مـوانـئ أبـوظـبـي
ABU DHABI PORTS

بوابة المقطع
MAQTA GATEWAY

JUL | Janela Única Logística

بوابـــة المقطـــع
MAQTA GATEWAY

JUL – Angola’s National Logistics Single Window (Phase 1)
High Level Design

DOCUMENT HISTORY & QUALITY CONTROL

Version	Date of Issue	Author	Summary Of Changes

Next Review Date:

Although this document is classified as FINAL it is still subject to the Continual Service Improvement process, therefore it is not necessary to wait until the annual review date to make improvements.

Note: As this document is a controlled document any updates or improvements must be implemented under strict change control via the process owner. All reviewers must review and approve all updates.

Document owner:

Role	Name
Document Owner	
Deputy Document Owner	

Signatures

This document has been reviewed by:

Role	Name	Signature	Date

This document has been approved by:

Role	Name	Signature	Date

Table of Contents

1 Introduction of HLD

1.1 Purpose and Audience

The purpose of this high-level design document is to provide a detailed blueprint and holistic understanding of the general structure of the application, the components that it contains, the capabilities offered by the design based on a phased delivery schedule, the integration points with other systems and the high-level definition of the functionality of each component. This document serves as a guiding reference for stakeholders, architects, developers, and other relevant parties involved.

1.2 Definitions, Acronyms and Abbreviations

[Instructions: Detail any terms, definitions, acronyms, and abbreviations introduced in this document or supported terms. List them in alphabetical order. This section may also simply link to the project glossary, if available.]

Abbreviation/ Term	Description
JUL	
ARRCLA	
CNCA	
DU	
DUP	
LPCO	

1.3 References

TBC

1.4 Business Vision

1.4.1 Business Domain

- Unified Trade Ecosystem: Deliver a single platform for import, export, and transit procedures.
- Regulatory Coordination: Strengthen collaboration among agencies to ensure compliance and risk-based decision-making.
- Business Imperative: Reduce transaction costs, accelerate clearance times, and enhance Angola's competitiveness in global trade.

1.4.2 Data Domain

- Data Harmonization: Standardize trade information using the WCO Data Model.
- Interoperability: Enable seamless data exchange across agencies and international partners.
- Business Imperative: Improve transparency, predictability, and trust in trade flows through consistent, reliable information.

1.4.3 Application Domain

- Single Window Portal: Provide a centralized communication and information system aligned with UNCTAD's framework.
- Workflow Automation: Digitize end-to-end trade processes, minimizing manual intervention.
- Business Imperative: Enhance efficiency, reduce delays, and support Angola's obligations under the WTO Trade Facilitation Agreement.

1.4.4 Technology Domain

- Scalable Infrastructure: Deploy modular, cloud-native architecture to support growth and resilience.
- Security & Trust: Implement strong cybersecurity and identity management to protect sensitive trade data.
- Business Imperative: Ensure system reliability, resilience, and stakeholder confidence in digital trade operations.

1.4.5 Supply Chain Domain

- Event Logging & Traceability: Monitor land and maritime supply chains with integrated tracking.
- Visibility & Accountability: Provide end-to-end transparency across logistics operations.
- Business Imperative: Strengthen resilience, reduce risks, and build trust among traders, regulators, and international partners.

2 Enterprise Architecture

[Instructions: State the architecture enterprise context and associated enterprise model. State the direction and relationship to the overall strategy.]

2.1 Project Phasing

[Instructions: Discuss the basic phases and their effect on the architecture and follow-on architecture decisions. Include any dependencies on other projects; give specific project name and owner. Specify technology release or technical issues that need to be proven before being used in another project.

State the project phasing related to an iterative design and build process and the associated framework approach supporting iterative development.

[Instructions: Give a complete list of development and testing tools.]

Tool	# License	Tool	# License
UML Modeling		Configuration	
IDE		Syntax Check	
HTML		Data/SQL Check	
Integration Architect		Web Check	
Unit Test		Deployment	
Performance Test			

2.2 Technology and Architecture Assumptions and Dependencies

[Instructions: Identify the basic assumptions or required resources and decisions necessary to allow the application to be successfully developed.]

Assumption & Dependencies	Architecture Decision

Assumption & Dependencies	Architecture Decision

3 Architecture Objectives

[Instructions: Describe the technical architecture objectives that must be met by this project. Emphasis should be on using a shared infrastructure and reuse. Cost reduction and control should be emphasized. Each technical objective should be tied to a specific business need related to strategy or technical dependencies.]

4 Architecture Constraints

*[Instructions**:** Describe the factors, standards-based or otherwise, that influence and constrain the architectural design. Give a specific statement on standards, rational and URL to website.]*

4.1 Standards

4.1.1 Technology Standards

4.1.2 Messaging Standards

4.1.3 XML standards

4.2 Portability

Instructions: *Discuss portability across application servers, operating systems, client devices, etc. This should address multi-channel, multi-content and environment independent portability*.

5 Architecture Model

[Instructions: Use the following template to diagram (in UML form) the major actors, the application control classes, the key business domain classes, the data stores for each major entity, and third party software packages. All of these high level components are shown from a logical view.

The diagram is not intended to show source code filenames or specific executable elements. This diagram is meant to represent a lower level of detail, interpreting components more as sub-systems. Specifically state the framework (UI, service, data access, infrastructure, and security) and design patterns implemented. Add hyperlink for ER Diagram and Process Flow Diagram.]

5.1 Component Descriptions

[Instructions: For each system component shown in the Architectural Model diagram, describe the component and its interactions, constraints, design requirements, etc.]

5.2 Presentation Layer

[Instructions: Discuss presentation requirements including issues such as different client viewers (browser, mobile phone, PDA, etc.), integration within existing portals, language support, and so on. Discuss the impact that these requirements have on the architecture. This section should specify the specific usage of presentation technologies such as XHTML, XML, XSLT or other presentation languages and technologies.

If the project requires complicated personalization, internationalization, portal requirements, specify the UI architecture that will achieve the objectives. The User Interface Design Specification contains the detailed design, functional flow, and user assistance design of the user interface for the application.]

5.3 Business Logic Layer

[Instructions: Insert the analysis-level Process Domain Model. The process domain is defined (in UML) as a model of the problem domain. Analysis focuses on what to do, design focuses on how to do it.

This model should include only business domains. (The Technical Design document completed later will provide the detailed drill down such as such as factories, view classes, local and shared services, etc.)

This model also presents a high-level use case, data and sequence flow diagrams that demonstrate how a particular business process is implemented by the interaction of various architectural components. Note that this is not the detailed sequence diagrams that involve low-level classes; this should be included in the technical document.]

5.4 Services and Integration Layer

[Instructions: List or diagram (in UML format) each remote or local service that the application uses. Include a short description of any new service. Indicate any enhancements needed to existing services. Indicate what type of reuse the service implements:

- **Shared Remote:** the application uses an existing or new, shared service which is designed to run remotely and service several applications. Be sure to indicate if enhancements are needed to the service API to accommodate application requirements.
- **Shared Local:** the application will use an existing or new shared local service. Shared, local services are shared code that is instantiated separately for each application (running as a local service within the CASPER framework).
- **Local:** the application builds its own service to run locally within the CASPER framework.]

Remote/Local Service	Reuse	Description/Enhancements

5.4.1 Integration Services

[Instructions: Show the high level data flows between systems, including the major data entity names (e.g. Purchase Order, Invoice) and system names (e.g. COPICS, COSDOM). This diagram should not include details such as physical machine names, but it must include geographical location.]

5.4.2 Data Flow

[Instructions: Describe in the table below the details of the Application Integration diagram. The columns are:

- **Data Flow** = a name of one object from the Application Integration diagram that generally identifies data being copied
- **Style** = Publish, subscribe, request reply, notification.
- **Source System** = the application or database the data is being extracted from.
- **Target System** = the application or database the data is being copied to.
- **Data Spec Reference** = URL or other reference that defines the data being transferred.
- **Transaction & Timing** = how often is the flow of data required (e.g. daily, every 3 minutes, on user demand). Specify **transactional requirements** & SSL requirements.
- **Replication Justification** = the reason why the data must be copied (e.g. "source system not available 24 x 7" or "data being merged with other data for analysis").]

Use Case	Data Flow	Type	Source	Target	Data Spec Reference	Transaction & Timing	Replication Justification

5.5 Data Access Layer

[Instructions: Discuss the data layer in terms of the different data stores. Add hyperlink for Data Dictionary.]

5.5.1 Data Access Control

Instructions: Discuss the data access control.

5.5.2 Data Storage

Instructions: Discuss data storage: English only, Unicode based data, blob data

5.5.3 Connection Pooling

Instructions: Discuss the specific connection pooling mechanism

5.5.4 Concurrent Access and Object Locking

Instructions: Discuss concurrent access strategies (optimistic or pessimistic locking, etc.)

5.5.5 Transactional Requirements

Instructions: Discussion specific boundaries at each data and database access layer.

5.5.6 Persistence

Instructions: For all date and objects that require a persistence mechanism, identify the object size, volume, duration, retrieval mechanism and identification, update frequency and reliability in the infrastructure.

5.6 Performance

[Instructions: Address each specific performance metric from the functional requirements. Discuss constraints, exceptions, and handling requirements.]

Performance Requirements	Use Case	Mechanism	Constraints

5.7 Security

[Instructions: Discuss the security design and integration requirements with Single Sign-on. Discuss any application-specific security enforced outside of Single Sign-on (e.g. user level authentication and authorization to access a database) and secure transmission (SSL) requirements.]

5.7.1 Authentication

Instructions: Discuss the type of authentication that must occur (i.e., password, certificate, hard tokens, etc.) for the application as a whole, for specific actors, or specific functional areas.

Keep in mind that only individual users may be authenticated at the Single Sign-on level. Generic accounts are not allowed. Users also cannot share login accounts because of the implications in sharing personal information across GE, including benefits, payroll, salary, medical, and executive compensation information.

5.7.2 Authorization

Instructions: Complete the following to describe the levels of authorization to the application for each actor and how this relates to Single Sign-on Groups and internal application data.

Group Name	No. of users	Authorization Level/Role

Describe the design of groups and roles for Single Sign-on to accomplish authorization and visibility requirements.

5.7.3 Access Request Process

Instructions: Discuss the process required to approve/disapprove users requests for access.

5.7.4 Encryption

Instructions: Discuss encryption or any additional security measures that should be taken within specific views (web page, card, etc.) or at the data level (in databases).

5.8 Scalability

Instructions: Discuss the techniques or issues for handling scaling of the application in terms of the number of sessions, database space, etc.

5.9 Extensibility

Instructions: Discuss, in relation to the MGSP, how extensibility has been allowed for in the architectural design.

6 Data Conversion and Migration

[Instructions: discuss the strategy, approach, high-level mapping for conversion and migration of business objects. This also identifies data sources, target tables and columns, validation, processing, translation, filter, and foreign key rules.]

Module	Business Object	Owner	Source Table	Constraints	Target Table	Interface	Rule Sequence

7 Reporting and Information

Instructions: Describe the strategy for creating reports for each use case that involves reports.

Use Case	Subject Domain	Business Object	Type Report	Description	Comment

8 Deployment Architecture

[Instructions: State the hardware and software configuration and installation for each layer of the infrastructure. Cover the physical environment reliability and availability, distribution of processing, capacity and performance, cost, and support administration. Specifically state the physical target production infrastructure environment for this application, including server names, physical geographic location, etc. Provide a system diagram, clearly stating where the new application/components will reside. This includes staging, testing and development environments.]

HW/SW	Location	Specification	Installation Costs	Requirements	Admin & License

9 Infrastructure Architecture

9.1 Backup, Fail-over and Recovery

9.1.1 Backup

Instructions: Specify what the backup need is of the application.

9.1.2 Fail-over

Instructions: Specify, if applicable, how fail-over has been designed.

9.1.3 Recovery

Instructions: Specify the means by which the application and services can be recovered in the event of failure.

9.2 Maintenance Windows

Instructions: Specify the outage windows that are permissible based on user needs.