

Documenting Functional & Volumetric Requirements

Instructions

SG-Network TF use case architecture functional and non-functional (volumetric) requirements, are a collection of application payload requirement sets that describe the interaction from the originating source actor(s) of a specific application payload, across intermediary actors, to the consuming actor(s), with primary focus on domain gateway actors and the telecomm communication paths.

Requirements Table “Reqmts-Combined” Tab

Each application payload requirement-set consists of multiple functional & volumetric table rows, where:

- requirements must be stated from a business perspective and not specific to what any technology or standard or protocol might theoretically specify
- payloads are stated as one directional, with any expected responses documented as a separate named application payload, unless specifically stated otherwise e.g. negotiation communication threads
- the sequence of the entered requirement rows, flows from the originating source actor(s), through any intermediary actors to the consuming actor(s).
- 1st row of a payload requirement set is the application payload parent for that requirement set, which identifies the actor(s) originating the payload, and the actor(s) that consume the payload. The parent rows from/to actor pairs may consist of a 1 to many or many to 1 relationship, but not a many to many relationship to make the parsing of the data flow “pseudo code” easier to handle. The parent row is to use bold text.
- the remaining rows (children rows), for that specific parent’s from/to actors, are entered in sequence as one traverses from the originating source actor(s) through the any alternative intermediary actor communication or interface paths, ending with row(s) for the from to actor pair for the consuming actor(s).
- Notes:
 - if there are no intermediary actors or alternative communication paths or interfaces, then only one requirement row is required, and it is treated as a parent
 - no child row shall have any embedded intermediary actors for that child’s from/to actor pairing.
 - it is acceptable for two child rows with the same data flow ref identifier, provided that the parent data flow requires an intermediary actor to have both a from and to role with the same “other” to and from actor for a specific subset of a payload communication path.
- application payload parent volumetric (architectural non-functional), requirements must mathematically be satisfied by the application payloads' children rows that are entered below the parent for top down data flows for any documented alternative communication path between the initiating actor to the target consuming actor(s).
- There are some columns (e.g. yellow shaded column heading), that are inserted and used for programmatic vetting of the requirements data (see Content Vetting – Formula Based Checking section later in this document). These columns are carried in the

work-in-progress candidate versions for a specific release version of the Requirements Table and are not included in the “clean” release version.

Version Control will be applied to the requirements table per the following:

- each release of the table will contain all the past and current documented requirements
- when changes or additional payload requirement-sets are added to previously issued tables, the release number will increment as in 2.0 to 2.1, 2.2 to 2.3
- when “new” use case requirements are added (no previous requirements for a named use case existed), the release number will increment by a whole number e.g. 2.2 to 3.0. This release may also include changes (adds, deletes, edits) to previously released payload requirement-sets.
- changes from one release to the next, will use some form of formatting indicating what data has changed, e.g. shading (light blue or light green, something that does not make entered text hard to read), of table cells or specific text in a cell e.g. red text. Any flagged changes (version change text formatting) from previous releases, will be removed e.g. using no cell shading and black text. This version change formatting syntax **only applies to the work-in-progress** versions and not the posted “clean” versions starting with Requirements Table 5.0.

The Requirements Table spreadsheet contains 2 main tabs, “Reqmts-Combined” and “Payload_attrib_LIC_CIA_rtnl”. A description and data entry syntax of the columns found in these two spreadsheet tabs, follows. Other tabs included are for general summary information and data e.g. “Diag-xref-table” used by embedded vetting macro purposes.

Rqmt Ref – This column is a unique fixed reference to a given application payload's actor to actor pairing functional and volumetric requirements row. Syntax and redaction:

- [short acronym (upper case) based on “Use Case” name | “- “ | zero left filled numeric 3 to 4 digits] applied after all of the use case's payload sets have been entered and the “normal” use case row sequencing applied:
- new requirement rows are manually inserted where appropriate (in the normal table sequence), with the next available use case “Rqmt Ref” sequence number
- Notes:
 - this reference number is not to be changed from one release of the requirements table to the next, and no significance is to be applied to the “Rqmt Ref” value other than it is unique for a given row and will remain with the requirement row regardless of what edits are made to that row
 - To be retired/deleted rows will be physically removed from the requirements table and will result in voids in the “Rqmt Ref” sequencing. If there is a need to retain these retired/deleted rows for possible future reuse, then the physical delete step must be changed to mark the row for deletion e.g. text strike-through, resulting in no anticipated voids in the use case “Rqmt Ref” numeric portion of the label.

Row Type – This column is used to indicate that the row is a “parent” row for the specific payload source actor(s) and the consuming actor(s), and is of primary use in data load to the Requirements Data Base. Syntax: [P (if parent) | else c for child]

Data-Flow Ref – This column is a reference to the Reference Diagram lines (interfaces) and data flows between actors shown and documented as pseudo logic code. Syntax:

- logic code separators – the following separators are to be used only as the parsing routines in the Requirement Database expects to find these separators. A simple data flow reference may not require any separators:
 - “[“ and “]” carry the outer most, highest level
 - “(“ and “)” carry the 2nd highest level of separation, with multiple entries allowed
- operators:
 - “+” logical and
 - “or” logical or
 - “&” used to denote the parent includes multiple originating actors or consuming actors and multiple clones of the payload must be accounted for in this specific payload requirement set
- the parent row includes the complete set of data flows or communication (comm.) paths from originating actor to consuming actor(s) including alternatives e.g. “[(ref1 or ref2-alternative)] + ref10 + (ref20 & (ref25 + ref26))”
- for children requirement rows:
 - originating actor to 1st intermediary actor e.g. “[ref1 or ref2]” or “ref9”
 - intermediary actor to intermediary actor e.g. “ref30”
 - intermediary actor to consuming actor e.g. “[ref40 or ref42]” or “ref45”
- no data flow reference and child row shall encompass more than 2 in-the-communication-path actors and must traverse at least one network cloud (per the Reference Diagram used as the design basis).

Data Flow From Actor – This column indicates the actor(s) that is/are considered the sender of information (application payload) noted in the Requirements and Payload Name columns. Syntax:

- actor(s) must be included in the Actors table and have the same spelling as used in the external Actor table and Reference Diagram entry as qualified by Domain or other qualifier to help maintain uniqueness,. e.g. [actor – domain-name]

Data Flow to Actor – This column indicates the actor(s) that receives the information (application payload) as noted in the Requirements and Payload Name columns. Syntax:

- actor(s) must be included in the Actors table and have the same spelling as used in the external Actor table and Reference Diagram entry as qualified by Domain or other qualifier to help maintain uniqueness,. e.g. [actor – domain-name]

Requirements – This column is the actual application requirement. Words like “shall” in this column are to be considered required, while words like “may” should be considered optional. Syntax:

- statement starts with the name of the “Data Flow From Actor”
- with action verbs coupled with other phrases that indicates:
 - originating source actor sends or processes & sends
 - intermediary “data flow from actor” [processes (if any reformatting of the application payload) & forwards | forwards | inspects and forwards] the information (fully qualified)
- statement ends with the name of the “Data Flow to Actor”

Payload Name – This column is a unique payload label used for any requirement rows or use

cases that specify the exact same application payload description and attributes as described in the "Payload_attrib_LIC_CIA_rtnl".tab. Syntax:

- [keyword1_ | keyword2_ | keywordn_ | payload type]

Payload Type – This column specifies the payload type and is unique to each payload, and generally denotes one directional flows. Syntax (acceptable values):

- ack – acknowledgement to a received command (request)
- alarm – message that originates from any actor indicating a condition that warrants immediate attention
- alert - message that originates from any actor indicating an condition that may require further attention
- cmd – command (request) to perform a specific action by the target (consuming) actor
- cmd-err – command execution or processing error by the target (consuming) actor
- comm-err – communications error in sending or forwarding the payload between actor pairs
- negot-thread – negotiation thread – used as shorthand to account for a variable number of passing different (but cyclical) payloads between the originating and consuming actor to reach agreement on the operating parameters or options for the two actors.
- rpns-data – response data to a command (request)

Daily Clock Periods – This column indicates the primary times during the day that the payloads are sent or forwarded. Syntax:

- [start hour – end hour (1st period using 12 hour format) + start hour – end hour (2nd period using 12 hour format)]

How Often – This column attempts to describe the quantity and frequency of the specific application payload as it moves between the stated actors across the interface (data flow) that this requirement row addresses. Syntax (also refer to the Communications Path Options Discussion Diagram):

- is separated into 3 components: quantity; of some specific item; frequency per unit of time or event e.g. ["quantity per" " | "of some specific item per" " | "frequency per unit of time or event"]
- quantity "of some specific item", e.g. "12 per 1000" or "1"
- of some specific item, e.g. "ElectrMtr" or "Sub(x)DAC" or "fdrVR". Refer to the "HowOften-abbrev-xref" tab of the Requirements Table spreadsheet for other actor cross references. Add to that tab additional How Often abbreviations as needed.
- frequency per unit of time or event, e.g. "week", "12 hours", "x zone isolation step event". Note: frequency of 1 per time unit unless stated otherwise
- make sure to fully qualify, either in the: of some specific item; frequency per unit of time or event components, exactly what is being accounted for in this payload. If this is a child it is really important to also make sure the child's parent qualifiers are included, e.g. "DAPjm-ESIUtl-PrePay-ElectrMtr-with-CustEMS per" or "LMS - REPi CPP price data broadcast event - 12 per year"
- for application payloads that are used in several use cases, make sure there are NO duplication of quantities across those use cases for that specific application payload
- make sure that mathematically the children rows "How Often" are consistent with the parent's "How Often"

Note: it is expected that the vast amount of parsing and scaling of the non-functional requirements (especially “How Often”), will be handled in the Requirements Database.

Reliability – The probability that an operation will complete without failure over a specific period or amount of time. Syntax (also refer to the Communications Path Options Discussion Diagram, at the end of this document):

- the parent row's value must be mathematically consistent with the product of children's reliability values chain that make up the originating actor to consuming actor communication path scenario.
- data center actor to actor reliabilities are generally higher values than those in networks where less control over environment and processes for the actor to actor interfaces may exist.

Latency – Summation of actor (including network nodes) processing time and network transport time measured from an actor sending or forwarding a payload to an actor, and that actor processing (or consuming) the payload. Syntax (also refer to the Communications Path Options Discussion Diagram), the parent's value must be equal or greater than:

- any communication path scenarios for the parent's stated from actor and to actor
- and must be mathematically consistent with the highest summation of children latencies specific to any communication path scenario

App Payload Size – This column is an estimation of how many bytes are needed for the requirement application payload, without any other overheads included. No explicit assumption of byte encoding to reduce payload size for transmission is made.

Getting Started – Suggested Steps

From the body of published or available collections of classically documented uses cases as information flows and/or illustrated in sequence diagram flows, select one that has pertinence to the network communication effort. Review the specific flows as they tend to identify specific business information movement from one business actor to another business actor.

1. From the selected use case and specific business information flows from above, a “new” use case and associated application payload functional & volumetric requirement payload sets can be created by outlining in a scratch spreadsheet tab the specific steps and the application payloads and the string of from to actors across the telecomm networks as illustrated in the Reference Diagram. From this scratch worksheet, the content can be copied into the Requirements Table template of columns and formatting. The rest of the Requirement Table columns can be added in. **Note:** the alternative is to jump to step 2 if the additional application payload requirement sets are very similar to any that are already documented in the Requirements Table.
2. Make a copy of the current requirements table, and at the bottom, create the payload requirement set, starting with the parent row for that selected application payload
 - a) Identify the various communication path scenarios the parent row might take, then select either the longest path e.g. number of actors because it should make creating the other scenarios easier, or pick the scenario you are most familiar with
 - b) Decompose the parent into the individual children rows traversing through one of

- the end-to-end communication paths, actor by actor
- c) Note, if the diagram data flows and correct communication paths have not already been identified in the current master diagram, continue on without that information. Once the use case payload requirement sets are drafted, then share that with the maintainers of the master diagram and the needed data flows references and other communication paths can be created and added to the master, diagram and then entered into the payload documentation sets.
3. Create the response payload requirement set (if the one selected in (2) was a “cmd” payload type, using the same steps identified in step 2a & 2b. This should make traversing backward through the communication paths and actors easier since you just finished the forward traverse in step 2a & 2b.
 4. With these two initial payload requirement sets completed, go into each one and incrementally and manually insert the alternative communication paths children for each payload as needed.
 5. Continue this process for the remaining payloads for that use case. If there are payloads that have already been created in the current release that traverse the same originating actor to consuming actor, copy paste that into this new use case, change the payload names and requirements as appropriate along with all the other volumetric data as appropriate to that use case and payload. At least the majority of the number of children and the references to the data flow references hopefully will still apply.

Remember to take advantage of the table filtering capability to help focus on the specific payload of interest or using custom filters to select two similar payload requirement sets as reference in creating or modifying new payload requirement sets. Note: having use of dual monitors really eases the steps above.

Content Vetting – Formula Based Checking

Other than simple visual inspection of each and every row and cell entry is entered correctly for the payload requirement set, another approach is to use some programmatic checking to flag inconsistencies (errors) especially for child requirement rows. “SG Network System Requirements Specification v5.0-draft1i.xls” file introduced a formula based set of calculations that checked a subset of the Requirement Table column contents. Additional vetting tools (VBA macros), were added in the “SG Network System Requirements Specification v5.1-drafta.xls” file. This programmatic vetting code will be retained in the SG Network TF shared documents via retention of the “SG Network System Requirements Specification v5.1-draft1a.xls” file for future reuse. Note, that the Requirements Database also contains some content vetting routines.

Specific to the Requirements Table checking, take note of the capabilities described below. **Hint**, to speed up routine editing, filtering, grouping operations of any work-in-progress Requirements Table, either **do not** load in these formula based checking code until needed for vetting, as they increase processing time for large quantities of requirement rows. Alternative is to change the spreadsheet calculation process to manual mode. Also, the recommendation is to perform the following cleanup of column entries, in the order presented below.

Data Flow: Ref, from Actor, to Actor Vetting (written by Juan Oliver and contributed by David Gosch IAPsolutions) – An additional tab “Diag-xref-table” needs to be inserted into the work-in-progress Requirements Table, which also needs to be kept in-sync with the Reference Diagram (Object ID, test, custom resource text extract from VISIO), and the current Requirements Tables use of Actor names. This tab is used in vetting formula calculations (e.g. refer to 4 manually added columns to the right of the “Data Flow to Actor” column as noted in “SG Network System Requirements Specification v5.0-draft1j.xls”, or in the latest work-in-progress Requirements Table draft. This checks for the existence of the Data Flow / Interface references in the “Diag-xref-table” tab and what the expected end-point actors should be as noted in column B of that tab, then compares that to what is found in the Requirement Table Data Flow from/to Actor cells for that row.

Expected vetting responses include: null e.g. names match; “Invalid” e.g. name mismatch that may require a change to the diagram object custom resource name, or rename of the diagram name, or ignore the “Invalid” flag; “N/A” e.g. the formulas are not coded to deal with fuzzy logic word / phrase matching, generally ignore these entries.

Requirements Column Vetting – 2 vetting columns are manually added to the right of the “Requirements” column as noted in “SG Network System Requirements Specification v5.0-draft1j.xls”, and in the latest work-in-progress Requirements Table draft, that flag errors in Data Flow from/to Actor entries being at the beginning and end of the “Requirements” cell text.

Expected responses includes: null e.g. Requirements text matches the from/ to Actor(s); “err” e.g. the length of text associated with the from/to Actor does not exactly match what is found in the Requirements cell. Either correct the Requirements cell text or accept the Requirements phrasing as is.

How Often – 2nd parsing component Vetting - An additional tab “HowOften-comp2-vet-ref” needs to be inserted into the work-in-progress Requirements Table, which also needs to be kept in-sync with the current Requirements Tables use of Actor names and abbreviations. This tab is used in vetting formula calculations (e.g. refer to 2 manually added vetting columns to the right of the “How Often” column as noted in “SG Network System Requirements Specification v5.0-draft1j.xls”), and in the latest work-in-progress Requirements Table draft, that flag errors in missing Data Flow from/to actor abbreviation entries in the 2nd parsed component of the “How Often” cell text. Note: this vetting is not expected to be as exact as the previous 2 vetting subsets, as the How Often 2nd parsed component needs to indicate exactly what Actor quantity drives the payload quantities in the stated frequency.

Expected responses includes: null e.g. 2nd parsed component of the How Often column matches exactly the from/to Actor name abbreviation; “#VALUE” or “#N/A” e.g. the 2nd parsed component does not match the from/to Actor name abbreviation or that the appropriate Actor reference is tied to an consuming or source Actor and this child row is for an intermediate Actor to Actor pairing. Take note of these just to question that the 2nd parsed component is appropriate and if not correct the 2nd parsed component Actor name abbreviation/combine as needed. Otherwise ignore.

Parent Row Data Flow Ref pseudo code Bracket/Parentetical Imbalance and Parent versus Child Row Checking (written by Sunjot Singh Hunjan and contributed by Cheong Siew BC Hydro) – 3 vetting columns are added at the end of the requirements data columns (via the

“brackets_child” VBA macro), that flags for: the quantity of open/close brackets and parenthetical(s) do not match and by how much; a child row is present but not referenced in the parent row Data Flow Ref pseudo code; a data flow is referenced in the parent row Data Flow Ref pseudo code but no matching child row is included in the payload requirement set. Note: before running this macro, either manually delete the cell contents of the existing e columns mentioned below, or run the “deleteColumns” macro.

Expected responses for:

- “Bracket Imbalance” includes: null e.g. left and right brackets/parenthetical(s) match in quantity; a +/- value and the bracket or parenthetical indicator of quantity imbalance
- “Child not in Parent” includes: null e.g. no orphaned child row(s); Data Flow Ref value(s) of orphaned child row(s) in the payload requirements set, but not included in the Parent row Data Flow Ref cell, if Parent row Data Flow Ref entry is correct, then flag the child row for deletion, otherwise correct Parent row Data Flow Ref entry to account for the child row; compound interface names in the child row Data Flow Ref cell, check out this entry for being valid and if so ignore as the macro currently does not handle this case, otherwise correct either the child row or the parent Data Flow Ref entry as needed.
- “Missing Child Row” includes: null e.g. no missing child row(s); Data Flow Ref value(s) of missing child row(s) in the payload requirements set, but not included in the Parent row Data Flow Ref cell, if Parent row Data Flow Ref entry is correct, then add the missing child row, otherwise correct Parent row Data Flow Ref entry to delete reference to the missing child row; compound interface names in the Parent row Data Flow Ref cell, check out this entry for being valid and if so ignore as the macro currently does not handle this case, otherwise correct either the child row or the parent Data Flow Ref entry as needed.

Note: there is a manually added column to the right of the “App Payload Size” column that is used to filter on those requirement rows that need to be addressed e.g. potential content redaction or discussion prior to final disposition.

All of the content checking columns and inserted tabs are removed from the clean release version of the Requirements table.

Requirements Table “Payload attrib LIC CIA rtnl” Tab

SGIP PAP02 and CSWG requested that SG Network TF perform a mapping of the Requirement Table payloads to specific NISTIR 7628 security Logical Interface Categories (LIC). This task was accomplished (and continues for new payloads being added to the Requirements table), by documenting the following information, where the SG Network TF proposed payload to the NISTIR 7628 LICs are vetted with CSWG and OpenSG SG Security WG. The following describes how the Requirements Table “Payload attrib LIC CIA rtnl” tab contents are to be documented.

Payload Name – Copied from the “Reqmts-Combined” tab “Payload Name”. Hint, the list of Payload Names is easily created by performing a pivot or datapilot of the “Reqmts-Combined” using “Payload Name” as the primary row element.

Payload Type- Copied from the “Reqmts-Combined” tab “Payload Type”. Hint, the list of Payload Types is easily created by performing a pivot or datapilot of the “Reqmts-Combined” using “Payload Name” and “Payload Type” as the primary and secondary row elements.

Payload Description – Short explanation of what is the application payload use and intent.

Payload Attributes – Lists the data elements that are included in the application payload. This excludes any additional security and/or telecommunication protocol(s) added data elements around that application payload. Though not specifically listed, date-time stamps are assumed for all Payloads.

Security LICs - NISTIR 7628 – Logical Interface Category (LIC) derived and mapped (as closely as possible to typically no more than 2-3 LICs) from the NISTIR 7628 document volume 1, section 2.3 “Table 2.2 Logical Interfaces by Category” and remaining sections 2.3.x to the specific application payload in question. Consideration of the originating and consuming actors, specific Domains, telecomm networks used, and application payload content are also necessary in selecting specific LICs. Syntax:

- [“numeric-alpha value 1” , “numeric-alpha value 2”/“numeric-alpha value n”]

Payload C-I-A Risk Values- Confidentiality, Integrity, Availability security risk levels as described in NISTIR 7628 document volume 1, section 3.2 “Table 3-1 Impact Levels Definitions” and assigned based on the application payload’s description, attributes, C-I-A rationale excluding other security or telecomm network protocol(s) overhead data elements. Syntax (acceptable values):

- [Confidentiality-risk-level “-“ Integrity-risk-level “-“ Availability-risk-level] where the risk levels have values:
 - L low
 - M moderate
 - H high

Security C-I-A Risk Values Rationale – This column is an attempt to document the business impacts of the payload being compromised as assessed against the security confidentiality, integrity, availability areas. Syntax:

- [“C – “ business impacts and severity due to compromised payload <in cell crt>

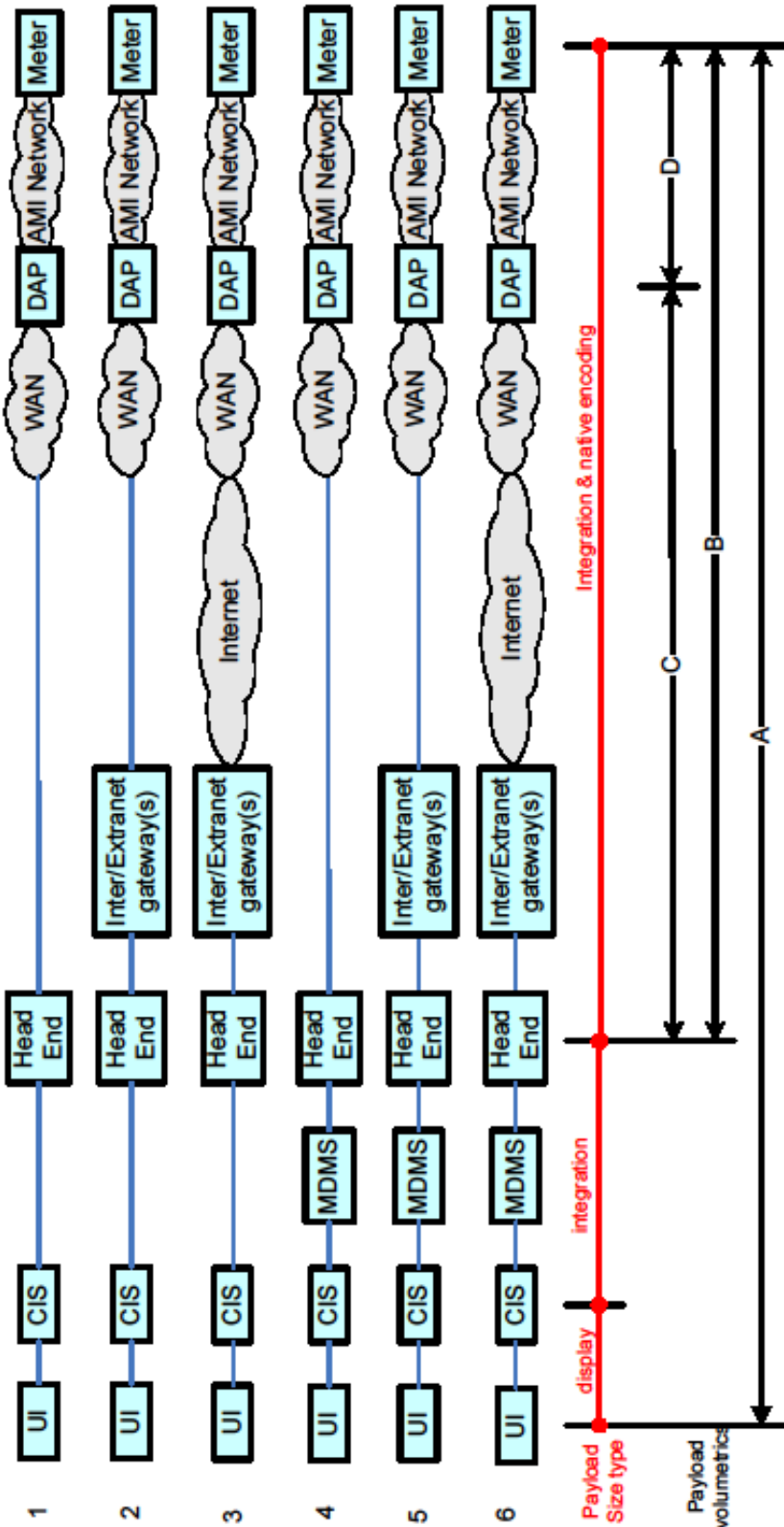
“I – “ business impacts and severity due to compromised payload <in cell crt>

“A – “ business impacts and severity due to compromised payload.

- if any comments on LIC mapping, include those as <in cell crt> “LIC – text”.

Note: to ease the loading of this content into the Requirements Database, do not use “:” in any of the cell entries.

Utility CIS <-> Meter - Comm Path Options



Notes:

- 1) Business volumetric requirements are being documented for the application payloads between specific actors for specific communications paths. Typically it is easier to document these volumetric requirements for the dataflows (ref boundary points above), in the following order: A, C, B or D.
- 2) Most volumetrics for dataflows for the same payload, MUST NOT be relaxed more than their parents volumetrics requirements. E.G.: a) the latency for dataflow B, MUST be < A's; b) C + D latency MUST be < B's; HOWEVER, the total amount of payloads per dataflow will diminish as one traverses towards a singular endpoint, e.g. specific payload qrys for CIS & HeadEnd probably will be equal, HeadEnd = sum of DAPs; a specific DAP = sum of meters that the specific DAP has been designed/deployed to handle.
- 3) For the CIS <-> Meter comm paths, several payloads will concurrently traverse, each with different business volumetric reqmts. E.G.: a) on-demand meter reads, b) meter read of multiple interval data time blocks, c) service switch ops & status, d) demand resets, e) meter last-gasp, f) and this same common path used to pass HAN, prepay, load control, type messages to HAN devices

Communications Path Options Discussion Diagram