# Technical Guidelines

To


For

# **Smart Services**
# Lusail Smart Services Interface Guidelines

Date of release: 5/15/2014
Proposal reference: NLOB-TG-V4--LUS-006
Version: 4.0

# Document Control

## Proposal information

| | | |
|---|---|---|
| Customer Name | : | LUSAIL CITY |
| Proposal Name | : | SMART SERVICES INTERFACE CONNECTION GUIDE |
| Document Version | : | 4.0 |
| Document Release Date | : | 15 May 2014 |
| Validity of Proposal | : | Till the next version of Lusail Smart Services Interface Guidelines Document |
| Ooredoo Reference | : | NLOB-TG-V4-LUS-006 |

| Version. | Date | Reviewed By | Changes made | Approved By |
|---|---|---|---|---|
| **1.0** | **26/11/2013** | **Gowri Bhuthanathan** | **-** | **Karam Alshorbassi** |
| **2.0** | **16/12/2013** | **Gowri Bhuthanathan** | **Each Section of the Document has been revised and updated** | **Karam Alshorbassi** |
| **3.0** | **03/04/2014** | **Christopher Ebenezer** | **Section 5** | **Karam Al Shorbassi** |
| **4.0** | **15/04/2014** | **Christopher Ebenezer** | **Section 8** | **Karam Al Shorbassi** |

## Commercial in confidence

This document contains Ooredoo Q.S.C (previously Qatar Telecom (Qtel) Q.S.C.) confidential and proprietary information and is supplied to Lusail City purely in order to enable Lusail City to evaluate details concerning Ooredoo' s Products and Services. This document (including any part hereof) is not to be disclosed or transferred outside your organisation without the prior written consent of Ooredoo, and may not be copied or reproduced in any form or by any means except internally within Lusail City organisation in order to enable such evaluation.

This document does not constitute a contract between Lusail City and Ooredoo. The only binding terms between Lusail City and Ooredoo shall be the terms of a binding customer order signed by both parties, and this document shall not be deemed to replace, override, or add to the terms of any such binding customer order.

Many of the products, services and company names referred to in this document are trademarks or registered trademarks, they are all hereby acknowledged.

# Table of Content

# 1. Introduction

The Lusail Command and Control Center (LCCC) built in Lusail Smart City enables integration, management and monitoring of integrated smart service systems based on diverse interface standards, centralized in a single location. Real-time monitoring of the smart service system process through the LCCC provides the facility to various stakeholders in Lusail Smart city to improve their system operation and efficiency.

"**Lusail-Smart Services Interface guidelines**" is a document that details the interfaces for smart services that will be managed and controlled by Lusail Command and Control Center (LCCC).

The document is targeted for a smart service designer and implementer.

All the smart services that are to be managed and controlled by LCCC should follow the interfacing requirements.

# 2. Disclaimer

This document specifies the interface specification for smart services to be implemented in Lusail Smart City.

This is not a design or work instruction document.

The information provided in the document is to help the designer and the implementation engineers to understand the type and nature of interfacing requirements for the smart services.

The interfacing of the External Stakeholder's System with the LCCC is out of scope of this document.

The classification of various interfacing scenarios in the document is based on Ooredoo's understanding on the Lusail Smart City project and is generalized to cover all/most of the smart services proposed.

# 3. Definition and Acronyms

In this document, unless the context otherwise requires:

"**Access Network**" means all the fiber contained in the company infrastructure relating to the **Telecommunications Apparatus** and the **ICTE Apparatus** required to carry **Telecommunication Services** and/or **ICTE Services** from Telecommunications Apparatus and/or ICTE Apparatus and/or **End-User(s);**

**"Core Apparatus"** means software and hardware such as routers, switches, repeaters, software, servers and other necessary hardware that may be provided by **Ooredoo** and that is designated for and facilitates the operation of **Smart Services** relating to the **Development**

"**Field Apparatus**" refers to the software and the hardware that may be provided by Ooredoo or by third party to enable provision to the smart services by Ooredoo or by a third party.

"**Lusail Command and Control Center (LCCC)**" means a control tower which integrates, manages, maintains and monitors **ICTE Services** and relevant systems based on diverse interface standards**.** It will be designed considering the efficiency of service operation and smooth monitoring of various services;

"**Network Interface Unit (NIU)**" means the apparatus employed to aggregate/protocol conversion or both if required and there by transmitting safe/reliable bi-directional data from/to any of the system modules employed in the Lusail smart services system architecture;

"**Stakeholder**" refers to the third party organization who shares their business interests in the Lusail smart city project.

# 4. Interfacing Scenarios

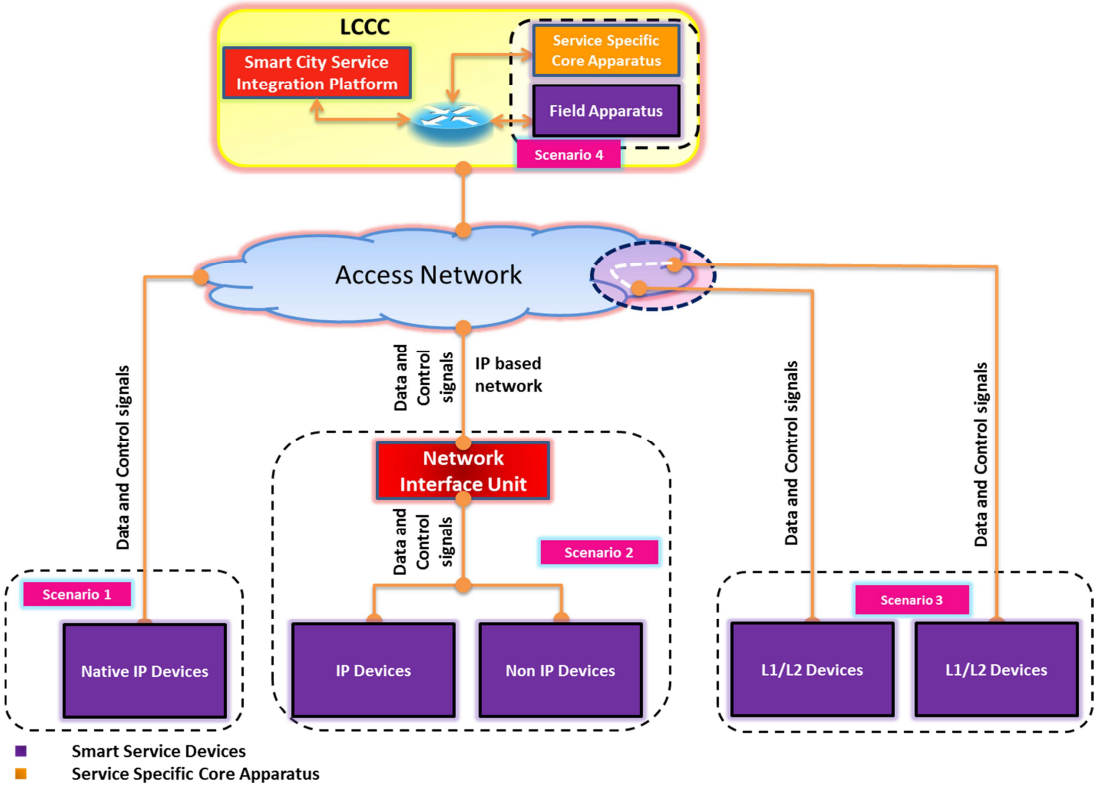


**Figure 1 : Interfacing Scenarios**

## 4.1. OSI Layers

International Organization for Standardization (ISO) - ISO/IEC 7498 began to develop its OSI framework architecture. OSI has two major components: an abstract model of networking, called the Basic Reference Model or seven-layer model, and a set of specific protocols.

As per the ITU-T as the X.200-series of recommendations, the concept of OSI, a real system is a set of one or more computers, associated software, peripherals, terminals, human operators, physical processes, information transfer means, etc., that forms an autonomous whole capable of performing information processing and/or information transfer. The basic structuring technique in the Reference Model of Open Systems Interconnection is layering. The layers are stacked this way:

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

### 4.1.1. LAYER 7 - APPLICATION

As the highest layer in the Reference Model of Open Systems Interconnection, the application Layer provides the sole means for the application process to access the Open System Interconnect Environment (OSIE).

In addition to information transfer, such facilities may include, but are not limited to the following:

a) Identification of the intended communication partners (for example by name, by address, by definite description, by generic description);

b) Determination of the acceptable quality of service (for example response time, tolerable error rate, cost vis-a-vis the previous considerations);

c) Synchronization of cooperating applications;

d) Agreement on responsibility for error recovery;

e) Agreement on security aspects (e.g. authentication, access control, data integrity);

f) Selection of mode of dialogue; and

g) Identification of abstract syntaxes.

### 4.1.2. LAYER 6 - PRESENTATION

The Presentation Layer provides for the representation of information that application-entities either communicate or refer to in their communication.

The Presentation Layer provides for common representation of the data transferred between application-entities. This relieves application-entities of any concern with the problem of “common” representation of information, i.e. it provides them with syntax independence.

The Presentation Layer ensures that the information content of the Application Layer data is preserved during transfer. Cooperating application-entities are responsible for determining the set of abstract syntaxes they employ in their communication. The Presentation Layer is informed of the abstract syntaxes that are to be employed. Knowing the set of abstract syntaxes to be used by the application-entities, the Presentation Layer is responsible for selecting mutually acceptable transfer syntaxes.

### 4.1.3. LAYER 5 - SESSION

The purpose of the Session Layer is to provide the means necessary for cooperating presentation-entities to organize and to synchronize their dialogue and to manage their data exchange. To do this, the Session Layer provides services to establish a session-connection between two presentation-entities, to support orderly data exchange interactions, and to release the connection in an orderly manner.

The only function of the Session Layer for connectionless-mode communication is to provide a mapping of transport-addresses to session-addresses.

The functions of session layer are described to be:

- **Session-connection to transport-connection mapping**

There is a one-to-one mapping between a session-connection and a transport-connection at any given instant. However, the lifetime of a transport-connection and that of a related session-connection can be distinguished so that a transport connection supports several consecutive session-connections.

- **Session-connection flow control**

There is no peer flow control in the Session Layer. To prevent the receiving presentation-entity from being overloaded with data, the receiving session-entity applies back pressure across the transport-connection using the transport flow control.

### 4.1.4. LAYER 4 - TRANSPORT

The transport-service provides transparent transfer of data between session-entities and relieves them from any concern with the detailed way in which reliable and cost effective transfer of data is achieved.

The Transport Layer optimizes the use of the available network-service to provide the performance required by each session-entity at minimum cost. This optimization is achieved within the constraints imposed by the overall demands of all concurrent session-entities and the overall quality and capacity of the network-service available to the Transport Layer

All protocols defined in the Transport Layer have end-to-end significance, where the ends are defined as transport entities having transport associations. Therefore, the Transport Layer is OSI end open system oriented and transport-protocols operate only between OSI end open systems.
The Transport Layer is relieved of any concern with routing and relaying since the network-service provides data transfer from any transport-entity to any other, including the case of tandem sub networks.
The transport functions invoked in the Transport Layer to provide a requested service quality depend on the quality of the network-service. The quality of the network-service depends on the way the network-service is achieved

### 4.1.5. LAYER 3 - NETWORK

The Network Layer provides the functional and procedural means for connectionless-mode or connection mode transmission among transport-entities and, therefore, provides to the transport-entities independence of routing and relay considerations.

The Network Layer provides the means to establish, maintain, and terminate network-connections between open systems containing communicating application-entities and the functional and procedural means to exchange network-service-data-units between transport-entities over network-connections.

It provides to the transport-entities independence from routing and relay consideration associated with the establishment and operation of a given network-connection. This includes the case where several sub networks are used in tandem or in parallel. It makes invisible to transport-entities how underlying resources such as data-link connections are used to provide network-connections. Any relay functions and hop-by-hop service enhancement protocols used to support the network-service between the OSI end systems are operating below the Transport Layer, i.e. within the Network Layer or below.

### 4.1.6. LAYER 2 - DATA LINK

The Data Link Layer provides functional and procedural means for connectionless-mode among network-entities, and for connection-mode for the establishment, maintenance, and release data-link-connections among network-entities and for the transfer of data-link-service-data-units. A data-link-connection is built upon one or several physical-connections.

The Data Link Layer detects and possibly corrects errors which may occur in the Physical Layer.

In addition, the Data Link Layer enables the Network Layer to control the interconnection of data-circuits within the Physical Layer.

### 4.1.7. LAYER 1 - PHYSICAL

The Physical Layer provides the mechanical, electrical, functional and procedural means to activate, maintain, and de-activate physical-connections for bit transmission between data-link-entities. A physical-connection may involve intermediate open systems, each relaying bit transmission within the Physical Layer. Physical Layer entities are interconnected by means of a physical medium.

## 4.2. Scenario 1: Native IP Devices

In this scenario, field apparatus are directly connected to the LCCC through the access Network.

The interfacing specifications for this type of field apparatus are listed below:

<table>
<tr><th rowspan="2">OSI Layer</th><th colspan="5">Interfacing Details</th></tr>
<tr><th>Description</th><th colspan="4">Standard</th></tr>
<tr><td rowspan="5">Layer 1</td><td>Distance</td><td><90m</td><td><5km</td><td><10km</td><td><40km</td></tr>
<tr><td>Media</td><td>CAT6/6a<br>(ANSI/TIA 568-C.2)</td><td colspan="3">SM Fiber Optic Cable<br>(ITU-T G652.D or G657.A2)</td></tr>
<tr><td>Connector</td><td>RJ45</td><td colspan="3">LC/APC</td></tr>
<tr><td>Transceiver</td><td>1000BASE-T<br>(IEEE 802.3ab)</td><td>1000BASE-LX<br>(IEEE 802.3z)</td><td>1000BASE-LH<br>(IEEE 802.3z)</td><td>1000BASE-ZX</td></tr>
<tr><td>Topology</td><td>Star</td><td colspan="3">Star or Ring</td></tr>
<tr><td rowspan="3">Layer 2</td><td>Protocol</td><td colspan="4">Ethernet (IEEE 802.3)</td></tr>
<tr><td>Topology</td><td colspan="4">Star</td></tr>
<tr><td>Redundancy Topology</td><td colspan="4">Double Star (two independent NICs)</td></tr>
<tr><td rowspan="4">Layer 3</td><td>Protocol</td><td colspan="4">• IPv4 (RFC 791)<br>• IPv6 (RFC 2460)</td></tr>
<tr><td>Addressing</td><td colspan="4">Dynamic, Static, APIPA (Automatic Private IP Addressing)</td></tr>
<tr><td>Routing</td><td colspan="4">• Routing Information Protocol (RIP) V2 (RFC 2453)<br>• Open Short Path Finder (OSPF) (RFC 2328)</td></tr>
<tr><td>Redundancy Topology</td><td colspan="4">VRRP(Virtual Router Redundancy Protocol) (RFC 5798)</td></tr>
<tr><td>Layer 4</td><td>Protocol</td><td colspan="4">• Transmission Control Protocol (TCP) (RFC 793)<br>• Telnet (RFC 854)<br>• File Transfer Protocol (FTP) (RFC 959)</td></tr>
</table>

| OSI Layer | Interfacing Details | |
|---|---|---|
| | Description | Standard |
| | | • Secure Shell (SSH) (RFC 4253) |
| Layer 5 | Not Required | |
| Layer 6 | Not Required | |
| Layer 7 | Protocol | • HTTP (Hyper Text Transfer Protocol) (RFC 2616)<br>• RPC (Remote Procedure Call) (RFC 5531) |
| | Encryption | • AES (Advanced Encryption Standard) (ISO/IEC 18033-3),<br>• 3DES (Triple Data Encryption Standard) (ISO/IEC 18033-3:2010)<br>• SSL (Secure Sockets Layer) (RFC 6101) |

**Table 1:Interfacing specifications of scenario where Native IP Devices connected to the LCCC**

## 4.3. Scenario 2: NIU

In this scenario, field apparatus devices are connected to the LCCC through a NIU based on the requirement of the designed architecture.

The interfacing specifications for such type of system are listed below:

| OSI Layer | Interfacing Details | | |
|---|---|---|---|
| | Description | Standard | |
| | | NIU to LCCC | NIU to field Apparatus |
| Layer 1 | Media | SM Fiber Optic Cable (ITU-T G652.D or G657.A2) | • RS485 (EIA/TIA-485-A)<br>• Zig Bee (IEEE 802.15.4)<br>• Wi-Fi (IEEE 802.11)<br>• LAN (IEEE 802.1) |
| | Connector | LC/APC | RJ11,D9 |
| | Transceiver | • 1000BASE-LX (IEEE 802.3z)<br>• 1000BASE-LH (IEEE 802.3z)<br>• 1000BASE-ZX | • 1000 BaseT (IEEE 802.3ab)<br>• M-Bus (Meter Bus) (EN 13757-3)<br>• PoE (Power over Ethernet) (IEEE 802.3at) |
| | Topology | Star | Star, Bus |
| Layer 2 | Protocol | Ethernet (IEEE 802.3) | Standard Specific |
| | Topology | Star | |
| | Redundancy Topology | Double Star (two independent NICs) | |
| Layer 3 | Protocol | • IPv4 (RFC 791)<br>• IPv6 (RFC 2460) | Standard Specific |
| | Addressing | Static, Dynamic, APIPA | |
| | Routing | RIP V2 (RFC 2453) | |
| | Redundancy Topology | VRRP (Virtual Router Redundancy Protocol) (RFC 5798) | |
| Layer 4 | Protocol | • Transmission Control Protocol (TCP) (RFC 793)<br>• Secure Shell (SSH) (RFC 4253) | |
| Layer 5 | Not Required | | |
| Layer 6 | Not Required | | |
| Layer 7 | Protocol | • Web Services (W3C)<br>• HTTP (RFC 2616) | • DLMS (IEC 62056)<br>• M-BUS (EN 13757-2) |

| OSI Layer | Interfacing Details | | |
|---|---|---|---|
| | Description | Standard | |
| | | NIU to LCCC | NIU to field Apparatus |
| | | • RPC (RFC 5531) | • BACNET (ISO 16484-5)<br>• HTTP (RFC 2616)<br>• RPC (RFC 5531) |
| | Encryption | • AES (ISO/IEC 18033-3)<br>• 3DES (ISO/IEC 18033-3 :2010)<br>• SSL (RFC 6101) | |

**Table 2: Interfacing specifications of the scenario where field apparatus are connected to LCCC through a NIU**

## 4.4. Scenario 3: L1 & L2 Devices

This section of the document covers the scenario where in it required by the field apparatus devices to communicate to each other accessing the Lusail fiber access network without being interfaced to the LCCC.

The interfacing specifications for such type of system are listed below:

| OSI Layer | Interfacing Details | | | | |
|---|---|---|---|---|---|
| | Description | Standard | | | |
| Layer 1 | Distance | <90m | <5km | <10km | <40km |
| | Media | CAT6/6a (ANSI/TIA 568-C.2) | SM Fiber Optic Cable (ITU-T G652.D or G657.A2) | | |
| | Connector | RJ45 | LC/APC | | |
| | Transceiver | 1000BASE-T (IEEE 802.3ab) | 1000BASE-LX (IEEE 802.3z) | 1000BASE-LH (IEEE 802.3z) | 1000BASE-ZX |
| | Topology | Star | Star or Ring | | |
| Layer 2 | Protocol | Ethernet (IEEE 802.3) | | | |
| | Topology | Star | | | |
| | Redundancy Topology | Double Star (two independent NICs) | | | |
| Layer 3 | Not Required | | | | |
| Layer 4 | Not Required | | | | |
| Layer 5 | Not Required | | | | |
| Layer 6 | Not Required | | | | |
| Layer 7 | Not Required | | | | |

**Table 3: Interfacing specifications of the scenario where L1 and L2 devices are connected to the LCCC**

## 4.5. Scenario 4: Service Specific Core Apparatus

This section of the document is drafted considering the scenarios where in a service specific core apparatus/Field Apparatus has to be integrated with in the LCCC based on the business requirement of the smart service.

The interfacing specifications for such type of system are listed below:

| OSI Layer | Interfacing Details | | | | |
|---|---|---|---|---|---|
| | Description | Standard | | | |
| Layer 1 | Distance | <90m | <5km | <10km | <40km |
| | Media | CAT6/6a (ANSI/TIA 568-C.2) | SM Fiber Optic Cable (ITU-T G652.D or G657.A2) | | |
| | Connector | RJ45 | LC/APC | | |
| | Transceiver | 1000BASE-T (IEEE 802.3ab) | 1000BASE-LX (IEEE 802.3z) | 1000BASE-LH (IEEE 802.3z) | 1000BASE-ZX |
| | Topology | Star | Star or Ring | | |
| Layer 2 | Protocol | Ethernet (IEEE 802.3) | | | |
| | Topology | Star | | | |
| | Redundancy Topology | Double Star (two independent NICs) | | | |
| Layer 3 | Protocol | • IPv4 (RFC 791)<br>• IPv6 (RFC 2460) | | | |
| | Addressing | Dynamic, Static, AIPPA | | | |
| | Routing | Routing Information Protocol(RIP) V2 (RFC 2453) | | | |
| | Redundancy Topology | VRRP (Virtual Router Redundancy Protocol) (RFC 5798) | | | |
| Layer 4 | Protocol | • Transmission Control Protocol (TCP) (RFC 793)<br>• Telnet (RFC 854)<br>• File Transfer Protocol (FTP) (RFC 959)<br>• Secure Shell (SSH) (RFC 4253)<br>• User Datagram Protocol (UDP) (RFC 768)<br>• SOAP (Simple Object Access Protocol) over HTTP (W3C) | | | |
| Layer 5 | Not Required | | | | |
| Layer 6 | Not Required | | | | |
| Layer 7 | As detailed in Section 6 of this document | | | | |

**Table 4: Interface specification of Service specific core apparatus interfaced to the smart city platform**

# 5. Smart City Platform (Ubi-Cahn) Link Overview

Smart city platform – Ubi-Cahn is proposed to be implemented for the integration of different urban management systems and services into a single unified platform in order to realize efficient city operation.

| Category | Functionality | Description |
|---|---|---|
| **Ubi-Cahn** | Fault Management and GIS | Module to manage lifecycle of Smart City facilities in outbreak, order, action, and termination stages in relation to breakdown information of facilities, and to receive and process measurement messages from onsite sensors<br><br>Display the state list of Device Fault<br><br>Display Information of Incident<br><br>Display device status through GIS based |

| Category | Functionality | Description |
|---|---|---|
| | | Administration UI. |
| **LCCC**<br><br>**- Data Center** | Asset Management | None |
| | SSO and Privilege Management | Operation Manager is detecting server fault, fault recovery process automation and setting up user authority individually regarding efficiently managing IT system resource which is scattered in the managed area. |
| | Reporting, | None |
| | Billing | None |
| | FMS | Facility management solution is not just managing IT device but also managing building and facility that is called Intelligent Building Management Solution (IBS) |
| | EMS | Regarding integrated control of IT running situation, it provides consolidated IT control, operation dashboard by associated system, operator editable and flexible dashboard and web report. |

**Table 5: Functionalities of Ubi-Cahn and LCCC**

## 5.1. Message Link Overview

Ubi-Cahn connection is divided into three types, situation information, alarm order information and connection information. Situation information is again divided into situation event, facility event and complex event. Alarm order information is divided into 'connection event' for alarm and order processing from outside through Event Propagation and Order (EPO) of Ubi-Cahn and 'transmission event' to send alarms and orders from Ubi-Cahn to outside services. Services are connected using connection data information.

## 5.2. Link Message Type

### 5.2.1. Message Definition

| Information Type | | Definition | Remarks |
|---|---|---|---|
| Situation Information | Situation Event | Event requiring response and action about situation outbreak and recognition | Vehicle accident, leakage from water supply facilities, excessive air pollution, emergency bell |
| | Facility Event | Event requiring response and action about sensor equipment breakdown | RTU defect, power defect, shielding opening, communication defect, power leakage |
| | Complex Event | Event generated as a result of correlation analysis between situation events | Cold wave watch & leakage from water supply facilities => water supply facilities freezing |

| Information Type | | Definition | Remarks |
|---|---|---|---|
| Alarm Order Information | Connection Event | Data to deliver alarm order to EPO | SMS, e-mail and fax transmission data |
| | Send Event | Data delivered to U-Service to display VMS and BIT data in AUI | VMS, BIT transmission data |
| Connection Information between services | Inter –Service Connection Data | Data requiring connection without relevance with platform process in between U-Service and an outside organization | Data directly exchanged between connected services |

**Table 6: Link Message Definition**

## 5.2.2. Message Header Format

The message header is a compulsory element in message type data transmission (socket, HTTP, etc.). When responding to a request, the part defined as 'copy' in an output compulsory item is copied and transmitted from a header where the message is received. For other items, the message is filled by the responding system and re-transmitted.

| Seq. | Field Name | Type | Input Compulsory | Output Compulsory | Remarks |
|---|---|---|---|---|---|
| 1 | Header Type | String | Y | Copy | 'U' fixed – A character indicating the start of a header |
| 2 | Data Type | String | Y | Copy | Data type |
| 3 | Event ID | String | Y | Copy | ID of event defined in platform |
| 4 | Trace ID | String | Y | Copy | Key for tracing |
| 5 | Message ID | String | Y | Copy | Value for tracing |
| 6 | Sending System Code | String | Y | Y | Message sending system code |
| 7 | Receiving System Code | String | Y | Y | Message receiving system code |
| 8 | Send Time | String | Y | Copy | Message sending time |
| 9 | Processing Result Code | String | | Y | |
| 10 | Processing Result Message | String | | | For normal processing, the processing result message is processed as blank. When error occurs, the error details are transmitted based on the error code. |
| 11 | Data Part Length | String | Y | Y | |

**Table 7: Link Message Header Format**

## 5.2.3. Message Body Format

### 5.2.3.1. Situation / Complex Event Information

The message body format used in single or complex situation event transmission is defined as of the following. Items excluded from web service are those excluded when web service of the External Service Information Interface (ESI) is called.

| Seq. | Field Name | Type | Compulsory | Remarks | Web Service Exclusion |
|---|---|---|---|---|---|
| 1 | U-Service situation event sequence (PK) | String | Y | ID of PK where situation event is managed (PK created by an entity that generated event) | |
| 2 | Situation event ID | String | Y | ID of internally defined situation event | |
| 3 | Situation event name | String | Y | Internally defined situation event name | |
| 4 | Situation event grade | String | Y | Situation event grade | |
| 5 | Processing state | String | Y | Lifecycle code value for situation event | |
| 6 | Outbreak location count | String | Y | Count of the bundles of situation event outbreak location arrays – '0' if N/A | V |
| 7 | Outbreak locating array | String[] | | Array containing coordinates of situation event outbreak location | |
| | X coordinates of outbreak location | String | | Situation event outbreak location X | |
| | Y coordinates of outbreak location | String | | Situation event outbreak location Y | |
| | Z coordinates of outbreak location | String | | Situation event outbreak location Z | |
| 8 | Name of outbreak place | String | Y | Name of place for situation event outbreak | |
| 9 | Situation event contents | String | Y | Contents of situation event outbreak | |
| 10 | Situation event outbreak date and time | String | Y | Date and time of situation event outbreak | |
| 11 | Situation event item count | String | Y | Count of the bundles of situation event item value arrays | V |
| 12 | Situation event item value array | String | | Array containing situation event item value | |
| | Situation event item name | String | | Situation event item name - Item name of a value that generates situation event | |
| | Situation event item value | String | | Situation event item value - Item value of a value that generates situation event | |
| 13 | Situation event action contents | String | | Contents of action about situation event occurring (only in case of action) | |
| 14 | Situation event action by | String | | Person to take action about situation event occurring (only in case of action) | |
| 15 | Situation event action date and time | String | | Date and time of action about satiation event occurring (only in case of action) | |
| 16 | Situation event closing date and time | String | | Date and time when situation event occurring is closed (only in case of closing) | |
| 17 | Outbreak scope radius | String | | Radius under the impact of complex event | |

Table 8: Message body format

### 5.2.3.2. Facility Event Information

Message body format used in facility event transmission is defined as of the following. Items excluded from web service are those excluded when web service of outside information system (ESI) is called.

| Seq. | Field Name | Type | Compulsory | Remarks | Web Service Exclusion |
|---|---|---|---|---|---|
| 1 | Facility event outbreak ID(PK) | String | Y | ID of PK where facility event is managed | |
| 2 | Facility event ID | String | Y | ID of internally defined facility event | |
| 3 | Facility type name | String | Y | Internally defined facility type name | |
| 4 | Facility ID | String | Y | ID of facility for facility management | |
| 5 | Facility name | String | Y | Name of facility for facility management | |
| 6 | Processing state | String | Y | Lifecycle code for facility event | |
| 7 | Outbreak location count | String | Y | | V |
| 8 | Installation location array | String[] | | Array containing facility installation location coordinates | |
| | X coordinates of installation location | String | | Facility installation location X | |
| | Y coordinates of installation location | String | | Facility installation location Y | |
| 9 | Name of installation place | String | Y | Name of place where facility is installed | |
| 10 | Breakdown cause code | String | Y | Code of breakdown cause for facility event outbreak | |
| 11 | Breakdown cause name | String | Y | Name of breakdown cause for facility event outbreak | |
| 12 | Facility event outbreak date and time | String | Y | Date and time of facility event outbreak | |
| 13 | Facility event action contents | String | | Contents of action about facility event occurring (only in case of action) | |
| 14 | Facility event action by | String | | Person to take action about facility event occurring (only in case of action) | |
| 15 | Facility event action date and time | String | | Date and time of action about facility event occurring (only in case of action) | |
| 16 | Facility event closing date and time | String | | Date and time when facility event occurring is closed (only in case of closing) | |

Table 9: Facility Event Information

### 5.2.3.3. Connection Event Information

The message body format used in alarm and order sending (Email, SMS, and FAX etc) of Ubi-Cahn is defined as of the following.

| Seq. | Field Name | Type | Compulsory | Remarks | Web Service Exclusion |
|---|---|---|---|---|---|
| 1 | Sending mail address/ number | String | Y | Sender's mail address or number | |
| 2 | Receiving mail address/ number count | int | Y | Receiver's mail address or number count | V |
| 3 | Receiving mail address/ number array | String[] | Y | Receiver's mail address or number | |
| 4 | Mail title | String | Y | Title of mail to send | |
| 5 | Message contents | String | Y | Contents of message to send | |

**Table 10: Connection Event Information**

### 5.2.3.4. Send Event Information

Ubi-Cahn will use the message format described below to deliver U-service data such as VMS and BIT.

- VMS (Vehicle Management System) Send Event

| Seq. | Field Name | Type | Compulsory | Remarks |
|---|---|---|---|---|
| 1 | Send Event ID | String | Y | Defined Send Event ID |
| 2 | VMS ID | String[] | Y | The array of VMS ID |
| 3 | VMS Present data | String | Y | Data body part to show on VMS system |

**Table 11: VMS Send Event Information**

- *BIT (Bus Information Terminal) Send Event*

| Seq. | Field Name | Type | Compulsory | Remarks |
|---|---|---|---|---|
| 1 | Send Event ID | String | Y | Defined Send Event ID |
| 2 | BIT ID | String[] | Y | The array of BIT ID |
| 3 | BIT Present data | String | Y | Data body part to show on BIT system |

**Table 12: BIT Send Event Information**

#### 5.2.3.5. Inter-service Connection Data Information

The message body format used in data transmission between services, which is not directly used in Ubi-Cahn, is defined as of the following.

| Seq. | Field Name | Type | Compulsory | Remarks |
|---|---|---|---|---|
| 1 | Destination system code count | int | Y | Count of destination system codes to receive message |
| 2 | Destination system code | String[] | Y | Destination system code to receive message |
| 3 | Transmission data | String | Y | Data body part completed of discussion between services |

**Table 13: Inter-service Connection Data Information**

### 5.2.4. Message Rules

#### 5.2.4.1. Message Rules

- Messages for Ubi-Cahn system link consist of a message header and a body.
- The message header and body are composed according to the format defined in Ubi-Cahn.

#### 5.2.4.2. Message Header Composition

- The message header has fixed length in each field. If the length is shorter than the designated, message header is aligned on the right. The remaining part is padded blank (‘ ‘). However, body length field must be padded as ‘0’ rather than left blank.
- As for characters used in a field, only capital English letters and numbers must be used.
- When the body is encoded, the length is set as the length of encoded body rather than of the body prior to encoding.

#### 5.2.4.3. Message Body Composition

- The length of a body field refers to the max value of each field. Unlike the header, it is not a fixed length character string and padding is not necessary.
- Even if the value of the body field is not a character string, it must be converted into a string and transmitted.
- Data are divided using field delimiter (Hex 0x1E) and array delimiter (Hex 0x1F).
- Delimiter is not inserted at the back of the last field or array.
  Ex.) aaa0x1Ea010x1Fa020x1Fa03bbb0x1Eccc0x1Eddd

- For compulsory items in the message body format, the corresponding values must be set and sent.

## 5.3. Link Interface

### 5.3.1. Socket Interface

- The socket interface is configured as non-blocking mode socket using a TCP socket.
- Data are transmitted while maintaining TCP socket session. Close the session after sending a response through the active session.
- A socket client must be drawn up in a connection system for service request and transmission of messages generated in the connection system.
- To process service request and transmission of messages generated in Ubi-Cahn ESI, a socket server must be drawn up in the connection system.
- A message using a socket interface contains a header and a body.
- As for a response to a message, the receiver header is transmitted to the receiving system by referring to the message header format.

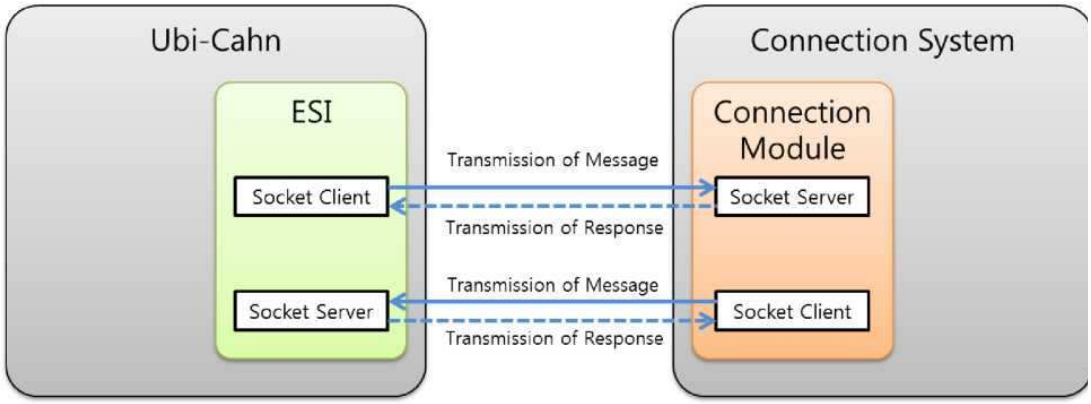


**Figure 2 - Socket Interface Connection**

### 5.3.2. HTTP Interface

- The transmission between Ubi-Cahn ESI and a connection system using HTTP is configured with the post type method. In addition, messages must implement URL encoding.
- A HTTP client must be drawn up for service request to Ubi-Cahn ESI and transmission of messages generated in the connection system.
- To process service request from Ubi-Cahn ESI to a connection system, HTTP (server) must be drawn up in the connection system. In Ubi-Cahn ESI, a HTTP client is created and connected to the connection system.
- Messages using a socket interface contain a header and a body.
- As for a response to a message, the received header is transmitted to the receiving system by referring to the message header format.

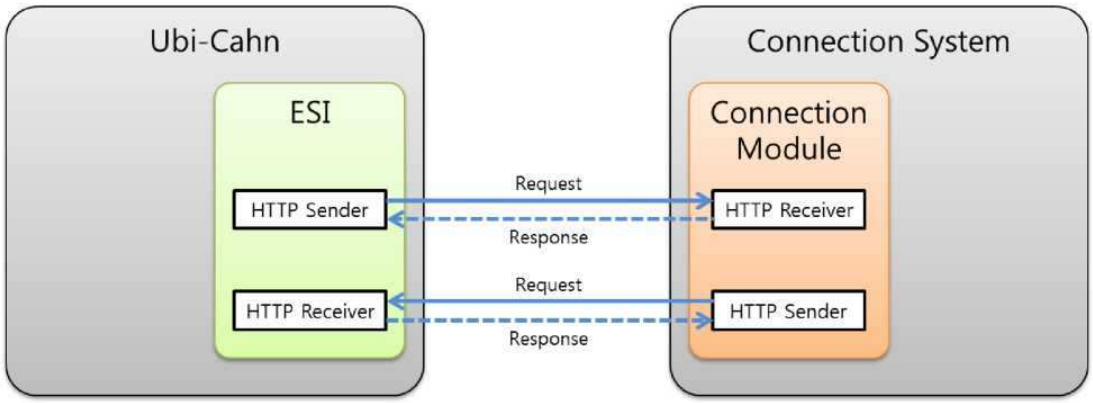


**Figure 3 - HTTP Interface Connection**

### 5.3.3. Web Service Interface

- Connection using web service is carried out not in a message type, but in a parameter type, the basic web service method.
- In the web service connection, a header used in a message type is not used. Instead, items are transmitted as parameters according to data formats in sending system codes, receiving system codes and messages.
- A web service consumer must be drawn up for service requests to Ubi-Cahn ESI and transmission of messages generated in the connection system.
- To process a service request from Ubi-Chan ESI to a connection system, web service provider is drawn up in the connection system. The web service provider must be drawn up so that it can be received with parameters according to the defined - message body format. In addition, XML scheme that is the same as Ubi-Cahn ESI must be used. In Ubi-Cahn ESI, a web service consumer is created and connected to the connection system.

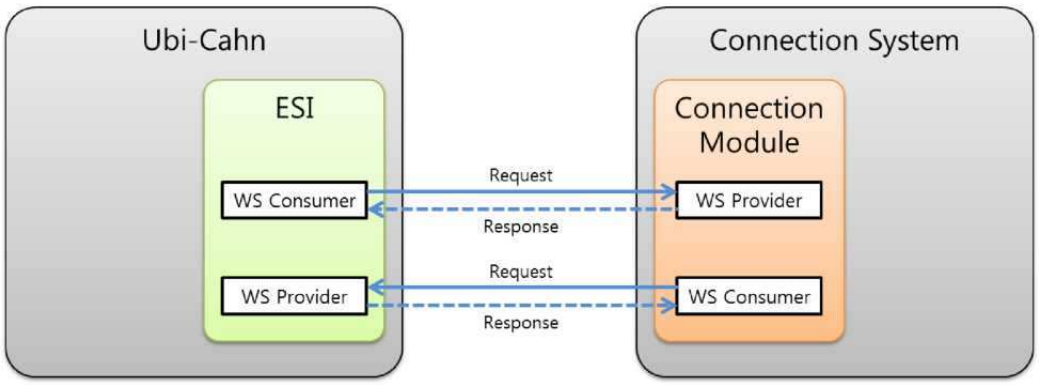


**Figure 4 – Web Service Interface Connection**

## 5.4. Message Encryption

- All data transmitted at data connection with outside U-Services and organizations are encrypted by applying the AES-128 algorithm, one of the standard symmetrical key algorithms of the U.S. and Europe. In case of socket and HTTP communication, the body of message excluding the header is encrypted at once. As for web service, all parameters excluding sending and receiving system codes are individually encrypted.

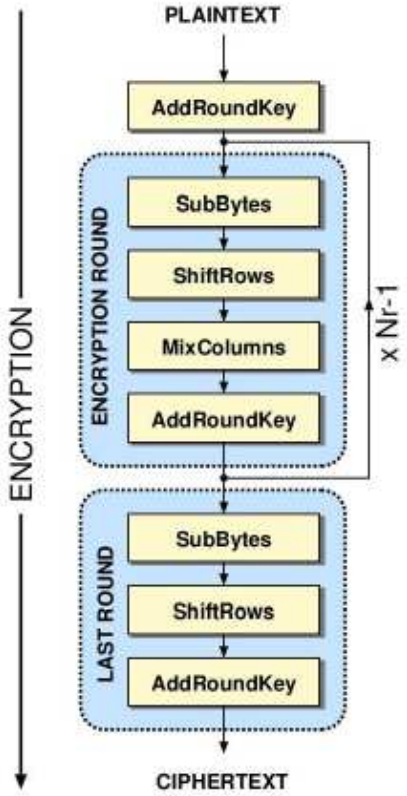


**Figure 5 - AES-128 encryption process**

- The AES-128 encryption creates encoded data by working in the order of key extension, startup round, middle round and final round. As for key extension, keys used in encryption are extended according to the key bits. In startup round, XOR operation is carried out for input data and the extended keys. In middle round, digit shift of each byte, rotation by set numbers in each row and inter-column change are carried out according to the prepared matrix. In the final round, operations with an exception of rotation generated in the middle round are conducted. In case of AES-128, encryption is carried out with one startup round, eight middle rounds and one final round.
- Encryption keys are defined by consultation between Ubi-Cahn and target systems for link. Encryption keys are managed by Ubi-Cahn and the target systems. In Ubi-Cahn, encryption keys decided through a consultation are registered for management at link target system registration in the link system management menu of the management UI. In this case, 16-digit (128bit) encryption keys must be used.
- Encryption is carried out in the following process. However, when using JAVA link library provided in Ubi-Cahn, the encoding is processed inside the library. In the C link library, UTF-8 encoding must be carried out in the link system.

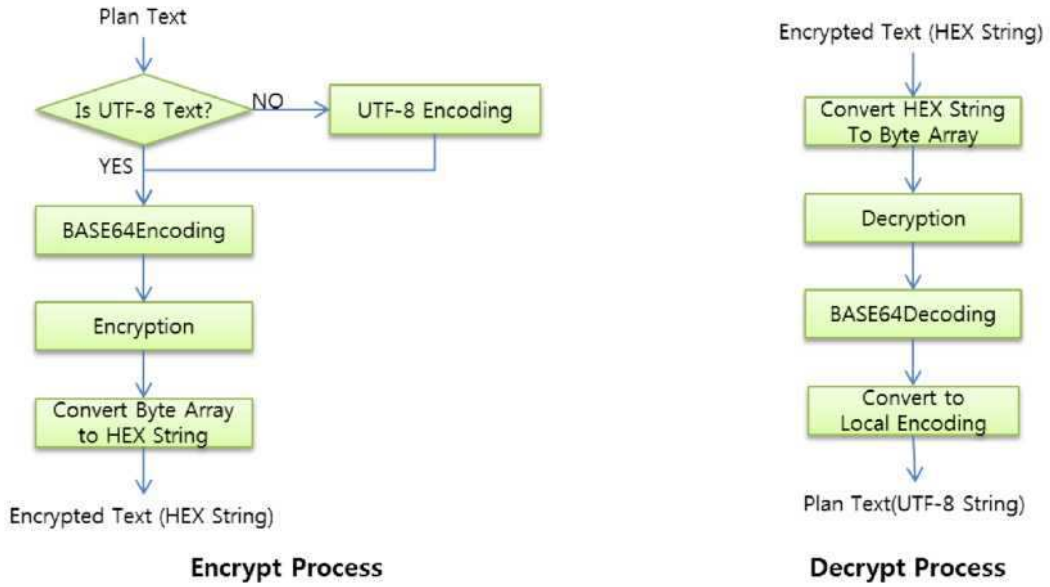


**Figure 6 - AES Encryption and Decryption process**

# 6. Acceptable network performance parameters

The acceptable network performance parameters for proposed smart services applications to be implemented in Lusail smart city are listed below in the table:

| Network Performance Parameters - Lusail Smart Services | |
|---|---|
| **Parameter** | **Threshold** |
| Packet Loss | 0.50% |
| Delay- Latency | 100ms |
| Jitter | 30ms |
| MTU Size | 576byte ~ 1500byte |

**Table 14: Acceptable Network Performance Parameters**

# 7. Appendix

## 7.1. Ubi-Cahn ESI link messages

The following is an example of WSDL provided in link messages of Ubi-Cahn ESI. In an actual environment, IP and port of endpoint URL must be checked according to the ESI installation environment.

* WSDL to receive Ubi-Cahn ESI situation event

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ns="http://rcvProc.otsid.msgProc.esi.kt.com" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<wsdl:documentation>
Please Type your service description here
</wsdl:documentation>
<wsdl:types>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<xs:element name="rcvOtsidStatEvet">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="sndSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="rcvSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="usvcStatEvetSeqn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetGdCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="procSt" nillable="true" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="outbPosAr" nillable=" false " type="xs:string"/>
<xs:element minOccurs="0" name="outbPosNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetCntn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetOutbDtm" nillable="true" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="statEvetItemAr" nillable="false" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetActnCntn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetActnMn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetActnDtm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="statEvetClrDtm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="outbMainGb" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rcvOtsidStatEvetResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="rcvOtsidStatEvetRequest">
<wsdl:part name="parameters" element="ns:rcvOtsidStatEvet"/>
</wsdl:message>
<wsdl:message name="rcvOtsidStatEvetResponse">
<wsdl:part name="parameters" element="ns:rcvOtsidStatEvetResponse"/>
</wsdl:message>
<wsdl:portType name="OtsidRcvStatEvetWsHandlerPortType">
<wsdl:operation name="rcvOtsidStatEvet">
<wsdl:input message="ns:rcvOtsidStatEvetRequest" wsaw:Action="urn:rcvOtsidStatEvet"/>
<wsdl:output message="ns:rcvOtsidStatEvetResponse" wsaw:Action="urn:rcvOtsidStatEvetResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="OtsidRcvStatEvetWsHandlerSoap11Binding" type="ns:OtsidRcvStatEvetWsHandlerPortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<wsdl:operation name="rcvOtsidStatEvet">
<soap:operation soapAction="urn:rcvOtsidStatEvet" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="OtsidRcvStatEvetWsHandler">
```

```
<wsdl:port name="OtsidRcvStatEvetWsHandlerHttpSoap11Endpoint"
binding="ns:OtsidRcvStatEvetWsHandlerSoap11Binding">
<soap:address
location="http://10.214.114.197:17030/ESI/services/OtsidRcvStatEvetWsHandler.OtsidRcvStatEvetWsHandlerHttpSoap11En
dpoint/"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

* WSDL to receive Ubi-Cahn ESI facility breakdown event

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ns="http://rcvProc.otsid.msgProc.esi.kt.com" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<wsdl:documentation>
Please Type your service description here
</wsdl:documentation>
<wsdl:types>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<xs:element name="rcvOtsidFacEvet">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="sndSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="rcvSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="usvcFacEvetId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facClfyNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="procSt" nillable="true" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="outbPosAr" nillable="false" type="xs:string"/>
<xs:element minOccurs="0" name="outbPosNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facBrkdnTypCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facBrkdnTypNm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetOutbDtm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetActnCntn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetActnMn" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetActnDtm" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="facEvetClrDtm" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rcvOtsidFacEvetResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="rcvOtsidFacEvetRequest">
<wsdl:part name="parameters" element="ns:rcvOtsidFacEvet"/>
</wsdl:message>
<wsdl:message name="rcvOtsidFacEvetResponse">
<wsdl:part name="parameters" element="ns:rcvOtsidFacEvetResponse"/>
</wsdl:message>
<wsdl:portType name="OtsidRcvFacEvetWsHandlerPortType">
<wsdl:operation name="rcvOtsidFacEvet">
<wsdl:input message="ns:rcvOtsidFacEvetRequest" wsaw:Action="urn:rcvOtsidFacEvet"/>
<wsdl:output message="ns:rcvOtsidFacEvetResponse" wsaw:Action="urn:rcvOtsidFacEvetResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="OtsidRcvFacEvetWsHandlerSoap11Binding" type="ns:OtsidRcvFacEvetWsHandlerPortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
```

```
<wsdl:operation name="rcvOtsidFacEvet">
<soap:operation soapAction="urn:rcvOtsidFacEvet" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="OtsidRcvFacEvetWsHandler">
<wsdl:port name="OtsidRcvFacEvetWsHandlerHttpSoap11Endpoint"
binding="ns:OtsidRcvFacEvetWsHandlerSoap11Binding">
<soap:address
location="http://10.214.114.197:17030/ESI/services/OtsidRcvFacEvetWsHandler.OtsidRcvFacEvetWsHandlerHttpSoap11En
dpoint/"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

* WSDL to receive Ubi-Cahn ESI connection event

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:ns1="http://org.apache.axis2/xsd"
xmlns:ns="http://rcvProc.otsid.msgProc.esi.kt.com" xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<wsdl:documentation>
Please Type your service description here
</wsdl:documentation>
<wsdl:types>
<xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://rcvProc.otsid.msgProc.esi.kt.com">
<xs:element name="rcvSvcToSvcTransfer">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="sndSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="rcvSysCd" nillable="true" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="destSysCdAr" nillable="false" type="xs:string"/>
<xs:element minOccurs="0" name="sndData" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rcvSvcToSvcTransferResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rcvOtsidConnEvet">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="sndSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="rcvSysCd" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="connEvetId" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="sender" nillable="true" type="xs:string"/>
<xs:element maxOccurs="unbounded" minOccurs="0" name="receiverAr" nillable="false" type="xs:string"/>
<xs:element minOccurs="0" name="title" nillable="true" type="xs:string"/>
<xs:element minOccurs="0" name="msgCntn" nillable="true" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rcvOtsidConnEvetResponse">
<xs:complexType>
<xs:sequence>
<xs:element minOccurs="0" name="return" nillable="true" type="xs:string"/>
</xs:sequence>
```

```
</xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="rcvSvcToSvcTransferRequest">
<wsdl:part name="parameters" element="ns:rcvSvcToSvcTransfer"/>
</wsdl:message>
<wsdl:message name="rcvSvcToSvcTransferResponse">
<wsdl:part name="parameters" element="ns:rcvSvcToSvcTransferResponse"/>
</wsdl:message>
<wsdl:message name="rcvOtsidConnEvetRequest">
<wsdl:part name="parameters" element="ns:rcvOtsidConnEvet"/>
</wsdl:message>
<wsdl:message name="rcvOtsidConnEvetResponse">
<wsdl:part name="parameters" element="ns:rcvOtsidConnEvetResponse"/>
</wsdl:message>
<wsdl:portType name="OtsidRcvConnEvetWsHandlerPortType">
<wsdl:operation name="rcvSvcToSvcTransfer">
<wsdl:input message="ns:rcvSvcToSvcTransferRequest" wsaw:Action="urn:rcvSvcToSvcTransfer"/>
<wsdl:output message="ns:rcvSvcToSvcTransferResponse" wsaw:Action="urn:rcvSvcToSvcTransferResponse"/>
</wsdl:operation>
<wsdl:operation name="rcvOtsidConnEvet">
<wsdl:input message="ns:rcvOtsidConnEvetRequest" wsaw:Action="urn:rcvOtsidConnEvet"/>
<wsdl:output message="ns:rcvOtsidConnEvetResponse" wsaw:Action="urn:rcvOtsidConnEvetResponse"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="OtsidRcvConnEvetWsHandlerSoap11Binding" type="ns:OtsidRcvConnEvetWsHandlerPortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<wsdl:operation name="rcvSvcToSvcTransfer">
<soap:operation soapAction="urn:rcvSvcToSvcTransfer" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="rcvOtsidConnEvet">
<soap:operation soapAction="urn:rcvOtsidConnEvet" style="document"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="OtsidRcvConnEvetWsHandler">
<wsdl:port name="OtsidRcvConnEvetWsHandlerHttpSoap11Endpoint"
binding="ns:OtsidRcvConnEvetWsHandlerSoap11Binding">
<soap:address
location="http://10.214.114.197:17030/ESI/services/OtsidRcvConnEvetWsHandler.OtsidRcvConnEvetWsHandlerHttpSoap11
Endpoint/"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

# 8. Standards References

1. **ANSI/TIA 568-C.2** - “Balanced Twisted Pair Telecommunications Cabling Standard”
2. **ITU-T G657.A2** – “Characteristics of Bending – Loss Insensitive Single –mode optical fibre and cable for the access network”
3. **ITU-T G652.D** – “Characteristics of a single-mode optical fibre and cable”
4. **IEEE 802.3** – Ethernet Working Group – develops standards for Ethernet networks.
5. **IEEE 802.3ab** – “Standard for Information Technology - Telecommunications and information exchange between systems - Local and Metropolitan Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications - Physical Layer Parameters and Specifications for 1000 Mb/s Operation over 4 pair of Category 5 Balanced Copper Cabling, Type 1000BASE-T”
6. **IEEE 802.3at** – “Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification Amendment : Data Terminal Equipment (DTE) Power Via the Media Dependent Interface (MDI) Enhancements.”
7. **IEEE 802.3z** – “Media Access Control Parameters, Physical Layers, Repeater and Management Parameters for 1,000 Mb/s Operation, Supplement to Information Technology - Local and Metropolitan Area Networks - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications”
8. **IEEE 802.15.4** –“Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)”
9. **IEEE 802.11** –“Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”
10. **RFC 791** - “Internet Protocol” - Commonly referred to as Internet Protocol Version 4 (IPv4)
11. **RFC 2460** – “Internet Protocol, Version 6 (IPv6) Specification”
12. **RFC 2453** – “RIP Version 2” – Routing Information Protocol (RIP) v2
13. **RFC 2328** – “OSPF Version 2” – Open Short Path Finder (OSPF) v2
14. **RFC 5798** – “Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6”
15. **RFC 793** – “Transmission Control Protocol” – TCP
16. **RFC 768** –“User Datagram Protocol”- UDP
17. **RFC 854** – “Telnet Protocol Specification”
18. **RFC 959** – “File Transfer Protocol” – FTP
19. **RFC 4253** –“The Secure Shell (SSH) Transport Layer Protocol”
20. **RFC 2616** –“Hypertext Transfer Protocol – HTTP/1.1”
21. **RFC 5531** –“RPC: Remote Procedure Call Protocol Specification Version 2”
22. **RFC 6101** –“The Secure Sockets Layer (SSL) Protocol Version 3.0”
23. **ISO/IEC 18033-3** –“Information technology – Security Techniques – Encryption Algorithms - Part3: Block Ciphers”
24. **ISO 16484-5** -**“**Building automation and control systems (BACS) -- Part 5: Data communication protocol” – BACNET Standard
25. **EIA/TIA-485-A** –“Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint ISO 16484-5Systems”
26. **EN 13757-3** –“Communication systems for and remote reading of meters. Dedicated application layer”- M-Bus Standard
27. **IEC 62056** -“Electricity Metering – Data exchange for meter reading, tariff and load control”- DLMS Standard.