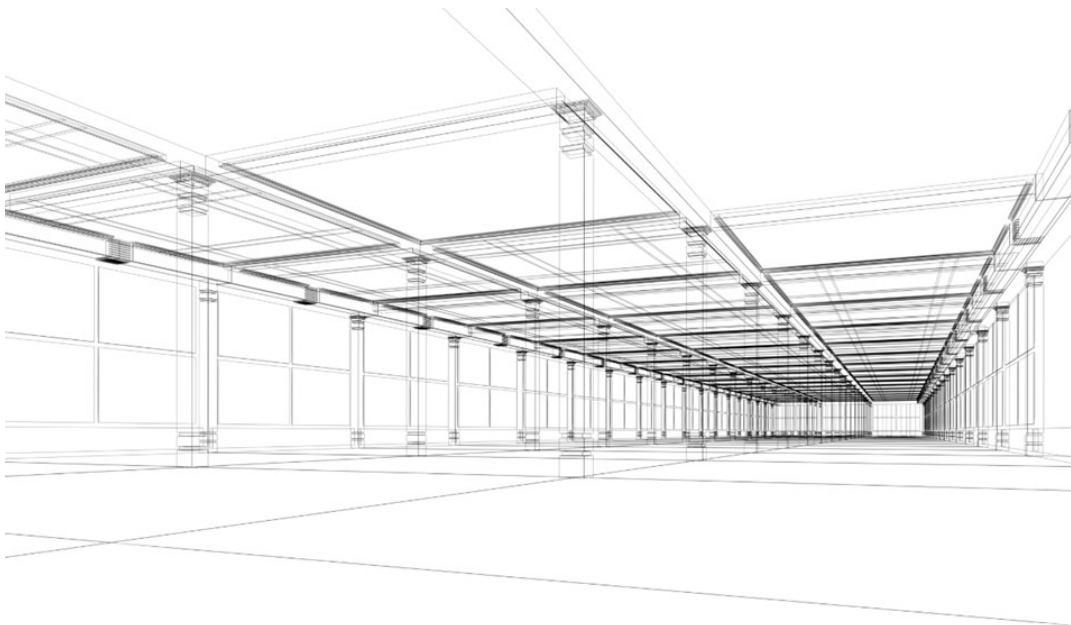


Architectural Thinking

Course Exercise Model Answer - Functional Aspect



Version: 7.12

Table of Contents

1	Introduction	5
1.1	Purpose of this document.....	5
1.2	Definitions	5
1.3	References and Related Documents.....	6
2	Overview of requirements	7
2.1	Summary of functional requirements.....	7
2.2	Summary of non-functional requirements	10
2.3	Out of scope statements.....	11
3	Architecture Overview	12
4	Functional Viewpoint: Component Model.....	15
4.1	Static View.....	15
4.2	Data Model	18
4.3	Dynamic Views	19
4.4	Realisation Decisions.....	21
5	Summary of key Architectural Decisions	22
6	Trademarks	25

Table of Figures

Figure 1 - System Context	8
Figure 2 - Use Case Model Diagram	9
Figure 3 - Architecture Overview Diagram	12
Figure 4 – ECS Static Functional View	15
Figure 5 – High Level ECS Data Model	19
Figure 6 – UC_01 Logon	19
Figure 7 – UC_12 Submit Form (from mobile device)	20
Figure 8 – UC_19 Transfer Management Information.....	20

Table of Tables

Table 1. List of Terms and Acronyms.	5
Table 2. References and related documents.	6
Table 3 - Use Case Overview.....	9
Table 4 – NFR Summary.....	10
Table 5: Functional responsibilities of components	15
Table 6: Component realisation decisions.....	21
Table 7: Summary of key Architectural Decisions	22
Table 8: Example of an architectural decision elaboration	23

1 Introduction

1.1 Purpose of this document

This document contains the model answers for the exercises offered as part of the Architectural Thinking 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
LOM	Logical Operational Model

Acronyms	Meaning
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.

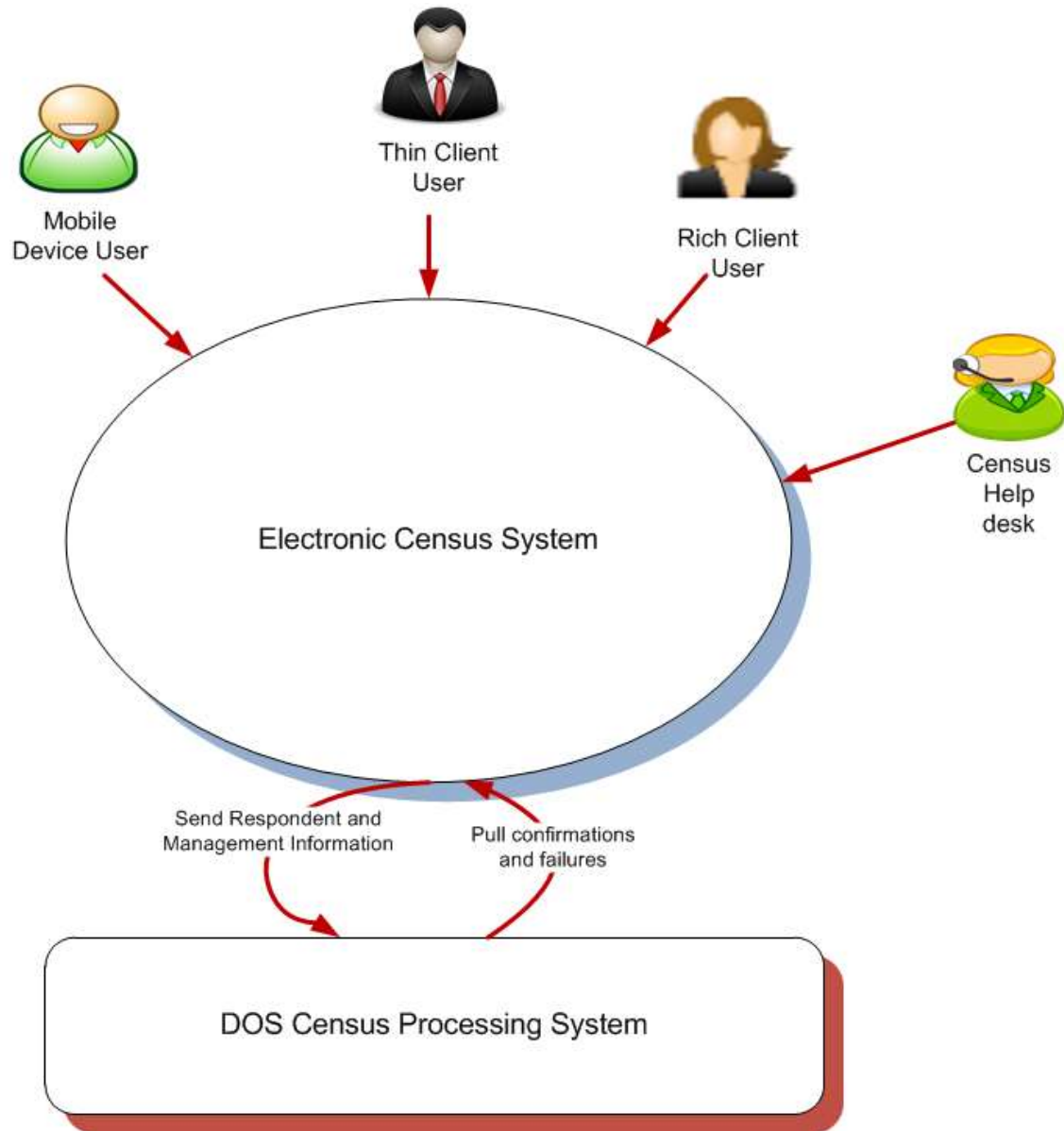


Figure 1 - System Context

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

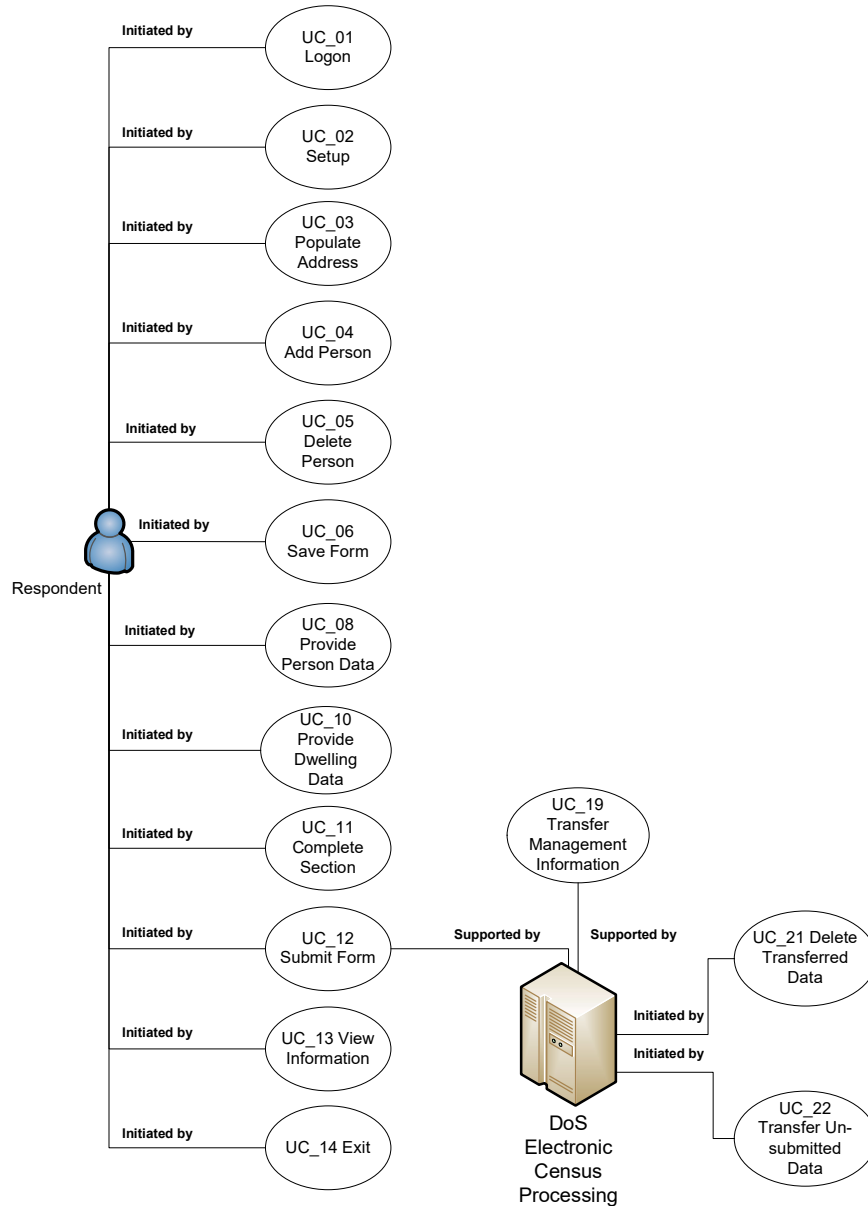


Figure 2 - Use Case Model Diagram

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
1	Volumetric	Capacity to support up to 9.5 Million households during the census enumeration period.

ID	Category	Requirement statement
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.

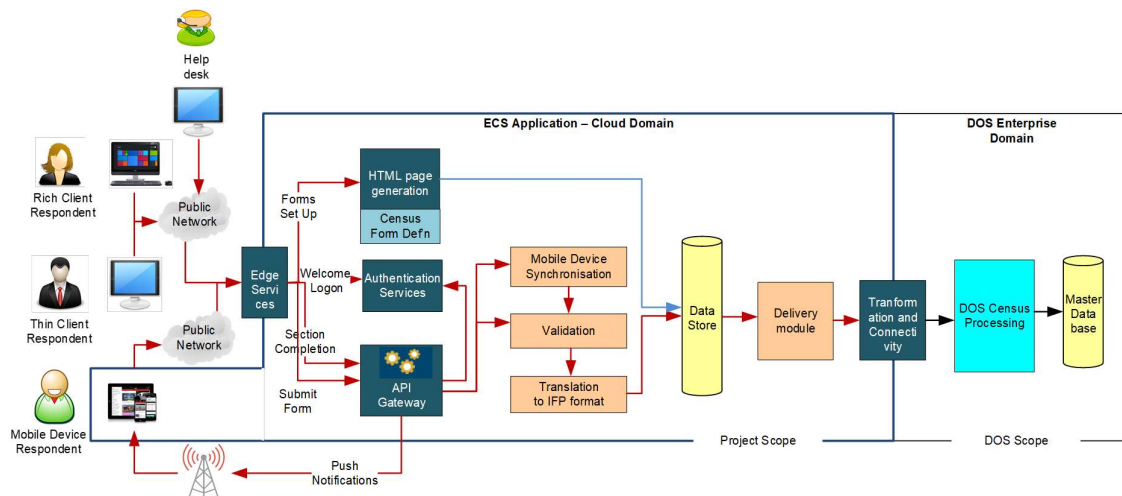


Figure 3 - Architecture Overview Diagram

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.

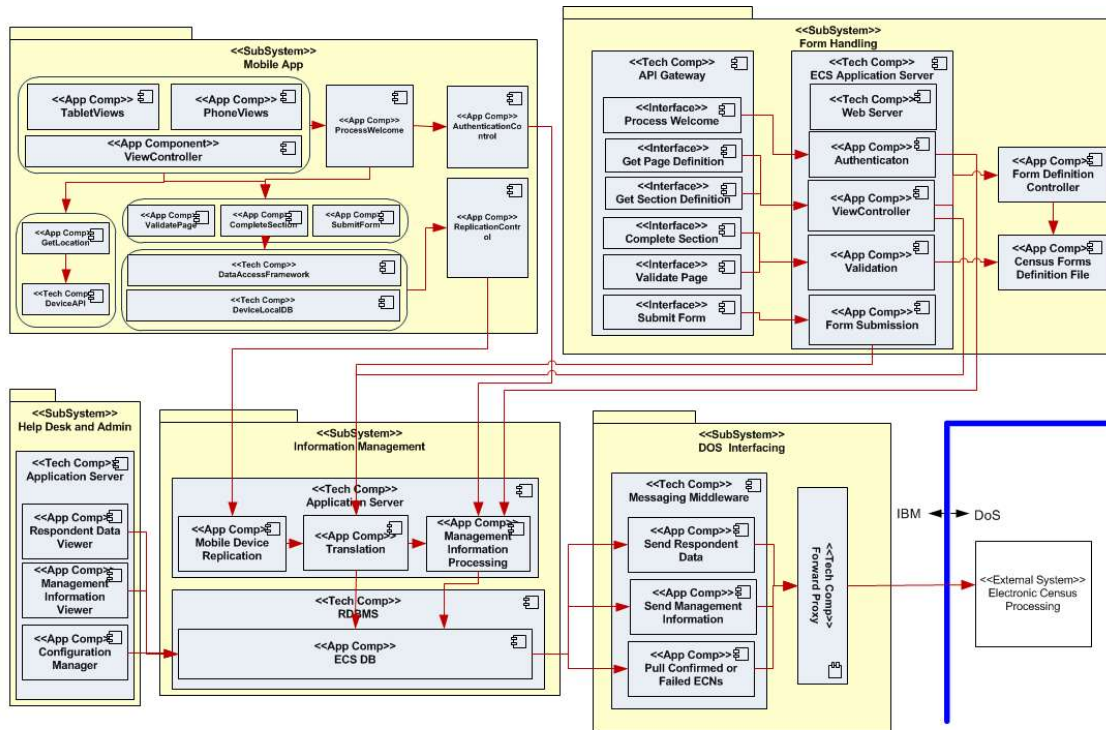


Figure 4 – ECS Static Functional View

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.
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.

Name	Type	Functionality supported
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.
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)

Name	Type	Functionality supported
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.

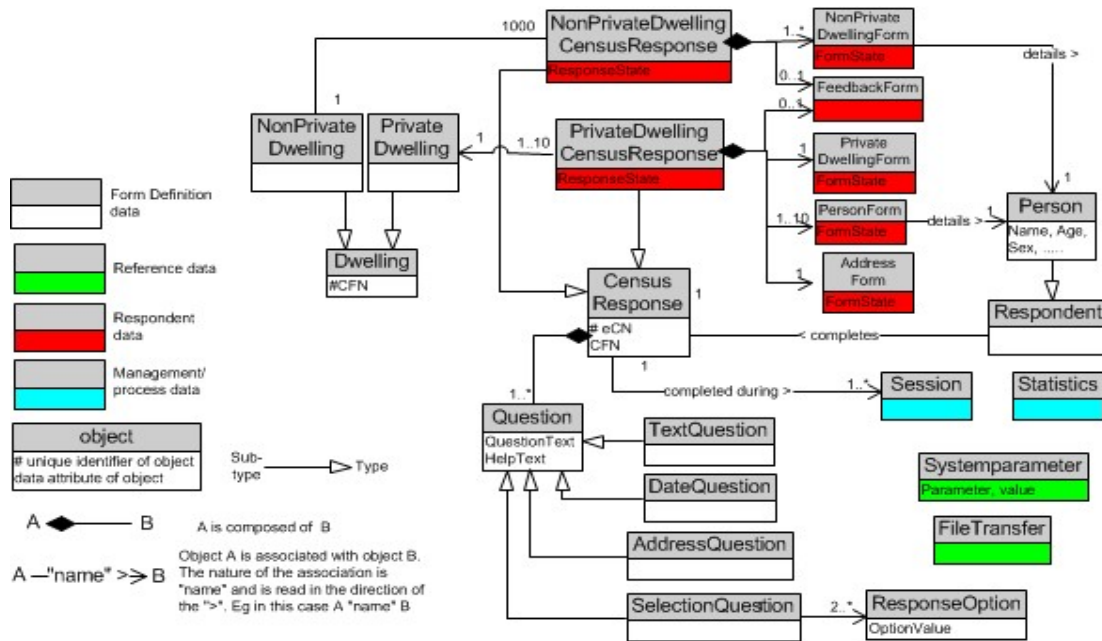


Figure 5 – High Level ECS Data Model

4.3 Dynamic Views

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

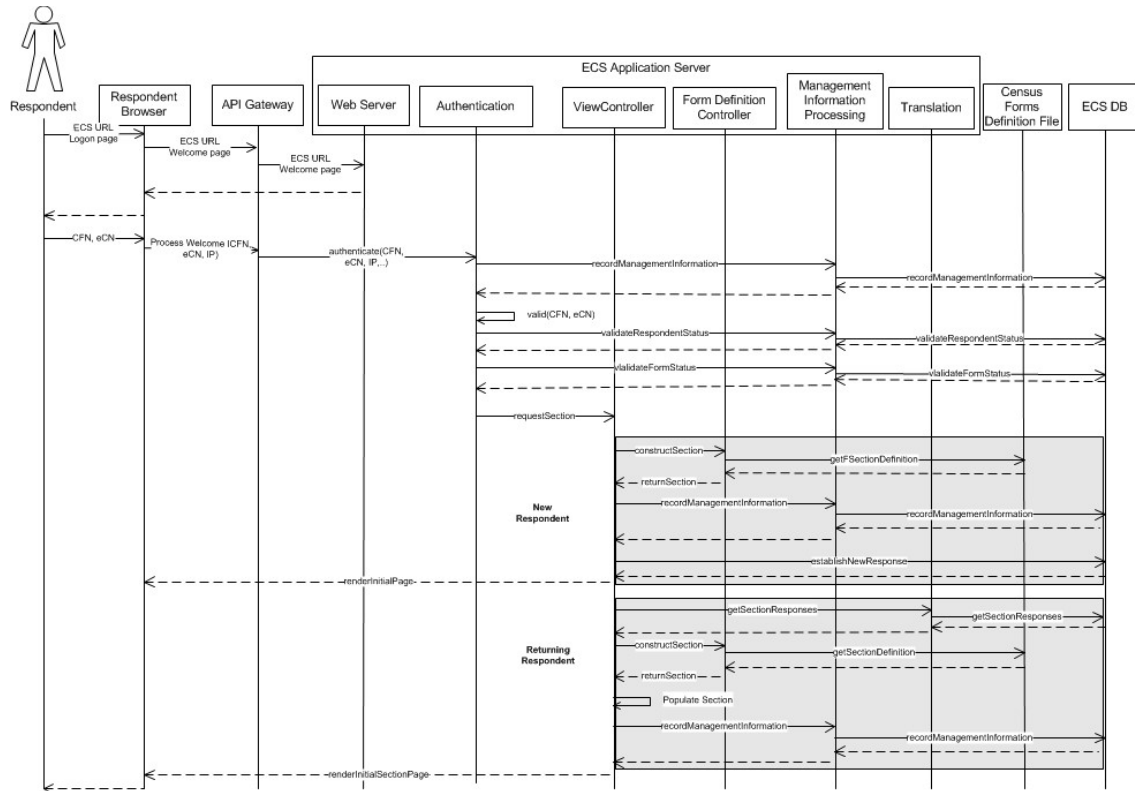


Figure 6 – UC_01 Login

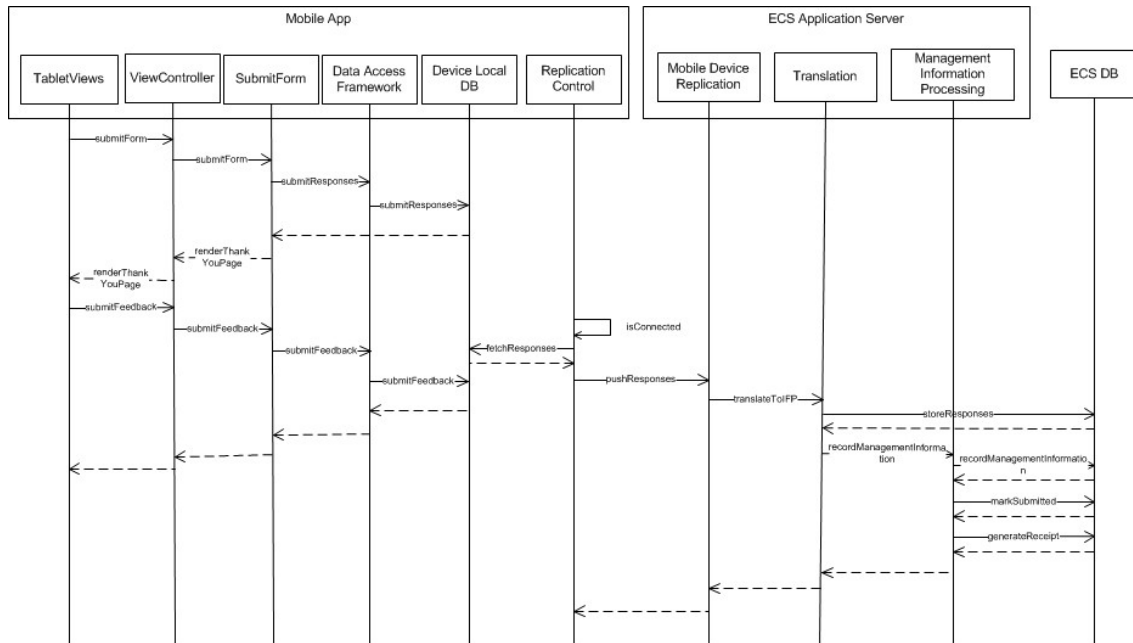


Figure 7 – UC_12 Submit Form (from mobile device)

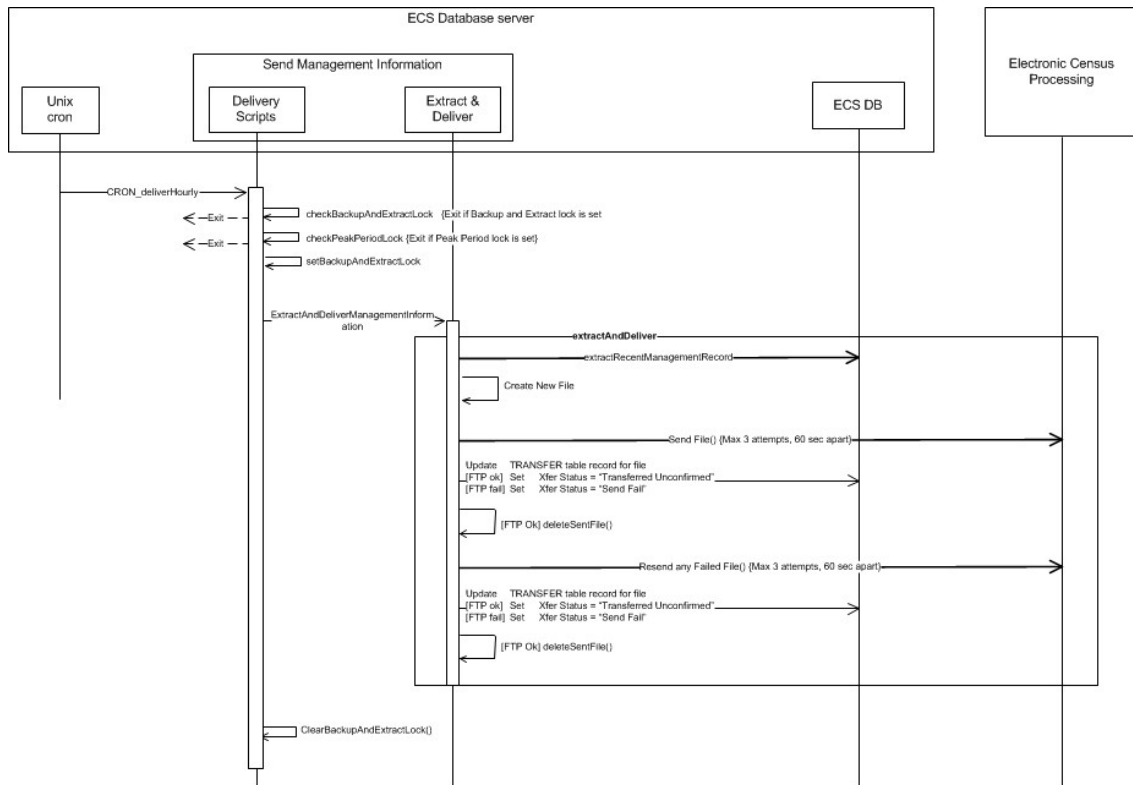


Figure 8 – UC_19 Transfer Management Information

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 Summary of key Architectural Decisions

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

Table 7: 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: <ul style="list-style-type: none"> 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 8: Example of an architectural decision elaboration

Issue or Problem	How should the ECS client web application be implemented?
Alternatives	<ol style="list-style-type: none"> 1. HTML-only client 2. AJAX Web 2.0 style application
Decision	<p>Both:</p> <ol style="list-style-type: none"> 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	<ul style="list-style-type: none">• 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.

6 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.