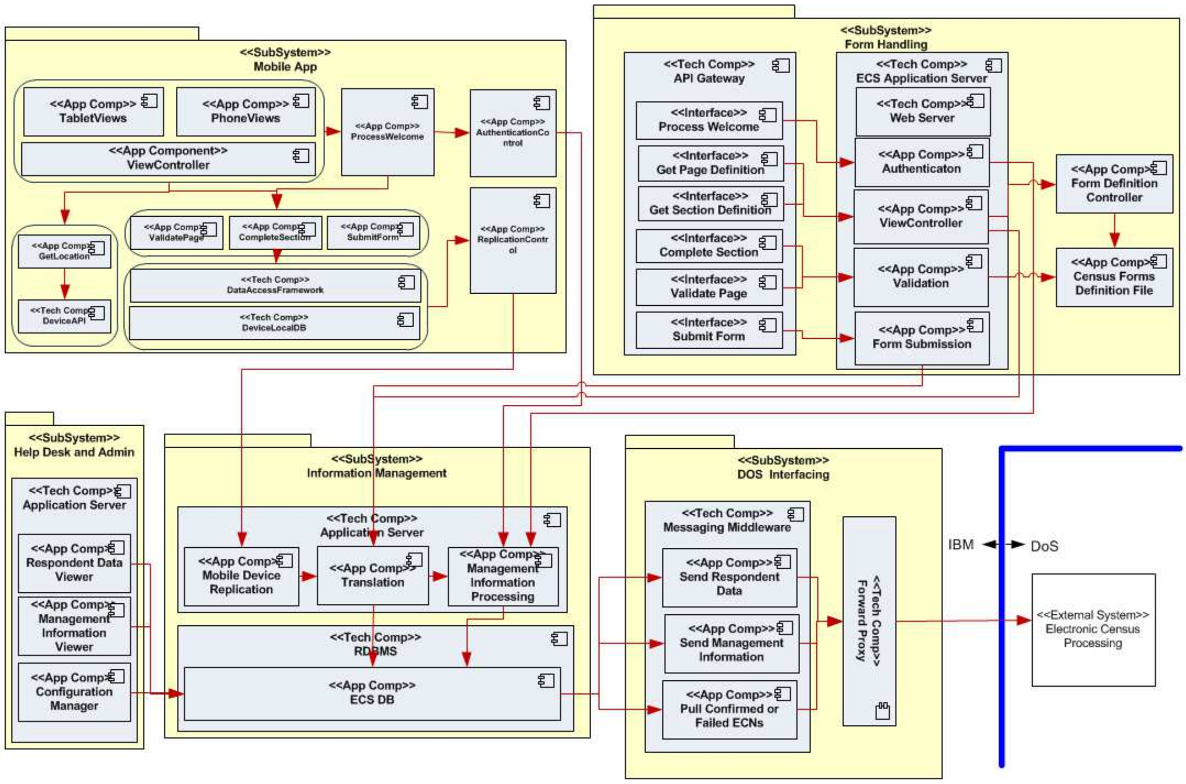

<<SubSystem>>
Mobile App
<<App Comp>> TabletViews
<<App Comp>> PhoneViews
<<App Component>> ViewController
<<App Comp>> ProcessWelcome
<<App Comp>> AuthenticationControl
<<App Comp>> GetLocation
<<Tech Comp>> DeviceAPI
<<App Comp>> ValidatePage
<<App Comp>> CompleteSection
<<App Comp>> SubmitForm
<<Tech Comp>> DataAccessFramework
<<Tech Comp>> DeviceLocalDB
<<App Comp>> ReplicationControl
<<SubSystem>>
Form Handling
<<Tech Comp>> API Gateway
<<Interface>> Process Welcome
<<Interface>> Get Page Definition
<<Interface>> Get Section Definition
<<Interface>> Complete Section
<<Interface>> Validate Page
<<Interface>> Submit Form
<<Tech Comp>> ECS Application Server
<<Tech Comp>> Web Server
<<App Comp>> Authenticaton
<<App Comp>> ViewController
<<App Comp>> Validation
<<App Comp>> Form Submission
<<App Comp>> Form Definition Controller
<<App Comp>> Census Forms Definition File
<<SubSystem>>
Help Desk and Admin
<<Tech Comp>> Application Server
<<App Comp>> Respondent Data Viewer
<<App Comp>> Management Information Viewer
<<App Comp>> Configuration Manager
<<SubSystem>>
Information Management
<<Tech Comp>> Application Server
<<App Comp>> Mobile Device Replication
<<App Comp>> Translation
<<App Comp>> Management Information Processing
<<Tech Comp>> RDBMS
<<App Comp>> ECS DB
<<SubSystem>>
DOS Interfacing
<<Tech Comp>> Messaging Middleware
<<App Comp>> Send Respondent Data
<<App Comp>> Send Management Information
<<App Comp>> Pull Confirmed or Failed ECNs
<<Tech Comp>> Forward Proxy
IBM
DoS
<<External System>> Electronic Census Processing

Figure 4 – ECS Static Functional View

Table 5 describes the responsibilities of each subsystem and component.

Table 5: Functional responsibilities of components

Name	Type	Functionality supported
Mobile App	**Subsystem**	ECS application installed on a mobile device.
Tablet Views	Application Component	A set of UI components (screens or pages and pop-ups) that support the functionality of the tablet style mobile device client application.
Phone Views	Application Component	A set of UI components (screens or pages and pop-ups) that support the functionality of the smart phone style mobile device client application.
View Controller	Application Component	Manages navigation through the UI (implementing controller part of the MVC pattern).

Name	Type	Functionality supported
Process Welcome	Application Component	Processes the initial page and calls authentication module.
Authentication Control	Application Component	Authentication module which validates CFN and ECN and if successful, initiates respondent's session and marks respondent's control status as "started".
Validate Page	Application Component	Performs page validations based on the page definitions and associated rules.
Complete Section	Application Component	Validates final page within section and saves the section data.
Submit Form	Application Component	Final form submission initiating the respondent data being marked as completed.
Data Access Framework	Technical Component	API providing access to the local device database.
Device Local DB	Technical Component	Local mobile device database.
Replication Control	Application Component	Once respondent successfully completes and submits the form, the respondent data is temporarily stored on the device local database until this component is able to establish connectivity with the ECS back end and send the respondent data to the ECS DB.
Get Location	Application Component	Calls device native API to obtain device location information attempting to validate that the respondent is at the dwelling as declared on the form.
Device API	Technical Component	Device native API which supports Hybrid Apps being able to access native device functions & sensors.
Form Handling	**Subsystem**	ECS application UI used by both, the rich client users and the thin client users
API Gateway	Technical Component	Invokes the APIs and connects the request from the front end application.
Process Welcome	Interface	Welcome page processing which in turn invokes authentication module.
Get Page Definition	Interface	Given the page id, returns page related questions and validations rules. Intended for use by thin client UI which operates at the page by page level.
Get Section Definition	Interface	Given the section id, returns section related questions and validations rules. Intended for use by rich client UI which operates at the section by section level.
Validate Page	Interface	Used by the thin client UI to validate pages and save page data before the navigation moves to the next page.
Complete Section	Interface	Used by the rich client UI to validate sections and save data before the navigation moves to the next section.
Submit Form	Interface	Invoked for final validation and submission of the responder data which results in Census form deemed completed.
ECS Application Server	Technical Component	Runtime container for the applications that accomplish business goals.

Name	Type	Functionality supported
Web Server	Technical Component	Technical component delivered as a part of application server. Servers static web content and handles HTTP/HTTPs requests from the client side.
Authentication	Application Component	Performs ECN and CFN validations as defined in the "UC_01 Logon" use case (see Requirements Specification document [2]).
View Controller	Application Component	Returns page level or section level question definitions and corresponding validation rules to be presented and executed by the respondent's browser.
Validation	Application Component	Server side page and cross page (section) level validations that couldn't be enforced by the UI.
Form Submission	Application Component	Final validation and submission of the responder data which results in Census form deemed completed. Updates management information accordingly and renders respondent data ready to be passed onto DoS.
Form Definition Controller	Application Component	Fetches either page or sections related questions, validation rules and any other meta data required by the UI to properly render screens and validate the data entry.
Census Forms Definition File	Application Component	XML file(s) containing both form definitions for each census form as well as related validation rules.
Help Desk and Admin	**Subsystem**	ECS help desk and administration functionality.
Application Server	Technical Component	Runtime container for the applications that accomplish business goals.
Respondent Data Viewer	Application Component	Makes it possible for the support staff to search and retrieve any respondent data currently save on the respondent database
Management Information Viewer	Application Component	Makes it possible for the support staff to retrieve any management data, such as session, number of attempted log ins, state of the respondent data, statistics etc.
Configuration Manager	Application Component	Allows System Admin staff to modify reference data, unlock respondent access etc.
Information Management	**Subsystem**	ECS database, data access framework and associated components
Application Server	Technical Component	Runtime container for the applications that accomplish business goals.
Mobile Device Replication	Application Component	Receives submitted responses from the mobile devices and stores them into the ECS DB.
Translation	Application Component	Translates respondent data from the input format into the IFP format before committing the data to the database.
Management Information Processing	Application Component	Maintains session, status and other meta-data from the respondents' data and records these on the database.
RDBMS	Technical Component	Relational Database Management System hosting ECS DB.

Name	Type	Functionality supported
ECS DB	Application Component	Respondents' data, management information, reference data and form questions and associated validation rules.
DoS Interfacing	**Subsystem**	Interfaces between ECS an DoS enterprise system (Electronic Census Processing)
Messaging Middleware	Technical Component	Specific class of middleware that supports the exchange of general-purpose messages in a distributed application environment. In ECS scenario it is used for interchange of data between ECS and DOS Census Processing.
Send Respondent Data	Application Component	Periodically sends any new respondent records that previously haven't been sent to the DOS Census Processing system, and also resends records that have failed to upload on the DOS side.
Send Management Information	Application Component	Periodically sends any new management information that previously hasn't been sent to the DOS Census Processing system.
Pull Confirmed or Failed ECNs	Application Component	Periodically pulls status records (acknowledgements or errors) indicating outcome of the data upload into the DOS Census Processing system.
Forward Proxy	Technical Component	A proxy configured to handle requests from the ECS to a specific group of resources that are available in the DOS domain supporting interfacing between ECS and DOS systems.

4.2 Data Model

The data model defines main business entities featuring on the ECS database and the way these entities are connected to each other and how they are processed and stored inside the system. The data model explicitly determines the structure of data.

The significance of the data model depends of the nature of the solution being implemented. In the case of bespoke solutions like ECS, the data model informs design of the components that operate on the data. In the case of COTS packages, the data model is defined by the package vendor and is used by architects and designers to better understand package functionality.

High level ECS data model is depicted in the Figure 5 – High Level ECS Data Model.

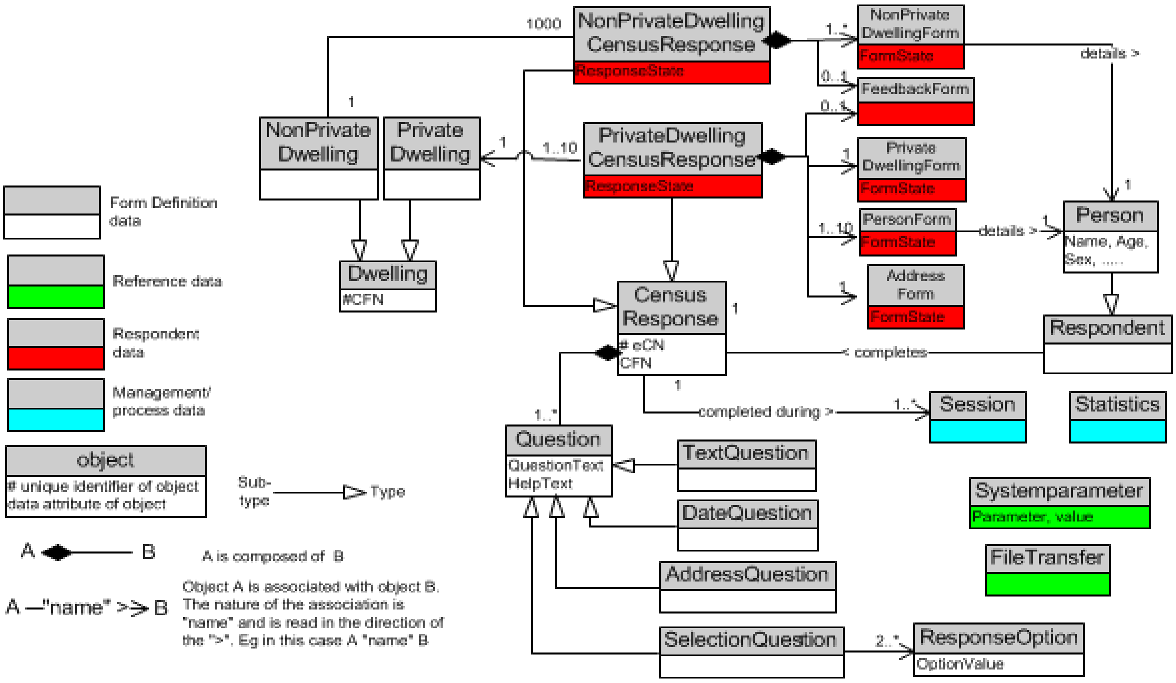


Figure 5 – High Level ECS Data Model

4.3 Dynamic Views

This section presents interaction diagrams for some of the most architecturally significant use cases.

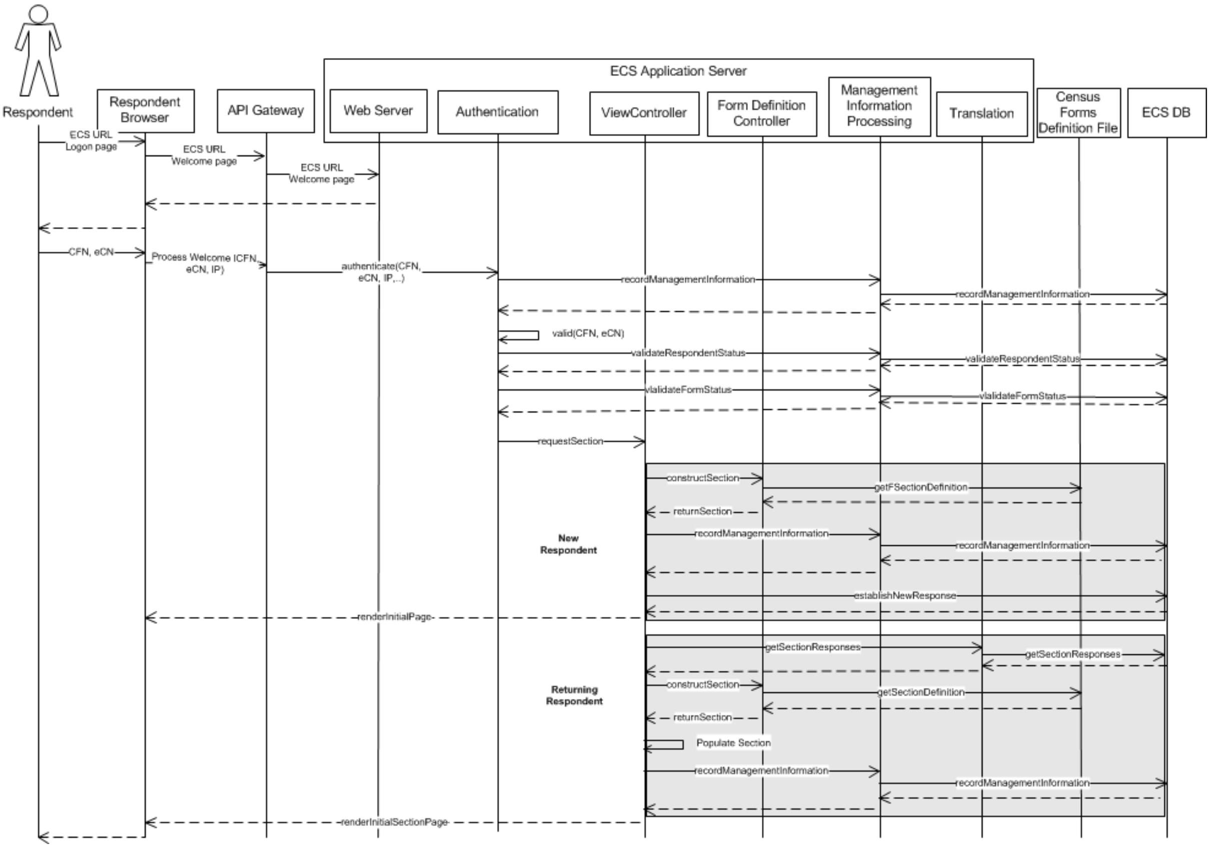


Figure 6 – UC_01 Logon

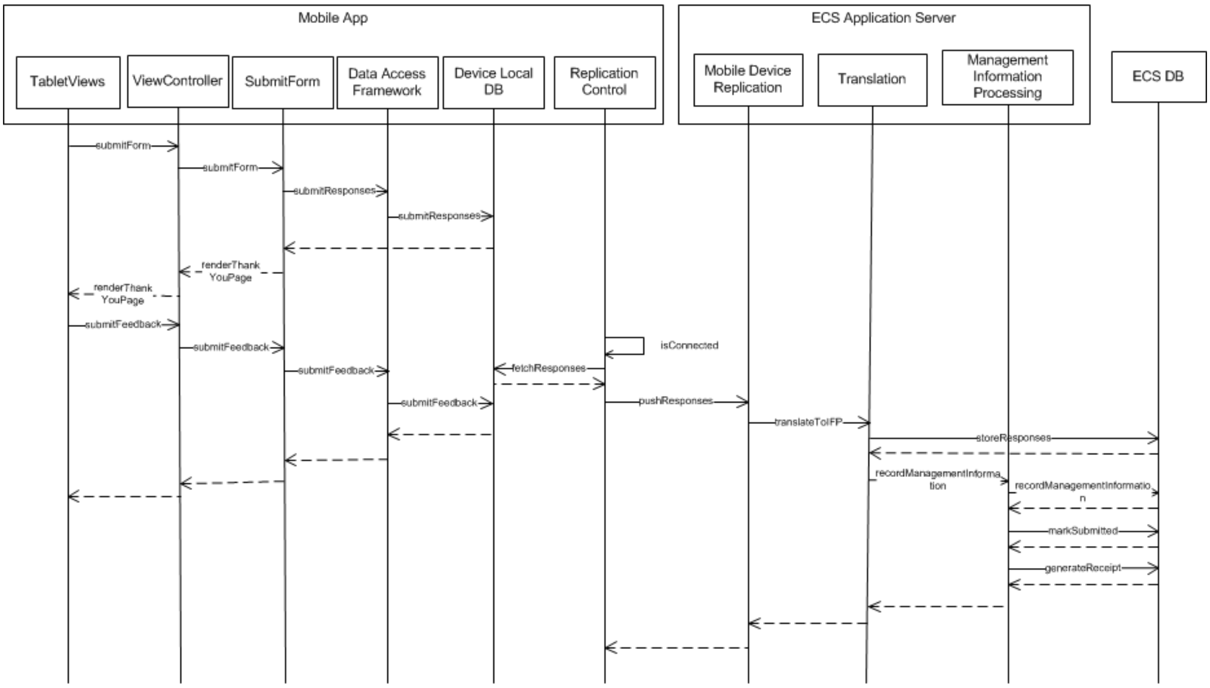


Figure 7 – UC_12 Submit Form (from mobile device)

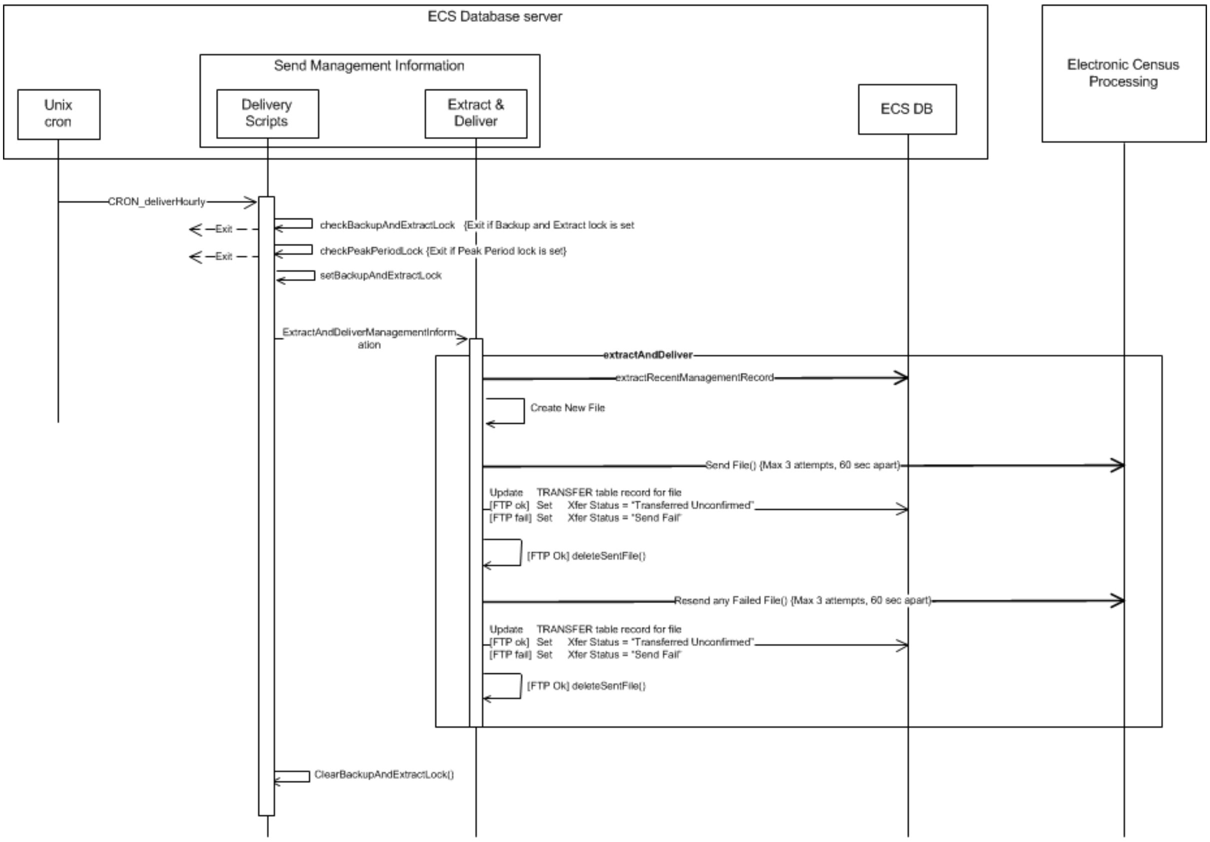

ECS Database server
Send Management Information
Unix cron
Delivery Scripts
Extract & Deliver
ECS DB
Electronic Census Processing
CRON_deliverHourly
Exit
Exit
checkBackupAndExtractLock {Exit if Backup and Extract lock is set
checkPeakPeriodLock {Exit if Peak Period lock is set}
setBackupAndExtractLock
ExtractAndDeliverManagementInformation
extractAndDeliver
extractRecentManagementRecord
Create New File
Send File() {Max 3 attempts, 60 sec apart}
Update TRANSFER table record for file
[FTP ok] Set Xfer Status = "Transferred Unconfirmed"
[FTP fail] Set Xfer Status = "Send Fail"
[FTP Ok] deleteSentFile()
Resend any Failed File() {Max 3 attempts, 60 sec apart}
Update TRANSFER table record for file
[FTP ok] Set Xfer Status = "Transferred Unconfirmed"
[FTP fail] Set Xfer Status = "Send Fail"
[FTP Ok] deleteSentFile()
ClearBackupAndExtractLock()

Figure 8 – UC_19 Transfer Management Information

4.4 Realisation Decisions

This section presents the realization decisions with regard to the major components of the solution. For each subsystem or component a decision is made whether it will be acquired, subscribed to (or bought if not cloud subscription model) or built. This information is summarized in Table 6.

Table 6: Component realisation decisions

No	Sub System / Component	Product / Platform	Acquire / Subscribe / Build
1.	Mobile App	**Apache Cordova** – using hybrid pattern with pre-packaged HTML 5 resources to allow for operation in the disconnected mode	Acquire (open source)
		SQLite	Acquire (public domain)
2.	API Gateway	**IBM API Connect**	Subscribe
3.	Application Server	**IBM WAS**	Subscribe
4.	Form Handling	**J2EE, Spring MVC**	Build
5.	RDBMS	**IBM dashDB**	Subscribe
6.	Forward Proxy	**IBM DataPower Gateway**	Subscribe
7.			

5 Operational Viewpoint

5.1 Logical Location View

The first step in developing the operational viewpoint is to develop a Logical Location View. The initial view of locations is listed in the following table:

Table 7 - Locations

Location ID	Description	Cardinality
LL_01	LL_External Respondents	10,000,000
LL_02	LL_External Client Data Centre	1
LL_03	LL_External Client Users	1
LL_04	LL_Cloud Secure (DMZ)	1
LL_05	LL_Cloud Data Centre Enterprise	1
LL_06	LL_External 3rd Parties	3

The diagram below shows a Logical Location View (Model) for the DoS ECS solution.

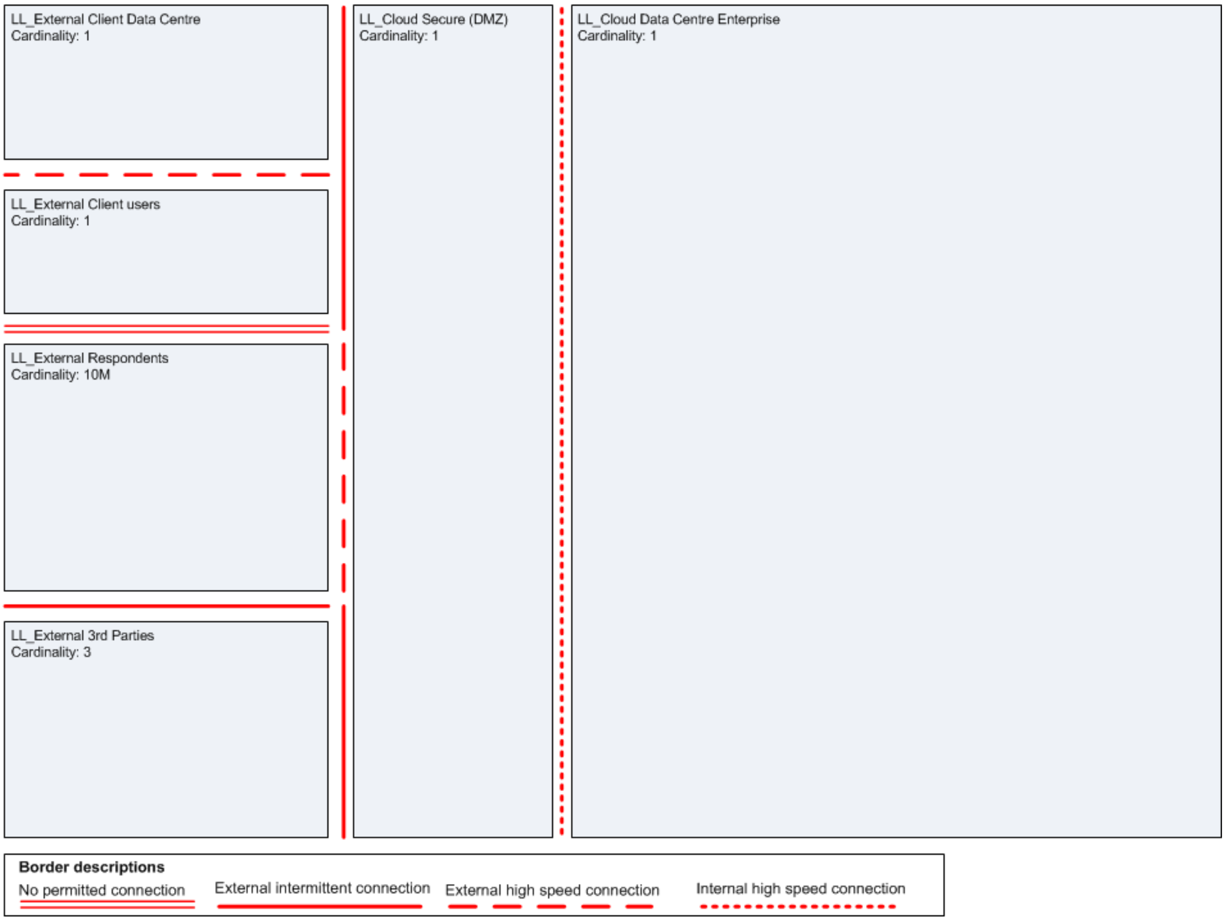
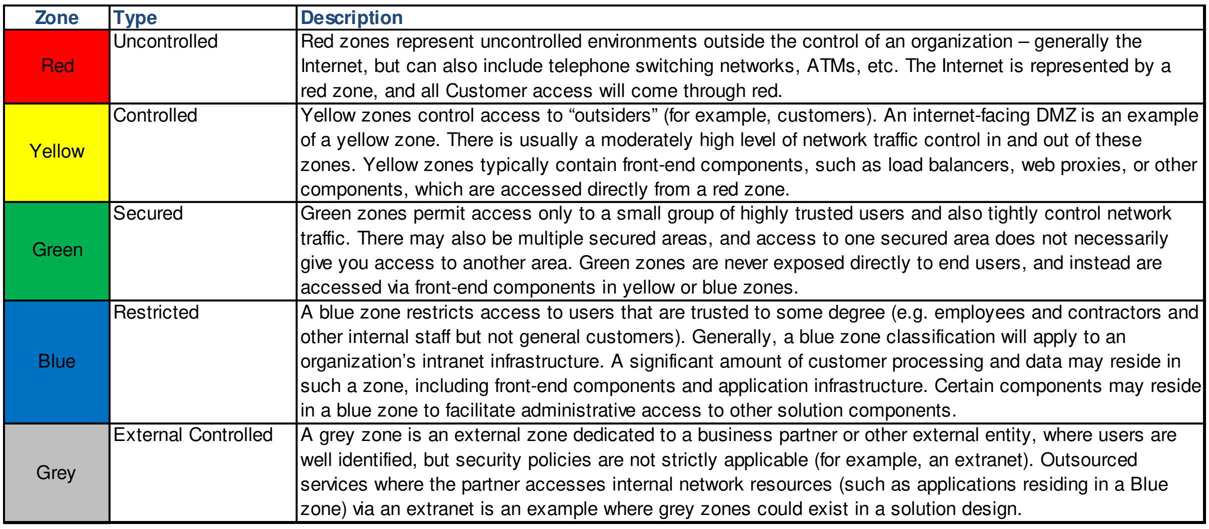


Figure 9 - Logical Location View

The logical locations will be mapped to security zones in the logical operational view later in this document. We use the IBM standard for modelling security zones. The following table shows a summary of the Zone Model.



Zone	Type	Description
Red	Uncontrolled	Red zones represent uncontrolled environments outside the control of an organization – generally the Internet, but can also include telephone switching networks, ATMs, etc. The Internet is represented by a red zone, and all Customer access will come through red.
Yellow	Controlled	Yellow zones control access to “outsiders” (for example, customers). An internet-facing DMZ is an example of a yellow zone. There is usually a moderately high level of network traffic control in and out of these zones. Yellow zones typically contain front-end components, such as load balancers, web proxies, or other components, which are accessed directly from a red zone.
Green	Secured	Green zones permit access only to a small group of highly trusted users and also tightly control network traffic. There may also be multiple secured areas, and access to one secured area does not necessarily give you access to another area. Green zones are never exposed directly to end users, and instead are accessed via front-end components in yellow or blue zones.
Blue	Restricted	A blue zone restricts access to users that are trusted to some degree (e.g. employees and contractors and other internal staff but not general customers). Generally, a blue zone classification will apply to an organization’s intranet infrastructure. A significant amount of customer processing and data may reside in such a zone, including front-end components and application infrastructure. Certain components may reside in a blue zone to facilitate administrative access to other solution components.
Grey	External Controlled	A grey zone is an external zone dedicated to a business partner or other external entity, where users are well identified, but security policies are not strictly applicable (for example, an extranet). Outsourced services where the partner accesses internal network resources (such as applications residing in a Blue zone) via an extranet is an example where grey zones could exist in a solution design.

Figure 10 - Zone Model

5.2 Deployment Unit Model

The following table shows the Deployment Unit Model, which contains a mapping between the Components (defined in the Functional Viewpoint: Component Model) to defined Deployment Units (Presentation, Execution, Data, or Installation).

Table 8 - Deployment Unit Mappings

Sub-system / Layer	Component	Presentation DU	Data DU	Execution DU	Installation DU	Comments
Mobile App	Tablet Views	U_01: U_TabletUI (Android) U_02: U_TabletUI (iOS) U_03: U_TabletUI (Windows)			I_01: I_MobileApplication (Android) I_02: I_MobileApplication (IOS) I_03: I_MobileApplication (Windows)	Authent&Author is included in UI
	Phone Views	U_04: U_PhoneUI (Android) U_05: U_PhoneUI (iOS) U_06: U_PhoneUI (Windows)				Authent&Author is included in UI
	View Controller			E_01: E_MobileApp (Android) E_02: E_MobileApp (iOS) E_03: E_MobileApp (Windows)		
	Process Welcome					
	Authentication Control					
	Replication Control					
	Validate Page					
	Complete Section					
	Submit Form					
	Get Location					
	Device API					
	Data Access Framework					
	Device Local DB		D_01: d_RespondentData D_02: d_FormData	E_04: E_DeviceLocalDB (Android) E_05: E_DeviceLocalDB (iOS)		
Form Handling	API Gateway			E_07: E_APIGateway		
	Application Server			E_08: E_FormHandlingAppServer		
	Web Server			E_09: E_FormHandlingWebServer		
	Authentication			E_10: E_Authentication		
	View Controller			E_11: E_ViewController		
	Validation			E_12: E_Validation		
	Form Submission			E_13: E_FormSubmission		
	Form Definition Controller			E_14: E_FormDefinitionController		
	Census Forms Definition File		D_03: D_FormData			
Information Management	Application Server			E_15: E_InfoManagementAppServer		
	Mobile Device Replication			E_16: E_MobileDeviceReplication		
	Translation			E_17: E_Translation		
	Management Information Processing			E_18: E_MgtInfoProcess		
	RDBMS			E_19: E_RDBMS		
	Respondent DB		D_04: D_RespondentData			
DOS Interfacing	Messaging Middleware			E_20: E_Middleware		Underlying technical
	Send Respondent Data			E_21: E_SendRespondentData		Custom component
	Send Management Information			E_22: E_SendManagementInfo		Custom component
	Pull Confirmed or Failed ECNs			E_23: E_PullECNs		Custom component
	Forward Proxy			E_24: E_ForwardProxy		
Help Desk & Admin	Application Server			E_25: E_HelpDeskAppServer		
	Respondent Data Viewer	U_07: U_RespondentViewer		E_26: E_RespondentViewer		
	Management Information Viewer	U_08: U_ManagementInfoViewer		E_27: E_ManagementInfoViewer		
	Configuration Manager	U_09: U_ConfigManagerUI		E_28: E_ConfigManager		

5.3 Logical Operational Model

The following diagram shows a Logical Operational View of the ECS solution. Identified Actors, along with Logical Nodes, are placed onto the Logical Location View developed earlier. In addition, the identified Deployment Units are "deployed" onto Nodes in the model. The ECS solution is consistent with a generic web application hosting pattern.

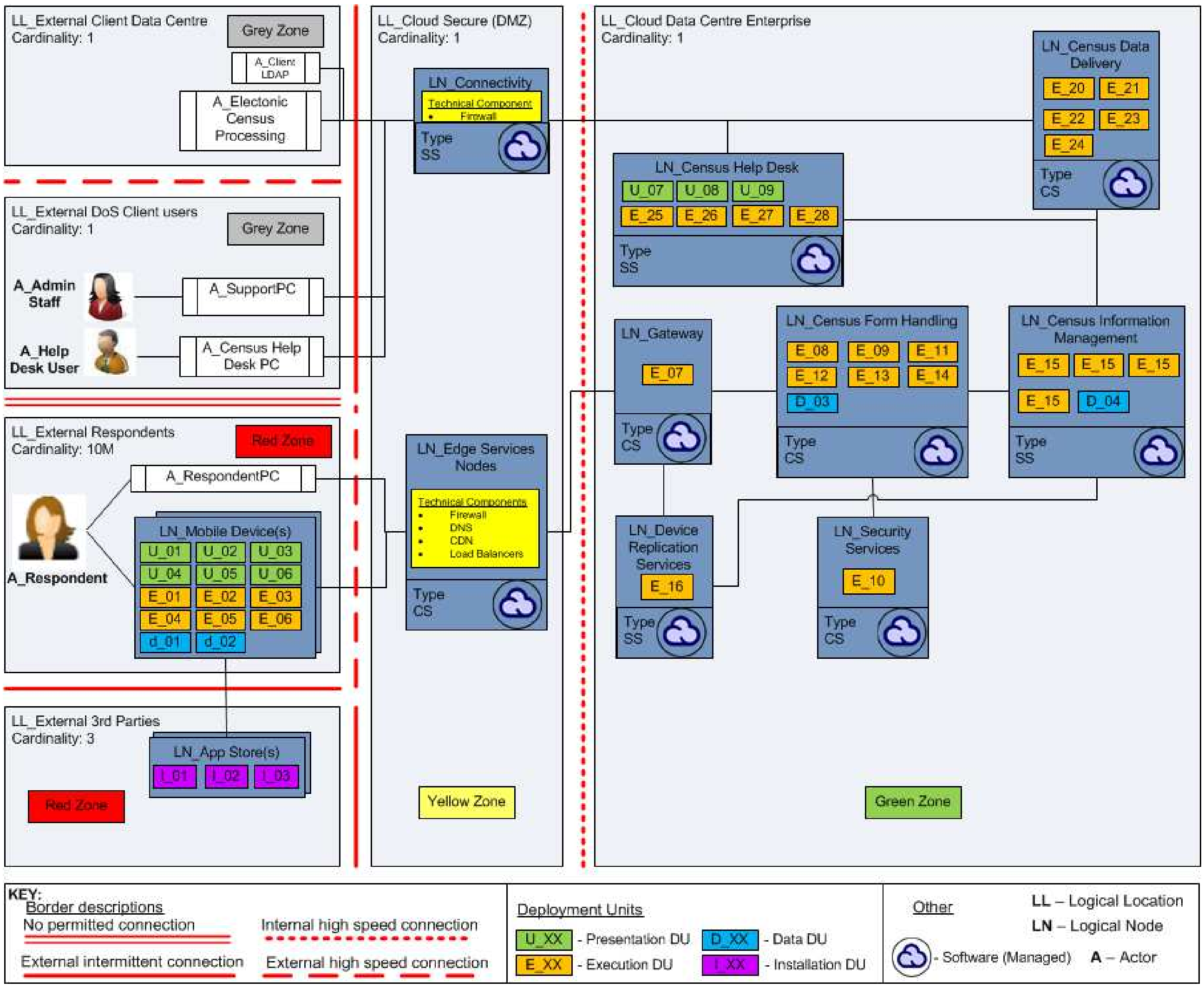


Figure 11 - Logical Operational Model

5.4 Cloud Computing Considerations

There are many choices to be made when considering a Cloud Computing-based deployment of a solution. Some of these are discussed in the subsections below.

5.4.1 Cloud Options

There are many benefits in using cloud based services over traditional in-house implementation, including lower TCO, faster deployment, more flexibility to scale, etc… However, there are also limitations, such as less flexibility in solution/software options, potentially more integration, more security to consider, etc… It is important to consider the pros and cons as they relate to a given situation and set of requirements. In the case of Bolumbia's ECS, there are requirements that impose limitations on the type of Cloud solution that can be used.

5.4.1.1 Cloud Deployment Model

What type of cloud infrastructure is needed for the solution? In Cloud Computing solutions, there are three basic infrastructure deployment models available[1] – Public Cloud, Private Cloud, or Hybrid Cloud.

Table 9 - Cloud Deployment Models

Public cloud is a publicly accessible cloud environment owned by a third-party cloud provider. The customer has no visibility and control over where the computing infrastructure is hosted and the infrastructure is usually shared amongst many organisations.

Private cloud is dedicated to a single organisation. The computing infrastructure is dedicated to that that organisation and is not shared with other clients. Such cloud infrastructure can be located on premises or externally hosted. Generally, private clouds are more expensive than public clouds but can also be more secure.

Hybrid cloud is a combination of either public cloud, private cloud, or in-house deployments.

The Bolumbian census enumeration period has a short duration so the infrastructure used to gather the responses will only be needed for a very limited period. Cloud Computing offers both flexibility and cost savings in comparison to deploying a dedicated solution within DoS or IBM data centres.

The client has stringent requirements relating to a cloud-based solution, including use of dedicated hardware, mandating deployment within Bolumbia, and using a private network. In addition, the solution needs to integrate with the client's ECP system that is hosted in the client's data centre. By definition, a cloud-based solution will use a hybrid cloud model. Dedicated infrastructure is required to implement the ECS within the cloud environment.

5.4.1.2 Cloud Service Model

Another key consideration is the type of cloud service model to use for the solution. The three basic cloud services models that are made available by cloud service providers are:

- **Infrastructure as a Service (IaaS)** – is the most flexible and requires selection of the infrastructure needed to deploy a cloud solution described by well-defined infrastructure requirements such as processor, memory, storage, and network. All elements of the infrastructure will need to be provisioned (or instantiated), although a service provider may offer some of the elements in a "ready to run" form.
- **Platform as a Service (PaaS)** - provides a platform allowing customers to develop, run, and manage applications without the complexity of building and maintaining the underlying infrastructure typically associated with developing and launching an application (e.g.

[1] Other variations exist (e.g. Community Cloud) but are not discussed here.

containers, middleware, development platforms, etc…). Although this model is simpler, and generally faster to implement, it sometimes has less flexibility than an IaaS service.

- **Software as a Service (SaaS)** – provides a complete software solution "in a box" that allows use of the software in cloud that is priced as "pay-per-use". It eliminates the requirements to manage the infrastructure, software license, and maintenance costs required to run the solution. This option provides the least flexibility.

Table 10 - Cloud Service Models

Traditional In-House	IaaS	PaaS	SaaS
Business Process	Business Process	Business Process	Business Process
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualisation	Virtualisation	Virtualisation	Virtualisation
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

Client manages
Cloud Provider manages

As the project scope clearly mandates using a cloud-based environment for this implementation, the traditional "in-house" implementation is not an option for the ECS. Also, since the ECS includes bespoke applications (web and mobile) this rules out SaaS as an option. That leaves only two options under consideration, namely – an IaaS or a PaaS.

Since the ECS is a new custom built solution it appears to be a good fit for a PaaS service model implementation. Using the PaaS model allows for a faster development and deployment of the application, as well as for a rapid expansion of capacity through elastic scaling capabilities (if the number of concurrent users happens to be higher than anticipated. The PaaS service model is also expected to result in a lower TCO.

It is proposed to implement Bolumbia's ECS using the PaaS cloud service model.

5.4.2 Selecting a cloud provider

Although most of the solution will be implemented on the cloud, the same architectural issues and challenges still need to be addressed as if it was implemented using a more traditional “in-house” on-premises deployment. Architects will still need to consider aspects such as component placement, availability, security, performance & capacity, governance, and operations.

In addition, a cloud service provider needs to be chosen to host a cloud-based solution. Whereas the traditional “in-house” solution option requires architects to expend the effort making decisions on the hardware and software (based on specific criteria), a cloud-based solution requires the architects to select the Cloud provider as well.

The relevant considerations may include:

- Availability of the cloud platform within the client’s jurisdiction. For ECS, one of the Non-Functional Requirements stipulates that the solution must be hosted within Bolumbia.
- Enterprise strength of the services offered by cloud providers, in particular security, availability, backup and disaster recovery.
- The level of support offered by the cloud provider
- Network connectivity to the cloud provider’s data centre. One of the disadvantages of a cloud based solution is the potential for increased latency.
- Content of the cloud provider’s service catalogue (e.g. operating system, databases, middleware, etc.).

5.4.3 Configuring the Cloud environment

Most cloud providers (e.g. AWS, IBM Bluemix, Azure, etc…) offer a set service catalogue and configuration options to choose from. One is then faced with the task of configuring the cloud-based platform in line with the solution requirements. This task is similar to that of defining a Bill of Materials for a data centre-based solution. The Operational Model of the proposed solution is one of the key inputs for this activity.

Operational modelling remains a key architectural activity (as well as a key artefact) for elaboration of the solution. This holds true even for such cloud-based solutions such as IaaS and PaaS.

Conventional operational modelling is mainly driven by the NFRs with some input from Functional Requirements. It is also largely influenced by constraints such as the client’s existing environment, standards, and guidelines. There exists an additional architectural constraint to consider when it comes to both IaaS and PaaS solutions. The physical operational model needs to conform to the services offered by the cloud service provider.

5.4.4 Software Options

From an IaaS perspective, there also may be constraints related to the type of operating system available within the cloud environment as well as its version/patch level that can be selected. From a PaaS perspective, the choice of a runtime platform is constrained to what is offered by the cloud provider. Other capabilities required to deliver the end-to-end solution are also limited by the service offerings made available by the cloud provider, e.g. firewalls, load balancers, secure connections, data storage, etc… These also present constraints that the solution needs to satisfy.

Selecting such services and then configuring them to work with the solution rather than implementing them from scratch constitutes a major departure from traditional implementation approaches for “in-house” infrastructures.

5.4.5 Other Considerations[2]

Lifecycle, operations, security, governance, and other requirements also need to be considered and addressed. Location of components (i.e. where they are deployed) will significantly influence the implementation of management and governance. Private cloud environments may be able to use existing internal management and governance tools if the tools have access to the target cloud infrastructure.

Similarly, operational monitoring and management capabilities – i.e. gathering metrics, checking SLAs, status, notifications, and negotiating changes in capacity – require that access to the relevant cloud service administrative interfaces are available and support for these is added to existing management tools. This may include integrating data, information, tools, and processes from multiple sources into common interfaces, reports, and automation tools for efficient and scalable operations.

Governance and compliance processes need to accommodate the change in control and risk over any externally hosted components, especially where changes are controlled by the cloud service provider.

5.4.6 Reference models

Depicted below is the Cloud Standards Customer Council (CSCC) web application hosting cloud reference architecture. It covers an end-to-end scenario of a user accessing a web application on cloud infrastructure. The ECS solution is consistent with this reference architecture, with some additional aspects (e.g. support for mobile and the need to securely transfer data across to a client hosted data centre).

[2] http://www.cloud-council.org/deliverables/CSCC-Cloud-Customer-Architecture-for-Web-Application-Hosting.pdf

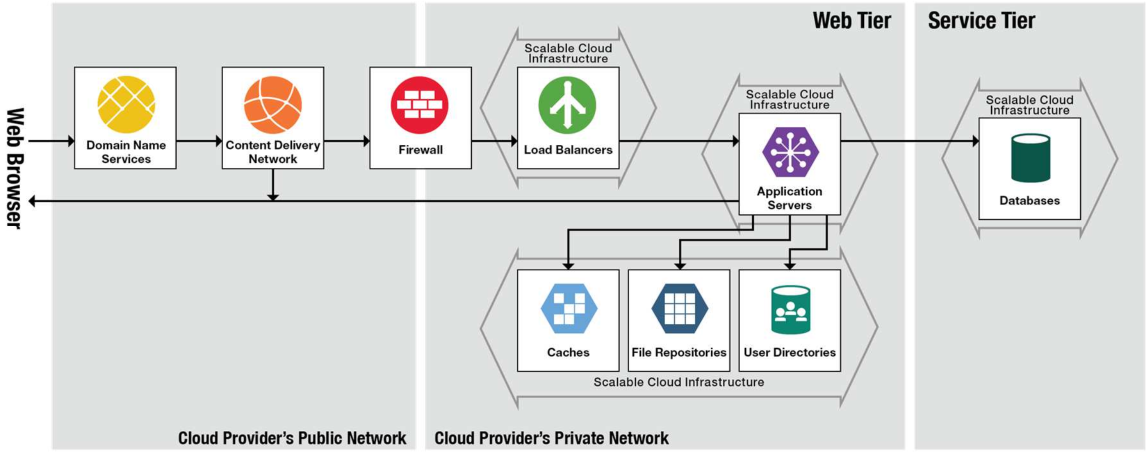
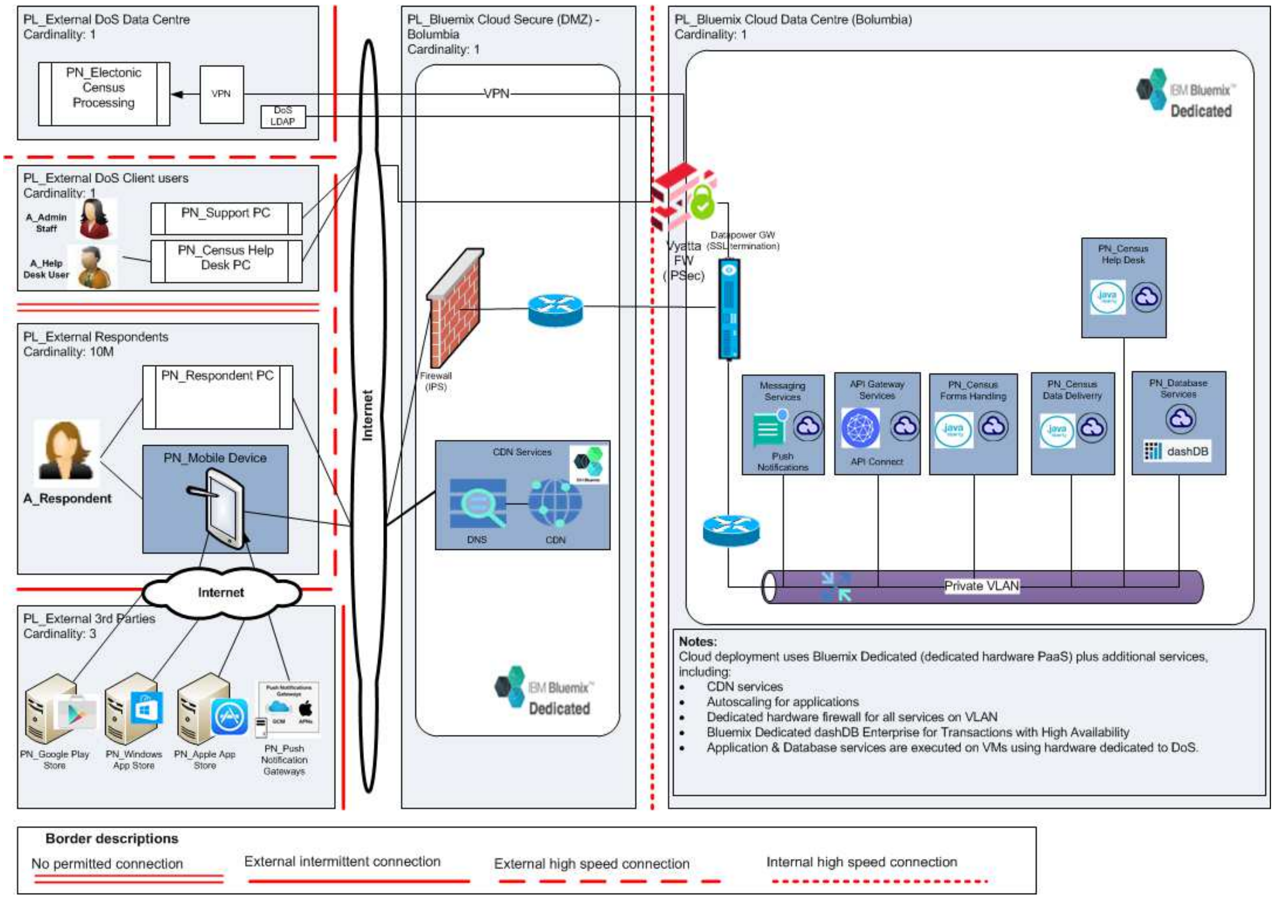

Web Browser
Domain Name Services
Content Delivery Network
Firewall
Scalable Cloud Infrastructure
Load Balancers
Web Tier
Scalable Cloud Infrastructure
Application Servers
Service Tier
Scalable Cloud Infrastructure
Databases
Caches
File Repositories
User Directories
Scalable Cloud Infrastructure
Cloud Provider's Public Network
Cloud Provider's Private Network

Figure 12 - CSCC Web App Hosting Cloud Reference Architecture

5.5 Physical Operational Model

The following diagram shows the physical operational view of the ECS solution.



Notes:
Cloud deployment uses Bluemix Dedicated (dedicated hardware PaaS) plus additional services, including:

- CDN services
- Autoscaling for applications
- Dedicated hardware firewall for all services on VLAN
- Bluemix Dedicated dashDB Enterprise for Transactions with High Availability
- Application & Database services are executed on VMs using hardware dedicated to DoS.

Figure 13 – Physical Operational Model

5.5.1 Key Operational Elements

Bluemix Dedicated is a cloud offering chosen for the ECS solution. Bluemix Dedicated is a PaaS offering that uses a dedicated hardware in a single tenanted cloud environment. The way all of the Bluemix infrastructure components, based on the Cloud Foundry framework, together with the IBM management components are deployed satisfies high availability requirements. Redundant infrastructure is used to achieve high availability.

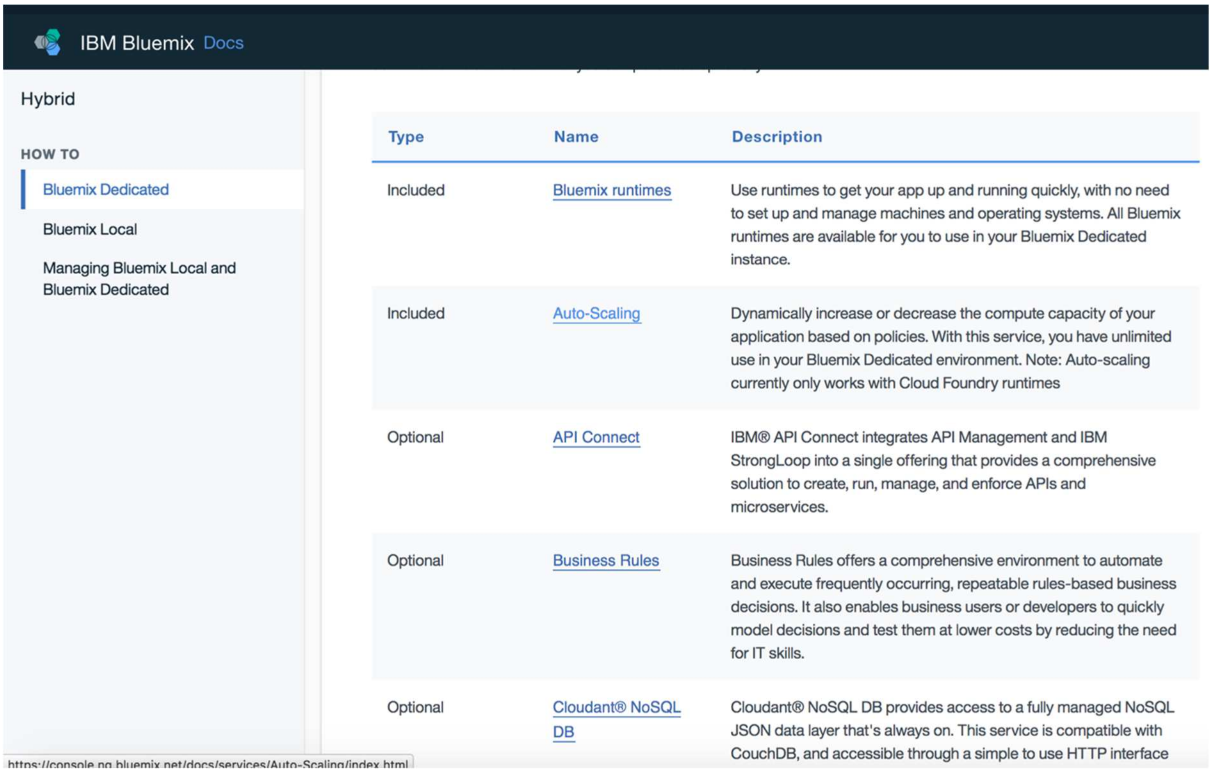
Key aspects of the Bluemix Dedicated offering include:

- VPN connectivity back to the DoS Data Centre to support encrypted transmission of respondent data (using a Dedicated 1 Gbps **Vyatta device**);
- Private Virtual Local Area Network (VLAN) to separate the ECS application from other tenants within the cloud hosting centre;
- The Bluemix hosting site is located within Bolumbia; and
- Dedicated Hardware;

The key elements making up the POM, as chosen from the Bluemix Dedicated catalogue, are discussed below (beginning with an introduction to the catalogue).

5.5.1.1 Cloud Provider Catalogues

As discussed earlier, one of the trade-offs in choosing a cloud provider, and a PaaS model, is the constraints this imposes on the decisions an architect can make when moving from a logical model to a physical one. Bluemix Dedicated provides a catalogue of services and components from which to choose the physical elements of an operational model (server hardware and OS platform are selectable). Some of these services are included as default with Bluemix Dedicated and others are optional offerings that will cost additional. A sample of the catalogue is depicted below:


IBM Bluemix Docs
Hybrid
HOW TO
Bluemix Dedicated
Bluemix Local
Managing Bluemix Local and Bluemix Dedicated
https://console.ng.bluemix.net/docs/services/Auto-Scaling/index.html

Type	Name	Description
Included	Bluemix runtimes	Use runtimes to get your app up and running quickly, with no need to set up and manage machines and operating systems. All Bluemix runtimes are available for you to use in your Bluemix Dedicated instance.
Included	Auto-Scaling	Dynamically increase or decrease the compute capacity of your application based on policies. With this service, you have unlimited use in your Bluemix Dedicated environment. Note: Auto-scaling currently only works with Cloud Foundry runtimes
Optional	API Connect	IBM® API Connect integrates API Management and IBM StrongLoop into a single offering that provides a comprehensive solution to create, run, manage, and enforce APIs and microservices.
Optional	Business Rules	Business Rules offers a comprehensive environment to automate and execute frequently occurring, repeatable rules-based business decisions. It also enables business users or developers to quickly model decisions and test them at lower costs by reducing the need for IT skills.
Optional	Cloudant® NoSQL DB	Cloudant® NoSQL DB provides access to a fully managed NoSQL JSON data layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface

Figure 14 - Bluemix Dedicated Catalogue (sample)

5.5.1.2 Edge Services

The Bluemix **Content Delivery Network (CDN)** is used to serve all static web content, reducing the capacity requirements of the entry firewalls. The CDN acts as a “pull caching proxy server” for the static content of the parent web address, in this case ECS. The CDN service includes content delivery network services and DNS services.

CDN services were not default options for Bluemix Dedicated and need to be arranged separately.

Firewall Services are included in the Bluemix Dedicated offering. A dedicated firewall is provided to protect all of the dedicated servers used to host the elements making up the ECS solution for DoS.

5.5.1.3 API Gateway Services

API Gateway services are implemented using **IBM API Connect.** It will provide the “front door” access to the ECS micro services for handling form collection and submission for both web and mobile device based access.

Bluemix Push Notification services will implement the messaging services to send notifications to users of the mobile ECS application.

5.5.1.4 Census Application Services

The application services are used for the ECS solution and cover three core application subsystems, these are:

- Forms Handling
- Information Management
- Interfacing with DOS (or Census Data Delivery)

These services are implemented using the **Liberty for Java** runtime platform in Bluemix Dedicated.

5.5.1.5 Database Services

The ECS database services will be provided using **IBM dashDB** on Bluemix. The Enterprise for Transactions High Availability configuration option was chosen to support the loads expected for ECS. This delivers a 128GB RAM and 1.4 TB SSD of storage for data and logs, with additional Standby server for high availability.

5.5.1.6 High Availability

There are multiple ways to provide high availability in this solution. There are 3 aspects to this:

- Bluemix Platform (Cloud Foundry)
- Bluemix Infrastructure
- Application components and HA choices

Bluemix Platform (Cloud Foundry)

Firstly, high availability is built into the solution, in part, through technologies included in Cloud Foundry (the open source PaaS sitting underneath Bluemix). Applications on Cloud Foundry run in VMs within execution pools. With enough compute resources available Bluemix can be configured to increase the pool of VMs running the ECS application instances. This can be for redundancy or scalability purposes.

Bluemix Infrastructure

The second aspect to HA in the ECS solution involves each cloud storage cluster, within a Bluemix Infrastructure dedicated environment, being written multiple times, and can be configured to attempt restart of a virtual server on another host if a failure is detected.

Application Components and HA choices

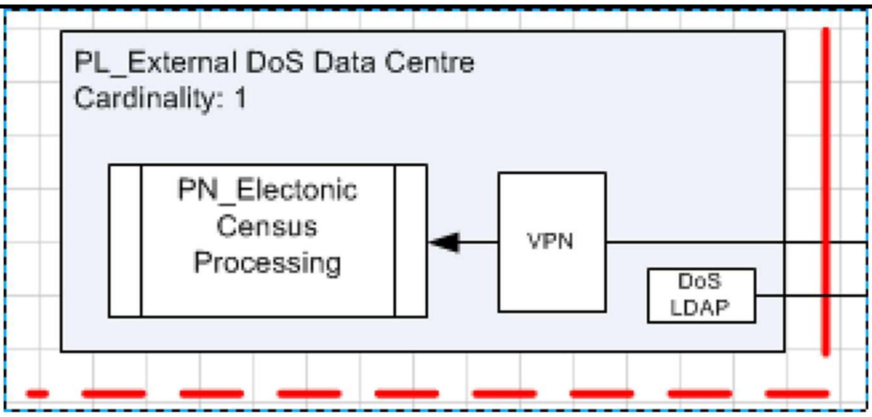
The final aspect involves the applications/services themselves. For the ECS solution the application components are implemented using the Liberty for Java runtime, which will be scaled for multiple instances (both for availability and redundancy). The dashDB service is configured with the High Availability option with a Standby in place. Other components of the solution are also configured with high availability options.

5.5.1.7 Performance & Scalability

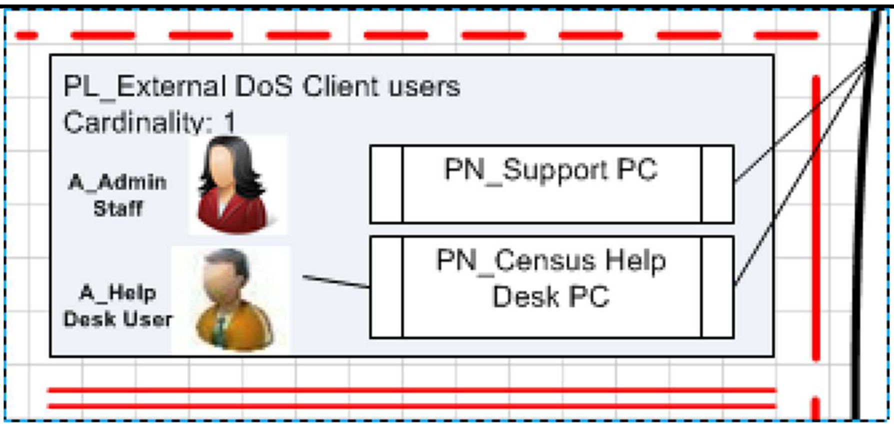
The Bluemix environment allows for automatically managing application capacity through an Auto-Scaling service**.** This service is being leveraged for the ECS solution and allows one to automatically increase or decrease the compute capacity of an application, with the number of application instances adjusted dynamically based on a defined Auto-Scaling policy.

For the ECS applications, we can define scaling rules using Heap, Memory, Response Time, and Throughput metrics. This allows for automatic increase capacity (e.g. application instances, vCPU, memory, etc…) based on the monitoring of these metrics.
Other components in the solution (e.g. API Connect) have high performance options that can be chosen to increase the capacity of those aspects of the solution.

5.5.2 Location: PL_External DoS Data Centre

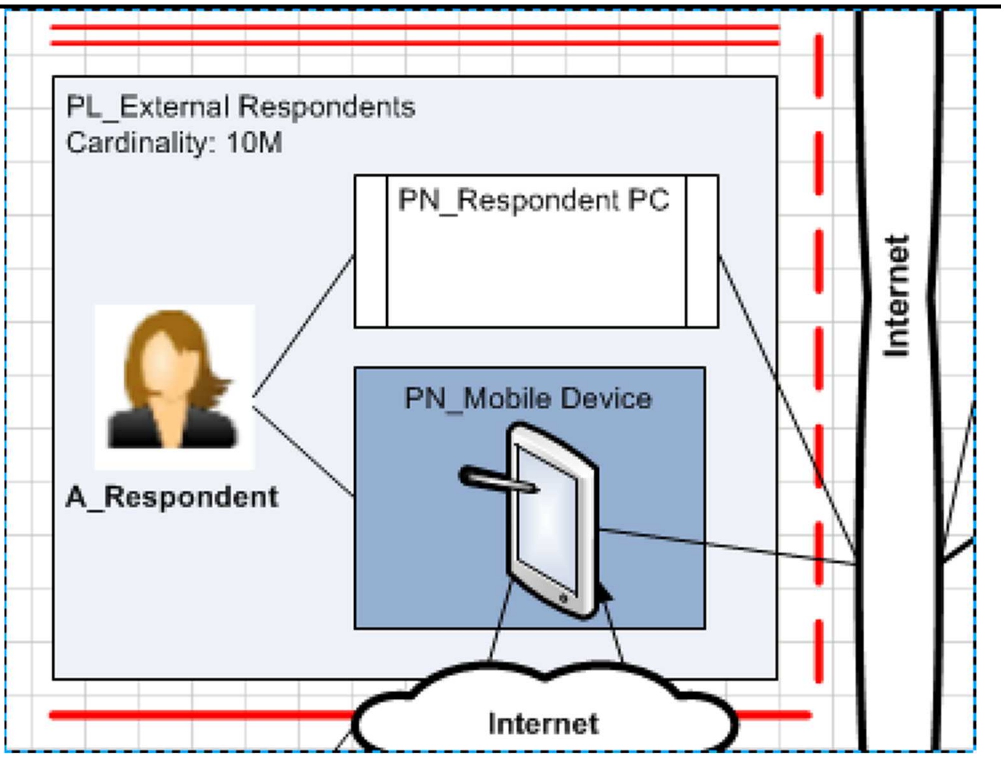
Location	 PL_External DoS Data Centre Cardinality: 1 PN_Electonic Census Processing VPN DoS LDAP			
Node	Description	DUs	System Software	Hardware
PN_Electronic Census Processing	Uncontrolled node that receives the Respondent data from ECS.	None	N/A	N/A

5.5.3 Location: PL_External DoS Client Users

Location	 PL_External DoS Client users Cardinality: 1 A_Admin Staff A_Help Desk User PN_Support PC PN_Census Help Desk PC

Node	Description	Dus	System Software	Hardware
PN_Support PC	Uncontrolled Personal Computer used by DoS and/or Application Support staff.	None	N/A	N/A
PN_Census Help Desk PC	Uncontrolled Personal Computer used by DoS Help Desk staff.	None	N/A	N/A

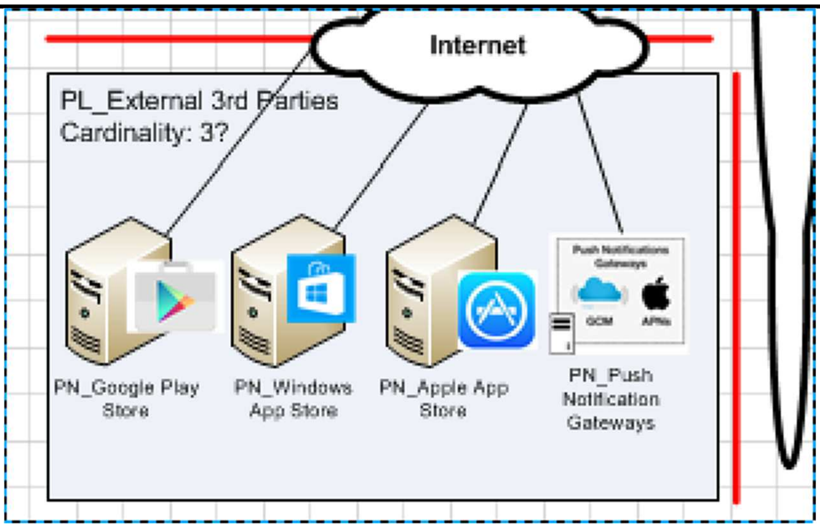
5.5.4 Location: PL_External Respondents

Location	
	

Node	Description	Dus	System Software	Hardware
PN_Respondent PC	Uncontrolled Respondent owned Personal Computer	None	Requires Chrome v35 or later, Firefox 30 or later, Internet Explorer 11 or later (Windows only)	Any PC that can execute the supported browsers.
PN_Mobile Devices	Uncontrolled Respondent owned	U_TabletUI (Android)	Supported OS: • Android	Supported Hardware: • Android

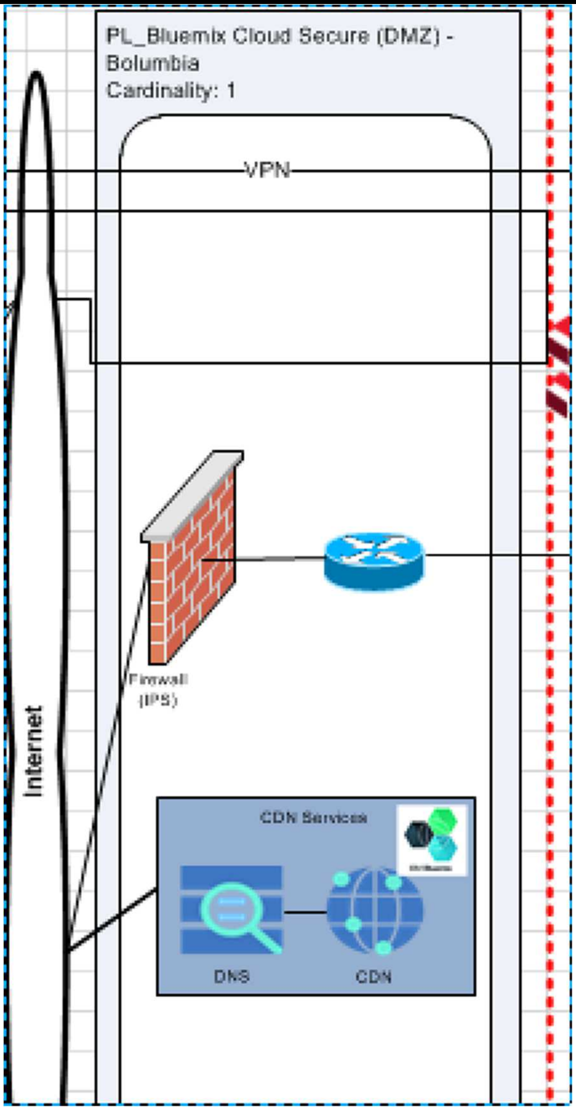
(Android)	Mobile Device	U_PhoneUI (Android) d_RespondentData d_FormData E_MobileApp (Android) E_DeviceLocalDB (Android)	4.3 or later	Phone/Tablet
PN_Mobile Devices (iOS)	Uncontrolled Respondent owned Mobile Device	U_TabletUI (iOS) U_PhoneUI (iOS) d_RespondentData d_FormData E_MobileApp (iOS) E_DeviceLocalDB (iOS)	Supported OS: • iOS7.0 or later	Supported Hardware: • iPhone/iPad
PN_Mobile Devices (Microsoft Surface)	Uncontrolled Respondent owned Mobile Device	U_TabletUI (Windows) U_PhoneUI (Windows) d_RespondentData d_FormData E_MobileApp (Windows) E_DeviceLocalDB (Windows)	Supported OS: • Microsoft Surface RT	Supported Hardware: • Microsoft Surface

5.5.5 Location: PL_External 3rd Parties

Location	 Internet PL_External 3rd Parties Cardinality: 3? PN_Google Play Store PN_Windows App Store PN_Apple App Store Push Notifications Gateways GCM APNs PN_Push Notification Gateways			
Node	Description	Dus	System Software	Hardware
PN_Apple Application Store	Uncontrolled environment that the respondent must access from their mobile device to download the ECS mobile application for iOS devices.	I_MobileApplication	N/A	N/A
PN_Google Application Store	Uncontrolled environment that the respondent must access from their mobile device to download the ECS mobile application for Android devices.	I_MobileApplication	N/A	N/A
PN_Windows App Store	Uncontrolled environment that the respondent must access from their mobile device to download the ECS mobile application for Windows devices.	I_MobileApplication	N/A	N/A
PN_Push Notification	Uncontrolled environment where messaging service	N/A	N/A	N/A

Gateway	notifications are sent to Mobile platform providers, for pushing onto the respondent's device.

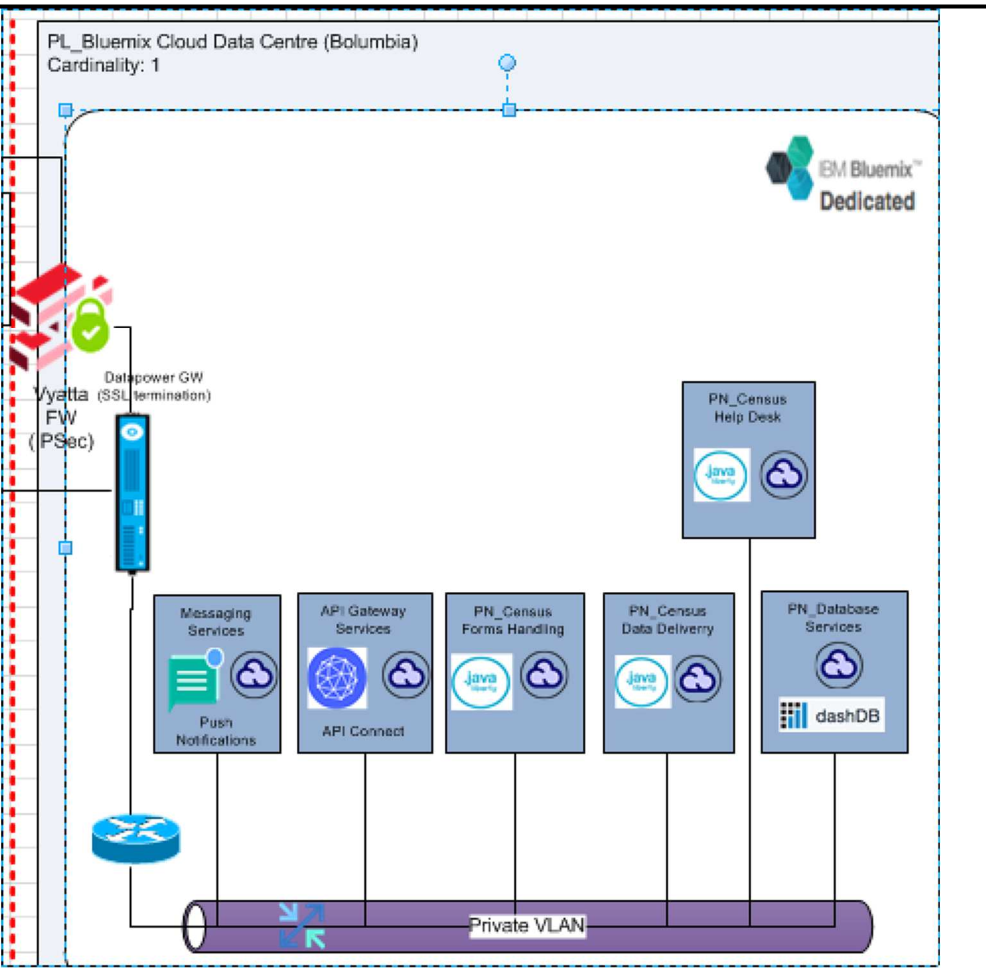
5.5.6 Location: PL_Bluemix Cloud Secure (DMZ)

Location	

Node	Description	Dus	System Software	Hardware
PN_Firewall	Entry Firewall	None	N/A	N/A

PN_CDN Services	SoftLayer Virtual service providing DNS and CDN services for static content	None	Bluemix CDN	N/A

5.5.7 Location: PL_Bluemix Cloud Data Centre (Bolumbia)

Location	 PL_Bluemix Cloud Data Centre (Bolumbia) Cardinality: 1 IBM Bluemix Dedicated Vyatta FW (IPSec) Datapower GW (SSL termination) PN_Census Help Desk Java Messaging Services Push Notifications API Gateway Services API Connect PN_Census Forms Handling Java PN_Census Data Delivery Java PN_Database Services dashDB Private VLAN			
Node	Description	Dus	System Software	Hardware
PN_Firewall (Vyatta)	Provides protection of ECS servers in the DoS private VLAN Bluemix Dedicated environment	None	Vyatta	Vyatta
PN_Gateway (DataPower)	Bluemix Gateway device providing SSL termination	None	DataPower Gateway	DataPower Gateway

	point			
PN_API Gateway	Dedicated API Gateway providing entry point for ECS API service calls from Web and Mobile applications	E_APIGateway	IBM API Connect	Bluemix IaaS
PN_Census Forms Handling	Virtual nodes that provide the core Census Collection Application (Form Handler) and other components such as Mobile Data Migration, Translation, and Authentication.	E_FormHandlingAppServer E_FormHandlingWebServer E_ViewController E_Validation E_FormSubmission E_FormDefinitionController E_Authentication E_Translation E_MgtInfoProcess E_InfoManagementAppServer D_FormData	Bluemix Liberty for Java	Bluemix IaaS
PN_Census Data Delivery	Virtual nodes that contain the messaging middleware and data delivery transfer.	E_Middleware E_SendRespondentData E_SendManagementData E_PullECNs E_ForwardProxy	Bluemix Liberty for Java	Bluemix IaaS
PN_Census Help Desk	Virtual nodes that contain application for census help desk users and admin.	U_ RespondentViewer U_ManagementInfoViewer U_ConfigManagerUI E_HelpDeskAppServer E_RespondentViewer	Bluemix Liberty for Java	Bluemix IaaS

		E_ManagementInfoViewer E_ConfigManager		
PN_Database Services	Virtual nodes that hold the database management server and storage.	E_RDBMS D_RespondentData	IBM dashDB	Bluemix IaaS

6 Summary of key Architectural Decisions

This section documents critical choices that have been made during creation of the solution architecture.

Table 11: Summary of key Architectural Decisions

ID	Problem Statement	Decision	Stage	Comments
AD005	How should the ECS client web application be implemented?	Implement two types for web clients: • HTML-only client • AJAX Web 2.0 style application	Architecture Overview	Most respondents will use AJAX web client thereby minimising server traffic, while HTML client is required to meet the accessibility requirements.
AD010	Interaction model between AJAX web client and the server.	The AJAX web client application posts the collected respondent and management information to the server at each form completion point and when the respondent submits their data, or creates feedback or a technical help request.	Architecture Overview	
AD015	Flexibility of the Census page and form definitions and validation rules	The ECS application is to be a generic mass scale internet based questionnaire solution that is configured to fulfil the specific DoS requirements.	Architecture Overview	Page and form definitions and validation rules externalised and defined within XML definition file(s).
AD020	Way of storing respondents' data in the ECS database	The respondents' data is stored in the ECS database using IFP format.	Architecture Overview	The respondents' data is required to be sent to DOS in the IFP format.
AD025	Lifecycle of the respondents' data on the mobile device	The data is temporarily stored on the mobile device and replicated back to the servers as soon as connection to a network is established. The data is removed from the mobile device once it is replicated to the server.	Architecture Overview	
AD030	Lifecycle of the respondents' data on the ECS database	The respondents' data (and management information) is temporarily stored on the ECS database. It is subsequently sent to the DoS' Electronic Census Processing system using asynchronous 'store & forward' pattern. Once acknowledgement is received from DOS, the corresponding respondents' records are removed from the ECS DB	Component Modelling	

ID	Problem Statement	Decision	Stage	Comments
AD035	Framework to be used for mobile application development	Apache Cordova – using hybrid pattern with pre-packaged HTML 5 resources to allow for operation in the disconnected mode. SQLite database.	Component Realisation Decision	
AD040	What cloud service model to use for deployment of the ECS?	A Platform as a Service (PaaS) model will be used to implement and deploy the ECS solution. ECS is a new application and required for only a short period of time. PaaS providers for fast development and deployment of the application.	Logical Operational Modelling	
AD045	Which Cloud offering to use?	Bluemix Dedicated is a PaaS offering that uses dedicated hardware in a single tenanted cloud environment.	Logical Operational Modelling	
AD050	How will respondents gain access to ECS private Bluemix environment from the Internet?	Provision for direct access to the ECS private Bluemix environment from the Internet.	Physical Operational Modelling	
AD055	How will the availability requirement be met?	The availability NFR is 98%. The standard Bluemix Dedicated SLAs provide for 99.5%.	Physical Operational Modelling	
AD060	How to implement the core ECS application components	Core components will be implemented using the Liberty for Java runtime. It is very light weight and quick, and leverages auto scaling so the capacity can be automatically "dialed up" when load increases.		

While in this section we present architectural decisions in a summary form, one should indeed have each decision elaborated as per templates presented in the lecture materials. The table below should be used as an example of one such elaboration.

Table 12: Example of an architectural decision elaboration

Issue or Problem	How should the ECS client web application be implemented?
Alternatives	1. HTML-only client 2. AJAX Web 2.0 style application
Decision	Both: 1. Implement the application as a Web 2.0 style application with most of the presentation logic running within the browser and copy all question, help and other text to definition/configuration files that can be rendered into HTML pages by the client locally. 2. HTML only interface to meet the accessibility requirements

Justification	• Much more responsive application with all actions occurring locally within the browser. • Substantial reduction in server end infrastructure • Ability to change question, help and message text without application changes. • Use only the core AJAX mechanism, avoiding the GUI presentation widget sets as these are more likely to have browser dependencies. • Substantial increase in application scalability.
Implications	Additional UI development effort.

7 Hot Spots

This section contains several gaps in the solution which can be discovered through techniques such as viability assessment, walkthroughs or similar "static testing" techniques.

These should be used early in the SDLC as it is well known that an early identification of significant gaps provides opportunity for those to be fixed without expensive rework.

This section is by no means intended to be a comprehensive catalogue of all possible gaps, but rather contains some of the notable ones illustrating the aim of exercise 6.

The gaps covered here have not been selected on the basis of any specific architectural principle or criteria. They are merely examples intended to maximize educational effect by illustrating different types of possible enhancements.

7.1 Availability Hot Spots

Issue or Problem	How do we resolve the DB tier single point of failure?
Alternatives	1. Implement high availability using Log Shipping. 2. Provision dashDB for Transactions High Availability Disaster Recovery (HADR) feature.
Decision	Option 2 - Provision dashDB for Transactions High Availability Disaster Recovery (HADR) feature.
Justification	This is a proven approach for dashDB HA in the Bluemix dedicated situation.
Implications	Requires HA testing as a part of non-functional testing cycle

Issue or Problem	There is no application level monitoring in the solution, therefore an unavailable or degraded service issue can pass undetected by the operations staff for a significant period of time.
Alternatives	1. Instrument the application via an instrumentation API like ARM 2. Use products (e.g. Tivoli) to instrument the application 3. Implement synthetic clients that exercise and monitor the system 4. Use an external monitoring service
Decision	We recommend a combination of option 1 to measure actual performance in key components of the solution and 3 to monitor availability.
Justification	• The combination of the two approaches provides operations information about system availability and how key components of the system are performing. • Minimises the additional load placed on the production environment.
Implications	None.

7.2 Sizing for performance and availability in a cloud-based solution

Non-functional requirements relating to performance, throughput, and availability drive most of the decisions about the sizing of the solution. This is the case for both traditional solutions that are hosted on premises as well as for different Cloud-based solutions.

The further one moves towards cloud managed services (e.g. IaaS->PaaS->SaaS->BPaaS) the less detailed information that is available about the infrastructure used in the solution. In the chosen model for ECS, PaaS, the majority of the end-to-end solution is delivered through subscription services (e.g. firewalls, load balancers, gateways, databases, etc..). These typically have varying options to choose a sizing (e.g. t-shirt size) and a high level of granularity. Knowing the number of expected transactions / service calls / etc. is still just as important for the architect to determine/calculate the sizing required in a cloud-based solution. It's just that rather than using this information to determine a hardware sizing specification, instead the architect uses the information to decide which "t-shirt" size is the most appropriate for the given situation

For the core application components of the ECS solution, in this case Liberty for Java runtime platform instances, there are options to select minimum (and maximum) application instances, CPU utilisation, memory utilisation, transaction throughputs, and performance response. Further, with capabilities such as Auto Scaling, the solution can be configured to adjust capacity to suit the circumstances. So, if the load increases substantially as the census approaches (peaking on census night) the solution can be configured to scale based on that load.

Note: ***Unlike a public cloud service where there is virtually unlimited capacity to leverage auto-scaling (or computing 'elasticity') capabilities that can automatically adapt to changing workloads, in a dedicated cloud service like Bluemix Dedicated there is a finite set of bare metal hardware underlying the dedicated environment. The architect must still determine the maximum expected workload for the solution and, working with the cloud service provider, estimate the underlying hardware required to support the expected workload.***

Because there aren't many details (not before purchasing the cloud service) about the specific hardware being used its difficult to get a benchmark to use as the basis for supporting performance / throughput targets. However, the advantage of a cloud-based solution is that one can deploy the application and assess the performance and using the outputs as input to the configuration for a predicted production load. All of this before having to commit to a production infrastructure. This would cost more and take longer in a traditional on premises solution.

The other key issue is that the architect needs to work more closely with the cloud provider in order to get the most efficient and effective solution as more of the management and hosting resides with the cloud provider.

7.3 Security Auditing, Logging and Reporting Hot Spot

The legislative compliance solution principles don't appear to be considered in the solution. The solution must be enhanced to account for following requirements to log audit data and perform audit reconciliations:

a) Equipment that processes respondent data should only be accessed in response to a problem or to implement an authorized change request.

b) All logons to devices within the ECS environment that process respondent data are logged.

c) A weekly audit process traces 'logins to devices within the ECS environment that process respondent data' to their authorizing Change or Problem record.

d) All actions taken by privileged accounts on the servers that process respondent data are logged where this is feasible with the supported technologies.

Issue or Problem	How will respondents gain access to ECS private Bluemix environment from the Internet?
Alternatives	1. Configure access to the ECS via the DoS Enterprise network, using a secured private bi-directional network link between DoS network and the DoS Bluemix Dedicated environment. 2. Provision for direct access to the ECS private Bluemix environment from the Internet.
Decision	**Option 2:** Provision for direct access to the ECS private Bluemix environment from the Internet.
Justification	• Ensures the ECS is hosted off premises in the Cloud as stipulated in the requirements. • Avoids an increase in server end infrastructure at the DoS hosting centre to support high levels of traffic to pass through the DoS. • Avoids the potential impact on performance, availability, and security that would result if ECS was access via the DoS infrastructure. • Ensures that the connection between the DoS enterprise network and the ECS private Bluemix environment is used only for what was intended (i.e. for transfer of completed respondent data to DoS for further processing and to allow DoS staff (and service providers) to access the ECS application.
Implications	An additional network security component is required at additional cost.

7.4 Disaster Recovery (DR)

The existing ECS solution does not currently address disaster recovery requirements. Most cloud providers offer disaster recovery options and this usually requires deploying the application at another of the provider's location. Disaster Recovery in a Bluemix (Public) environment is supported through provision of a continuously available platform. Applications/services must be deployed to multiple geographic regions to enable continuous availability so that the application/services can continue to run in the event of a catastrophic failure, such as the loss of a data centre.

To enable this for a Bluemix Dedicated environment, the ECS solution (applications/services) must be deployed to a second Bluemix geographic location within Bolumbia. A second dedicated environment also needs to be created to allow this.

In addition, the solution will require configuration of a dedicated Global Load Balancer that can point to the ECS solution in both locations**.** Global Load Balancing services are provided as part of the Bluemix Dedicated offering.

8 Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at *Copyright and trademark information*.

Other product and service names might be trademarks of IBM or other companies.