COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

MASTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MASTER

ONESAIT METERING FOR EDP

Autor: Eduardo Benavente Martínez

Director: Miguel Díaz Rodríguez

I declare, under my own responsibility, that the Project submitted under the title
Onesait Metering for EDP
at the ETS de Ingeniería - ICAI of the Universidad Pontificia Comillas in the
academic year 2021/22 is by me, original and unpublished, and
has not previously been submitted for other purposes.
The Project is not plagiarised from another, either in whole or in part, and information that
has been
taken from other documents is duly referenced.

S/ Eduardo Benavente Martínez Date: 25/ 08/ 2022

Project delivery authorised
THE PROJECT MANAGER

S.D.: Miguel Díaz Rodríguez Date: 25/ 08/ 2022

ONESAIT METERING PARA EDP

Autor: Benavente Martínez, Eduardo.
Director: Diaz Rodriguez, Miguel.
Entidad Colaboradora: Minsait (Indra IT Solutions)

RESUMEN DEL PROYECTO

La lucha contra el cambio climático es uno de los retos más importantes que debe afrontar la humanidad, y la apuesta por una economía descarbonizada basada en las energías renovables en un nuevo mix energético es una tarea de todos. Así nació en España el Plan Nacional Integrado de Energía y Clima (PNIEC), que tiene como objetivo mínimo alcanzar el 32% de energías renovables en 2030. Este objetivo es alcanzable en un escenario de alta descarbonización y electrificación de la economía, utilizando combustibles descarbonizados en aquellos nichos de difícil electrificación [1]. El reto de extender estas redes a millones de puntos finales, potencialmente dispersos geográficamente en grandes áreas de servicio, es inherente a la red inteligente [1].

Ante este nuevo paradigma, las empresas de servicios públicos se enfrentan a una serie de retos de transformación digital marcados por la migración de la cadena de suministro de energía a un nuevo modelo distribuido. Dentro del desarrollo de las redes inteligentes, el Smart Metering es uno de los pilares fundamentales, impulsado por la sostenibilidad y la satisfacción del usuario. Minsait ha desarrollado la tecnología Onesait Utilities Metering, una solución de medición inteligente diseñada para las utilities que permite una ventaja competitiva en un mercado energético afectado por las tendencias globales, permitiendo a la Utility recoger, certificar y exportar datos de información de diferentes fuentes de información [15]. EDP (Energías de Portugal), uno de los mayores grupos eléctricos de Europa, ha solicitado los servicios de Minsait para integrar la solución OUM en sus utilities en España. Se integrarán tres entidades bajo un único gestor (EDP-España, Viesgo y Begasa). Este proyecto se centrará en el análisis y desarrollo funcional de la integración de OUM bajo la especificación STG-DC de PRIME en EDP.

Al integrar diferentes instituciones en la infraestructura de EDP España, surge un problema para la empresa a la hora de explotar la información de forma ágil, sencilla y transversal con el objetivo de ofrecer los nuevos servicios introducidos en el sector de la distribución eléctrica. Este problema consiste en la interoperabilidad y centralización de los diferentes equipos que componen el sistema. Es decir, cada una de las empresas que se han integrado en la cartera de EDP utilizaba diferentes tecnologías, diferentes protocolos de comunicación, diferentes tipos de dispositivos y diferentes soluciones de telegestión.

La parte adquirida de E-Networks tiene aproximadamente 700.000 contadores Prime que se comunican con el protocolo 1.3.6. También cuenta con 7.000 concentradores Prime 1.3.6 que se comunican con el AMI utilizando el estándar STG-DC en su versión 3.4. Además, el

sistema incluye varios tipos de equipos de monitorización avanzada (que utilizan el estándar STG-DC), así como equipos como routers e interfaces de fibra óptica.

Por parte de Begasa + Viesgo hay 700.000 contadores que utilizan Telegestore como protocolo de comunicación con los concentradores y 12.000 concentradores que actualmente se comunican con el AMM a través de un estándar privado propiedad de Endesa.

El objetivo de este proyecto es colaborar en la integración de estos sistemas en la herramienta multiprotocolo Minsait OUM, con el fin de centralizar todas las comunicaciones para permitir una única gestión remota de todo el parque de equipos de forma que el usuario sea capaz de monitorizar, mejorar y automatizar todos los procesos de comunicación que se producen con cada uno de los equipos de medida que componen la red.

Estado del arte

El equipo de medición de electricidad utilizado en todo el mundo es el conocido contador de electricidad, que consta de tres elementos principales, a saber, el sistema de medición, el elemento de memoria y el dispositivo de información. En este sentido, el contador de electricidad realiza la función de interfaz de la red con el usuario, es el front-end de la red. Estos equipos de medida se pueden clasificar según sus características tecnológicas (si son electromecánicos o electrónicos), por su funcionalidad monofásica o trifásica, por su energía y por sus características operativas como dispositivo de tipo registrador programable que permite la gestión a distancia. Estos contadores han evolucionado notablemente en el mercado eléctrico a medida que se han ido desarrollando nuevas tecnologías.

Estos dispositivos están basados en la tecnología AMI (Advanced Metering Infrastructure), estos dispositivos permiten la lectura de la energía acumulada o de la potencia instantánea, permiten el registro de la demanda o la programación de intervalos de carga previamente definidos con cada usuario. De estas nuevas e incipientes tecnologías nace el concepto de Smart Meter. Estos dispositivos proporcionan información y control de los parámetros de calidad y programación del servicio a través del centro de gestión, junto con la actualización telemática del software de medición. En base a estas razones y a las descritas anteriormente, se justifica el interés por el despliegue e implantación de esta tecnología, así como la importancia de todos los estudios que engloban este mundo y entre los que se encuentra este proyecto.

Arquitectura de comunicaciones AMI

El acrónimo AMI, como ya se ha mencionado, significa Advanced Metering Infrastructure y representa el corazón y el alma de esta tecnología. Para empezar a describir esta arquitectura estándar para los sistemas de comunicaciones basados en la medición avanzada, es necesario discernir a alto nivel cuáles son los principales elementos que la componen [17]:

- **Contador inteligente:** Es un contador electrónico inteligente capaz de registrar el consumo de electricidad a intervalos de tiempo configurables y comunicar la información de este registro a la unidad de control. Se trata, por tanto, de un instrumento que cumple la función de sensor y punto de interacción simultáneamente, es decir, un nodo inteligente.

- **Concentrador de datos (DCU):** Es la columna vertebral de la Infraestructura de Medición Avanzada (AMI) que soporta la adquisición de datos, la transferencia de datos de energía a la base de datos central y la lectura automatizada de los contadores electrónicos de energía. Las DCU se utilizan principalmente en los sistemas de energía eléctrica de baja tensión.

- **Sistemas de información:** Estos sistemas de información actúan como enlace entre la adquisición de datos y su posterior procesamiento. Están diseñados como bases de datos (tanto relacionales como no relacionales) y tienen capacidad de gestión remota. Estos sistemas de información se caracterizan por tener una visión completa de la infraestructura.

Estos sistemas suelen ser infraestructuras de comunicación heterogéneas, en las que intervienen diferentes tecnologías en función de los requisitos específicos de cada segmento de la red. En España, la comunicación entre los contadores y los concentradores de datos se basa en tecnologías PLC de banda estrecha.

Las comunicaciones por línea eléctrica (PLC) son una tecnología de telecomunicaciones que se basa en el uso de la infraestructura de distribución de energía eléctrica (normalmente BT o MT) para la transmisión de señales y datos a alta velocidad. Proporciona comunicaciones de datos de banda ancha a través de conductores que ya se utilizan para la transmisión de energía eléctrica mediante una señal modular. Esta tecnología PLC se considera un medio de comunicación fiable para aplicaciones como el Internet de las cosas (IoT) y las redes inteligentes [3].

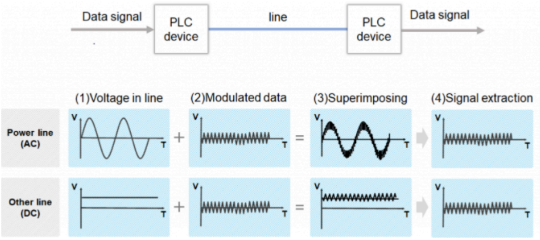


Figura 1: Principio básico de funcionamiento del PLC [3].

Aplicación de PRIME para AMI

PRIME (PoweRline Intelligent Metering Evolution) es una norma abierta, libre y no propietaria de una tecnología PLC OFDM establecida a nivel mundial para aplicaciones avanzadas de medición, control de redes y supervisión de activos, y ha cumplido el objetivo de establecer un conjunto de normas PLC abiertas a nivel internacional. Hay más de 20 millones de contadores PRIME instalados en más de 15 países de todo el mundo [5]. Desde el inicio de la desregulación de la electricidad y la fijación de precios basada en el mercado en todo el mundo, las empresas de servicios públicos han buscado un medio para ajustar el consumo a la generación. Los contadores tradicionales de electricidad y gas sólo miden el consumo total y, por lo tanto, no proporcionan información sobre cuándo se ha consumido energía en cada lugar. El PRIME AMI ofrece una forma de medir esta información específica de cada emplazamiento, lo que permite establecer diferentes precios para el consumo en función de la hora del día y la estación del año [5]. Una de las ventajas de implementar el protocolo PRIME es que garantiza la interoperabilidad entre diferentes equipos y sistemas, no está sujeto a ninguna propiedad intelectual y está diseñado para conseguir un bajo coste de implementación y un alto rendimiento, acelerando así la entrada de nuevos operadores.

La capa física, la capa MAC y la capa de convergencia de la versión 1.3.6 de la especificación han sido consideradas estándar por el UIT-T desde 2012. La versión 1.4 amplía el espectro de frecuencias para permitir el funcionamiento en los mercados norteamericano y asiático e incluye algunas mejoras destinadas a mejorar la solidez a nivel físico y MAC [7, 8]. En el nivel MAC, una subred tiene una estructura en forma de árbol, en la que se distinguen dos tipos principales de nodos:

- **Nodo base (BN):** El nodo base es el nodo maestro que gestiona una subred PRIME. Sólo hay un BN por subred, y es el responsable de gestionar los recursos y las conexiones, actúa como nodo maestro y está en la raíz de la estructura del árbol, es el elemento más importante. En la terminología de AMI, el nodo base es el concentrador de datos [7, 14].
- **Nodo de servicio (SN):** Los nodos de servicio son el elemento central de la subred. Pueden verse como hojas o ramas en la estructura de árbol y están inicialmente en un estado desconectado, siendo sus funciones mantener la conectividad a la subred para las diferentes capas de aplicación y conmutar la información de otros nodos. Los contadores inteligentes deben registrarse en la subred para convertirse en nodos de servicio que dependen de su correspondiente BN. En la terminología de AMI, los nodos de servicio son contadores inteligentes. [7, 14].

Inicialmente la subred está compuesta sólo por el nodo base y el resto de nodos que quieran formar parte de ella tienen que realizar un procedimiento de registro.

- **Desconectado:** Este es el estado inicial de cualquier nodo de servicio antes de registrarse en la subred PRIME. Mientras está en este estado, el contador inteligente no puede comunicar sus propios datos ni conmutar los de otros SN y su único objetivo es buscar una subred a la que pueda llegar y registrarse en ella.

- **Terminal:** Este estado permite a la SN establecer nuevas conexiones y transmitir datos, pero no conmutar datos de otras SN.
- **Conmutación:** Este estado es el más complejo y permite a la SN establecer nuevas conexiones, transmitir sus propios datos y conmutar datos de otras SN.

Por último, se lleva a cabo un proceso de registro y baja.

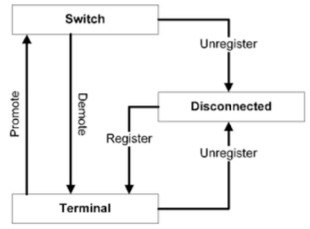


Figura 2: Estados funcionales de un nodo de servicio [11]

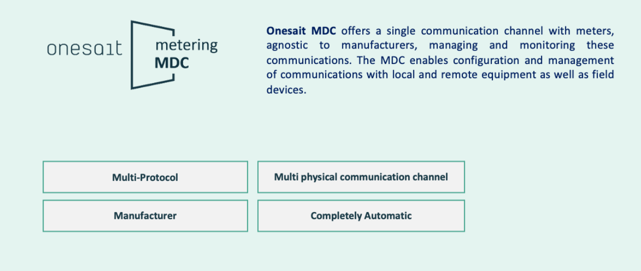
Tecnología de medición de servicios públicos de Onesait

OUM (Onesait Utilities Metering) es el producto de medición para utilities desarrollado por Minsait (energía, agua y gas) que permite la captura, almacenamiento, explotación y exportación de datos de medición en el ámbito de las Smart Grids. Este sistema de software permite a la compañía una interacción bidireccional con los dispositivos de medición en campo, tanto contadores como concentradores, a través de cualquier red de comunicaciones WAN.

Una de las características diferenciadoras de OUM es que se trata de un sistema multifabricante, es decir, el sistema permite la comunicación bidireccional con cualquier tipo de dispositivo de medición, independientemente del fabricante y del protocolo requerido. Una de las bases de la solución OUM es la automatización de los procesos, una vez configurado el sistema es capaz de operar de forma automatizada en función de lo establecido por el usuario del sistema. Todas las tareas realizadas por OUM pueden ser ejecutadas de forma manual o automática para reducir la intervención del usuario.

Por otro lado, OUM también tiene la capacidad de actuar como maestro de inventario de los equipos de medición desplegados en el campo. Además, el sistema permite la sincronización de inventarios con sistemas de terceros. Por lo tanto, dentro del mapa de aplicaciones de Smart Metering de cualquier negocio energético, OUM puede ser considerado como el maestro de datos de medición, ya que centraliza la recepción y el procesamiento de todos los datos de medición del negocio que representa. Este sistema ha sido desarrollado para ser utilizado por cualquier tipo de empresa de servicios públicos que requiera una gestión y tratamiento centralizado y eficiente de los datos de medición [14].

El MDC de Onesait Utilities Metering se encarga de gestionar el proceso de captura de datos, a través de un único canal de comunicación con los equipos de medida, gestionando y monitorizando estas comunicaciones. El MDC permite la configuración y gestión de las comunicaciones con los equipos locales y remotos, así como con los dispositivos de campo. En la página web Figure 16 muestra las funcionalidades del MDC descritas anteriormente.

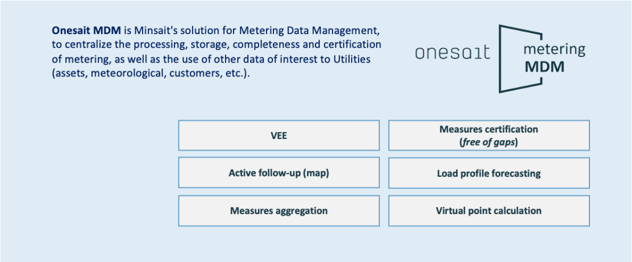


onesait metering MDC

Onesait MDC offers a single communication channel with meters, agnostic to manufacturers, managing and monitoring these communications. The MDC enables configuration and management of communications with local and remote equipment as well as field devices.

Figura 3: Descripción del MDC [14].

Onesait Utilities Metering MDM ha sido concebido para gestionar y procesar los datos recogidos por el sistema para centralizar el tratamiento, almacenamiento, integridad y certificación de las mediciones, así como el uso de otros datos de interés para las Utilities.



Onesait MDM is Minsait's solution for Metering Data Management, to centralize the processing, storage, completeness and certification of metering, as well as the use of other data of interest to Utilities (assets, meteorological, customers, etc.).

onesait metering MDM

Figura 4: Descripción del MDM [14].

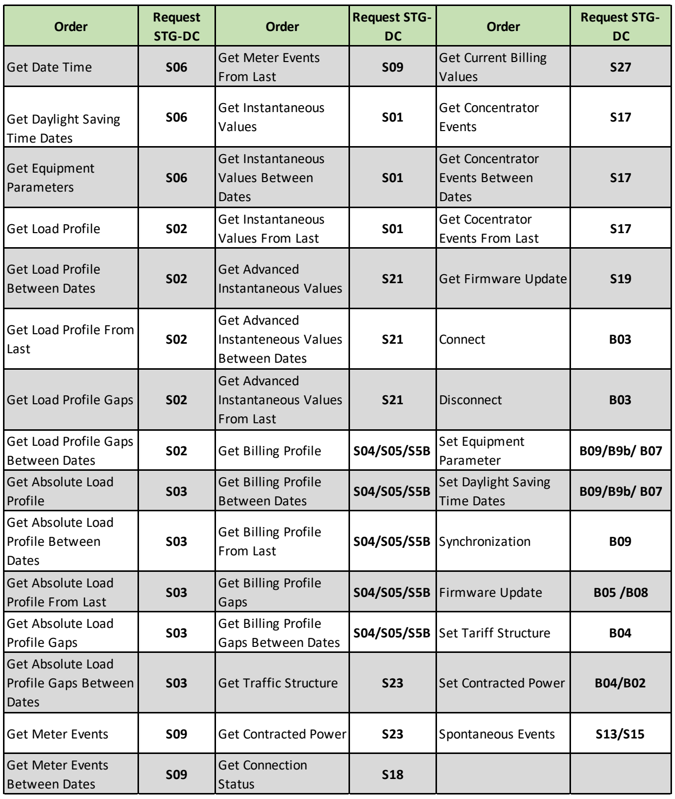
El módulo de gestión de las mediciones representa la parte más representativa de lo que se considera un módulo o funcionalidad MDM, que incluye VEE (Validación, Edición y Estimación). Esta funcionalidad incluye la configuración del ciclo de medición de cada punto o grupo de puntos de medición.

Desarrollo

En este capítulo se definen las diferentes peticiones a los equipos que se registrarán en el sistema, tanto para la telemedida como para la telegestión. También se presentarán los diagramas de flujo de comunicación que conforman el objetivo principal del proyecto.

Como se ha mencionado anteriormente, cada una de estas peticiones tendrá como objetivo solicitar o enviar información al contador o contadores correspondientes. Cada petición tendrá un conjunto de parámetros que deberán ser introducidos a través de una ejecución manual o de un planificador. En primer lugar, en una primera subsección, se proporcionará un mapeo de las peticiones cruzadas con el protocolo STG-DC, que sirve como lista de todas estas peticiones que se definirán inmediatamente después.

La tabla 2 representa el mapeo de las órdenes a integrar en la herramienta con la correspondiente referencia dada en el protocolo STG-DC.



Order	Request STG-DC	Order	Request STG-DC	Order	Request STG-DC
Get Date Time	S06	Get Meter Events From Last	S09	Get Current Billing Values	S27
Get Daylight Saving Time Dates	S06	Get Instantaneous Values	S01	Get Concentrator Events	S17
Get Equipment Parameters	S06	Get Instantaneous Values Between Dates	S01	Get Concentrator Events Between Dates	S17
Get Load Profile	S02	Get Instantaneous Values From Last	S01	Get Cocentrator Events From Last	S17
Get Load Profile Between Dates	S02	Get Advanced Instantaneous Values	S21	Get Firmware Update	S19
Get Load Profile From Last	S02	Get Advanced Instanteneous Values Between Dates	S21	Connect	B03
Get Load Profile Gaps	S02	Get Advanced Instantaneous Values From Last	S21	Disconnect	B03
Get Load Profile Gaps Between Dates	S02	Get Billing Profile	S04/S05/S5B	Set Equipment Parameter	B09/B9b/ B07
Get Absolute Load Profile	S03	Get Billing Profile Between Dates	S04/S05/S5B	Set Daylight Saving Time Dates	B09/B9b/ B07
Get Absolute Load Profile Between Dates	S03	Get Billing Profile From Last	S04/S05/S5B	Synchronization	B09
Get Absolute Load Profile From Last	S03	Get Billing Profile Gaps	S04/S05/S5B	Firmware Update	B05 /B08
Get Absolute Load Profile Gaps	S03	Get Billing Profile Gaps Between Dates	S04/S05/S5B	Set Tariff Structure	B04
Get Absolute Load Profile Gaps Between Dates	S03	Get Traffic Structure	S23	Set Contracted Power	B04/B02
Get Meter Events	S09	Get Contracted Power	S23	Spontaneous Events	S13/S15
Get Meter Events Between Dates	S09	Get Connection Status	S18		

Tabla 2: Asignación de pedidos

La definición de las órdenes se encuentra en el documento largo.

Una vez presentadas y descritas las órdenes a integrar en el sistema para el proyecto EDP, el siguiente paso es desglosar el proceso de comunicaciones que tiene lugar entre los diferentes equipos y sistemas implicados. Para representar este proceso y las interacciones entre las diferentes entidades existentes, se utilizarán diagramas de flujo con el fin de describir todo el proceso de una forma muy visual que facilite al lector la correcta comprensión y seguimiento del proceso de comunicaciones.

Para lograr el objetivo de representar todas las órdenes que aparecen en el mapeo de la Tabla 2, es necesario agrupar las órdenes que aparecen en la columna *"Solicitud STG-DC" según* las características de su proceso de comunicación:

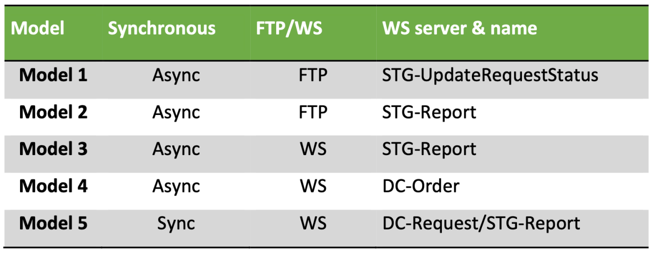
En primer lugar, es necesario distinguir entre las tareas *"Pull"* y *"Push"*.

- **Pull: Las** tareas pull son tareas que se originan a partir de una petición específica del GTS y suelen ser ejecuciones manuales solicitadas por el usuario.

- **Push:** Las tareas push son tareas o acciones que se realizan por iniciativa del DC, o por tareas que han sido programadas en el dispositivo.

La diferencia estructural entre estas dos tipologías es sencilla. Si la tarea es Pull, el diagrama de flujo correspondiente se lee en su totalidad, pero si la tarea es Push, el diagrama de flujo sería el mismo, saltando el primer nivel. Este primer nivel consiste en la primera comunicación entre el STG y el DC y consiste en la creación del registro *"Telemetry_user_request"*. Lógicamente, si la tarea es una tarea push, y por lo tanto no hay petición del STG, este registro no existe y se pasa directamente a la fase en la que el Job correspondiente interviene y actualiza los registros.

Se pueden distinguir cinco grupos diferentes, que se utilizarán para agrupar los organigramas. El cuadro 3 muestra estos grupos y asigna una referencia a cada uno, que es la que utilizaremos en los siguientes subapartados.



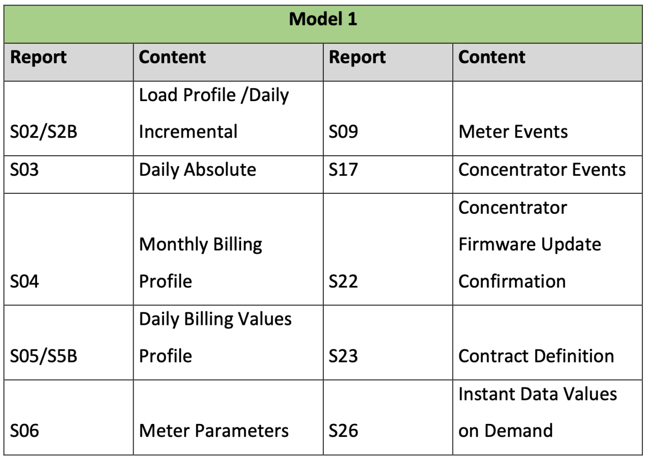
Model	Synchronous	FTP/WS	WS server & name
Model 1	Async	FTP	STG-UpdateRequestStatus
Model 2	Async	FTP	STG-Report
Model 3	Async	WS	STG-Report
Model 4	Async	WS	DC-Order
Model 5	Sync	WS	DC-Request/STG-Report

Tabla 3: Clasificación de las solicitudes

A modo de ejemplo, en este documento se describirán con detalle el modelo 1 y el modelo 4. Sin embargo, la descripción de los 5 modelos representados en la Tabla 2 puede encontrarse en el documento largo.

Modelo 1

El modelo 1 es el conjunto de tareas asíncronas, que utilizan el protocolo FTP para el envío y la recepción de información y utilizan el estado de solicitud de actualización como WS de comunicación.
El cuadro 4 contiene una lista de los informes que se han elaborado en relación con este modelo:



Model 1			
Report	Content	Report	Content
S02/S2B	Load Profile /Daily Incremental	S09	Meter Events
S03	Daily Absolute	S17	Concentrator Events
S04	Monthly Billing Profile	S22	Concentrator Firmware Update Confirmation
S05/S5B	Daily Billing Values Profile	S23	Contract Definition
S06	Meter Parameters	S26	Instant Data Values on Demand

Cuadro 3: Informes del modelo 1

Este modelo utiliza en su protocolo de comunicación el WS STG-UpdateRequestStatus. Este WebService está diseñado para informar al STG sobre el estado de una solicitud.
Debe ser llamado por el DC después de que una petición asíncrona del STG haya finalizado en el DC, para informar al STG de por qué ha finalizado la petición (finalización normal, tiempo de espera...) si la información aún no se ha pasado al STG (por ejemplo, si el informe se envió utilizando Report WS). En el caso de los informes periódicos (tareas programadas) entregados por el DC se debe utilizar este servicio web para notificar la carga del informe y poner el IdPet a 0.

En este tipo de peticiones, paralelamente a la URS, interviene el WS UpdateMeterStatus. Este WebService está diseñado para informar al GTS sobre el estado de una solicitud.
Debe ser llamado por el DC después de que una petición asíncrona del STG haya finalizado en el DC, para informar al STG de por qué ha finalizado la petición (finalización normal, tiempo de espera...) si la información aún no se ha pasado al STG (por ejemplo, si el informe se envió utilizando Report WS). En el caso de los informes periódicos (tareas programadas) entregados por el DC se debe utilizar este servicio web para notificar la carga del informe y poner el IdPet a 0.

A continuación, se ha decidido utilizar un ejemplo de la lista anterior para describir el proceso de comunicación de este modelo. El ejemplo seleccionado es el de los informes S09 y S17, Eventos de Contadores y Concentradores, ya que además tiene la particularidad de contener otro WS *"STG-Report Query"*.

Hay que tener en cuenta que el resto de los informes enumerados en la Tabla 4 están representados en sus correspondientes diagramas de flujo en el Anexo IV. Cuando se inicia esta tarea, recoge la información de los contadores si es S09 o de los DC si es S17 y la envía al STG.
La ejecución de esta tarea puede incluir como atributos los Grupos y los códigos de evento a entregar. El plazo por defecto viene dado por la periodicidad de la tarea. Estas peticiones

son las encargadas de ejecutar las órdenes de tipo *"Obtener xxx Eventos"* de la herramienta OUM.

La descripción del diagrama de flujo de estas órdenes se hará en dos secciones:

1. **Ejecución de la solicitud y recepción del FTP.**

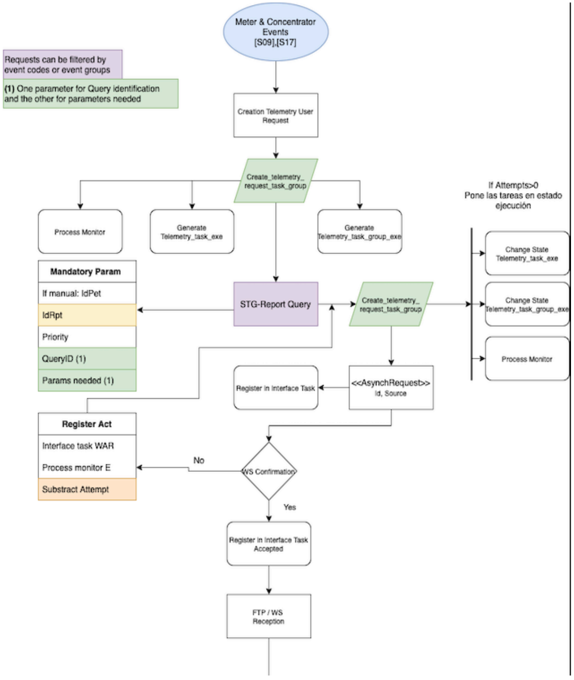

Meter & Concentrator Events [S09],[S17]
Requests can be filtered by event codes or event groups
(1) One parameter for Query identification and the other for parameters needed
Creation Telemetry User Request
Create_telemetry_ request_task_group
If Attempts>0
Pone las tareas en estado ejecución
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Mandatory Param
If manual: IdPet
IdRpt
Priority
QueryID (1)
Params needed (1)
STG-Report Query
Create_telemetry_ request_task_group
Register in Interface Task
<<AsynchRequest>> Id, Source
Register Act
Interface task WAR
Process monitor E
Substract Attempt
No
WS Confirmation
Yes
Register in Interface Task Accepted
FTP / WS Reception

Figura 5: Diagrama de flujo de contadores y eventos de CC Parte 1

Cuando el usuario ejecuta la solicitud, lo primero que se crea es el registro *"Create_Telemetry_User_Request",* interviene el Job *"Create_telemetry_request _task"* y se generan los siguientes registros en estado READY.

- Monitor de procesos
- Telemetry_Task_execute
- Telemetría_Grupo_de_tareas_Ejecutar

Una vez generados estos registros, interviene el WS STG-ReportQuery. Este WS asíncrono se utiliza para recuperar eventos en contadores (S09) y concentradores (S17).

Las solicitudes pueden filtrarse por códigos o grupos de eventos. Para ello, este WS incluye parámetros diferentes al anterior: un parámetro para la identificación de la consulta y otro para los parámetros necesarios.

A continuación, una vez que OUM tiene capacidad para procesar la orden, entra el siguiente JOB, que se encarga de confirmar la WS con el concentrador y pretende poner los tres registros indicados en estado de EJECUCIÓN. Si la confirmación del WS no ha llegado, se restará un reintento en los registros, el estado del Monitor de Proceso cambiará a Error y el estado de la Tarea de Interfaz cambiará a Advertencia hasta que llegue la confirmación. Una vez confirmada esta WS, el sistema espera y el concentrador envía el fichero FTP al sistema OUM. Este sistema se almacena en un registro de la herramienta y contiene una referencia a su solicitud para que, cuando se vaya a procesar, se sepa a qué proceso pertenece la generación del fichero.

El siguiente paso será el procesamiento del archivo y la interacción del sistema con el WS URS y UMS.

2. Tramitación de expedientes y resolución de comunicaciones

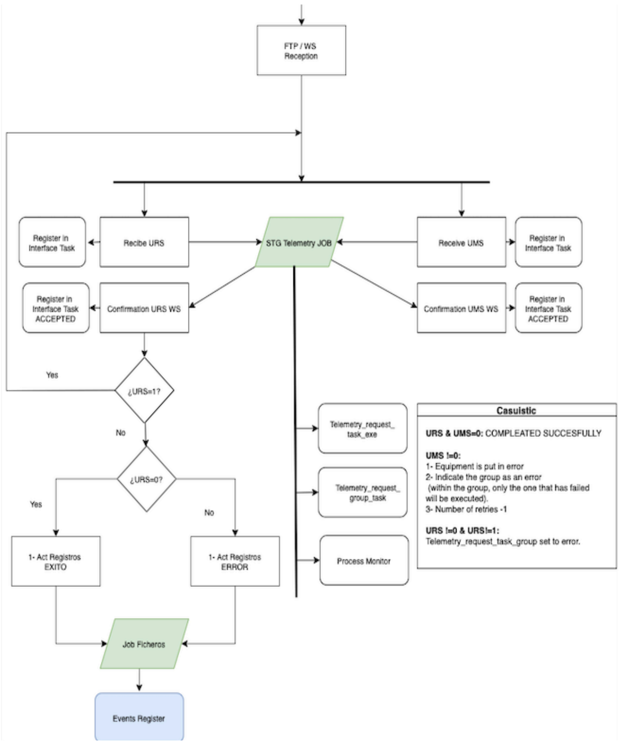

FTP / WS Reception
Register in Interface Task
Recibe URS
STG Telemetry JOB
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
Yes
¿URS=1?
No
¿URS=0?
Yes
No
1- Act Registros EXITO
1- Act Registros ERROR
Job Ficheros
Events Register
Telemetry_request_ task_exe
Telemetry_request_ group_task
Process Monitor
Casuistic
URS & UMS=0: COMPLEATED SUCCESFULLY
UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error (within the group, only the one that has failed will be executed).
3- Number of retries -1
URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

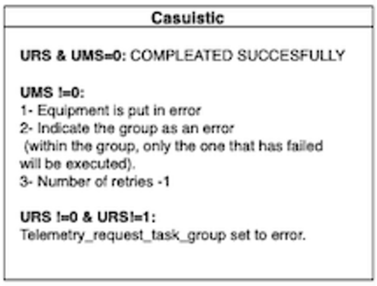
Figura 6: Diagrama de flujo de contadores y eventos de CC Parte 2

Una vez recibido el archivo FTP, *entra en juego* el *JOB STG-Telemetry.* Este trabajo busca las tareas pendientes del protocolo PRIME, las agrupa, crea las correspondientes telemetry_requests e interactúa con el AsynchRequest para procesar el mensaje.

En este caso concreto, este puesto de trabajo se encargará de interpretar y responder a los registros entrantes por medio de WS (URS y UMS) como se muestra en el diagrama de flujo representado en la Figura 29.

Como se puede ver, cada vez que el concentrador lo notifica con un URS y su referencia de petición, o con un UMS y su referencia de petición, interviene el *STG-Telemetry JOB,* que da lugar al *STG-ServiceJob*. Este job, según la llegada de WS recibidos (URS, UMS) actualiza el estado de las peticiones en las tablas, y procesa los ficheros con el nombre especificado en cada UMS del proceso enviado por el concentrador.

A partir de aquí se interpreta el contenido del URS, que es el WebService que gobierna el sistema y que indica cuando la comunicación ha finalizado. Cuando URS=1 significa que la comunicación sigue en curso, mientras que cuando URS=0 significa que la comunicación ha finalizado con éxito, el resto de valores posibles indican mensajes de error o advertencia que se especifican en el Anexo II. Estos valores se cruzan con los valores devueltos por el UMS y la casuística sigue el proceso indicado en la tabla del diagrama de flujo de la Figura 30.



Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed
will be executed).
3- Number of retries -1

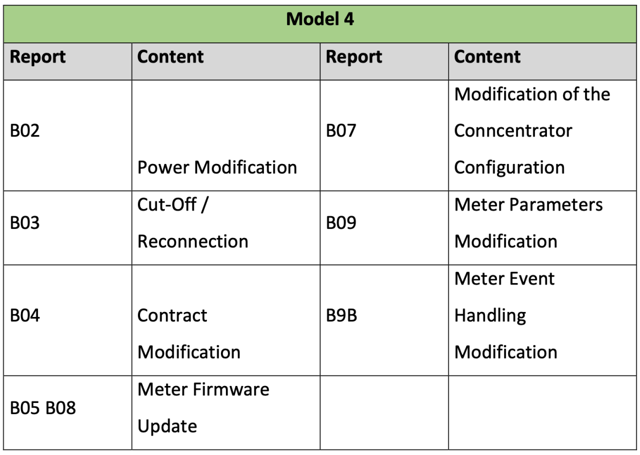
URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figura 7: Casuística URS y UMS

Finalmente, los registros se actualizan con el estado correspondiente, y los Ficheros JOB almacenan la información en el registro apropiado.

Modelo 4

El modelo 4 es el conjunto de tareas asíncronas, que utilizan el WS para enviar y recibir información y que utilizan el DC-Order como WS de comunicación.
El cuadro 4 contiene una lista de los infomes que se han construido en relación con este modelo:



Model 4			
Report	Content	Report	Content
B02	Power Modification	B07	Modification of the Conncentrator Configuration
B03	Cut-Off / Reconnection	B09	Meter Parameters Modification
B04	Contract Modification	B9B	Meter Event Handling Modification
B05 B08	Meter Firmware Update		

Cuadro 4: Informes del modelo 4

Este modelo se encarga de gestionar las órdenes de actuación directa del STG al concentrador, es decir, las órdenes marcadas con Bxx. Al igual que las solicitudes de informe, estas órdenes utilizan mensajes en formato XML enviados a través de un WS en forma de carga útil.

Este modelo utiliza el tipo de orden DC como confirmación WS. Se trata de un protocolo en el que se prepara una acción en un equipo de medida concreto, se extrae la información necesaria mediante peticiones de informe y, finalmente, se ejecuta la acción.

Una vez ejecutada con éxito la orden DC, el concentrador debe verificar explícitamente la orden solicitada a cada contador mediante una petición del STG.

El envío de una Solicitud de Orden desde el GTS a un contador puede dar lugar a la ejecución de diferentes fases o pasos (actualización del archivo de descarga, preinformación, fase de ejecución, postinformación, etc.). Estas fases se ejecutarán secuencialmente en el CC, de forma que no se pueda iniciar una fase hasta que se haya completado la anterior.

En caso de fallo en la ejecución de una fase, la secuencia de ejecución se detendrá y el estado de la orden se comunicará al GTS a través de UMS y URS con la categoría y el código de error adecuados.

Para describir el Modelo 4, utilizaremos el diagrama de flujo correspondiente al comando de *Actualización de Firmware,* contenido en las peticiones B05 y B08.

Es importante tener en cuenta que durante la ejecución de este comando, el sistema solicita al GTS que genere los informes S19 o S22 (dependiendo de si la actualización del firmware se está realizando en un contador o en el concentrador). El proceso de recogida y ejecución de estos informes se realiza según el modelo 3.

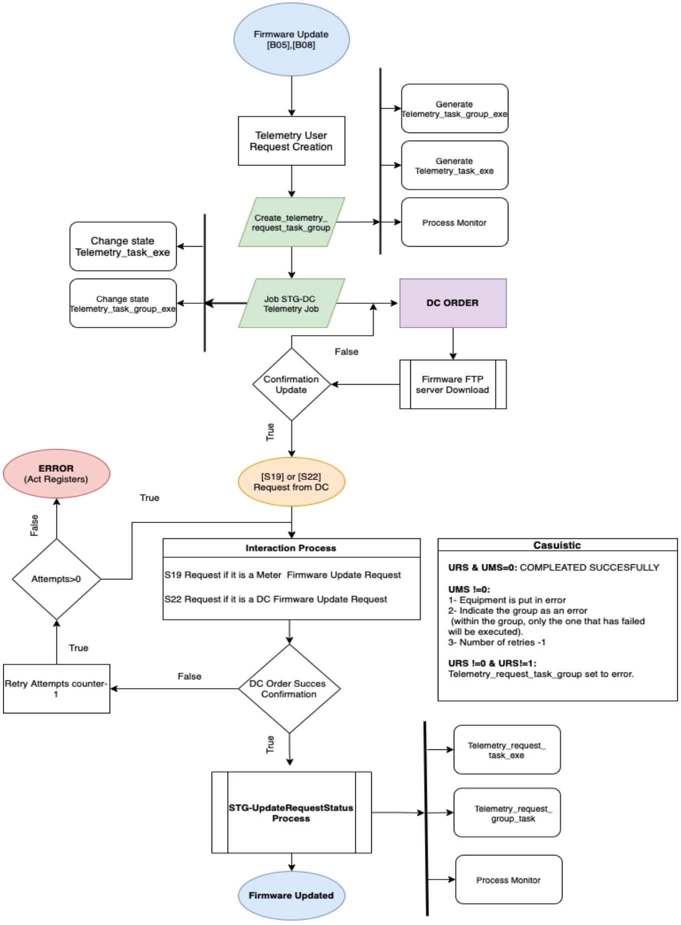

Firmware Update
[B05],[B08]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_request_task_group
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC ORDER
False
Confirmation Update
Firmware FTP server Download
True
ERROR
(Act Registers)
True
[S19] or [S22] Request from DC
False
Interaction Process
S19 Request if it is a Meter Firmware Update Request
S22 Request if it is a DC Firmware Update Request
Casuistic
URS & UMS=0: COMPLEATED SUCCESFULLY
UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1
URS !=0 & URS!=1:
Telemetry_request_task_group set to error.
Attempts>0
True
Retry Attempts counter-1
False
DC Order Succes Confirmation
True
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
STG-UpdateRequestStatus Process
Firmware Updated

Figura 8: Diagrama de flujo de la actualización del firmware

Como en todos los casos anteriores, el primer paso para ejecutar el comando es generar los tres registros principales del sistema OUM a través del script *Telemetry_User_Request*.
El trabajo *de telemetría STG-DC* se ejecuta exactamente igual que en el caso de los modelos 2 y 3 sólo que esta vez precede a un WS de tipo *DC-Order*. Este WS, una vez confirmado correctamente, solicita un informe S19 o S22 para guardar un registro anterior (una copia de seguridad) por si algo va mal en el proceso de ejecución del trabajo.

Una vez finalizado el proceso de generación de los informes anteriores, el sistema debe confirmar los WS. Para ello, dispone de un número limitado de reintentos y lleva al sistema al estado de error en caso de que se completen.

Por último, interviene el *STG-Service Job* y mediante el procesamiento de URS y UMS siguiendo exactamente el mismo proceso descrito en el Modelo 1, se actualizan los estados de los registros y se finaliza la comunicación realizando la acción de actualización en el equipo solicitado.

CASO DE USO

Al dibujar los diagramas de flujo necesarios para caracterizar la integración de OUM con la especificación STG-DC, se genera un patrón al ejecutar los comandos. Este patrón consiste en una serie de registros de la base de datos que permiten la trazabilidad del mensaje para comunicar correctamente el sistema de cabecera con el MDM. Contienen la información necesaria para identificar la petición, el destino y el formato de la misma para su correcto tratamiento y para que el usuario pueda monitorizar y actuar en consecuencia.

Se puede ver en los diagramas del Anexo X cómo aparecen tres registros en el primer nivel y cómo se actualizan en los siguientes. Estos tres registros son:

- *Monitor de procesos*
- *Grupo de Ejecución de Tareas de Solicitudes de Telemetría*
- *Telemetría_Solicitud_Ejecución_de_tareas*

Sin embargo, aunque no aparece directamente, intrínsecamente antes de generar estos tres registros principales, cuando el MDM ejecuta una tarea manual o programada, genera otro registro:

- *Telemetry_User_Request:* Este registro es el encargado de crear las peticiones en la base de datos. Estas peticiones se crean en estado "Preparado" para que posteriormente el Job correspondiente pueda actualizar los registros cambiando su estado para cada etapa y caso que presente el protocolo de comunicación.

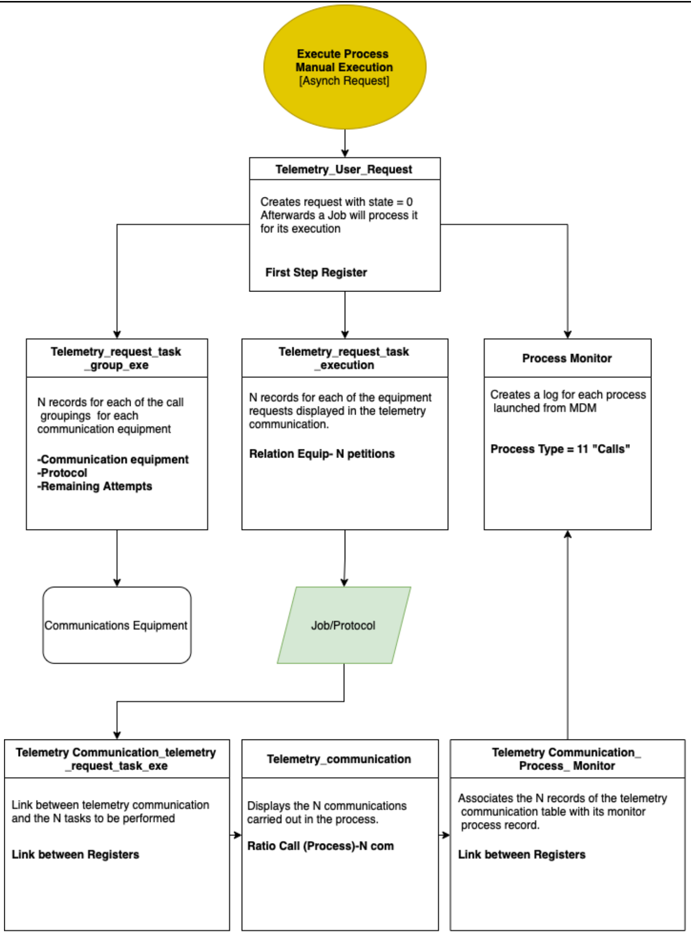

Execute Process
Manual Execution
[Asynch Request]
Telemetry_User_Request
Creates request with state = 0
Afterwards a Job will process it
for its execution
First Step Register
Telemetry_request_task
_group_exe
N records for each of the call
groupings for each
communication equipment
-Communication equipment
-Protocol
-Remaining Attempts
Telemetry_request_task
_execution
N records for each of the equipment
requests displayed in the telemetry
communication.
Relation Equip- N petitions
Process Monitor
Creates a log for each process
launched from MDM
Process Type = 11 "Calls"
Communications Equipment
Job/Protocol
Telemetry Communication_telemetry
_request_task_exe
Link between telemetry communication
and the N tasks to be performed
Link between Registers
Telemetry_communication
Displays the N communications
carried out in the process.
Ratio Call (Process)-N com
Telemetry Communication_
Process_ Monitor
Associates the N records of the telemetry
communication table with its monitor
process record.
Link between Registers

Figura 9: Diagrama de flujo de las interacciones de los registros

Conocer estos procesos a nivel de base de datos es fundamental para entender la lógica de programación y la creación de los diagramas de flujo propuestos en el proyecto. El esquema genérico de interacción que siguen estos registros es el representado en la Figura 9.

Este estudio de caso se presenta en el documento largo correspondiente a este proyecto.

ONESAIT METERING FOR EDP

Author: Benavente Martínez, Eduardo.
Supervisor: Diaz Rodriguez, Miguel.
Collaborating Entity: Minsait (Indra IT Solutions)

ABSTRACT

The fight against climate change is one of the most important challenges that humanity must face, and the commitment to a decarbonised economy based on renewable energies in a new energy mix is a task for all of us. This is how the Spanish National Integrated Energy and Climate Plan (PNIEC) was born in Spain, which has the minimum objective of reaching 32% renewable energy by 2030. This objective is achievable in a scenario of high decarbonisation and electrification of the economy, using decarbonised fuels in those niches that are difficult to electrify [1]. The challenge of extending these networks to millions of endpoints, potentially geographically dispersed over large service areas, is inherent to the Smart Grid [1].

Faced with this new paradigm, utilities companies are facing a series of digital transformation challenges marked by the migration of the energy supply chain to a new distributed model. Within smart grid development, Smart Metering is one of the fundamental pillars, driven by sustainability and user satisfaction. Minsait has developed Onesait Utilities Metering technology, a smart metering solution designed for utilities that enables a competitive advantage in an energy market affected by global trends, allowing the Utility to collect, certify and export information data from different information sources [15]. EDP (Energías de Portugal), one of the largest electricity groups in Europe, has requested Minsait's services to integrate the OUM solution in its utilities in Spain. Three institutions will be integrated under a single manager (EDP-Spain, Viesgo and Begasa). This project will focus on the analysis and functional development of OUM integration under PRIME's STG-DC specification in EDP.

When integrating different institutions into the EDP Spain infrastructure, a problem arises for the company when it comes to exploiting the information in an agile, simple and transversal way with the aim of offering the new services introduced in the electricity distribution sector. This problem consists of the interoperability and centralisation of the different equipment that makes up the system. In other words, each of the companies that have been integrated into the EDP portfolio used different technologies, different communication protocols, different types of devices and different remote management solutions.

The acquired part of E-Networks has approximately 700,000 Prime counters that communicate with the 1.3.6 protocol. It also has 7,000 Prime 1.3.6 concentrators that communicate with the AMI using the STG-DC standard in its 3.4 version. In addition, the system includes various types of advanced monitoring equipment (using the STG-DC standard) as well as equipment such as routers and fibre optic interfaces.

On the Begasa + Viesgo side there are 700,000 meters that use Telegestore as a communication protocol with the concentrators and 12,000 concentrators that currently communicate with the AMM via a private standard owned by Endesa.

The aim of this project is to collaborate in the integration of these systems in the Minsait OUM multiprotocol tool, with the aim of centralising all communications to enable a single remote management of the entire fleet of equipment so that the user is able to monitor, improve and automate all communications processes that occur with each of the measuring devices that make up the network.

State of Art

The electricity metering equipment used worldwide is the well-known electricity meter, which consists of three main elements, namely the metering system, the memory element and the information device. In this sense, the electricity meter performs the function of the network interface with the user, it is the front-end of the network.
This metering equipment can be classified according to its technological characteristics (whether electromechanical or electronic), by its single-phase or three-phase functionality, energy, and operational characteristics as a programmable recorder type device that allows remote management. These meters have evolved significantly in the electricity market as new technologies have been developed.

These devices are based on AMI (Advanced Metering Infrastructure) technology, these devices allow the reading of accumulated energy or instantaneous power, they allow demand records or programming of load intervals previously defined with each user. The Smart Meter concept was born from these new incipient technologies. These devices provide information and control of the quality and programming parameters of the service via the management centre, together with telematic updating of the metering software. Based on these reasons and those described above, the interest in the deployment and implementation of this technology is justified, as well as the importance of all the studies that encompass this world and among which this project is included.

AMI Communications Architecture

The acronym AMI, as mentioned above, stands for Advanced Metering Infrastructure and represents the heart and soul of this technology. To begin to describe this standard architecture for advanced metering-based communications systems, it is necessary to discern at a high level what are the main elements that comprise it [17]:

- **Smart Meter:** This is an electronic smart meter capable of recording electricity consumption at configurable time intervals and communicating the information from this recording to the control unit. It is therefore an instrument that fulfils the function of a sensor and an interaction point simultaneously, i.e. a smart node.

- **Data Concentrator (DCU):** It is the backbone of the Advanced Metering Infrastructure (AMI) that supports data acquisition, transfer of energy data to the central database and automated reading of electronic energy meters. DCUs are mainly used in low-voltage electric power systems.

- **Information systems:** These information systems act as a link between data acquisition and further processing. They are designed as databases (both relational and non-relational) and have remote management capabilities. These information systems are characterised by having a complete view of the infrastructure.

These systems are usually heterogeneous communication infrastructures, involving different technologies depending on the specific requirements of each network segment. In Spain, communication between meters and data concentrators is based on narrowband PLC technologies.

Power Line Communications (PLC) is a telecommunications technology that relies on the use of the electrical power distribution infrastructure (typically LV or MV) for high-speed transmission of signals and data. It provides broadband data communications over conductors that are already in use for the transmission of electrical power using a modular signal. This PLC technology is seen as a reliable communication medium for applications such as the Internet of Things (IoT) and Smart Grids [3].

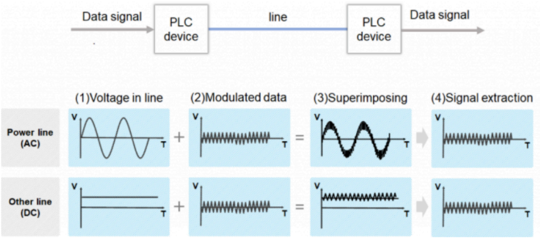


Figure 1: Basic PLC working principle [3].

PRIME for AMI Application

PRIME (PoweRline Intelligent Metering Evolution) is an open, free, non-proprietary standard for a globally established OFDM PLC technology for advanced metering, network control and asset monitoring applications, and has fulfilled the goal of establishing a set of open PLC standards on an international basis. There are more than 20 million PRIME meters installed in more than 15 countries around the world [5]. Since the beginning of electricity deregulation and market-based pricing around the world, utilities have been looking for a means to match consumption with generation. Traditional electricity and gas meters only measure total consumption and therefore do not provide information on when energy has been consumed at each site. PRIME AMI offers a way to measure this site-specific information, allowing different prices to be set for consumption based on time of day and season [5]. One of the advantages of implementing the PRIME protocol is that it ensures interoperability between different equipment and systems, is not subject to any intellectual property, and is designed to achieve a low implementation cost and high performance, thus accelerating the entry of new operators.

The physical layer, MAC layer and convergence layer of version 1.3.6 of the specification have been considered standard by ITU-T since 2012. Version 1.4 extends the frequency spectrum to allow operation in the North American and Asian markets and includes some enhancements aimed at improving robustness at the physical and MAC level [7, 8]. At the MAC level, a sub-network has a tree-like structure, where two main types of nodes can be distinguished:

- **Base Node (BN):** The base node is the master node that manages a PRIME subnetwork. There is only one BN per subnetwork, and it is responsible for managing resources and connections, it acts as the master node and is at the root of the tree structure, it is the most important element. In AMI terminology, the base node is the data concentrator [7, 14].
- **Service Node (SN):** Service nodes are the core element of the subnetwork. They can be seen as leaves or branches in the tree structure and are initially in a disconnected state, their functions being to maintain connectivity to the subnetwork for the different application layers and to switch information from other nodes. Smart meters must register in the sub-network to become service nodes that depend on their corresponding BN. In AMI terminology, service nodes are smart meters. [7, 14].

Initially the sub-network is composed only by the base node and the rest of the nodes that want to be part of it have to perform a registration procedure.

- **Disconnected:** This is the initial state of any service node before it registers into the PRIME subnetwork. While in this state, the smart meter cannot communicate its own data or switch data from other SNs and its only goal is to search for a subnetwork that it can reach and register on it.

- **Terminal:** This state allows the SN to establish new connections and transmit data, but not to switch data from other SNs.
- **Switch:** This state is the most complex one and allows the SN to establish new connections, transmit its own data and switch data from other SNs.

Finally, a process of registration and deregistration is carried out.

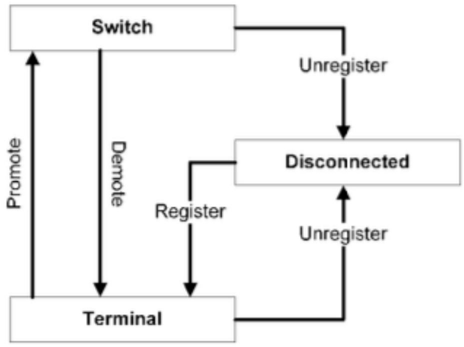


Figure 2: Functional states of a Service Node [11]

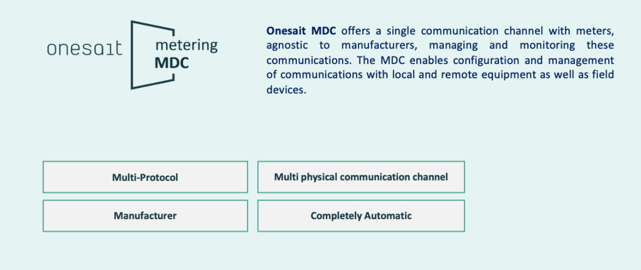
Onesait Utilities Metering Technology

OUM (Onesait Utilities Metering) is the metering product for utilities developed by Minsait (energy, water and gas) that enables the capture, storage, exploitation and export of metering data in the field of Smart Grids. This software system allows the company a bidirectional interaction with field metering devices, both meters and concentrators, through any WAN communications network.

One of the differentiating features of OUM is that it is a multi-manufacturer system, i.e. the system allows bidirectional communication with any type of measuring device, regardless of the manufacturer and protocol required. One of the bases of the OUM solution is the automation of processes, once the system has been configured it is able to operate in an automated way depending on what is set by the user of the system. All tasks performed by OUM can be executed manually or automatically in order to reduce user intervention.

On the other hand, OUM also has the capability to act as an inventory master of the measuring equipment deployed in the field. Additionally, the system enables the synchronisation of inventories with third-party systems. Therefore, within the Smart Metering application map of any energy business, OUM can be considered as the master of metering data, as it centralises the reception and processing of all the metering data of the business it represents. This system has been developed for use by any type of utility that requires centralised and efficient management and processing of metering data [14].

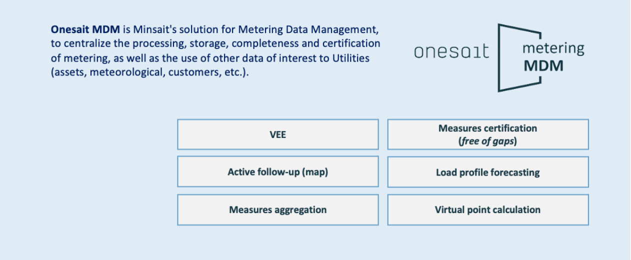
Onesait Utilities Metering MDC is responsible for managing the data capture process, through a single communication channel with the metering equipment, managing and monitoring these communications. The MDC enables the configuration and management of communications with local and remote equipment as well as field devices. The Figure 16 shows the MDC functionalities described above.



Onesait MDC offers a single communication channel with meters, agnostic to manufacturers, managing and monitoring these communications. The MDC enables configuration and management of communications with local and remote equipment as well as field devices.

Figure 3: MDC description [14].

Onesait Utilities Metering MDM has been conceived to manage and process the data collected by the system to centralise the processing, storage, completeness, and certification of measurements, as well as the use of other data of interest to Utilities.



Onesait MDM is Minsait's solution for Metering Data Management, to centralize the processing, storage, completeness and certification of metering, as well as the use of other data of interest to Utilities (assets, meteorological, customers, etc.).

Figure 4: MDM description [14].

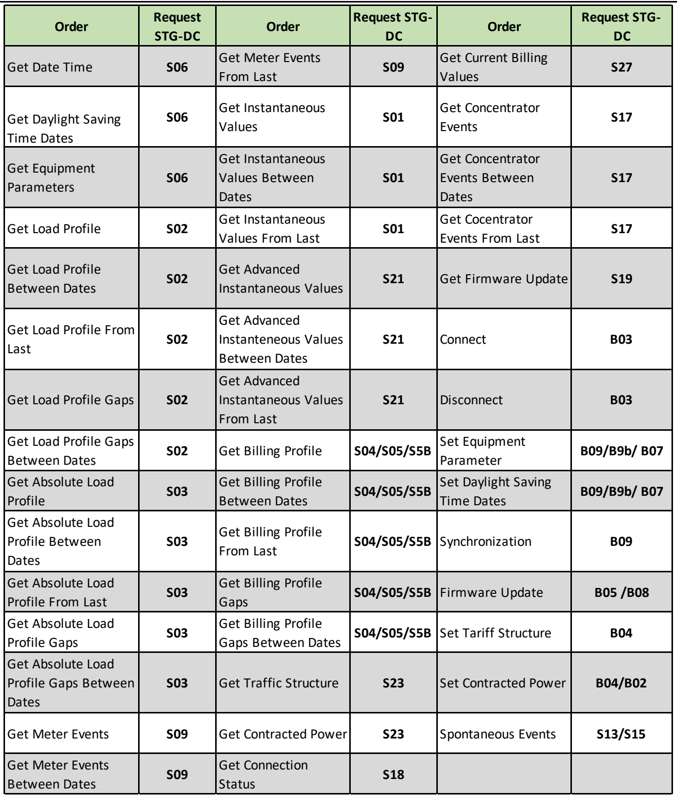
The measurement management module represents the most representative part of what is considered an MDM module or functionality, which includes VEE (Validation, Editing and Estimation). This functionality includes the configuration of the measurement cycle of each measurement point or group of points.

Development

This chapter defines the different requests to the equipment that will be registered in the system, both for remote metering and remote management. The communication flow diagrams that make up the main objective of the project will also be presented.

As mentioned above, each of these requests will have the objective of requesting or sending information to the corresponding meter(s). Each request will have a set of parameters that must be entered through a manual execution or a scheduler. First of all, in a first subsection, a mapping of the cross requests with the STG-DC protocol will be provided, which serves as a list of all these requests that will be defined immediately afterwards.

Table 2 represents the mapping of the orders to be integrated in the tool with the corresponding reference given in the STG-DC protocol.



Order	Request STG-DC	Order	Request STG-DC	Order	Request STG-DC
Get Date Time	S06	Get Meter Events From Last	S09	Get Current Billing Values	S27
Get Daylight Saving Time Dates	S06	Get Instantaneous Values	S01	Get Concentrator Events	S17
Get Equipment Parameters	S06	Get Instantaneous Values Between Dates	S01	Get Concentrator Events Between Dates	S17
Get Load Profile	S02	Get Instantaneous Values From Last	S01	Get Cocentrator Events From Last	S17
Get Load Profile Between Dates	S02	Get Advanced Instantaneous Values	S21	Get Firmware Update	S19
Get Load Profile From Last	S02	Get Advanced Instanteneous Values Between Dates	S21	Connect	B03
Get Load Profile Gaps	S02	Get Advanced Instantaneous Values From Last	S21	Disconnect	B03
Get Load Profile Gaps Between Dates	S02	Get Billing Profile	S04/S05/S5B	Set Equipment Parameter	B09/B9b/ B07
Get Absolute Load Profile	S03	Get Billing Profile Between Dates	S04/S05/S5B	Set Daylight Saving Time Dates	B09/B9b/ B07
Get Absolute Load Profile Between Dates	S03	Get Billing Profile From Last	S04/S05/S5B	Synchronization	B09
Get Absolute Load Profile From Last	S03	Get Billing Profile Gaps	S04/S05/S5B	Firmware Update	B05 /B08
Get Absolute Load Profile Gaps	S03	Get Billing Profile Gaps Between Dates	S04/S05/S5B	Set Tariff Structure	B04
Get Absolute Load Profile Gaps Between Dates	S03	Get Traffic Structure	S23	Set Contracted Power	B04/B02
Get Meter Events	S09	Get Contracted Power	S23	Spontaneous Events	S13/S15
Get Meter Events Between Dates	S09	Get Connection Status	S18		

Table 2: Orders Mapping

Definition of orders can be found in the long document.

Once the orders to be integrated in the system for the EDP project have been presented and described, the next step is to break down the communications process that takes place between the different equipment and systems involved. To represent this process and the interactions between the different existing entities, flow diagrams will be used in order to describe the whole process in a very visual way that will make it easier for the reader to understand and follow the communications process correctly.
In order to achieve the objective of representing all orders appearing in the mapping of Table 2, it is necessary to group the orders appearing in the column *"Request STG-DC" according to* their characteristics of their communications process:
First of all, it is necessary to distinguish between *"Pull"* and *"Push"* tasks.

- **Pull:** Pull tasks are tasks that originate from a specific request from the GTS and are usually manual executions requested by the user.

- **Push:** Push tasks are tasks or actions that are performed at the initiative of the DC, or by tasks that have been programmed in the device.

The structural difference between these two typologies is simple. If the task is a Pull task, the corresponding flowchart is read as a whole, but if the task is a Push task, the flowchart would be the same, skipping the first level. This first level consists of the first communication between the STG and the DC and consists of the creation of the record *"Telemetry_user_request.* Logically, if the task is a push task, and therefore there is no request from the STG, this record does not exist and it goes directly to the phase where the relevant Job intervenes and updates the records.

Five different groups can be distinguished, which will be used to group the flowcharts. Table 3 shows these groups and assigns a reference to each one, which is the one we will use in the following subsections.

Model	Synchronous	FTP/WS	WS server & name
Model 1	Async	FTP	STG-UpdateRequestStatus
Model 2	Async	FTP	STG-Report
Model 3	Async	WS	STG-Report
Model 4	Async	WS	DC-Order
Model 5	Sync	WS	DC-Request/STG-Report

Table 3: Requests Clasification

As an example, in this document Model 1 and Model 4 will be described to detail. However, the description of the 5 models represented in Table 2 can be found in the long document.

Model 1

Model 1 is the set of Asynchronous tasks, which use the FTP protocol for sending and receiving information and use Update Request Status as communication WS.
Table 4 contains a list of the reports that have been constructed pertaining to this model:

Model 1			
Report	**Content**	**Report**	**Content**
S02/S2B	Load Profile /Daily Incremental	S09	Meter Events
S03	Daily Absolute	S17	Concentrator Events
S04	Monthly Billing Profile	S22	Concentrator Firmware Update Confirmation
S05/S5B	Daily Billing Values Profile	S23	Contract Definition
S06	Meter Parameters	S26	Instant Data Values on Demand

Table 3: Model 1 Reports

This model uses in its communication protocol the WS STG-UpdateRequestStatus. This WebService is designed to inform the STG about the status of a request.
It has to be called by the DC after an asynchronous request from the STG has finished at the DC, to inform the STG why the request has finished (normal completion, timeout...) if the information has not yet been passed to the STG (e.g. if the report was sent using Report WS). In the case of periodic reports (scheduled tasks) delivered by the DC you must use this web service to notify the report upload and set IdPet to 0.

In this type of requests, parallel to the URS, the WS UpdateMeterStatus intervenes.
This WebService is designed to inform the GTS about the status of a request.
It has to be called by the DC after an asynchronous request from the STG has finished at the DC, to inform the STG why the request has finished (normal completion, timeout...) if the information has not yet been passed to the STG (e.g. if the report was sent using Report WS). In the case of periodic reports (scheduled tasks) delivered by the DC you must use this web service to notify the report upload and set IdPet to 0.

In the following, it has been decided to use an example from the list above in order to describe the communication process of this model. The selected example is the reports S09 and S17, Meter and Concentrator Events as it also has the particularity of containing another WS *"STG-Report Query"*.

It should be noted that the rest of the reports listed in Table 4 are represented in their corresponding flow charts in Annex IV. When this task is started, it collects the information from the counters if S09 or from the DCs if S17 and sends it to the STG.
The execution of this task can include as attributes the Groups and event codes to be delivered. The default deadline is given by the periodicity of the task. These requests are in charge of executing the *"Get xxx Events"* type orders of the OUM tool.
The description of the flowchart of these orders will be done in two sections:

3. **Execution of the request and receipt of the FTP.**

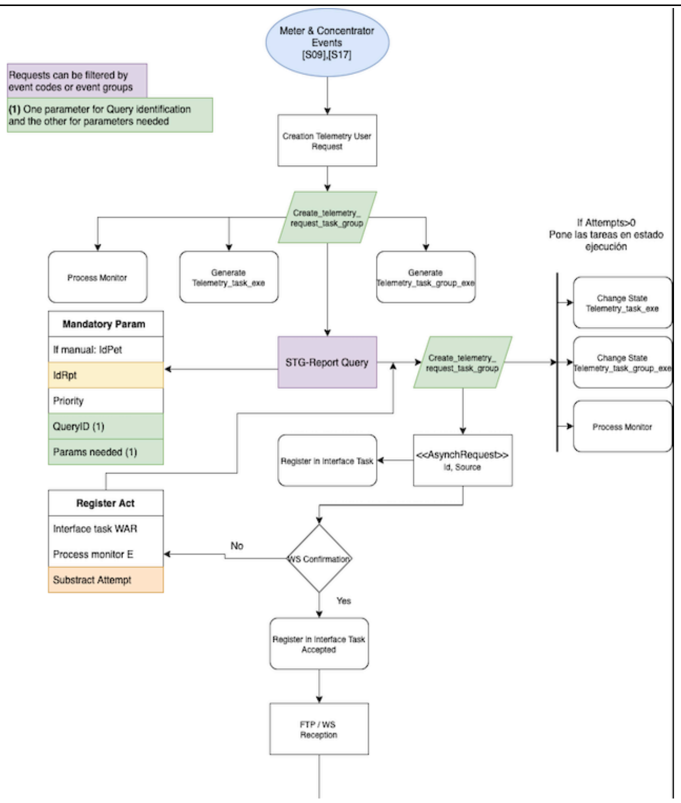

Meter & Concentrator Events [S09],[S17]
Requests can be filtered by event codes or event groups
(1) One parameter for Query identification and the other for parameters needed
Creation Telemetry User Request
Create_telemetry_ request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
If Attempts>0
Pone las tareas en estado ejecución
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Mandatory Param
If manual: IdPet
IdRpt
Priority
QueryID (1)
Params needed (1)
STG-Report Query
Create_telemetry_ request_task_group
Register in Interface Task
<<AsynchRequest>> Id, Source
Register Act
Interface task WAR
Process monitor E
Substract Attempt
No
WS Confirmation
Yes
Register in Interface Task Accepted
FTP / WS Reception

Figure 5: Meter and DC Events FlowChart Part 1

When the user executes the request, the first thing that is created is the record *"Create_Telemetry_User_Request"*, the Job *"Create_telemetry_request _task"* intervenes and the following records are generated in READY status.

- Process Monitor
- Telemetry_Task_execute
- Telemetry_Task_Group_Execute

Once these registers have been generated, the WS STG-ReportQuery intervenes. This asynchronous WS is used to retrieve events in counters (S09) and concentrators (S17).
Requests can be filtered by codes or groups of events. For this purpose, this WS includes different parameters than the previous one: one parameter for the identification of the query and another one for the necessary parameters.

Then, once OUM has the capacity to process the order, the next JOB enters, which is in charge of confirming the WS with the concentrator and aims to put the three indicated registers in the EXECUTION status. If the confirmation of the WS has not arrived, a retry will be subtracted in the registers, the status of the Process Monitor will change to Error and the status of the Interface Task will change to Warning until the confirmation arrives.
Once this WS is confirmed, the system waits and the concentrator sends the FTP file to the OUM system. This system is stored in a record of the tool and contains a reference to its

request so that when it is going to be processed, it is known to which process the generation of the file belongs.

The next step will be the processing of the file and the interaction of the system with the WS URS and UMS.

4. File processing and communication resolution

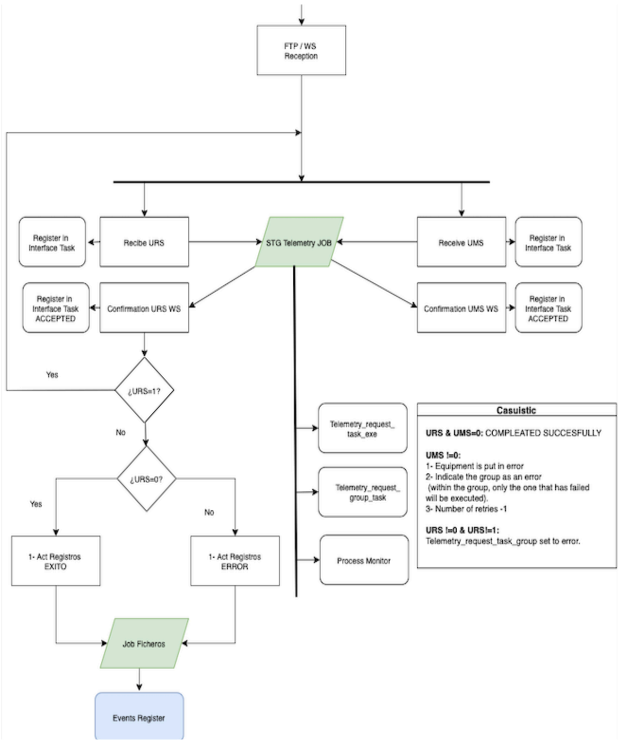
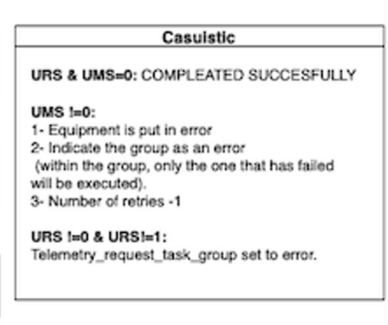

FTP / WS Reception
Register in Interface Task
Recibe URS
STG Telemetry JOB
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
Yes
¿URS=1?
No
¿URS=0?
Yes
No
1- Act Registros EXITO
1- Act Registros ERROR
Job Ficheros
Events Register
Telemetry_request_ task_exe
Telemetry_request_ group_task
Process Monitor
Casuistic
URS & UMS=0: COMPLEATED SUCCESFULLY
UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1
URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figure 6: Meter and DC Events FlowChart Part 2

Once the FTP file is received, the *STG-Telemetry JOB comes into play.* This job looks for pending PRIME protocol tasks, groups them, creates the corresponding telemetry_requests and interacts with the AsynchRequest to process the message.

In this particular case, this job will be in charge of interpreting and responding to incoming records by means of WS (URS and UMS) as shown in the flowchart depicted in Figure 29. As can be seen, each time the concentrator notifies it with a URS and its request reference, or with a UMS and its request reference, the *STG-Telemetry JOB* intervenes*,* which gives rise to the *STG-ServiceJob*. This job, according to the arrival of WS received (URS, UMS) updates the status of the requests in the tables, and processes the files with the name specified in each UMS of the process sent by the concentrator.

From here, the content of the URS is interpreted, which is the WebService that governs the system and indicates when the communication is finished. When URS=1 it means that the communication is still in progress, while when URS=0 it means that the communication has ended successfully, the rest of possible values indicate error or warning messages that are specified in Annex II. These values are crossed with the values returned by the UMS and the casuistry follows the process indicated in the flowchart table in Figure 30.



Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figure 7: URS and UMS Casuistic

Finally, the registers are updated with the corresponding status, and the JOB Ficheros stores the information in the appropriate register.

Model 4

Model 4 is the set of Asynchronous tasks, which use WS for sending and receiving information and which use the DC-Order as communication WS.
Table 4 contains a list of the infoms that have been constructed pertaining to this model:

Model 4			
Report	**Content**	**Report**	**Content**
B02	Power Modification	B07	Modification of the Conncentrator Configuration
B03	Cut-Off / Reconnection	B09	Meter Parameters Modification
B04	Contract Modification	B9B	Meter Event Handling Modification
B05 B08	Meter Firmware Update		

Table 4: Model 4 Reports

This model is in charge of managing the direct action orders from the STG to the concentrator, i.e. the orders marked with Bxx. Like the report requests, these orders use messages in XML format sent through a WS in the form of a payload.

This model uses the DC-Order type as confirmation WS. It is a protocol in which an action is prepared in a specific measuring equipment, the necessary information is extracted by means of report requests, and finally the action is executed.

Once the DC-Order has been successfully executed, the concentrator must explicitly verify the requested order to each counter by means of a request from the STG.

The sending of an Order Request from the GTS to a counter may result in the execution of different phases or steps (download file update, pre-reporting, execution phase, post-reporting, etc.). These phases shall be executed sequentially in the CC, so that a phase cannot be started until the previous one is completed.

In case of failure in the execution of a phase, the execution sequence shall be stopped and the status of the order shall be reported to the GTS via UMS and URS with the appropriate category and error code.

To describe Model 4, we will use the flowchart corresponding to the *Firmware Update* command*,* contained in requests B05 and B08.

It is important to note that during the execution of this command, the system prompts the GTS to generate reports S19 or S22 (depending on whether the firmware upgrade is being performed on a counter or on the concentrator). The process of collecting and executing these reports is performed according to Model 3.

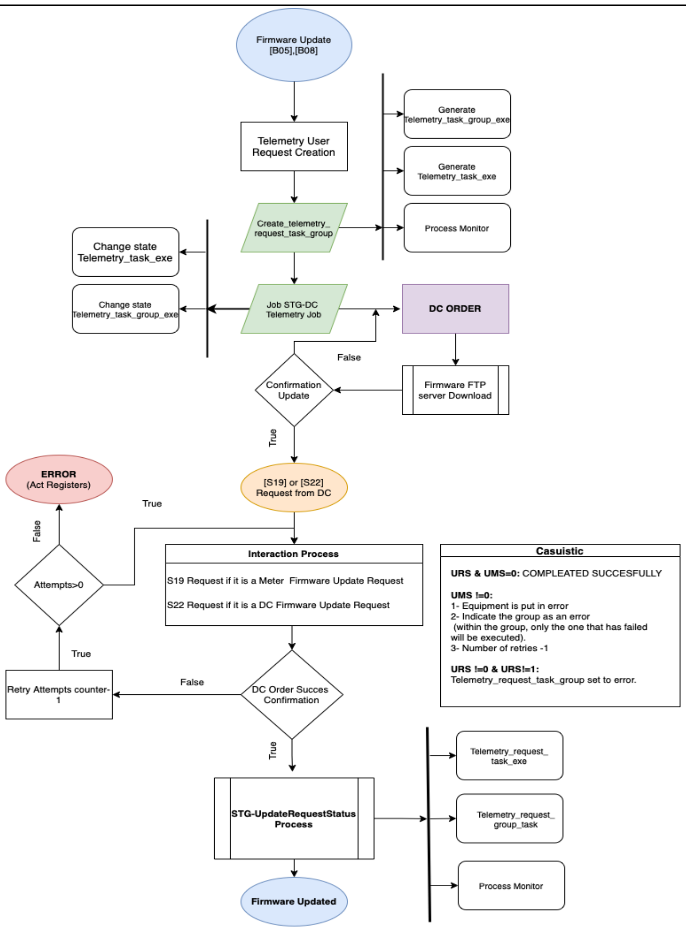


Figure 8: Firmware Update FlowChart

As in all previous cases, the first step in executing the command is to generate the three main OUM system registers through the *Telemetry_User_Request* script.
The *STG-DC Telemetry Job* is executed in exactly the same way as in the case of Model 2 and 3 only this time it precedes a WS of type *DC-Order*. This WS, once it is correctly confirmed, requests an S19 or S22 report to save a previous record (a backup) in case something goes wrong in the process of executing the job.

Once the process of generating the above reports is finished, the system must confirm the WS. To do so, it has a limited number of retries and takes the system to error status in case they are completed.

Finally, the *STG-Service Job* intervenes and by processing URS and UMS following exactly the same process described in Model 1, the states of the registers are updated and the communication is finalised by carrying out the update action in the requested equipment.

Case Study

When drawing the flowcharts necessary to characterise the OUM integration with the STG-DC specification, a pattern is generated when executing the commands. This pattern consists of a series of database records that allow the traceability of the message in order to correctly communicate the head-end system with the MDM. They contain the necessary information to identify the request, the destination, and the format of the request for its correct processing and to allow the user to monitor and act accordingly.

It can be seen in the diagrams shown in Annex X how three registers appear in the first level and how they are updated in the following levels. These three registers are:

- *Process Monitor*
- *Telemetry_Request_Task_Execution_Group*
- *Telemetry_Request_Task_Execution*

However, although it does not appear directly, intrinsically before generating these three main records, when the MDM executes a manual or scheduled task, it generates another record:

- *Telemetry_User_Request:* This register is in charge of creating the requests in the database. These requests are created in "Prepared" status so that the corresponding Job can later update the records by changing their status for each stage and case that the communication protocol presents.

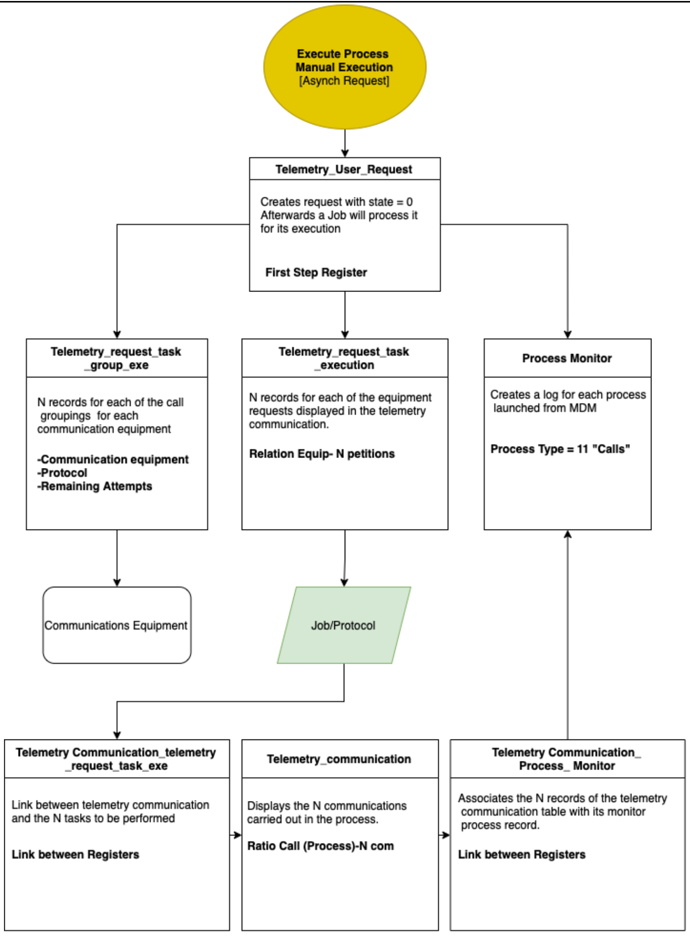

Execute Process
Manual Execution
[Asynch Request]
Telemetry_User_Request
Creates request with state = 0
Afterwards a Job will process it
for its execution
First Step Register
Telemetry_request_task
_group_exe
N records for each of the call
groupings for each
communication equipment
-Communication equipment
-Protocol
-Remaining Attempts
Telemetry_request_task
_execution
N records for each of the equipment
requests displayed in the telemetry
communication.
Relation Equip- N petitions
Process Monitor
Creates a log for each process
launched from MDM
Process Type = 11 "Calls"
Communications Equipment
Job/Protocol
Telemetry Communication_telemetry
_request_task_exe
Link between telemetry communication
and the N tasks to be performed
Link between Registers
Telemetry_communication
Displays the N communications
carried out in the process.
Ratio Call (Process)-N com
Telemetry Communication_
Process_ Monitor
Associates the N records of the telemetry
communication table with its monitor
process record.
Link between Registers

Figure 9: Registers Interactions FlowChart

Knowing these processes at the database level is fundamental to understand the programming logic and the creation of the flowcharts proposed in the project. The generic interaction scheme that these registers follow is as represented in Figure 9.

This Case Study is presented in the long document corresponding to this project.

Table of contents

Index of figures

Index of tables

CHAPTER 1. INTRODUCTION

The fight against climate change is one of the most important challenges that humanity must face, and the commitment to a decarbonised economy based on renewable energies in a new energy mix is a task for all of us. This is how the Spanish National Integrated Energy and Climate Plan (PNIEC) was born in Spain, which has the minimum objective of reaching 32% renewable energy by 2030. This objective is achievable in a scenario of high decarbonisation and electrification of the economy, using decarbonised fuels in those niches that are difficult to electrify [1].

The PNIEC plan for the period 2021-2030 contemplates as basic lines of action, among others, the electrification of the end uses of energy and the commitment to a new model of electric mobility and firm support for self-consumption as the basis of a new model of distributed generation. In the coming years, demand management, storage, digitisation of grids and new grid operation schemes will be essential to provide flexibility to the system [newspaper]. In this scenario, the term Smart Grid acquires great relevance [2].

The term Smart Grid refers to the smart electricity distribution network. Smart grids are bi-directional grids, capable of transmitting electricity in both directions. This allows the integration of new electricity consumption and generation profiles by introducing concepts such as DG (Distributed Energy).

A Smart Grid is based on the application of ICT (Information and Communication Technologies) to the electricity grid. Telecommunications networks are already an important part of utility operations. The challenge of extending these networks to millions of endpoints, potentially geographically dispersed over large service areas, is inherent to the Smart Grid [1].

Faced with this new paradigm, utilities companies are facing a series of digital transformation challenges marked by the migration of the energy supply chain to a new distributed model. Within smart grid development, Smart Metering is one of the fundamental pillars, driven by sustainability and user satisfaction.

Minsait has developed Onesait Utilities Metering technology, a smart metering solution designed for utilities that enables a competitive advantage in an energy market affected by global trends, allowing the Utility to collect, certify and export information data from different information sources [15].

EDP (Energías de Portugal), one of the largest electricity groups in Europe, has requested Minsait's services to integrate the OUM solution in its utilities in Spain. Three institutions will be integrated under a single manager (EDP-Spain, Viesgo and Begasa). This project will focus on the analysis and functional development of OUM integration under PRIME's STG-DC specification in EDP.

PROJECT MOTIVATION

EDP is currently responsible for the distribution and maintenance of the electricity distribution network in Cantabria, Asturias, and areas in the north-east of Castilla y León and Galicia, although it also operates in specific areas in Madrid, Zaragoza, Valencia and Huesca. The scope of the project entrusted to Minsait will cover the management of the advanced metering infrastructure of EDP Spain's electricity distribution companies: E-Redes, Viesgo and Begasa [6].

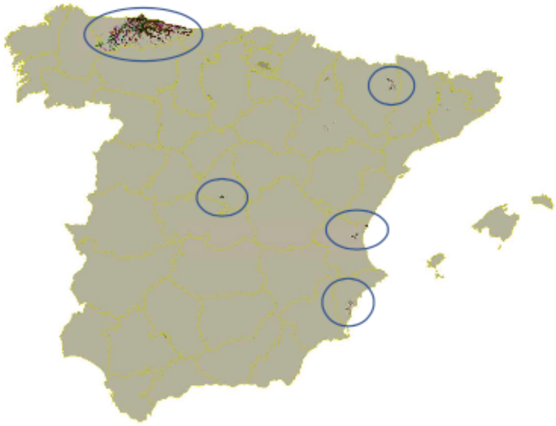


Figure 1: EDP Spain Operation Map [6]

When integrating different institutions into the EDP Spain infrastructure, a problem arises for the company when it comes to exploiting the information in an agile, simple and transversal way with the aim of offering the new services introduced in the electricity distribution sector. This problem consists of the interoperability and centralisation of the different equipment that makes up the system. In other words, each of the companies that have been integrated into the EDP portfolio used different technologies, different communication protocols, different types of devices and different remote management solutions:

The acquired part of E-Networks has approximately 700,000 Prime counters that communicate with the 1.3.6 protocol. It also has 7,000 Prime 1.3.6 concentrators that communicate with the AMI using the STG-DC standard in its 3.4 version. In addition, the system includes various types of advanced monitoring equipment (using the STG-DC standard) as well as equipment such as routers and fibre optic interfaces.

On the Begasa + Viesgo side there are 700,000 meters that use Telegestore as a communication protocol with the concentrators and 12,000 concentrators that currently communicate with the AMM via a private standard owned by Endesa.

The aim of this project is to collaborate in the integration of these systems in the Minsait OUM multiprotocol tool, with the aim of centralising all communications to enable a single remote management of the entire fleet of equipment so that the user is able to monitor, improve and automate all communications processes that occur with each of the measuring devices that make up the network.

PROJECT OBJECTIVES

As mentioned above, Minsait will provide a technological solution with high penetration in the company's technologies to address the problems posed by EDP Spain.

Although OUM is a multi-protocol system, the integration of the system with the STG-DC specification in its version 3.4 had not been carried out to date in the tool. For the EDP project, it is essential to develop all the necessary modules and communication links to integrate the application and provide MDC and MDM solutions in the EDP infrastructure and for future projects that need to integrate this technology.

The STG-DC specification appears in the E-Networks infrastructure, and there is also the possibility of replacing Viesgo's proprietary AMI protocol with this specification in the future. For this reason, this project will focus on the integration of the STG-DC specification with OUM.
The objectives of this project are as follows:

- To present a joint study of the main features of the STG-DC specification and the Onesait Utilities Metering application in order to understand and take advantage of their synergies in their integration.

- Classification of the different types of requests and reports that should be able to communicate between the user and the measuring point in question. These depend on the requirements of the customer and the possibilities provided by the specification.

- Finally, and as the main point of the project, a functional analysis of this integration will be carried out. In this functional analysis, the communications process that will take place for each type of request between the concentrator and the MDM system will be broken down in the form of flow diagrams that will have to be developed taking into account all the casuistry that may occur in each interaction in these communication processes. These diagrams shall be accompanied by a brief description of the type of communication being carried out, identifying the file format exchanged and the parameters involved in the process.

The aim of this work is to structure and capture in a document all the information necessary to identify possible inconsistencies or gaps to be filled in the system and to be able to successfully execute the technical implementation of the specification in the OUM application.

STATE OF THE ART

The electricity metering equipment used worldwide is the well-known electricity meter, which consists of three main elements, namely the metering system, the memory element and the information device. In this sense, the electricity meter performs the function of the network interface with the user, it is the front-end of the network.
This metering equipment can be classified according to its technological characteristics (whether electromechanical or electronic), by its single-phase or three-phase functionality, energy, and operational characteristics as a programmable recorder type device that allows remote management.

These meters have evolved significantly in the electricity market as new technologies have been developed. However, nowadays, different types of electricity meters coexist in households, which allow, among other things, the accounting of energy expenditure. In this sense, we will distinguish between two types of meters depending on their basic technology.

- **Electromechanical Counters**

  The basic idea for the electromechanical induction meter comes from the studies of Galileo Ferraris who made a key discovery, that by means of two alternating current fields out of phase, a solid metallic disc can be made to rotate. This discovery contributed to the later development of induction motors. The most common electric meter is the single-phase meter patented by Elihu Thomson in 1889, this model is considered the standard and is the basis for most modern electromechanical meters.

  These meters only allow the measurement of one type of energy, accumulated kWh and accumulated KVAh, they do not have tariff discrimination and are standard electromechanical induction meters. This type of meter will become obsolete and will disappear from the market.

- **Electronic counters:**
  They operate electronically and are also known as solid-state meters. They are able to measure consumption in real time, take advantage of tariffs based on hourly discrimination and know at all times what is happening with the energy at each supply point.

Among the digital meters, the programmable remote management meter stands out, known today as the new era of the digital meter due to the innumerable advantages it offers compared to its "little brothers".

These devices are based on AMI (Advanced Metering Infrastructure) technology, these devices allow the reading of accumulated energy or instantaneous power, they allow demand records or programming of load intervals previously defined with each user. The Smart Meter concept was born from these new incipient technologies. These devices provide information and control of the quality and programming parameters of the service via the management centre, together with telematic updating of the metering software. It includes extended network communication with the manager and Home Area Network (HAN) with the local consumption equipment.

In Spain, the installation of this type of meters will have to be carried out by all users with power equal to or less than 15kW before 2019 as defined in Real Decreto 1110/2007, Order ITC 3860/2007 and IET 290/2012.

Based on these reasons and those described above, the interest in the deployment and implementation of this technology is justified, as well as the importance of all the studies that encompass this world and among which this project is included.

CHAPTER 2. DESCRIPTION OF THE TECHNOLOGIES

2.1. AMI COMMUNICATIONS ARCHITECTURE

The acronym AMI, as mentioned above, stands for Advanced Metering Infrastructure and represents the heart and soul of this technology. To begin to describe this standard architecture for advanced metering-based communications systems, it is necessary to discern at a high level what are the main elements that comprise it [17]:

- **Smart Meter:** This is an electronic smart meter capable of recording electricity consumption at configurable time intervals and communicating the information from this recording to the control unit. It is therefore an instrument that fulfils the function of a sensor and an interaction point simultaneously, i.e. a smart node.

- **Data Concentrator (DCU):** It is the backbone of the Advanced Metering Infrastructure (AMI) that supports data acquisition, transfer of energy data to the central database and automated reading of electronic energy meters. DCUs are mainly used in low-voltage electric power systems.

- **Information systems:** These information systems act as a link between data acquisition and further processing. They are designed as databases (both relational and non-relational) and have remote management capabilities. These information systems are characterised by having a complete view of the infrastructure.

On the other hand, these systems are usually heterogeneous communication infrastructures, involving different technologies depending on the specific requirements of each network segment. In Spain, communication between meters and data concentrators is based on narrowband PLC technologies.

Nowadays, there are many options for implementing communication technologies in the distribution network. Some examples of these technologies are GPRS, GSM, DSL or fibre optics, or through technologies such as WiMax or ZigBee. However, of all these technologies, the most appropriate for the last mile of communication between meters and concentrators is the mentioned narrowband PLC technology.

One of the advantages of this technology is that it offers lower infrastructure costs, as the communication medium (low and medium voltage cables) is already deployed. In addition, it offers the possibility of communication between different devices by simply plugging them into the network. However, this technology also presents obvious problems, as the medium used was not designed for data transmission where electromagnetic interference, noise, or impedance variations in the channel are present.

Power Line Communications (PLC) is a telecommunications technology that relies on the use of the electrical power distribution infrastructure (typically LV or MV) for high-speed transmission of signals and data. It provides broadband data communications over conductors that are already in use for the transmission of electrical power using a modular signal. This PLC technology is seen as a reliable communication medium for applications such as the Internet of Things (IoT) and Smart Grids [3].

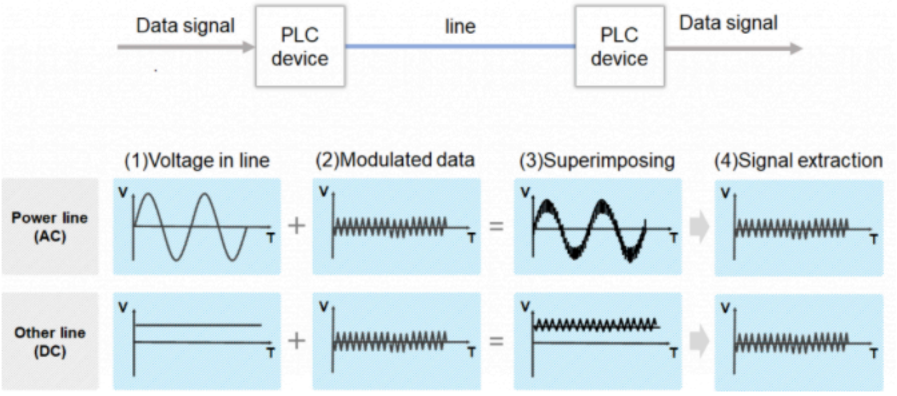


Figure 2: Basic PLC working principle [3].

There are several types of PLC technologies, the main difference being the frequency range in which they operate and the communications protocol they follow, which determines their data transmission speed: Narrowband PLC (3G-PLC, PRIME, Meters & More), Mediumband PLC (HPLC) and Wideband PLC (HD-PLC, HomePlug, G.hn) [3]. In a generic AMI communications architecture, narrowband PLC technology is used at the physical layer (PHY), the data link layer (Data Link Layer), and the convergence layer (CS). At the application level, DLMS/COSEM-based protocols are used, while broadband and long-range technologies such as GPRS or Optical Fibre are used for communication between the concentrator and the information system. In the case of PRIME, at the application level STG-DC is the web services based specification used and on which this project is based [10].

2.2. EUROPEAN-WIDE DEPLOYMENT OF PLC TECHNOLOGIES

In this section we will analyze the deployment of narrowband PLC technologies at European level, with the aim of understanding where this type of technology stands and which are the different technologies used. Figure 3 shows a map of Europe showing the penetration of this technology in the different countries.

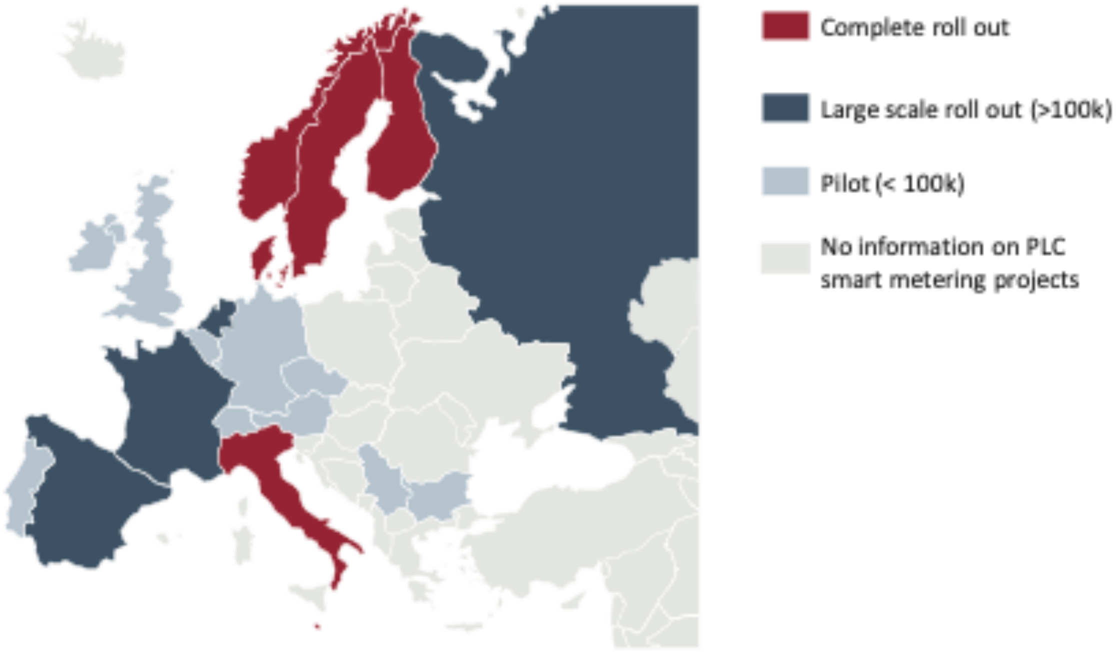

Complete roll out
Large scale roll out (>100k)
Pilot (< 100k)
No information on PLC smart metering projects

Figure 3: Deployment of AMI in Europe based on PLC [6].

It can be seen that some of the Nordic countries and Italy have already completed the roll-out of smart meters, although other projects are still under consideration. Spain and France, which have some of the largest Distribution System Operators (DSOs) in Europe, also have large-scale deployments. Germany and England, because of the number of DSOs, have many medium-sized deployments.

On the other hand, regarding the penetration of the different NB-PLC standards in the European market it can be seen that the first countries interested in PLC smart metering have chosen Echelon, Meters and More or G1-PLC. Current pilot projects are now largely with Prime, G3-PLC or proprietary technologies.

PRIME stands out in the market as NB-PLC technology as it is a consolidated standard in the Iberian Peninsula and is also being implemented in countries such as the UK, Australia and Brazil [5].

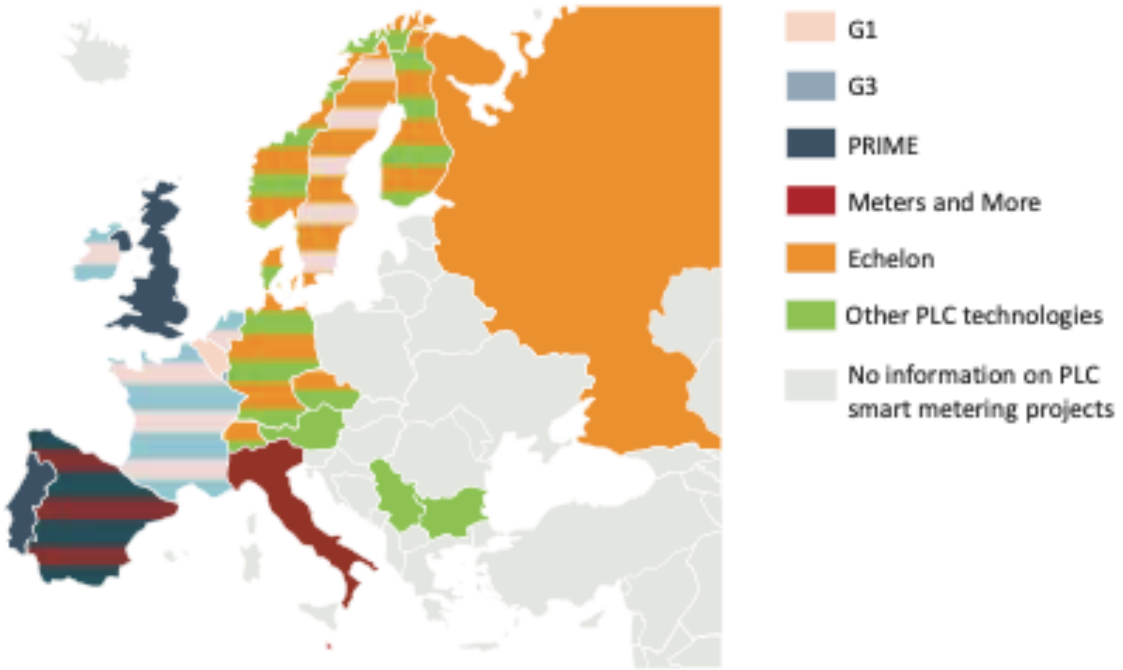


Figure 4: Technologies distribution of PLC smart metering projects in Europe [6].

Figure 4 represents the degree of penetration of each technology in the European market according to a study conducted by Accenture.

2.2.1. PLC TECHNOLOGIES

In this section, the main PLC technologies available in Europe will be described in a table.

PLC Variants	Main Characteristics
G1	Open specifications for PLC communication S-FSK based. Max. bit rate 2.4 kbps IEC 61334
G3	Open specifications OFDM for the CENELEC A Band Frequency Band: 36-90.6kHz Bit Rate (kbps): 5.6-45 36 channel-based OFDM system 6Lo WPAN adaptation layer to transmit IPv6 over PLC Signal crossing the MV/LV transformers
PRIME	Open specifications 97 channel-based OFDM system Frequency Band: 42-89 kHz Bit rates up to 128kbps
Meters and More	Open technology Founded by Enel and Endesa. OPEN project BPSK modulation Bit rate 4.8 kbps
Echelon	BPSK-based NB-PLC solution CENELEC A Band (In utility applications)

Table 1: Deployed PLC variants [6].

2.3. PRIME FOR AMI APPLICATIONS

PRIME (PoweRline Intelligent Metering Evolution) is an open, free, non-proprietary standard for a globally established OFDM PLC technology for advanced metering, network control and asset monitoring applications, and has fulfilled the goal of establishing a set of open PLC standards on an international basis. There are more than 20 million PRIME meters installed in more than 15 countries around the world [5].

Since the beginning of electricity deregulation and market-based pricing around the world, utilities have been looking for a means to match consumption with generation. Traditional electricity and gas meters only measure total consumption and therefore do not provide information on when energy has been consumed at each site. PRIME AMM offers a way to measure this site-specific information, allowing different prices to be set for consumption based on time of day and season [5].

One of the advantages of implementing the PRIME protocol is that it ensures interoperability between different equipment and systems, is not subject to any intellectual property, and is designed to achieve a low implementation cost and high performance, thus accelerating the entry of new operators.

The physical layer, MAC layer and convergence layer of version 1.3.6 of the specification have been considered standard by ITU-T since 2012. Version 1.4 extends the frequency spectrum to allow operation in the North American and Asian markets and includes some enhancements aimed at improving robustness at the physical and MAC level [7, 8].

2.3.1. PRIME PHYSICAL LAYER (PHY)

As described above, the PHY entity corresponds to an OFDM modulation scheme. This layer uses frequencies in the 3KHz to 95 KHz band. It should be noted that when working in a frequency range below 40KHz, low voltage power lines present a number of problems:

- The magnitude of the load impedance seen by transmitters is sometimes below 1Ω , especially for nodes located near transformers.
- Background colour noise appears, present in the power lines.
- Consumer behaviour operates at low frequencies causing interference with these signals.

For this reason, the specification for the PHY layer uses a frequency range for PRIME OFDM from 42KHz to 89KHz in its version 1.3.6. This is achieved by using OFDM modulation with a signal loaded with 97 equally spaced subcarriers, transmitted in 2240us symbols, of which 192us correspond to a cyclic prefix, i.e. such a signal is made up of 97 subcarriers and lasts 2.24ms[10].

For data, differential modulation is used, with one of the following three possible constellations: DBPSK, DQPSK or D8PSK. Thus, uncoded rates of approximately 47 Kbps, 94 Kbps and 141 Kbps (without taking into account the cyclic prefix), respectively, can theoretically be obtained.

On the other hand, a scrambler is used to avoid long sequences of identical bits. In addition, a 1/2-ratio convolutional encoder will also be used and may be accompanied at its output by a bit interleaver, which may be enabled/disabled by higher layers if the channel is good enough and higher performances are needed [10, 11].

On the transmitter side, the PHY layer receives an MPDU from the MAC layer, and from the latter generates a "PHY Frame", the structure of which is shown in Figure 5:

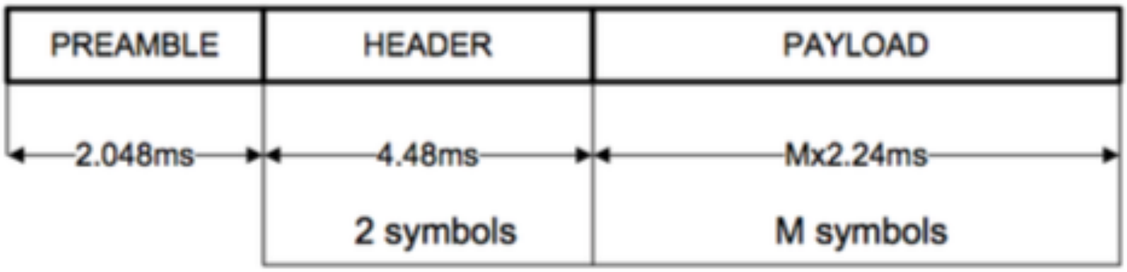


Figure 5: PHY Frame format [10, 11]

Each PHY Frame starts with a preamble of 2048 ms duration, followed by a number of OFDM symbols, each lasting 2.24ms. The first two OFDM symbols correspond to the PHY Frame header. The remaining M OFDM symbols carry the payload. The value of M is´ indicated in one of the header fields and is at most equal to 63 [11].

On the other hand, for the processing of the PPDU it is necessary to follow a sequence represented in the form of a block diagram, where this signal goes through different stages, as shown in Figure 6 below:

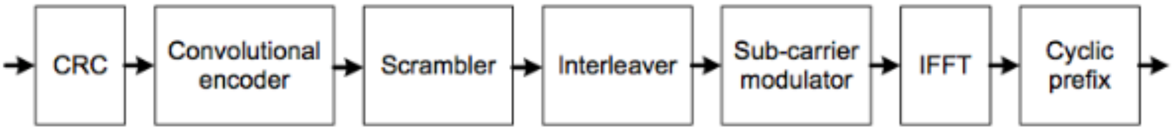


Figure 6: PDU processing unit in the transmitter [11, 14]

A CRC is added to the header in order to detect possible transmission errors in certain fields of the header. Then, if the FEC is enabled, convolutional coding is performed on the payload. The header, regardless of how FEC is enabled, is always encoded. The next step is scrambling, which is performed for both the header and the payload. At the output of the scrambler, the header is interleaved, while the payload is only interleaved if the FEC is set to "ON". The bits are then modulated using a DBPSK, DQPSK or D8PSK modulation scheme. Then, an IFFT (Inverse Fast Fourier Transform) is performed on the modulated data and finally in the last block a cyclic prefix is added to each symbol.

Therefore, when the header and payload bits enter the blockchain shown in Figure 6, at the output of the blockchain you have a concatenation of OFDM symbols that constitute the payload and header portions of the PPDU. The header portion contains two OFDM symbols, while the payload portion contains M OFDM symbols, whose maximum value is M = 63.

2.3.2. MAC PRIME LAYER

The MAC layer belongs to the Data Link Layer. It is responsible for defining the procedures that enable different devices to share the use of the radio spectrum. Each station and access point in the wireless LAN standard provides a number of services through the MAC layer to the higher layers by exchanging service data units (MSDUs) [4].

The MAC layer that is defined, as mentioned above, can be placed between the physical layer and the LLC link sublevel. The MAC architecture of the 802.11 standard consists of two basic functionalities: The Point Coordination Function (PCF) and the Distributed Coordination Function (DCF) [4].

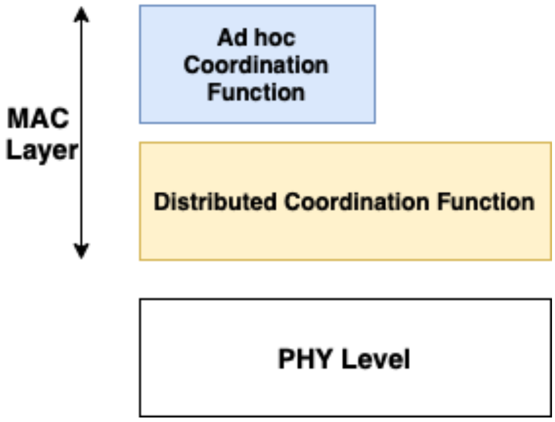

MAC Layer
Ad hoc Coordination Function
Distributed Coordination Function
PHY Level

Figure 7: MAC Inalambric Functions [4]

At the MAC level, a sub-network has a tree-like structure, where two main types of nodes can be distinguished:

- **Base Node (BN):** The base node is the master node that manages a PRIME subnetwork. There is only one BN per subnetwork, and it is responsible for managing resources and connections, it acts as the master node and is at the root of the tree structure, it is the most important element. In AMI terminology, the base node is the data concentrator [7, 14].

- **Service Node (SN):** Service nodes are the core element of the subnetwork. They can be seen as leaves or branches in the tree structure and are initially in a disconnected state, their functions being to maintain connectivity to the subnetwork for the different application layers and to switch information from other nodes. Smart meters must register in the sub-network to become service nodes that depend on their corresponding BN. In AMI terminology, service nodes are smart meters. [7, 14].

Initially the sub-network is composed only by the base node and the rest of the nodes that want to be part of it have to perform a registration procedure.

- **Disconnected:** This is the initial state of any service node before it registers into the PRIME subnetwork. While in this state, the smart meter cannot communicate its own data or switch data from other SNs and its only goal is to search for a subnetwork that it can reach and register on it.

- **Terminal:** This state allows the SN to establish new connections and transmit data, but not to switch data from other SNs.

- **Switch:** This state is the most complex one and allows the SN to establish new connections, transmit its own data and switch data from other SNs. The switching function allows some SNs in switch state to act as repeaters so that other SNs that cannot communicate with the BN otherwise can also be part of the subnetwork. Consequently, the tree topology of a PRIME subnetwork has the switches as branches that join the different levels of the subnetwork [11, 14].

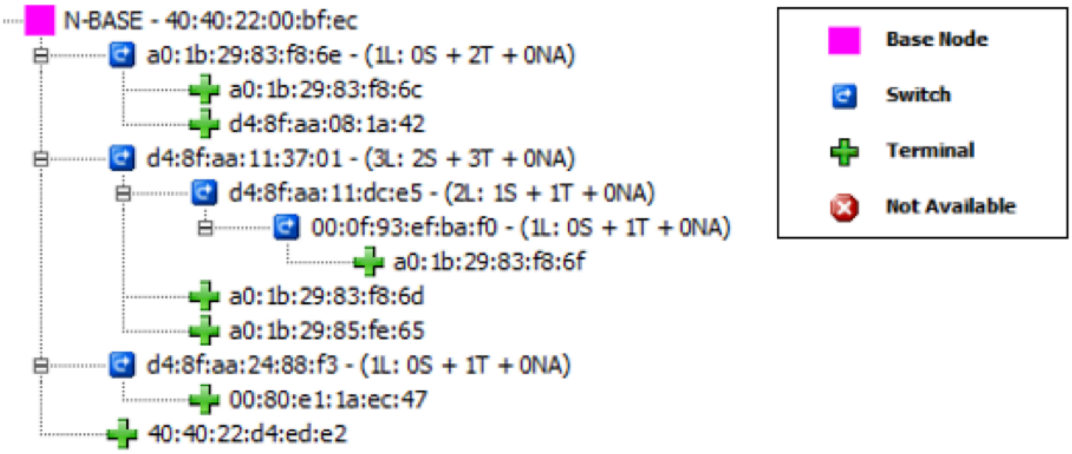

N-BASE - 40:40:22:00:bf:ec
a0:1b:29:83:f8:6e - (1L: 0S + 2T + 0NA)
a0:1b:29:83:f8:6c
d4:8f:aa:08:1a:42
d4:8f:aa:11:37:01 - (3L: 2S + 3T + 0NA)
d4:8f:aa:11:dc:e5 - (2L: 1S + 1T + 0NA)
00:0f:93:ef:ba:f0 - (1L: 0S + 1T + 0NA)
a0:1b:29:83:f8:6f
a0:1b:29:83:f8:6d
a0:1b:29:85:fe:65
d4:8f:aa:24:88:f3 - (1L: 0S + 1T + 0NA)
00:80:e1:1a:ec:47
40:40:22:d4:ed:e2
Base Node
Switch
Terminal
Not Available

Figure 8: PRIME subnetwork topology [11].

Finally, transitions between one functional state and the next are made according to the following two operations:

- **Registration:** This is the transition between the offline and the terminal state. The BN includes the SN as part of the list of registered nodes.

- **De-registration:** This is the reverse of registration, and can occur from either the offline or the terminal state. The BN removes the SN from the list of registered nodes. Deregistration can be initiated by the SN (if it wants to find a better coupling to the subnetwork) or by the BN if the SN does not meet the conditions for being connected and defined by some MAC procedures.

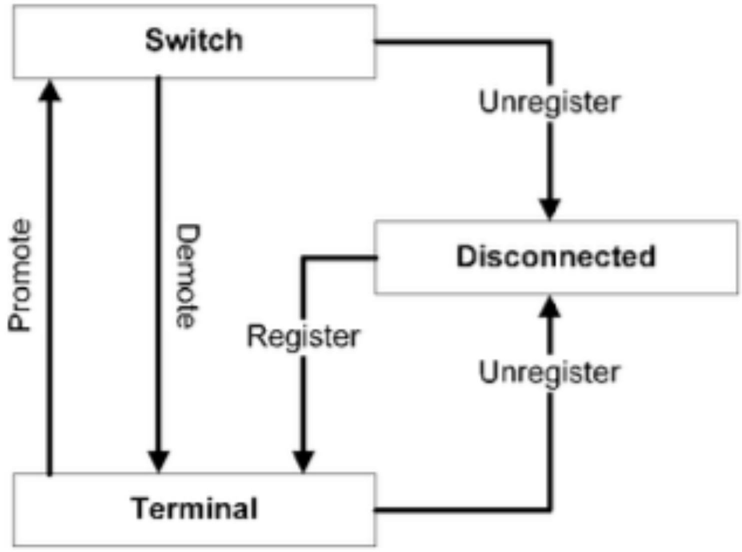


Figure 9: Functional states of a Service Node [11]

Each node has a 48-bit MAC address called EUI-48, used to identify it during the registration process. The EUI of the base node is used to identify the subnetwork and is called SNA (SubNetwork Address). The LSID (Local Switch IDentifier) is an 8-bit unique identifier for each switch node. Each service node receives a 14-bit local identifier called LNID (*Local Node IDentifier*). The LNID and the SID (LSID of the *switch* node it is associated with) form a 22-bit identifier called NID (*Node IDentifier*). A 9-bit Local Connection Identifier (LCID) is reserved during connection establishment that identifies a single connection. The NID and LCID form a 31-bit connection identifier (CID) [7].

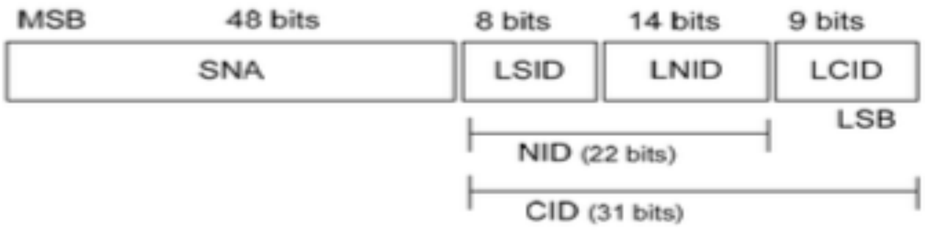


Figure 10: Addresing Structure [7]

2.3.3. PRIME LOGICAL LINK CONTROL LAYER (LLC)

The Logical Link Control (LLC) layer is the layer responsible for the identification of the logical form of the different protocol types and their encapsulation. A type code, or a Service Access Point (SAP) identifier, performs the logical identification. The type of LLC frame used by an end station depends on the identifier expected by the upper layer protocol.

The Logical Link Control (LLC) layer, belonging to the Convergence layer, is responsible for handling logical connections. It identifies each transaction with an identification number and performs the flow control processes. Flow control is implemented in PRIME by setting a Maximum Transmission Unit (MTU) and using a sliding window procedure. The MTU defines the length in bytes of the largest data packet that the MAC layer can encapsulate. In case the application tries to send a larger message, the LLC layer fragments the message into several packets, none of which will be larger than the MTU. Each of these packets is labelled with an identifier so that the receiving party can reassemble them [8].

With respect to the sliding window procedure, PRIME sets different allowed values for the Window Size (WS). The WS value can play an important role in the network performance in terms of latency. In addition, PRIME devices may or may not implement Repeat Request (ARQ) capabilities to ensure correct reception of all messages. Since ARQ parameters are negotiated at the connection stage, this process works end-to-end between the transmitter and the final receiver, thus making the switches transparent to it [8]

2.3.4. DLMS/COSEM APPLICATION LAYER

For the application layer, the DLMS/COSEM standard is used in PRIME. The following is a general definition of what the DLMS/COSEM-based technology is about.

DLMS/COSEM (IEC 62056) is an application layer protocol specifically designed for communication with smart meters. COSEM is the data model and DLMS is the PRIME application layer protocol that defines the exchange of data between metering equipment. DLMS/COSEM supports multiple lower layer protocols. It supports communication over TCP, UDP, RS-232, RS-485 and multiple PLC protocols. [7]

DLMS (*Distribution Line Message Specification*) is the protocol used in the PRIME application layer. It is independent of the lower layers and therefore of the communication channel and is designed to support messages to and from power distribution devices in a computing environment. Its acronym has evolved to *Device Language Message Specification* with the aim of providing an interoperable environment for structured modelling and data exchange between meters. It supports functionalities such as remote meter reading, remote control and value-added services for metering any type of energy, such as electricity, water, gas or heating, which makes it a key factor in the digitalisation of the energy sector [7].

COSEM (*COmpanion Specification for Energy Metering*) is the data model. It provides a functional description of the meter based on its interface. The modelling uses an object-oriented approach. An object consists of attributes and methods according to an *interface class*. Attributes are of different types (*data type, integer, array*, etc.) and can be organised in profiles (load, historical, event, etc.). The OBIS (*OBject Identification System*) is used for object identification.

The xDLMS (*extended dlms*) application layer protocol is also defined. This DLMS extension specifies how to construct messages to access methods and attributes of COSEM objects.

2.3.5. STG-DC INTERFACE SPECIFICATION

STG-DC is a protocol developed by the Prime Alliance to meet the requirements of the Spanish legislation regarding the deployment of smart meters. This specification consists of a web service protocol based on two communication standards: SOAP (Simple Object Access Protocol) and FTP (File Transfer Protocol). The purpose of this specification is to define the entire bidirectional communication process between the Remote Management System (STG) in question and the Data concentrators (DC or Data concentrator).

The aim is to ensure that the DC can be managed through the remote management system. The flow of information exchanged between the concentrator and the remote management system will be bidirectional [12]:

- The AMM system may send an order or report request to the concentrator, which will proceed to execute the necessary actions to comply and reply with the files or values required by the system.
- The concentrator shall be able to send information to the remote management system in two different modes:
    - Reports scheduled to be sent to the GTS as internal tasks of the DC
    - Events occurring spontaneously in a counter or in the concentrator itself.

This exchange of messages between STG and DC is typically done through XML type files, which are classified into two types of messages:

- Bxx for orders from STG to DC
- Sxx for planned reports from DC to ST

Thus, when one of these actions is performed at the initiative of the concentrator, it is called Push mode, and when it is a request from the STG, it is called Pull mode.

This specification has a list of standardised messages and reports to carry out these communications.

2.3.5.1. STG-DC Request Identification

All requests made by the STG to a DC will include an IdPet parameter or field which is the numeric identifier of the request. The IdPet value is unique within each hub: it is assigned by the GTS in the request, and inherited by all reports generated by the DC related to that request. The IdPet field is the identifier number of any request made to a concentrator [12].

This value will be unique for each concentrator and must comply with the following logical rules:

- Two different requests made to different concentrators at the same time by the STG may have the same IdPet code.
- Two different requests made at the same time to the same concentrator by the STG can never have the same IdPet code.

This standard will allow the GTS to link each request to each concentrator and to track the time taken to deliver responses and keep track of all messages transmitted.

However, reports generated at the initiative of the DC (scheduled tasks) will not have an associated IdPet code, they will have a value equal to zero, because they do not correspond to any associated GTS request.

2.3.5.2. DC Data Keeping and Programmed Tasks

The concentrator shall store the historical information of the counters (collected through the scheduling of the task or explicit request from the GTS) and keep it in its database locally according to the TpStore parameter. This information can be sent to the STG when requested or if the scheduled task includes sending it to the STG.

The instantaneous value information shall be read from the meters when a request is received. Such requests may take some time depending on the number of counters to be queried.

In each data request made by the STG, the STG can explicitly ask the DC to collect the data from the DC, from the counter or give priority to the DC but access the counter if it is not available in the DC [13].

On the other hand, it is important to mention that the "Fh" attribute is used in numerous places throughout the document, xml reports, orders, request for reports, etc. Some clarification is needed on the correct mapping time in the different situations.

For counter reports such as S02, S2B, S03, S04, S05, S5B, S09 and S14 (supervisor counters only), the field 'Fh' shall be mapped to the timestamp of the periodic data record embedded in the OBIS of the recorded data.

For DC or counter events "Fh" is the time stamp of the event record within the equipment.

2.3.5.3. Data Flow Principles and Main Parameters

The data flow specified in this protocol between the GTS and the DC shall follow the following principles:

- Requests and notifications use WebServices on both sides.
- The data concentrator provides the GTS with a main web service to allow a request to be made:
    - Get the results immediately (synchronous request).
    - Get results later (asynchronous request)

- The STG provides a primary WebService to the DC for notification of
    - DC or counter events

    - Application processing status
    - In some cases reports generated by the DC

- In other cases, reports are also sent via the FTP protocol.

In order to reduce the complexity of WS, requests to the concentrator, encoded as Bxx, as well as the reports generated for a given request, encoded as Sxx, consist of messages defined in XML that are collected as a "payload" by WebServices invocations. Thus, the added value of WebServices is the control of the flow, regardless of the type of data transported.

On the other hand, with regard to the GTS parameters, it should be noted that this protocol includes a parameter that indicates the priority of the communication order by defining three different priorities (Very High (1), High (2), Normal (3)).

This specification does not include Time Outs in the protocol, all these mechanisms are configured directly in the DC data concentrator based on the parameterisation of the PLC communication.

The DC must manage all order requests, report requests and scheduled tasks by their parameterisation across these three categories. This means that STG orders with the same priority code as scheduled tasks are considered to have the same priority.

Txx tasks must be executed first by the DC, before other requests or scheduled tasks, regardless of the assigned priority. Tasks or requests of similar priority order are managed under a FIFO (First in first out) philosophy: the first task that arrives at the DC, is the first to be performed.

It should be clarified that, when a low priority task is running and another task with higher priority arrives at the DC, the execution of the running task shall terminate the operation being performed for the current counter and then start the execution of the higher priority one. In other words, the access to the counters for the running task shall stop and the DC shall start accessing the counters to perform the higher priority task. Upon completion of the higher priority task, access to the counters for the lower priority task is resumed from the point where it was previously stopped.

Regarding the management of the priorities when the retry mechanism is activated, please refer to the DC specification [12].

2.3.5.4. Management of Data Flow by the DC

The DCs shall have a replica of the information stored in the counters according to the TpStore parameter.

Each STG request shall be answered according to the "source" parameter of the request. This means that the data can be collected from the DC database, from direct access to the counter or from a combination of both, as specified (taking into account where the data is available).

Only instantaneous value requests and counter parameterisation requests shall imply that the DC always has to go to the counter to collect this information in order to respond to the GTS.

DCs will have scheduled tasks to send information to the STG through push mechanisms.

- **Petitions to the DC:**

Some STG requests require multiple responses from the DC. Each of these responses must carry complete information for one counter and one day. The DC includes two global parameters to define the information splitting logic:

- Maximum number of counters included in a message (NumMeters)
- Maximum time (seconds) that the DC can spend collecting partial information to build a report to be sent to the STG (TimeSendReq).

- **Tasks programmed in the DCs**

The STG can schedule and manage tasks in the DC. There are a number of tasks that typically exist, which will be included in the configuration of the DC. Note that for all data requests from the counter, if the data is already in the database of the CC, unless otherwise specified, it should be collected from there instead of requesting it again from the counter (as defined in the "Source" flag of the AsynchRequest).

This project will focus on the integration and implementation of this specification (STG-DC) from a functional point of view with Minsait's own tool, Onesait Utilities Metering solutions, which will be described in the following section.

2.4. ONESAIT UTILITIES METERING TECHNOLOGY

OUM (Onesait Utilities Metering) is the metering product for utilities developed by Minsait (energy, water and gas) that enables the capture, storage, exploitation and export of metering data in the field of Smart Grids. This software system allows the company a bidirectional interaction with field metering devices, both meters and concentrators, through any WAN communications network.

One of the differentiating features of OUM is that it is a multi-manufacturer system, i.e. the system allows bidirectional communication with any type of measuring device, regardless of the manufacturer and protocol required.

One of the bases of the OUM solution is the automation of processes, once the system has been configured it is able to operate in an automated way depending on what is set by the user of the system. All tasks performed by OUM can be executed manually or automatically in order to reduce user intervention.

On the other hand, OUM also has the capability to act as an inventory master of the measuring equipment deployed in the field. Additionally, the system enables the synchronisation of inventories with third-party systems.

Therefore, within the Smart Metering application map of any energy business, OUM can be considered as the master of metering data, as it centralises the reception and processing of all the metering data of the business it represents. This system has been developed for use by any type of utility that requires centralised and efficient management and processing of metering data [14].

2.4.1. OUM FUNCTIONALITY

This section will describe the main features of the functionality of the OUM solution to be implemented in the project [14].

2.4.1.1. GUI access and functionality

The access will be done through the Login screen where the User and Password are entered. The solution will be integrated with the corporate user authentication system LDAP or Active Directory.

Once the user is inside the application, he/she will find a main dashboard with the following elements.

- Main window: The information is displayed in the different views, depending on the selected navigation option. One of these views is the Home Dashboard that the user finds when accessing the system.

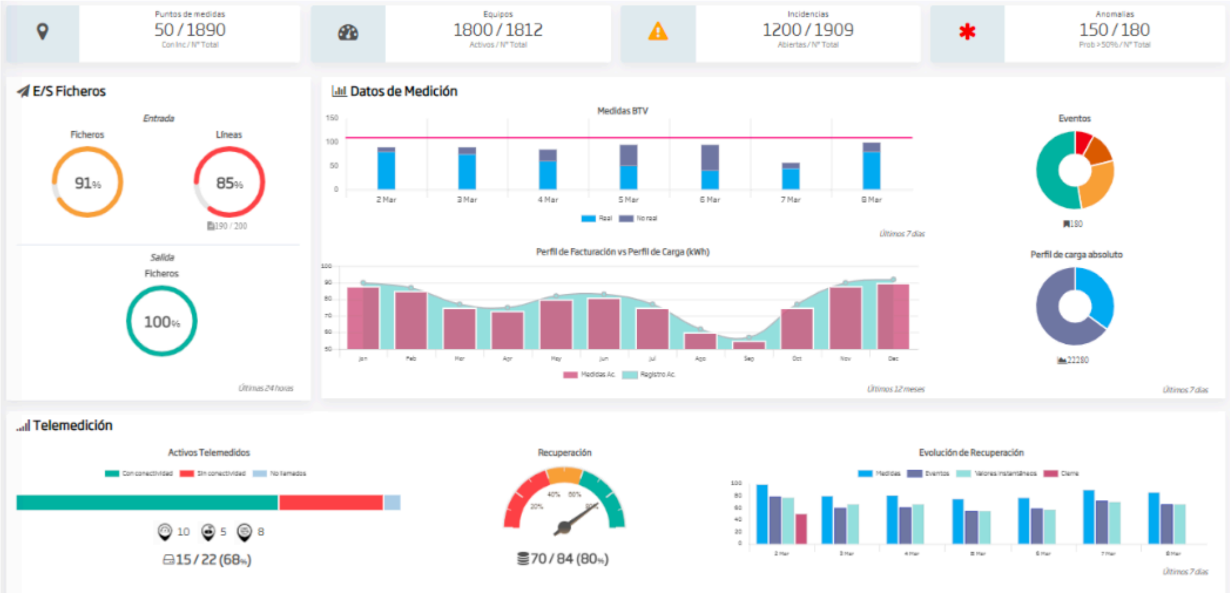


Figure 11: Main Menu Display [Indra]

  - Main Menu: This menu is located on the left side of the window and contains access to the different main modules of the application.
  - Shortcuts to the main modules: The shortcuts are located at the top right of the interface.
  - Other components: Search widget, navigation, satellite...

- **Navigation and display views**

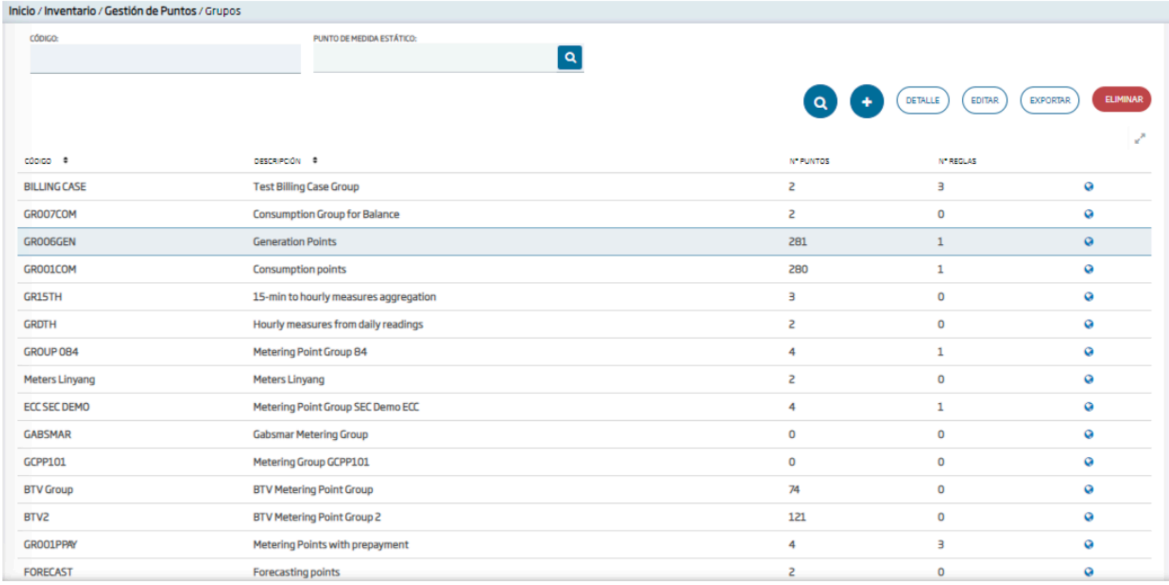
  These windows and visualisation systems are responsible for representing the entities and measurement data required by the user. A representative example of these windows is shown in Figure 11 and consists of three elements:

  - Search forms, to filter the list of items displayed in the main results table according to different criteria.
  - Main results table, where the list of elements that make up the entity will be displayed. The order of these tables is configurable.
  - Action buttons, to perform different actions on the displayed and selected elements.

2.4.1.2. Inventory Management

This functionality will manage the main system entities that are part of the system, together with their attributes and relationships. This module manages the master data or entities necessary for the general operation of the system. The OUM inventory and the entities that are part of it are divided into two main categories:

- **Main Entities.** They are defined as data that are related to the energy network and business infrastructure (metering points, equipment or customers), which may undergo changes due to the increase or reduction of the equipment installed in the field, or to possible modifications in its configuration.



Inicio / Inventario / Gestión de Puntos / Grupos

CÓDIGO:

PUNTO DE MEDIDA ESTÁTICO:

CÓDIGO	DESCRIPCIÓN	Nº PUNTOS	Nº REGLAS	
BILLING CASE	Test Billing Case Group	2	3	🌐
GR007COM	Consumption Group for Balance	2	0	🌐
GR006GEN	Generation Points	281	1	🌐
GR001COM	Consumption points	280	1	🌐
GR15TH	15-min to hourly measures aggregation	3	0	🌐
GRDTH	Hourly measures from daily readings	2	0	🌐
GROUP 084	Metering Point Group 84	4	1	🌐
Meters Linyang	Meters Linyang	2	0	🌐
ECC SEC DEMO	Metering Point Group SEC Demo ECC	4	1	🌐
GABSMAR	Gabsmar Metering Group	0	0	🌐
GCPP101	Metering Group GCPP101	0	0	🌐
BTV Group	BTV Metering Point Group	74	0	🌐
BTV2	BTV Metering Point Group 2	121	0	🌐
GR001PPAY	Metering Points with prepayment	4	3	🌐
FORECAST	Forecasting points	2	0	🌐

Figure 12: Inventory Display [Indra]

- **Master Data.** This type of data is in charge of supporting the previous entities and internal processes of the system.

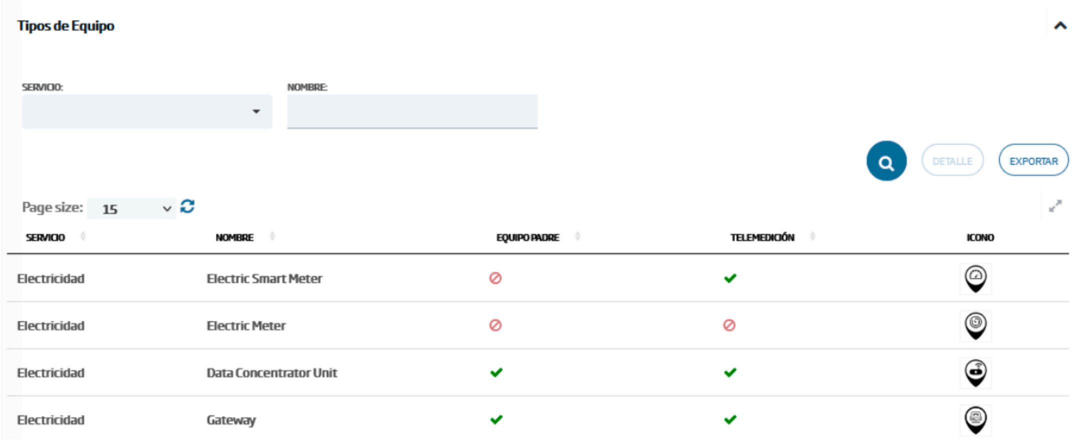
On the other hand, the following are the main inventory entities:

- **Equipment:** Physical devices deployed along the customer's energy network in order to interact with the customer's energy network.
- **Measurement Point:** This is the main entity of the system, to which the previous one is associated. It mainly represents a position in the network where the measurement data are taken.

This distinction between these two entities is of great importance. If the OUM system were directly referenced to the metering equipment, there would be a risk of losing and declassifying the information associated with it, since there is a possibility that this specific equipment could be updated, be decommissioned or have an incident. However, when this information is referenced to a "Measuring Point", where the data will normally be associated to this entity that will include a series of equipment in order to be able to keep a continuous traceability of the processed information.

For the management of equipment, which will be shown below, OUM proposes a classification into equipment types and models. In this way, the following windows can be distinguished:

- *Types of Equipment:*



Page size: 15

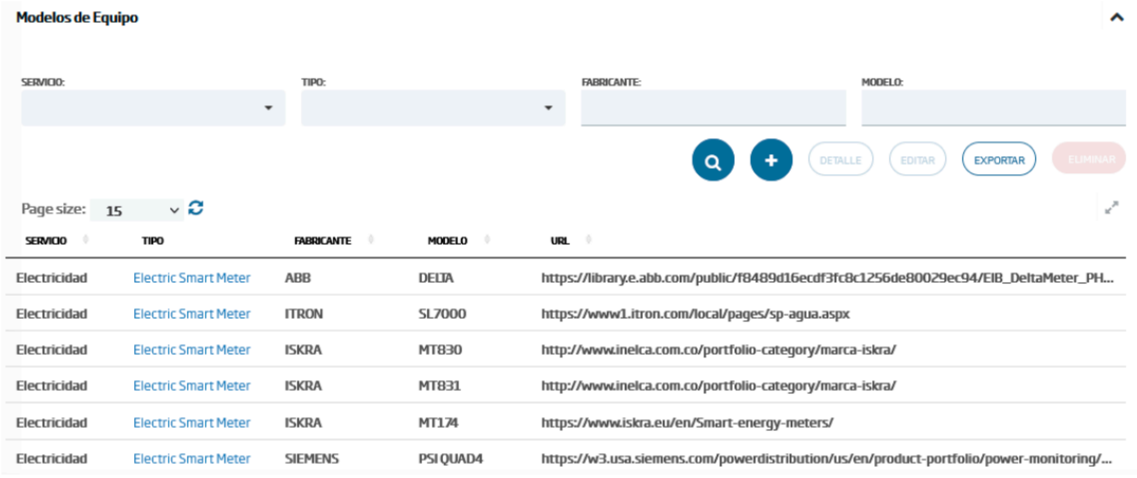
SERVICIO	NOMBRE	EQUIPO PADRE	TELEMEDICIÓN	ICONO
Electricidad	Electric Smart Meter	⊘	✓	📍
Electricidad	Electric Meter	⊘	⊘	📍
Electricidad	Data Concentrator Unit	✓	✓	📍
Electricidad	Gateway	✓	✓	📍

Figure 13: Remote Equipment Display [Indra]

Each type of equipment will have its corresponding attributes or properties, which will be inherited by the models or equipment corresponding to these types. Those that indicate the telemetering capability of the equipment type, such as Smart Meters, and the parent equipment type within a 1 to n topology, stand out.

- *Equipment models:*

This section distinguishes equipment by manufacturer according to the multi-manufacturer and multi-protocol feature that distinguishes the OUM tool.



SERVICIO	TIPO	FABRICANTE	MODELO	URL
Electricidad	Electric Smart Meter	ABB	DELTA	https://library.e.abb.com/public/f8489d16ecdf3fc8c1256de80029ec94/EIB_DeltaMeter_PH...
Electricidad	Electric Smart Meter	ITRON	SL7000	https://www1.itron.com/local/pages/sp-agua.aspx
Electricidad	Electric Smart Meter	ISKRA	MT830	http://www.inelca.com.co/portfolio-category/marca-iskra/
Electricidad	Electric Smart Meter	ISKRA	MT831	http://www.inelca.com.co/portfolio-category/marca-iskra/
Electricidad	Electric Smart Meter	ISKRA	MT174	https://www.iskra.eu/en/Smart-energy-meters/
Electricidad	Electric Smart Meter	SIEMENS	PSI QUAD4	https://w3.usa.siemens.com/powerdistribution/us/en/product-portfolio/power-monitoring/...

Figure 14: Remote Equipment Models Registered [Indra]

As mentioned above, each piece of equipment registered in the system is usually associated with the corresponding **measurement point**, although this relationship is not mandatory, especially in circumstances of equipment decommissioning or intermediate equipment. Therefore, the different equipment will have a set of properties or attributes, which will depend on their nature and typology.

- <u>Identification and Characterisation Properties: </u>Holds properties of the associated type and model, codes, serial numbers, or associations.

- <u>Telemetering Properties: Within the </u>telemetering configuration the equipment includes the following fundamental properties for interaction with the equipment through communication protocols and technologies. These properties are divided into two:

  - <u>Type of measurement: </u>Represents whether the communication with the field device is done directly or indirectly.
  - <u>Communications:</u> Types of technologies such as GPRS or GSM/RTC and associated properties.

These different measuring devices will have different **statuses, in order** to identify the status of the device and to establish a history. Likewise, these statuses establish a communication flow containing the following elements:

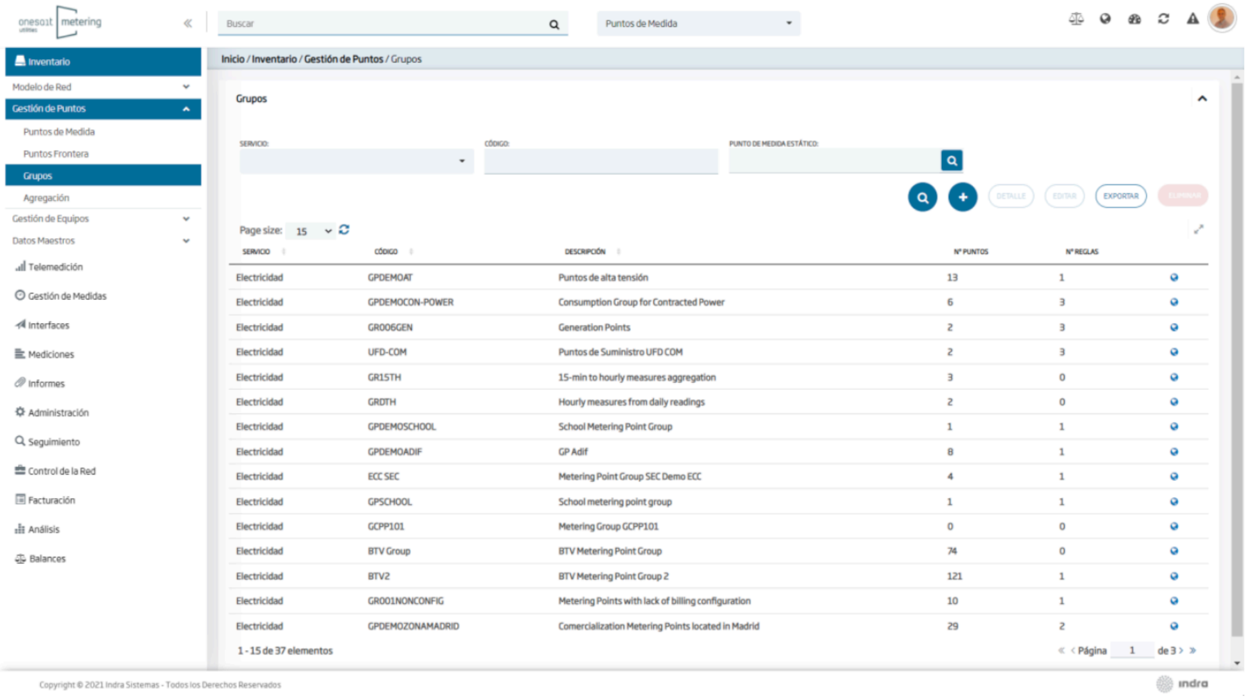
- Discovered: Equipment detected and identified in the field.
- Pre-loaded: Equipment with the corresponding identification and configuration information sent from the asset management system not put into production.
- Active: Active or operational equipment, with its complete configuration.
- Test Mode: Equipment in test mode.
- Inactive: Equipment temporarily inactive.

Therefore, OUM measurement data, regardless of the origin, shall always be associated to a metering point which shall be classified according to its scope of application and the activity it performs within the scope referred to above and belonging to the energy network.

These measuring points have different types of properties, of which the following are the main ones:

- Identification and characterisation properties, such as its coordinates, client, owner or calendar.
- Characteristics of the measuring point according to its classification.
- VEE configuration of algorithms or processes applied to ensure their certification and completeness in addition to the ranking or priority of origin for the calculation of the best value.
- Measurement of data such as the association of equipment according to the respective validity.

On the other hand, the Measurement Points can be grouped into what are known as Point Groups. The purpose of these groupings is to be able to massively use the whole set of points to perform certain operations, such as assigning access permissions per group or applying a business process to all the points in the group.



SERVICIO	CÓDIGO	DESCRIPCIÓN	Nº PUNTOS	Nº REGLAS
Electricidad	GPDEMOAT	Puntos de alta tensión	13	1
Electricidad	GPDEMOCON-POWER	Consumption Group for Contracted Power	6	3
Electricidad	GROO6GEN	Generation Points	2	3
Electricidad	UFD-COM	Puntos de Suministro UFD COM	2	3
Electricidad	GR15TH	15-min to hourly measures aggregation	3	0
Electricidad	GRDTH	Hourly measures from daily readings	2	0
Electricidad	GPDEMOSCHOOL	School Metering Point Group	1	1
Electricidad	GPDEMOADIF	GP Adif	8	1
Electricidad	ECC SEC	Metering Point Group SEC Demo ECC	4	1
Electricidad	GPSCHOOL	School metering point group	1	1
Electricidad	GCPP101	Metering Group GCPP101	0	0
Electricidad	BTV Group	BTV Metering Point Group	74	0
Electricidad	BTV2	BTV Metering Point Group 2	121	1
Electricidad	GROO1NONCONFIG	Metering Points with lack of billing configuration	10	1
Electricidad	GPDEMOZONAMADRID	Comercialization Metering Points located in Madrid	29	2

1 - 15 de 37 elementos

Página 1 de 3

Figure 15: Measuring Points Registered [Indra]

These groups can be formed according to any criteria the user deems appropriate, although they usually correspond to some criterion or business logic that defines a common treatment and relationship.

This allocation of Points to Groups can be done in two ways:

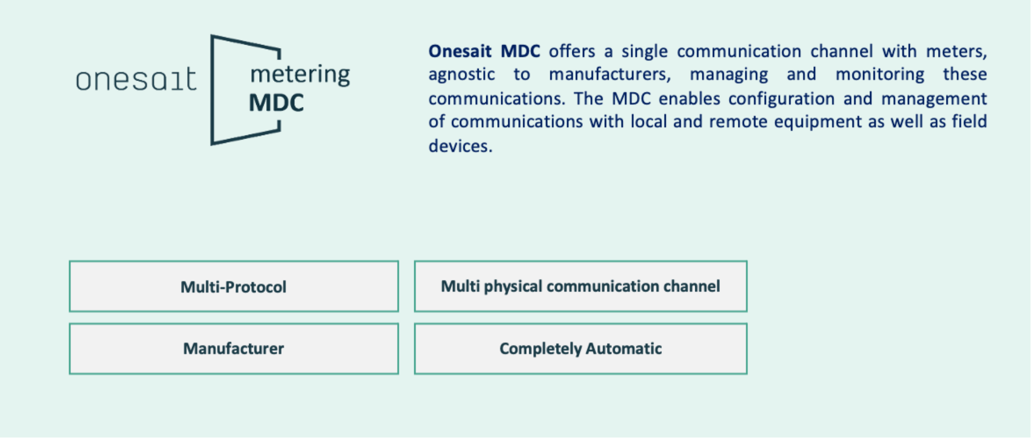
- Statically: Where it is carried out by direct and manual assignment of measurement points, either point to point or in bulk by uploading the corresponding file.

  - Dynamically: Where points are assigned based on the configuration of certain business rules. Different rules can be configured, which all associated points must comply with.

Importantly, there are also user-configurable mixed groups.

2.4.2. ONESAIT UTILITIES METERING MDC

Onesait Utilities Metering MDC is responsible for managing the data capture process, through a single communication channel with the metering equipment, managing and monitoring these communications. The MDC enables the configuration and management of communications with local and remote equipment as well as field devices. The Figure 16 shows the MDC functionalities described above.



Onesait MDC offers a single communication channel with meters, agnostic to manufacturers, managing and monitoring these communications. The MDC enables configuration and management of communications with local and remote equipment as well as field devices.

Figure 16: MDC description [14].

The logical implementation process of OUM is based on the following sections:

- <u>Inventory and communications configuration</u>
- <u>Data collection</u>: This acquisition can be done manually or programmed data acquisition through various communication technologies.
- <u>Data storage</u>: Captured data is stored in the database, ready for export.

- Data export: Involves the exchange of data with other corporate systems through configured files.

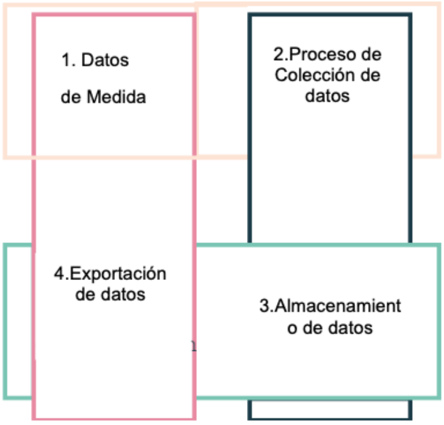


Figure 17: Onesait Utilities Metering Operational Flow [14].

On the other hand, the MDC data capture module has several features that differentiate it in the market and give the OUM solution a competitive advantage over other products on the market.

- It is a **multi-protocol** system, thanks to its protocol architecture it is able to manage equipment using different protocols and versions of these protocols.
- Likewise, it is capable of managing **any type of measuring equipment**, of any brand, manufacturer and corresponding model.
- It also has **telemetering and remote management** capabilities, i.e. it can both retrieve the information available in the meters and remotely manage them.
- OUM has the capability to manage any type of equipment, which also offers the ability to define and operate over different communications **topologies.**
  - 1-to-1 or p2p topologies
  - Topologies 1 to n

- The OUM information model within the inventory and measurement data management consists of the association of the different devices or equipment to the measurement points of the system, and the measurement data is stored on this entity.

Regarding telemetering inventory management, the OUM tool offers the views represented in the following figure:

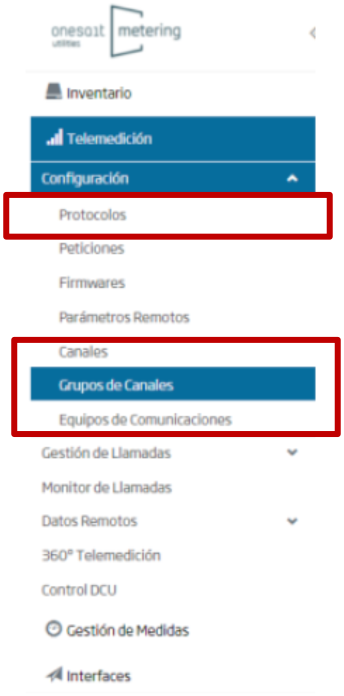


Figure 18: Telemetry Inventory Menu [Indra]

- Protocols: Represents the different protocols used within the system for interaction with connector-based equipment.
- Remote communications equipment: Represents the different remote communications equipment installed in the different devices in the field.

- Channels: Represents the different local modems or channels required for interaction with the corresponding field equipment.
- Channel groups: Allows different channels to be grouped into sets that can be assigned to ECRs, and thus to the corresponding equipment.

2.2.2.2. Call Management

Once the inventory of assets and communications has been configured, the processes and details of the interaction with these assets are configured through the Call Management functionality. This section will briefly describe the functionality and main features of this call management module.

First of all, the concept of requests and commands must be highlighted. Requests represent the different requests that can be executed on the devices configured within the system that have telemetry capability.

Such requests can be classified according to the type of target equipment:

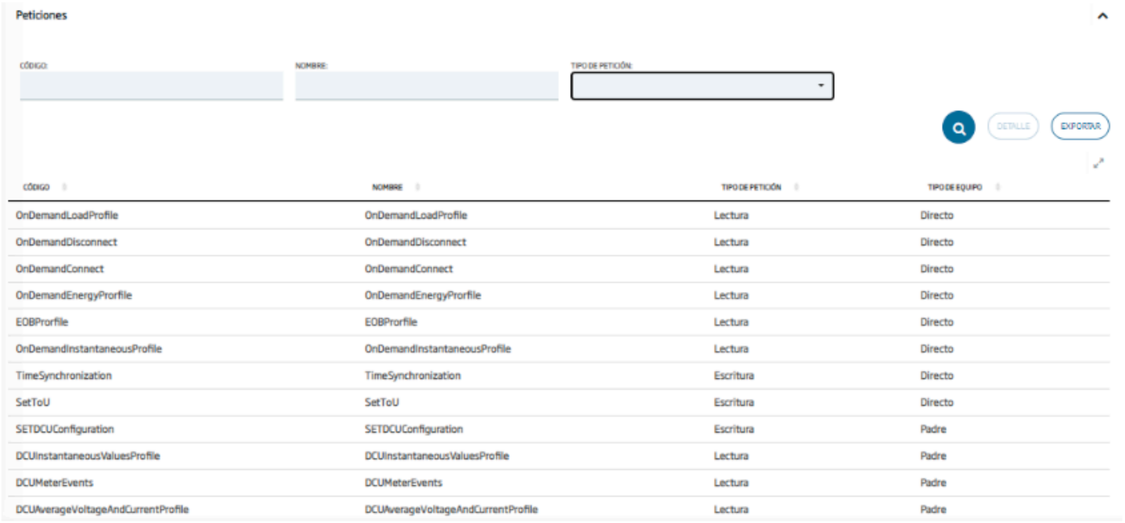
- Parents: Requests addressed to parent machines, in topologies from 1 to n.
- Endpoints: Requests oriented to end equipment, which may depend on a DCU (1 to n) or not (p2p), represented by Smart Meter in the EDP project.

Similarly, petitions can also be classified according to the form or impact of the petitions on these devices:

- Reading (telemetry): Requests oriented to retrieve information, both metering and configuration data.
- Writing (remote management): Requests aimed at the operation of these devices, in order to modify their parameters and operation.

Each of the requests may be made up of different parameters that allow the configuration and personalisation of the requests.

It is important to highlight that it is precisely from this possibility of configuring requests that the functional proposal that will be put forward in the development of this project is born.



Peticiones

CÓDIGO:

NOMBRE:

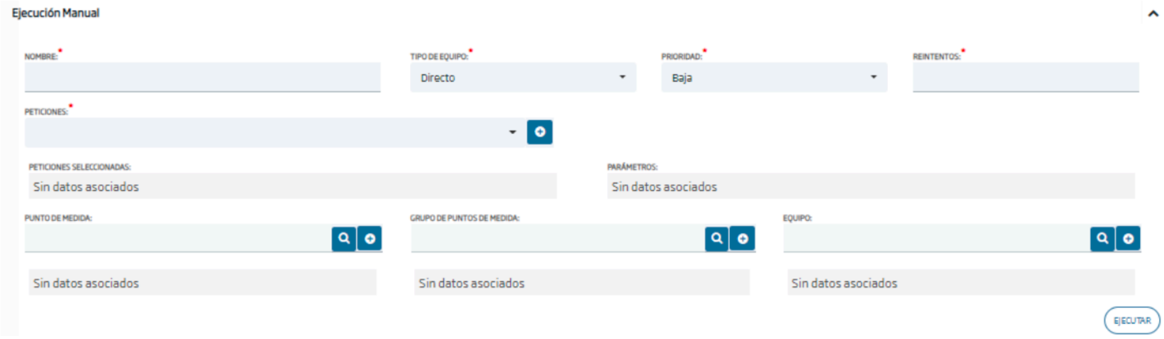
TIPO DE PETICIÓN:

CÓDIGO	NOMBRE	TIPO DE PETICIÓN	TIPO DE EQUIPO
OnDemandLoadProfile	OnDemandLoadProfile	Lectura	Directo
OnDemandDisconnect	OnDemandDisconnect	Lectura	Directo
OnDemandConnect	OnDemandConnect	Lectura	Directo
OnDemandEnergyProrfile	OnDemandEnergyProrfile	Lectura	Directo
EOBProrfile	EOBProrfile	Lectura	Directo
OnDemandInstantaneousProfile	OnDemandInstantaneousProfile	Lectura	Directo
TimeSynchronization	TimeSynchronization	Escritura	Directo
SetToU	SetToU	Escritura	Directo
SETDCUConfiguration	SETDCUConfiguration	Escritura	Padre
DCUInstantaneousValuesProfile	DCUInstantaneousValuesProfile	Lectura	Padre
DCUMeterEvents	DCUMeterEvents	Lectura	Padre
DCUAverageVoltageAndCurrentProfile	DCUAverageVoltageAndCurrentProfile	Lectura	Padre

Figure 19: Petition Interface [Indra]

2.2.2.3. Configuration and Execution of Requests

The OUM environment offers the possibility of configuring and executing requests manually on demand, on the different equipment or measurement points required. These requests can be executed individually or en masse on several groups of equipment or measuring points. In addition, any type or set of requests can be selected and the corresponding parameters for each of these requests can be configured.



Ejecución Manual

NOMBRE:*

TIPO DE EQUIPO:* Directo

PRIORIDAD:* Baja

REINTENTOS:*

PETICIONES:*

PETICIONES SELECCIONADAS: Sin datos asociados

PARÁMETROS: Sin datos asociados

PUNTO DE MEDIDA: Sin datos asociados

GRUPO DE PUNTOS DE MEDIDA: Sin datos asociados

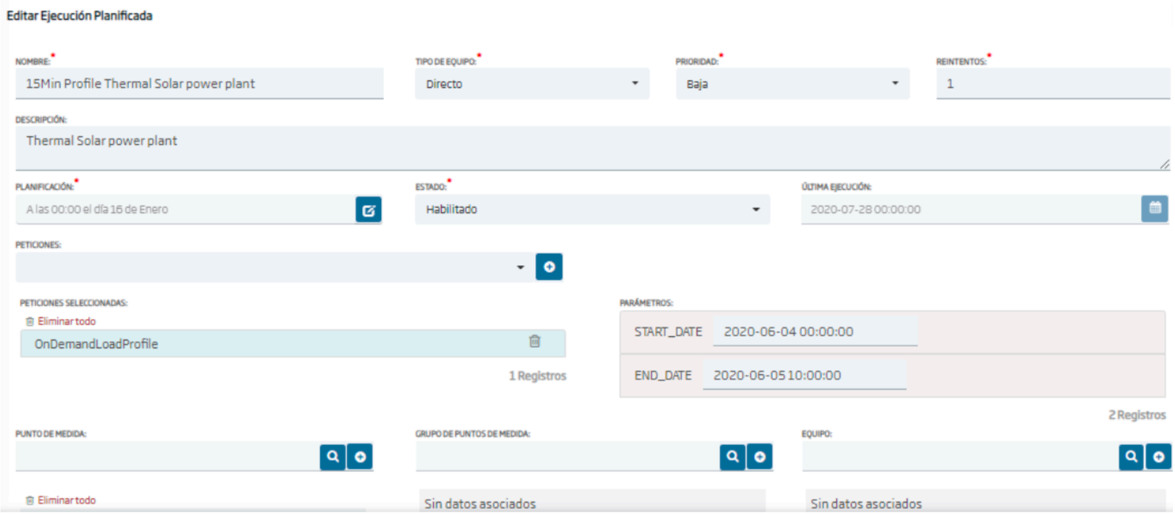
EQUIPO: Sin datos asociados

Figure 20: Request Manual Execution Interface [Indra]

Figure 20 represents the menu of options for launching a manual request. As can be seen, the user assigns a name to the process identifier, selects the type of equipment, a priority

(low, medium or high) and a number of retries to try to communicate with the selected measurement points. Then you can choose which measuring points or which devices you want to communicate with and finally select the desired request or set of requests to launch. <u>In this way, a manual execution is generated, which will be the first step that will appear in the flowcharts represented in this project.</u>

As with manual execution, the tool is able to configure the execution of any type of request on different elements automatically according to the indicated schedule. All the properties of the manual execution apply to the planned execution.



Editar Ejecución Planificada

NOMBRE: 15Min Profile Thermal Solar power plant

TIPO DE EQUIPO: Directo

PRIORIDAD: Baja

REINTENTOS: 1

DESCRIPCIÓN: Thermal Solar power plant

PLANIFICACIÓN: A las 00:00 el día 16 de Enero

ESTADO: Habilitado

ÚLTIMA EJECUCIÓN: 2020-07-28 00:00:00

PETICIONES:

PETICIONES SELECCIONADAS:

Eliminar todo

OnDemandLoadProfile

1 Registros

PARÁMETROS:

START_DATE	2020-06-04 00:00:00
END_DATE	2020-06-05 10:00:00

2 Registros

PUNTO DE MEDIDA:

Eliminar todo

GRUPO DE PUNTOS DE MEDIDA:

Sin datos asociados

EQUIPO:

Sin datos asociados

Figure 21: Planned Execution Request [Indra]

In contrast to manual execution, these configurations are stored in the system and are activated automatically according to the specified schedule.

Finally, regarding the generation of tasks, it is important to highlight that these are generated in time and execution, and correspond to each request and measurement point or corresponding equipment, depending on the configuration. In other words, if we have a set of requests and a set of equipment in the launch configuration, a task is generated for each equipment and request, which will be executed individually, although within the same communication or call.

2.2.2.4. Interaction Between Teams

The interaction with the different field measurement equipment can be done in several ways, depending on the characteristics and configuration of the field measurement equipment:

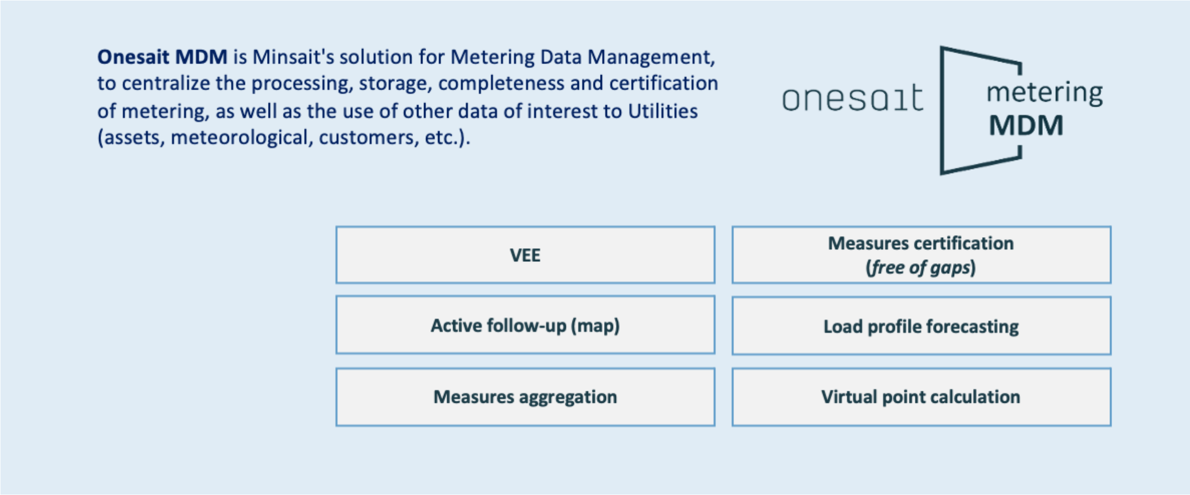
- Synchronous: Typical of 1-to-1 topologies, in the same communication the request and the response are sent, waiting for completion and keeping the channel open.
- Asynchronous: Typical of 1 to n topologies with PRIME protocol, where it is the field device (DCU) that sends the information, after configuration in the same.

- Asynchronous on demand: On the same elements and configuration as above, in this case the request is sent from OUM, and the configuration is closed, returning the response asynchronously.

All these variants shall be considered in the system, depending on the configuration and capacity of the related equipment.

2.2.3. ONESAIT UTILITIES METERING MDM

Onesait Utilities Metering MDM has been conceived to manage and process the data collected by the system to centralise the processing, storage, completeness, and certification of measurements, as well as the use of other data of interest to Utilities.



Onesait MDM is Minsait's solution for Metering Data Management, to centralize the processing, storage, completeness and certification of metering, as well as the use of other data of interest to Utilities (assets, meteorological, customers, etc.).

Figure 22: MDM description [14].

The measurement management module represents the most representative part of what is considered an MDM module or functionality, which includes VEE (Validation, Editing and Estimation). This functionality includes the configuration of the measurement cycle of each measurement point or group of points.

We can define the Measurement Cycle of a given Measurement Point as the different phases through which the measurement data of a given Measurement Point pass and which define their treatment. The BTV calculation is part of this treatment or Measurement Cycle, and the main measurement data of a given measurement point pass through this phase.

This processing cycle is independent for each measurement point according to its characteristics and configurable by the user. To document this cycle, it is important to start by defining what a measurement data is.

A measurement data is any type of data that represents the value of a given quantity. The main types of measurement data taken into account in the OUM tool are listed below:

- Absolute load profile readings or measurements
- Curve measurements or load profiles
- Closing records or billing measure

On the other hand, there are secondary measurement data types which are divided into events which represent an anomaly recorded by a meter, and instantaneous values which are instrumentation measurements that measure instantaneous electrical quantities.

There are two stages in the system for processing the measurement data discussed above:

- **Origin**: Gross measures entering the system from the different types of origins.
- **Optimal or BTV:** The optimal measurement or BTV is the best value of the measurement, in the case of multiple sources of the same measurement.

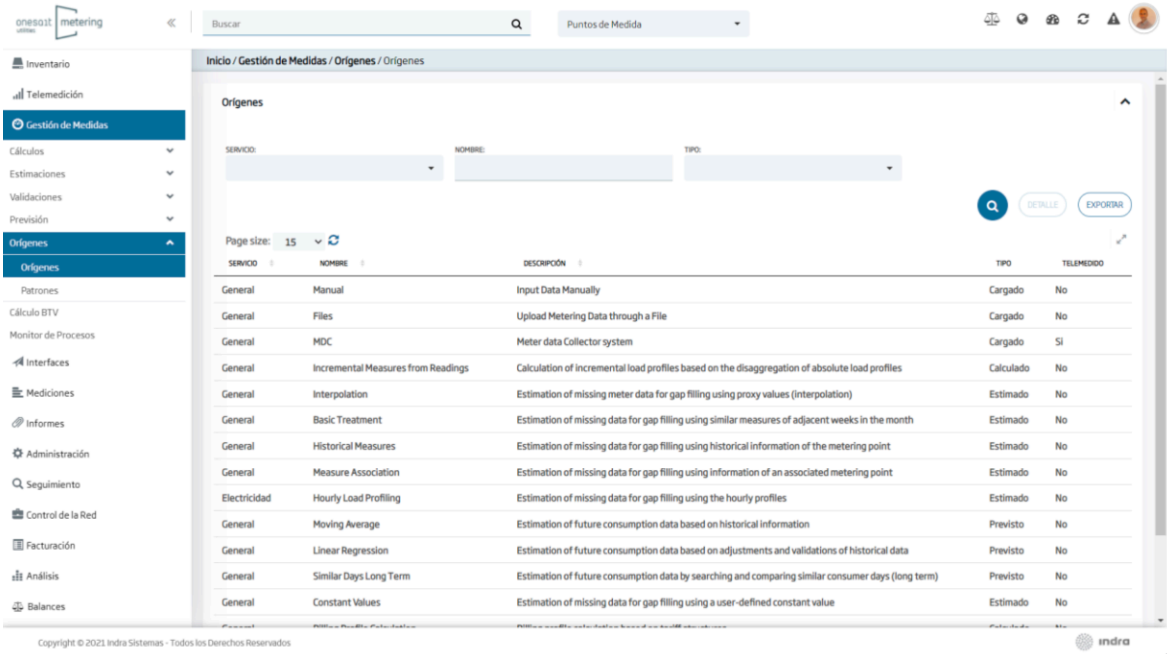
Each measurement data collected by the system has a specific origin or source associated with it. Therefore, data will be registered in the system according to these origins which, depending on the project and its application, will be assigned a certain weight so that the system will subsequently be able to discern whether a measurement is valid or not, and if it is valid, whether it is more or less reliable than another data source.

Note that these measurement data can be marked or qualified depending on the validations applied to them; Valid, invalid and discarded.

OUM MDM is a multi-source system, as it is able to store and manage measurement data from different sources, which means that the same data can come from different sources of information.

These origins can be classified according to the following topology:

- **Actual or Loaded:** These are origins from a source external to the system, such as a commercial system or an MDC system.
- **Calculated:** The origins of this type are the different calculation methods of the system, e.g. a measurement generated by the calculation method of hourly aggregated measurements.
- **Estimate:** Refers to the different estimation methods of the system, e.g. the origin of a measurement generated by the interpolation estimation method.

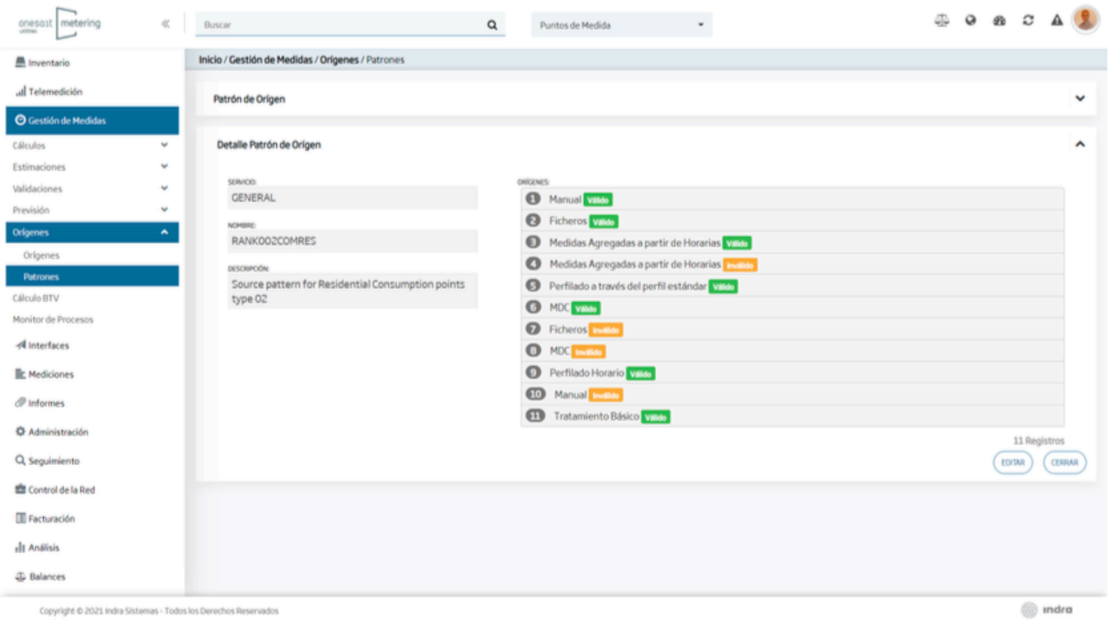


SERVICIO	NOMBRE	DESCRIPCIÓN	TIPO	TELEMEDIDO
General	Manual	Input Data Manually	Cargado	No
General	Files	Upload Metering Data through a File	Cargado	No
General	MDC	Meter data Collector system	Cargado	Si
General	Incremental Measures from Readings	Calculation of incremental load profiles based on the disaggregation of absolute load profiles	Calculado	No
General	Interpolation	Estimation of missing meter data for gap filling using proxy values (interpolation)	Estimado	No
General	Basic Treatment	Estimation of missing data for gap filling using similar measures of adjacent weeks in the month	Estimado	No
General	Historical Measures	Estimation of missing data for gap filling using historical information of the metering point	Estimado	No
General	Measure Association	Estimation of missing data for gap filling using information of an associated metering point	Estimado	No
Electricidad	Hourly Load Profiling	Estimation of missing data for gap filling using the hourly profiles	Estimado	No
General	Moving Average	Estimation of future consumption data based on historical information	Previsto	No
General	Linear Regression	Estimation of future consumption data based on adjustments and validations of historical data	Previsto	No
General	Similar Days Long Term	Estimation of future consumption data by searching and comparing similar consumer days (long term)	Previsto	No
General	Constant Values	Estimation of missing data for gap filling using a user-defined constant value	Estimado	No

Figure 23: Group Measurement Data Display [Indra]

Figure 23 represents the interface where the origin of each data or group of measurement data can be displayed. This screen shows the service, the reference or name of this source and a brief description of the data in question. In addition, the status of the data can be viewed.

On the other hand, it is necessary to define the Ranking of Origins, which is a set of origins ordered by priority. In addition, each of these origins and priority will have a certain status or measurement qualification associated with it. The following figure shows the view of the Source Pattern menu.



Inicio / Gestión de Medidas / Orígenes / Patrones

Patrón de Origen

Detalle Patrón de Origen

SERVICIO: GENERAL

NOMBRE: RANK002COMRES

DESCRIPCIÓN: Source pattern for Residential Consumption points type 02

ORÍGENES:

1. Manual Válido
2. Ficheros Válido
3. Medidas Agregadas a partir de Horarias Válido
4. Medidas Agregadas a partir de Horarias Inválido
5. Perfilado a través del perfil estándar Válido
6. MDC Válido
7. Ficheros Inválido
8. MDC Inválido
9. Perfilado Horario Válido
10. Manual Inválido
11. Tratamiento Básico Válido

11 Registros

Copyright © 2021 Indra Sistemas - Todos los Derechos Reservados

Figure 24: Measurement Qualification [Indra]

Once the main entities involved and having an important role in the MDM system have been defined, the different phases through which the measurement data pass, i.e. the measurement life cycle, will be detailed.

In this sense, La Figure 25 represents the life cycle of the measurement, from the time it enters the system until it leaves the system. The data is collected from two measurement sources and then undergoes a first set of validations, so that the system links the data to each source and validation resource. By configuration, each point has a hierarchy based on precisely that pair of attributes (source, validation result), and perhaps the version of the data (if versioning is enabled). Based on this hierarchy (ranking), the system chooses the best available value (the first in the ranking), marking it as BTV (Best Time Value), the BTVs are processed with new validations and specific estimations, to fill all the existing gaps. Once the BTV processes are done, the data is ready to be processed, e.g. to calculate billing determinants, to be included in reports or to be sent to other downstream systems.

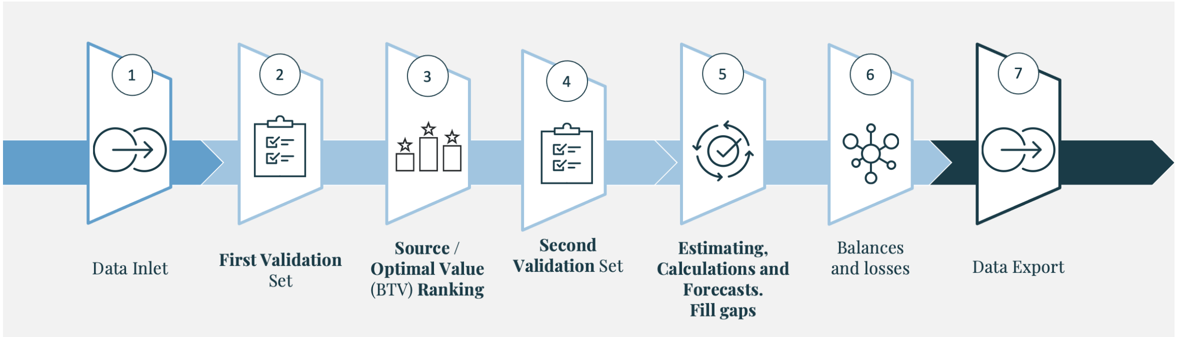


Figure 25: Measure Life Cycle [14]

On the other hand, we can define the measurement cycle or measurement cycle as the set of phases through which the different measurement data of the system pass. This cycle will depend on the nature and characteristics of each measurement point, according to which the user will have established the appropriate configuration of this cycle, i.e. the measurement data processing cycle for each point is configured and parameterised by the user according to his needs.

The Figure 26 represents the measurement cycle as the set of phases through which the different measurement data of the system pass. The user must configure and parameterise the characteristics of each measurement point according to his needs in order to define the behaviour of this measurement cycle.

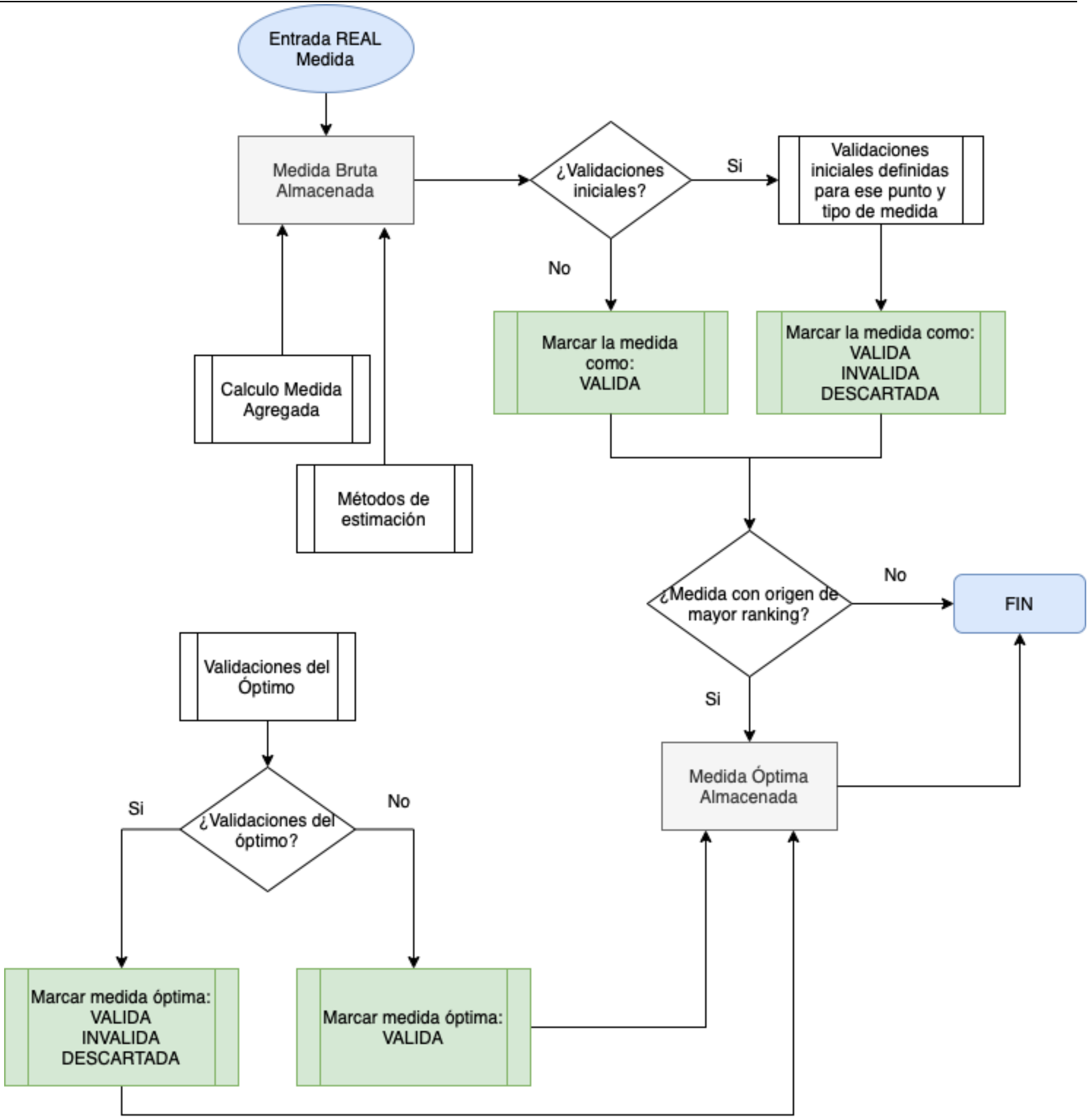

Entrada REAL Medida
Medida Bruta Almacenada
¿Validaciones iniciales?
Si
Validaciones iniciales definidas para ese punto y tipo de medida
No
Marcar la medida como: VALIDA
Marcar la medida como: VALIDA INVALIDA DESCARTADA
Calculo Medida Agregada
Métodos de estimación
¿Medida con origen de mayor ranking?
No
FIN
Si
Medida Óptima Almacenada
Validaciones del Óptimo
Si
¿Validaciones del óptimo?
No
Marcar medida óptima: VALIDA INVALIDA DESCARTADA
Marcar medida óptima: VALIDA

Figure 26: Measurement cycle

The integration of these two modules of the Onesait Utilities Metering (MDC & MDM) solution will be the key factor in providing users with a complete, quality service that will enable them to make the most of their data and provide complete and transparent traceability of their installations.

CHAPTER 3. JOB DEFINITION

4.1. JUSTIFICATION

One of the problems that the technical teams of companies dedicated to providing IT solutions usually face is that it is common that, despite the fact that there is a lot of programming work done behind the scenes, nothing has been documented beforehand. This means that when it is necessary to carry out maintenance, update functionalities, adapt the tool to a new client... it is difficult for someone other than the programmer of the previous code to take the reins of the work, or simply that it has been stopped for a long time and not even the programmer himself is clear about what he did in his day, which translates into a huge inefficiency as time is wasted in relearning what has already been done. For this reason, it has been decided to document the main features of these communicative processes in flow chart format.

This document has been prepared in a format that corresponds to a functional analysis of the technologies involved and their integration, but at the same time has a technical component that allows the reader to use this project as if it were a user manual where they can find a clear structure of the communication processes and the main interactions between measurement equipment to carry out efficient and organised low-level programming of the different OUM modules in order to achieve the final objective, which is none other than to ensure that the customer's expectations are met and that the project is profitable for Minsait.

In summary, the value of this document, apart from its nature as a detailed functional analysis, is that it provides the necessary information that a programming technician who has no experience working with the OUM tool can use as a guide to structure his scripts and have a clear view of the complex interaction of the equipment involved.

4.2. METHODOLOGY

For the satisfactory achievement of the objectives set out in this project, a path marked by different milestones has been defined, which can be broadly summarised in chronological order as follows:

1. First, an extensive study of the state of the art in the world of Smart Meetering was carried out, in which the different existing technologies, the different communication protocols and the main challenges faced by this industry were extensively investigated.

2. Next, the communication technologies that were to be applied in the interests of this project were discussed in depth. That is, the PRIME protocol and more specifically the STG-DC specification in its version 3.4. It was necessary to study the different orders or requests offered by this specification and to see how the different measuring equipment involved in the remote management system communicated.

3. Once the protocols to be implemented were understood, the next step was to study the operation of the Onesait Utilities Meetring (OUM) tool, for which it was necessary to learn how to use the user interface, differentiate the main blocks that make up the system, and be able to keep track of the communications offered by this system.

4. To conclude the preparation stage, the generic structure of the OUM tool was analysed at the lowest level. It is necessary to know how the information records are structured in the database in order to understand the logic of interaction between measurement equipment to implement the protocol.

5. The development stage started by making a complete list of the functionalities to be implemented in the EDP project by cross-checking the possibilities offered by the OUM tool with the customer's requirements.

6. A mapping was then performed to link these functionalities in the form of commands to the request references specified in the STG-DC protocol.

7. The different types of orders were then structured according to their nature and their particular communications processes. This classification is summarised in Annex III of this document.

8. Finally, they started to create the one-by-one flowcharts that will help the technical team to programme these requests and integrate them into the tool, and which are the main value of this document.

9. Finally, a case study of a real execution was carried out to extract all the records involved in the database. This example serves to verify that the flowcharts are structured correctly and helps the reader to understand in a more detailed way the communication process that takes place and how to monitor it correctly.

CHAPTER 4. SYSTEM/MODEL DEVELOPED

This chapter defines the different requests to the equipment that will be registered in the system, both for remote metering and remote management. The communication flow diagrams that make up the main objective of the project will also be presented.

As mentioned above, each of these requests will have the objective of requesting or sending information to the corresponding meter(s). Each request will have a set of parameters that must be entered through a manual execution or a scheduler.

First of all, in a first subsection, a mapping of the cross requests with the STG-DC protocol will be provided, which serves as a list of all these requests that will be defined immediately afterwards.

The following sub-section will contain all information related to flowcharts.

4.1. SYSTEM ANALYSIS

Table 2 represents the mapping of the orders to be integrated in the tool with the corresponding reference given in the STG-DC protocol.

Order	Request STG-DC	Order	Request STG-DC	Order	Request STG-DC
Get Date Time	**S06**	Get Meter Events From Last	**S09**	Get Current Billing Values	**S27**
Get Daylight Saving Time Dates	**S06**	Get Instantaneous Values	**S01**	Get Concentrator Events	**S17**
Get Equipment Parameters	**S06**	Get Instantaneous Values Between Dates	**S01**	Get Concentrator Events Between Dates	**S17**
Get Load Profile	**S02**	Get Instantaneous Values From Last	**S01**	Get Cocentrator Events From Last	**S17**
Get Load Profile Between Dates	**S02**	Get Advanced Instantaneous Values	**S21**	Get Firmware Update	**S19**
Get Load Profile From Last	**S02**	Get Advanced Instanteneous Values Between Dates	**S21**	Connect	**B03**
Get Load Profile Gaps	**S02**	Get Advanced Instantaneous Values From Last	**S21**	Disconnect	**B03**
Get Load Profile Gaps Between Dates	**S02**	Get Billing Profile	**S04/S05/S5B**	Set Equipment Parameter	**B09/B9b/ B07**
Get Absolute Load Profile	**S03**	Get Billing Profile Between Dates	**S04/S05/S5B**	Set Daylight Saving Time Dates	**B09/B9b/ B07**
Get Absolute Load Profile Between Dates	**S03**	Get Billing Profile From Last	**S04/S05/S5B**	Synchronization	**B09**
Get Absolute Load Profile From Last	**S03**	Get Billing Profile Gaps	**S04/S05/S5B**	Firmware Update	**B05 /B08**
Get Absolute Load Profile Gaps	**S03**	Get Billing Profile Gaps Between Dates	**S04/S05/S5B**	Set Tariff Structure	**B04**
Get Absolute Load Profile Gaps Between Dates	**S03**	Get Traffic Structure	**S23**	Set Contracted Power	**B04/B02**
Get Meter Events	**S09**	Get Contracted Power	**S23**	Spontaneous Events	**S13/S15**
Get Meter Events Between Dates	**S09**	Get Connection Status	**S18**		

Table 2: Orders Mapping

Next, all the orders collected in the previous mapping are defined.

FINAL EQUIPMENT TELEMEASUREMENT

Get Date Time

This request sends a command to any meter to return the date and time it is working with. In case there is a desynchronisation with respect to the system time, the latter must send a synchronisation request to the meter to modify it.

Parameters:

This request lacks specific parameters.

Get Daylight Saving Time Dates

The time change date request allows you to send a command on any meter to return the date and time of the time change days you are working with. It is essential that the system knows these dates in order to properly record and process the measurement data. In the event that the dates returned are not as expected, a request can be made from the system to modify the time change dates.

Parameters:

This request lacks specific parameters.

Get Equipment Parameters

This request sends an order on any meter to return a set of parameters of interest of the equipment in reading mode. The values received from the meters must be properly mapped to their corresponding attributes and the request must be executed with all the attributes requested by the specification.

Parameters:

This request lacks specific parameters

- **Load Profile**

The system requests load profiles from the equipment.

Get Load Profile

This request sends a command on any meter to return all stored load profile data.

Parameters:

This request lacks specific parameters

Get Load Profile Between Dates

This request sends a command to any meter to return all load profile data it has stored between the two user-defined dates.

Parameters:

- **Start Date:** Date from which the load profile data will be requested.
- **End Date:** Date until which the load profile data will be requested.

Get Load Profile from Last

This request sends a command to any meter to return all load profile data it has stored since the last measurement recorded in the system.

Parameters:

This request lacks specific parameters

Get Load Profile Gaps

This request sends a command on any meter to return all the profile data of the gaps it finds in the respective curves. The system will find all the gaps from the user-defined n days back, and retrieve the load profiles of the gaps found.

Parameters:

- **Days:** Days back from the day of the request in which gaps will be searched for and missing values will be requested.

Get Load Profile Gaps Between Dates

This request sends a command on any meter to return all the profile data for the gaps it finds in the respective curves. The system will find all the gaps between the dates selected by the user, and retrieve the missing load profiles.

Parameters:

- **Start Date:** Date from which gaps will be searched for, and missing load profile data will be requested.
- **End Date:** Date up to which gaps will be searched for, and missing load profile data will be requested.

- **Absolute Load Profile (Readings)**

The system requests absolute load profiles from the equipment.

Get Absolute Load Profile

This request sends a command on any meter to return all stored absolute load profile data.

Parameters:

This request lacks specific parameters

Get Absolute Load Profile Between Dates

This request sends a command on any meter to return all absolute load profile data it has stored between the two user-defined dates.

Parameters:

- **Start Date:** Date from which the absolute load profile data will be requested.
- **End Date:** Date until which the absolute load profile data will be requested.

Get Absolute Load Profile from Last

This request sends a command on any meter to return all absolute load profile data it has stored since the last measurement recorded in the system.

Parameters:

This request lacks specific parameters

Get Absolute Load Profile Gaps

This request sends a command on any meter to return all the absolute load profile data of the gaps it finds in the respective curves. The system will find all the gaps from the user-defined n days back, and retrieve the load profiles of the gaps found.

Parameters:

- **Days:** Days back from the day of the request in which the gaps will be found and missing values will be requested.

Get Absolute Load Profile Gaps Between Dates

This request sends a command on any meter to return all the profile data of the gaps it finds in the respective curves. The system will find all the gaps between the dates selected by the user, and retrieve the load profiles of the gaps found.

Parameters:

- **Start Date:** Date from which gaps will be searched for, and missing absolute load profile data will be requested.
- **End Date:** Date up to which gaps will be searched for, and missing absolute load profile data will be requested.

- **Events**

Get Meter Events

This request sends a command on any meter to return all stored events.

Parameters:

This request lacks specific parameters

Get Meter Events Between Dates

This request sends a command to any meter to return all stored events between two user-defined dates.

Parameters:

- **Start Date:** Date from which events will be requested.
- **End Date:** Date until which events will be requested.

Get Meter Events from Last

This request sends a command to any meter to return all stored events since the last event recorded in the system.

Parameters:

This request lacks specific parameters

- **Instantaneous Values**

Get Instantaneous Values

This request sends a command on any meter to return all stored instantaneous values.

Parameters:

This request lacks specific parameters

Get Instantaneous Values Between Dates

This request sends a command to any meter to return all the instantaneous values it has stored between the two user-defined dates.

Parameters:

- **Start Date:** Date from which the instantaneous values will be requested.
- **End Date:** Date until which the instantaneous values will be requested.

Get Instantaneous Values from Last

This request sends a command to any meter to return all stored events since the last instantaneous value recorded in the system.

Parameters:

This request lacks specific parameters

Get Advanced Instantaneous Values

This request sends a command to any meter to return all stored instantaneous values. This request will request the same data as the "Get Instantaneous Values" request plus additional instantaneous values typically found in multi-phase or supervisory equipment.

Parameters:

This request lacks specific parameters

Get Advanced Instantaneous Values Between Dates

This request sends a command to any meter to return all instantaneous values it has stored between the two user-defined dates. This request will request the same data as the "Get Instantaneous Values Between Dates" request plus additional instantaneous values typically found in multi-phase or supervisory equipment.

Parameters:

- **Start Date:** Date from which the instantaneous values will be requested.
- **End Date:** Date until which the instantaneous values will be requested.

Get Advanced Instantaneous Values from Last

This request sends a command to any meter to return all events it has stored since the last event recorded in the system. This request will request the same data as the "Get Instantaneous Values from Last" request plus additional instantaneous values typically found in multi-phase or supervisory equipment.

Parameters:

This request lacks specific parameters

- **Invoicing Records**

Get Billing Profile

This request sends an order on any meter to return all stored billing records.

Parameters:

This request lacks specific parameters

Get Billing Profile Between Dates

This request sends a command to any meter to return all billing records stored between the two user-defined dates.

Parameters:

- **Start Date:** Date from which billing records will be requested.
- **End Date:** Date up to which billing records will be requested.

Get Billing Profiles from Last

This request sends a command on any meter to return all billing records it has stored since the last event recorded in the system.

Parameters:

This request lacks specific parameters

Get Billing Profile Gaps

This request sends a command on any meter to return all the profile data of the gaps it finds in the respective curves. The system will find all the gaps from the user-defined n days back, and retrieve the load profiles of the gaps found.

Parameters:

- **Days:** Days back from the day of the request in which the gaps will be found and missing values will be requested.

Get Absolute Load Profile Gaps Between Dates

This request sends a command on any meter to return all the profile data of the gaps it finds in the respective curves. The system will find all the gaps from between the dates selected by the user, and retrieve the load profiles of the gaps found.
Change by billing.

Parameters:

- **Start Date:** Date (measurement date) from which the gaps will be searched for, and missing billing data will be requested.
- **End Date:** Date (measurement date) up to which gaps will be searched for, and missing billing data will be requested.

Get Tariff Structure

This request sends an order on any meter to return the tariff structure installed in the equipment. This request will retrieve among other things the totalisers associated to the tariff and the hourly discrimination of these.

Parameters:

This request lacks specific parameters.

Get Contracted Power

This request sends an order to any meter to return the contracted power at that meter.

Parameters:

This request lacks specific parameters.

- **Connection Status**

Get Connection Status

This request sends a command on any meter to return its current connectivity status.

Parameters:

This request lacks specific parameters.

INTERMEDIATE EQUIPMENT TELEMEDICIÓN

Get Current Billing Value

This request sends a synchronous character command on any intermediate equipment (DCU, Gateway...) to return the absolute values of the current register per contract and tariff period only for contract 1. It is composed of up to 6 tariff periods + total.

- **Timestamp:** Date and time of the requested readings.
- **Meter data reflecting.**
- **Events**

Get Concentrator Events

This request sends an order to any intermediate equipment (DCU, Gateway...) to return all the events it has stored.

Parameters:

This request lacks specific parameters

Get Concentrator Events Between Dates

This request sends an order to any intermediate equipment (DCU, Gateway...) to return all the events it has stored between the two dates defined by the user.

Parameters:

- **Start Date:** Date from which events will be requested.
- **End Date:** Date until which events will be requested.

Get Concentrator Meter Events from Last

This request sends an order to any intermediate equipment (DCU, Gateway...) to return all the events it has stored since the last event registered in the system.

Parameters:

This request lacks specific parameters

- **Firmware Update Configuration**

Get FTP DCU Firmware Update Configuration

This request sends a command to any intermediate equipment (DCU, Gateway...) to retrieve the firmware update schedule configured in the parent equipment.

Parameters:

This request lacks specific parameters.

Get FTP Meter Firmware Update Configuration

This request sends an order to any intermediate equipment (DCU, Gateway...) to retrieve the configured firmware update schedule of the equipment associated to it.

Parameters:

This request lacks specific parameters

FINAL EQUIPMENT TELEMANAGEMENT

Connect

This request sends a command on any equipment to connect the meter.

Parameters:

This request lacks specific parameters.

Disconnect

This request sends a command on any equipment to disconnect the meter.

Parameters:

This request lacks specific parameters.

Set Equipment Parameter

This request sends a command to any meter to define one or several parameters of interest of the equipment (writing).

Parameters:

This request lacks specific parameters.

Set Daylight Saving Time Dates

This request sends a command on any meter to set the Summer-Winter time changeover dates and vice versa.

- **Date W-S:** Date on which the equipment will switch to daylight saving time.
- **Date S-W:** Date on which the equipment will switch to winter time.

Synchronization

This request sends a command on any meter to modify the date and time it is working with and set the same date and time of the system.

Parameters:

This request lacks specific parameters.

- **Firmware update**

Get Firmware Update

This request sends a command on any meter to upgrade the meter's firmware to the version indicated by the user.

Parameters:

- **Start Date:** Number of the version to which the equipment is to be upgraded.

Get Firmware Update from File

This request sends a command on any meter to update the meter's firmware via the file uploaded by the user.

Parameters:

- **Firmware**: The file corresponding to the update to be performed shall be loaded.

Set Tariff Structure

This request sends an order to any meter to configure a new tariff structure. The user will be able to configure up to 3 tariffs per equipment, indicating the tariff code, the activation date and the calendar code to be used.

Parameters:

- **Tariff Structure Name C1:** Code of the first tariff configured.
- **Activation Date C1: Activation** date of the first tariff configured.
- **Calendar Name C1:** Calendar Code of the first tariff configured.
- **Tariff Structure Name C2:** Code of the second tariff configured.
- **Activation Date C2: Activation** date of the second tariff configured.
- **Calendar Name C2:** Calendar Code of the second tariff configured.
- **Tariff Structure Name C3:** Code of the third tariff configured.
- **Activation Date C3: Activation** date of the third tariff configured.

- **Calendar Name C3:** Calendar Code of the third tariff configured.

- **Contracted Power Configuration**

Set Contracted Power

This request sends an order to any meter to configure the contracted power. The user will be able to configure up to 6 contracted powers, depending on the number of hourly discrimination totalisers the equipment in question has.

Parameters:

- **Tariff Rate 1:** Contracted power in the first totaliser (kW).
- **Tariff Rate 2:** Contracted power in the second totaliser (kW).
- **Tariff Rate 3:** Contracted power in the third totaliser (kW).
- **Tariff Rate 4:** Contracted power in the fourth totaliser (kW).
- **Tariff Rate 5:** Contracted power in the fifth totaliser (kW).
- **Tariff Rate 6:** Contracted power in the sixth totaliser (kW).

4.2. IMPLEMENTATION

Once the orders to be integrated in the system for the EDP project have been presented and described, the next step is to break down the communications process that takes place between the different equipment and systems involved. To represent this process and the interactions between the different existing entities, flow diagrams will be used in order to describe the whole process in a very visual way that will make it easier for the reader to understand and follow the communications process correctly.

In order to achieve the objective of representing all orders appearing in the mapping of Table 2, it is necessary to group the orders appearing in the column *"Request STG-DC" according to* their characteristics of their communications process:

First of all, it is necessary to distinguish between *"Pull"* and *"Push"* tasks.

- **Pull:** Pull tasks are tasks that originate from a specific request from the GTS and are usually manual executions requested by the user.

- **Push:** Push tasks are tasks or actions that are performed at the initiative of the DC, or by tasks that have been programmed in the device.

The structural difference between these two typologies is simple. If the task is a Pull task, the corresponding flowchart is read as a whole, but if the task is a Push task, the flowchart would be the same, skipping the first level. This first level consists of the first communication between the STG and the DC and consists of the creation of the record *"Telemetry_user_request.* Logically, if the task is a push task, and therefore there is no request from the STG, this record does not exist and it goes directly to the phase where the relevant Job intervenes and updates the records.

For this reason, the flowcharts to be presented in this document are all drawn in Pull format, because if it were Push, the diagram can still be interpreted without any additional complication.

However, in Figure 27, the process of a standard push task is described with the most common examples being Meter Events and Concetrator Events.

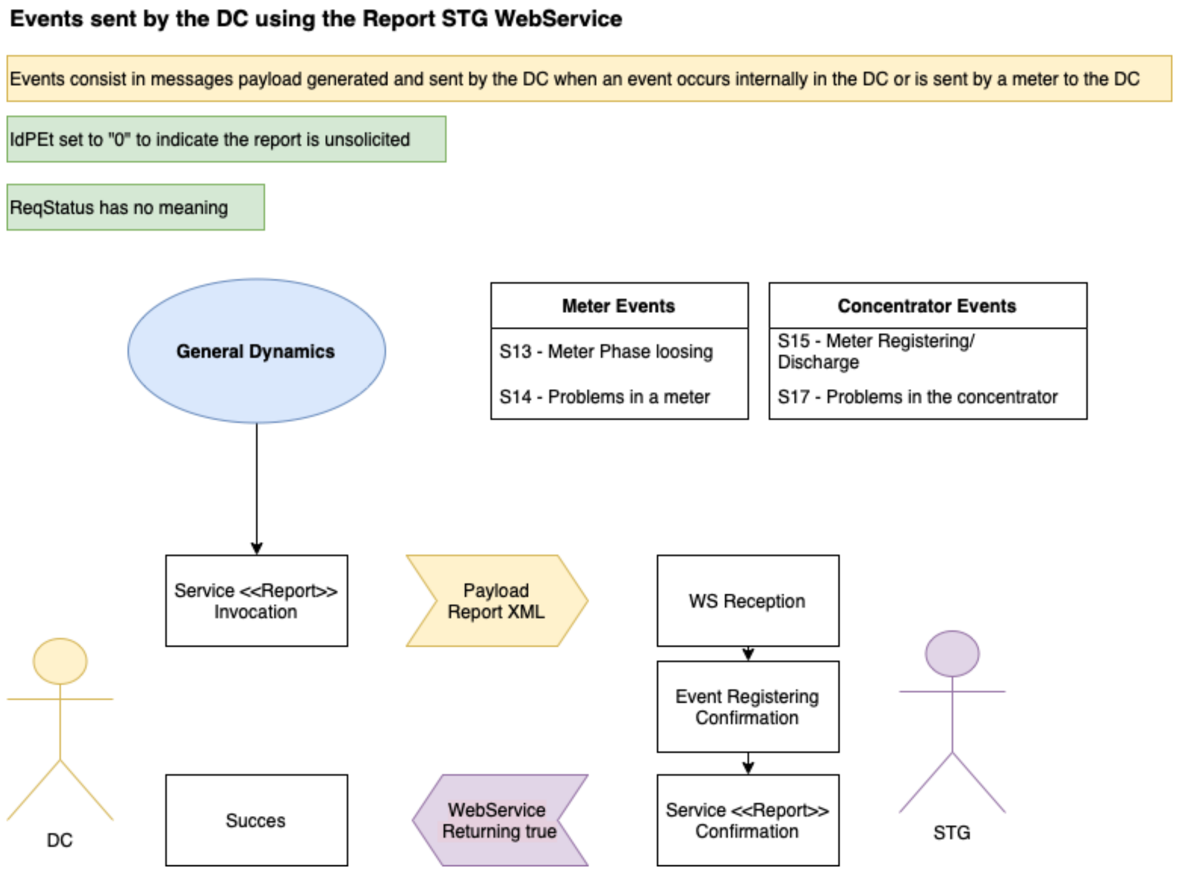

Events sent by the DC using the Report STG WebService
Events consist in messages payload generated and sent by the DC when an event occurs internally in the DC or is sent by a meter to the DC
IdPEt set to "0" to indicate the report is unsolicited
ReqStatus has no meaning
General Dynamics
Meter Events
S13 - Meter Phase loosing
S14 - Problems in a meter
Concentrator Events
S15 - Meter Registering/ Discharge
S17 - Problems in the concentrator
Service <<Report>> Invocation
Payload Report XML
WS Reception
Event Registering Confirmation
Service <<Report>> Confirmation
WebService Returning true
Succes
DC
STG

Figure 27: General Dynamics of Events sent by the DC WS

As can be seen in Figure 27, the DC executes a *"Report Invocation"* that sends a payload in XML format to the STG. Once the information is received, it is necessary to perform the registration confirmation that is returned to the DC in the form of a WebService.

In the specific case of tasks S13, S14, S15, and S17, this process is fundamental, as these are events occurring in the system that need to be reported to the user automatically. Having made this distinction, the next classification method to be applied is to distinguish the task according to three parameters:

- Its synchronicity.
  - Asynchronous: Requests the task, disregards the communication and when the DC returns the request a notification is sent to notify the STG that it is ready.
  - Synchronous: The STG requests the task and keeps the communication open until the task is completed.
- The information transfer format, which can be:
  - WS (WebService)
  - FTP (File Transfer Protocol)
- Parameters involved in the WS server.

Annex III contains a table showing all the applications offered by the STG-DC protocol in which these three characteristics are indicated for each type of application.
Annex II contains the description of all types of communication WS that exist in the GTS-DC specification.
Once this classification has been made, five different groups can be distinguished, which will be used to group the flowcharts. Table 3 shows these groups and assigns a reference to each one, which is the one we will use in the following subsections.

Model	Synchronous	FTP/WS	WS server & name
Model 1	Async	FTP	STG-UpdateRequestStatus
Model 2	Async	FTP	STG-Report
Model 3	Async	WS	STG-Report
Model 4	Async	WS	DC-Order
Model 5	Sync	WS	DC-Request/STG-Report

Table 3: Requests Clasification

4.2.1. MODEL 1

Model 1 is the set of Asynchronous tasks, which use the FTP protocol for sending and receiving information and use Update Request Status as communication WS.

Table 4 contains a list of the reports that have been constructed pertaining to this model:

Model 1			
Report	**Content**	**Report**	**Content**
S02/S2B	Load Profile /Daily Incremental	S09	Meter Events
S03	Daily Absolute	S17	Concentrator Events
S04	Monthly Billing Profile	S22	Concentrator Firmware Update Confirmation
S05/S5B	Daily Billing Values Profile	S23	Contract Definition
S06	Meter Parameters	S26	Instant Data Values on Demand

Table 4: Model 1 Reports

This model uses in its communication protocol the WS STG-UpdateRequestStatus. This WebService is designed to inform the STG about the status of a request.

It has to be called by the DC after an asynchronous request from the STG has finished at the DC, to inform the STG why the request has finished (normal completion, timeout...) if the information has not yet been passed to the STG (e.g. if the report was sent using Report WS). In the case of periodic reports (scheduled tasks) delivered by the DC you must use this web service to notify the report upload and set IdPet to 0.

In this type of requests, parallel to the URS, the WS UpdateMeterStatus intervenes.

This WebService is designed to inform the GTS about the status of a request.
It has to be called by the DC after an asynchronous request from the STG has finished at the DC, to inform the STG why the request has finished (normal completion, timeout...) if the information has not yet been passed to the STG (e.g. if the report was sent using Report WS). In the case of periodic reports (scheduled tasks) delivered by the DC you must use this web service to notify the report upload and set IdPet to 0.

In the following, it has been decided to use an example from the list above in order to describe the communication process of this model. The selected example is the reports S09 and S17, Meter and Concentrator Events as it also has the particularity of containing another WS *"STG-Report Query"*.

It should be noted that the rest of the reports listed in Table 4 are represented in their corresponding flow charts in Annex IV.
When this task is started, it collects the information from the counters if S09 or from the DCs if S17 and sends it to the STG.

The execution of this task can include as attributes the Groups and event codes to be delivered. The default deadline is given by the periodicity of the task. These requests are in charge of executing the *"Get xxx Events"* type orders of the OUM tool.
The description of the flowchart of these orders will be done in two sections:

5. Execution of the request and receipt of the FTP.

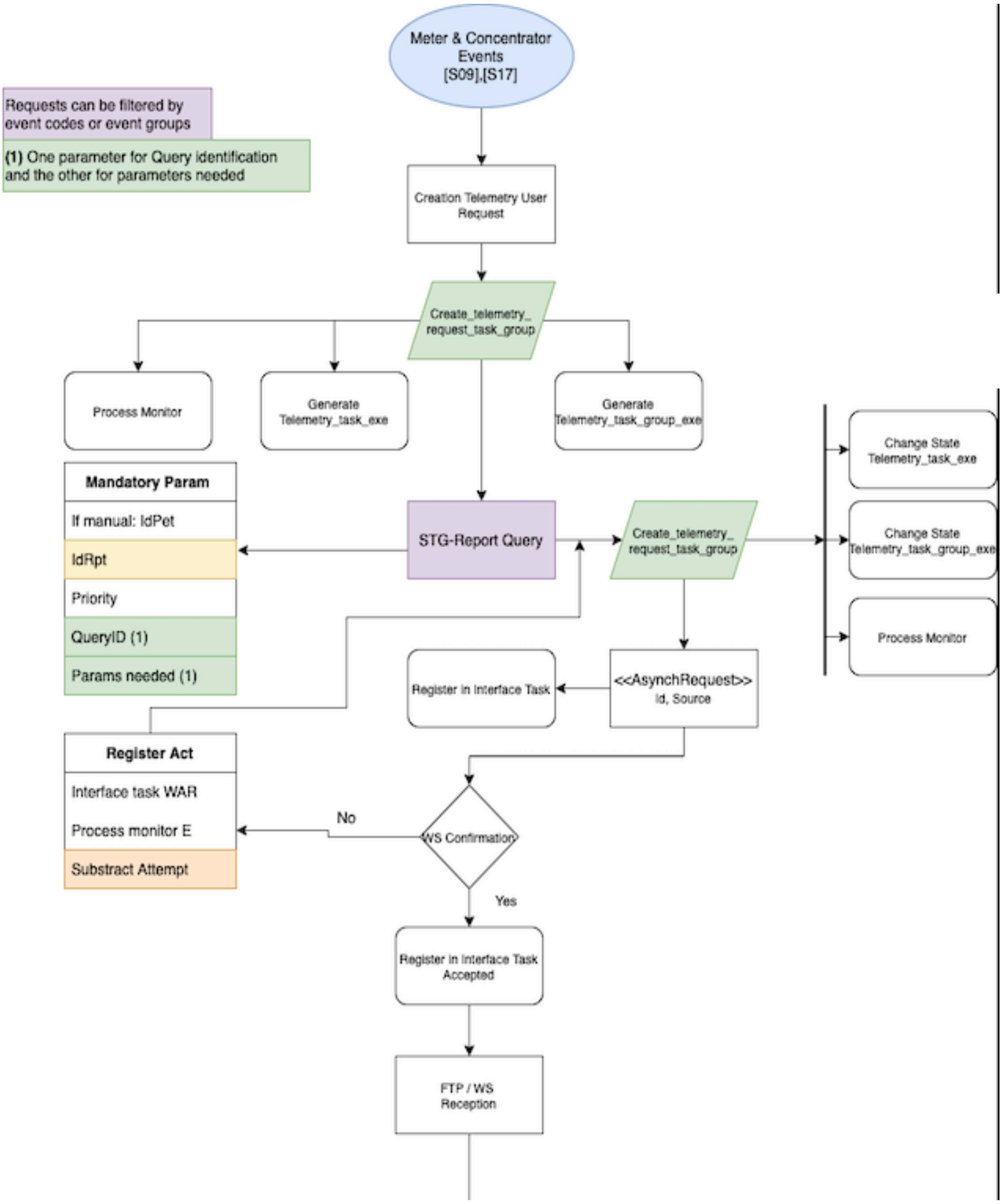

Meter & Concentrator Events [S09],[S17]
Requests can be filtered by event codes or event groups
(1) One parameter for Query identification and the other for parameters needed
Creation Telemetry User Request
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Mandatory Param
If manual: IdPet
IdRpt
Priority
QueryID (1)
Params needed (1)
STG-Report Query
Create_telemetry_request_task_group
Register in Interface Task
<<AsynchRequest>> Id, Source
Register Act
Interface task WAR
Process monitor E
Substract Attempt
No
WS Confirmation
Yes
Register in Interface Task Accepted
FTP / WS Reception

Figure 28: Meter and DC Events FlowChart Part 1

When the user executes the request, the first thing that is created is the record *"Create_Telemetry_User_Request",* the Job *"Create_telemetry_request _task"* intervenes and the following records are generated in READY status.

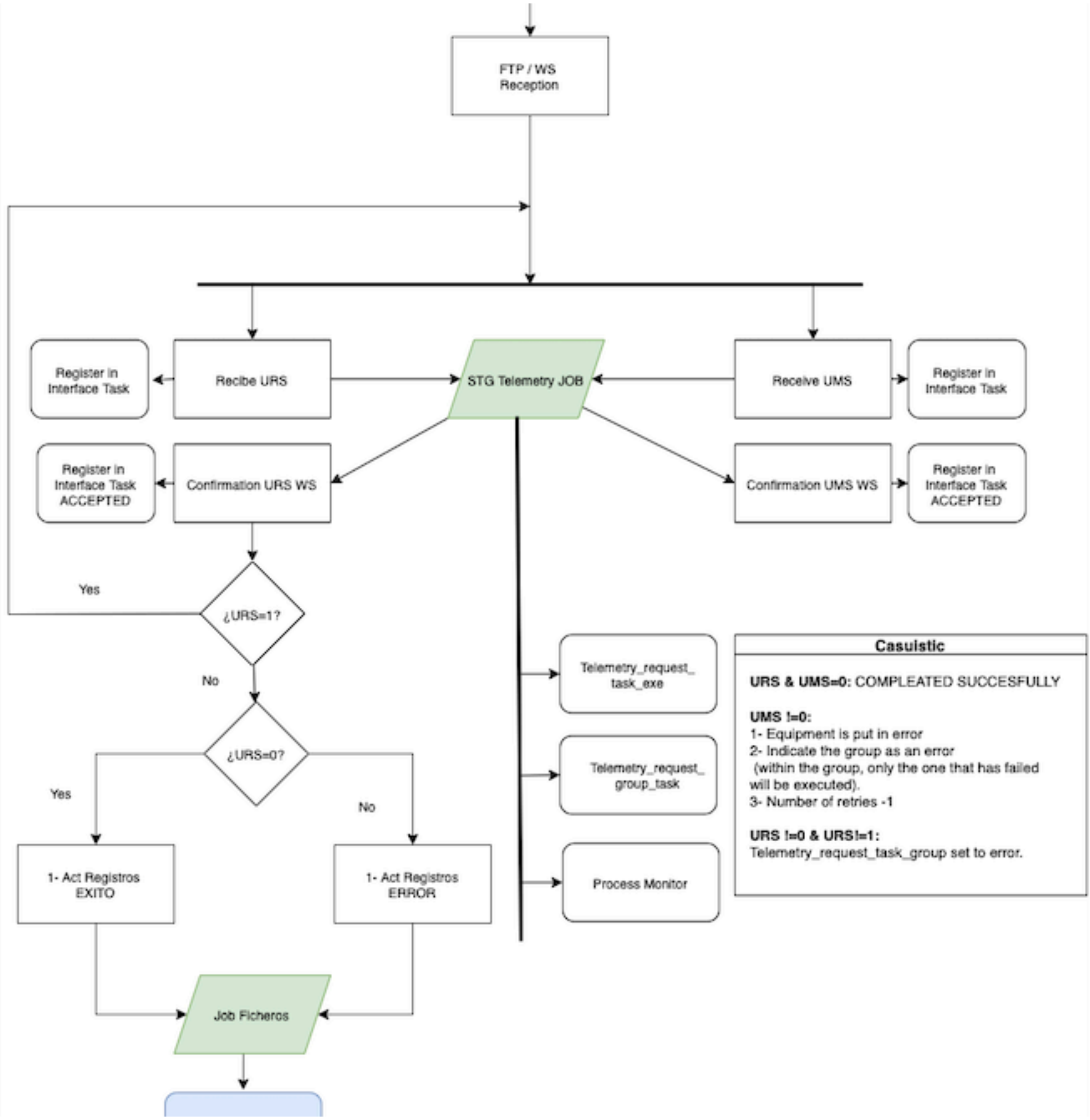
- Process Monitor
- Telemetry_Task_execute
- Telemetry_Task_Group_Execute

Once these registers have been generated, the WS STG-ReportQuery intervenes. This asynchronous WS is used to retrieve events in counters (S09) and concentrators (S17). Requests can be filtered by codes or groups of events. For this purpose, this WS includes different parameters than the previous one: one parameter for the identification of the query and another one for the necessary parameters.

Then, once OUM has the capacity to process the order, the next JOB enters, which is in charge of confirming the WS with the concentrator and aims to put the three indicated registers in the EXECUTION status. If the confirmation of the WS has not arrived, a retry will be subtracted in the registers, the status of the Process Monitor will change to Error and the status of the Interface Task will change to Warning until the confirmation arrives. Once this WS is confirmed, the system waits and the concentrator sends the FTP file to the OUM system. This system is stored in a record of the tool and contains a reference to its request so that when it is going to be processed, it is known to which process the generation of the file belongs.

The next step will be the processing of the file and the interaction of the system with the WS URS and UMS.

6. File processing and communication resolution



Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

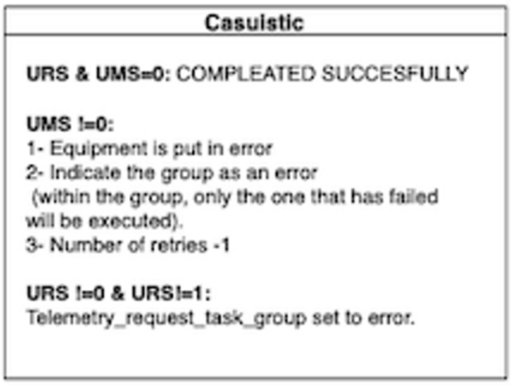
URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figure 29: Meter and DC Events Flowchart Part 2

Once the FTP file is received, the *STG-Telemetry JOB comes into play.* This job looks for pending PRIME protocol tasks, groups them, creates the corresponding telemetry_requests and interacts with the AsynchRequest to process the message.

In this particular case, this job will be in charge of interpreting and responding to incoming records by means of WS (URS and UMS) as shown in the flowchart depicted in Figure 29. As can be seen, each time the concentrator notifies it with a URS and its request reference, or with a UMS and its request reference, the *STG-Telemetry JOB* intervenes*,* which gives rise to the *STG-ServiceJob*. This job, according to the arrival of WS received (URS, UMS) updates the status of the requests in the tables, and processes the files with the name specified in each UMS of the process sent by the concentrator.

From here, the content of the URS is interpreted, which is the WebService that governs the system and indicates when the communication is finished. When URS=1 it means that the communication is still in progress, while when URS=0 it means that the communication has ended successfully, the rest of possible values indicate error or warning messages that are specified in Annex II. These values are crossed with the values returned by the UMS and the casuistry follows the process indicated in the flowchart table in Figure 30.



Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed
will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figure 30: URS and UMS Casuistic

Finally, the registers are updated with the corresponding status, and the JOB Ficheros stores the information in the appropriate register.

4.2.2. MODEL 2

Model 2 is the set of Asynchronous tasks, which use the FTP protocol for sending and receiving information and use the STG-Report as communication WS.

Table 5 contains a list of the informs that have been constructed pertaining to this model:

Model 2	
Report	**Content**
S14	Voltage and Current Profile
S18	Cut/Off Confirmation

Table 5:Model 2 Reports

This model uses the WS STG-Report in its communications protocol. Any request for information from a counter or the concentrator is considered a report.

The information report must be delivered asynchronously, the DC will respond with an FTP.

This service allows the DC to send to the GTS the content of a report, in terms of:

- Unsolicited report: DC event, counter event, DC reset, counter registration.
- Requested report : Reports generated after a given request

In the following it has been decided to use the report S14 *Voltage and Current Profile* as an example to describe the communicative process in Model 2.

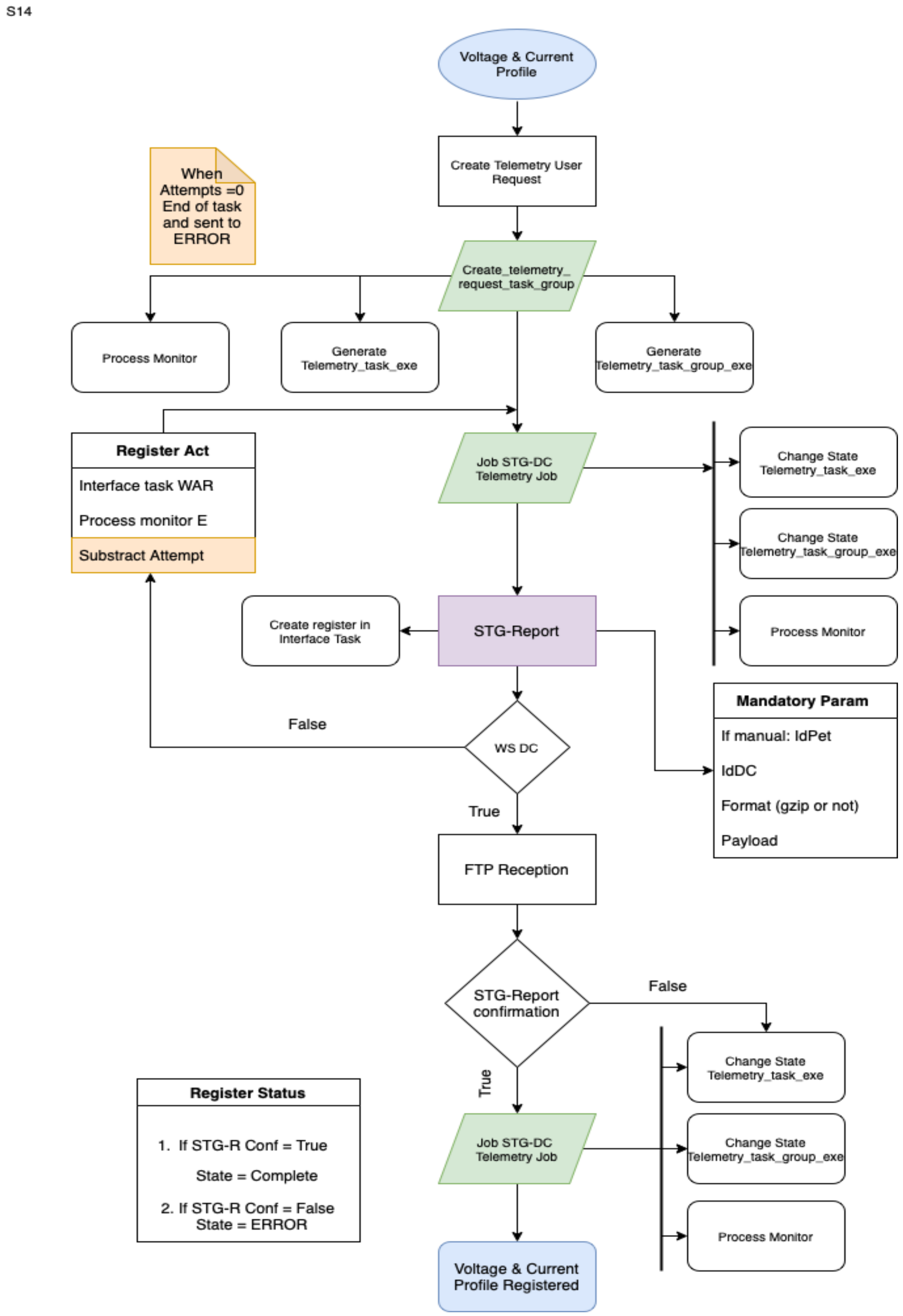

S14
Voltage & Current Profile
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
STG-Report
Mandatory Param
If manual: IdPet
IdDC
Format (gzip or not)
Payload
False
WS DC
True
FTP Reception
STG-Report confirmation
False
True
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Register Status
1. If STG-R Conf = True
State = Complete
2. If STG-R Conf = False
State = ERROR
Job STG-DC Telemetry Job
Voltage & Current Profile Registered

Figure 31: Voltage and Current Profile FlowChart

When the user executes the request, the first thing that is created is the record *"Create_Telemetry_User_Request",* the Job *"Create_telemetry_request _task"* intervenes and the following records are generated in READY status.

- Process Monitor
- Telemetry_Task_execute
- Telemetry_Task_Group_Execute

Once these registers have been created and the command has been executed, the *STG-Telemetry JOB* intervenes. This Job searches for pending PRIME protocol tasks, groups them and creates the corresponding *telemetry_requests* and prepares to launch the *STG-Report.*

This *STG-Report is* firstly responsible for managing the parameters received when executing the order and for managing the confirmation process with the concentrator by means of WS that the order has been launched correctly. The flowchart in Figure 31 shows the main cases to be taken into account.

The FTP file is received, and it is then that the *STG-Report* handles the read confirmation of the file according to its inherited status:

- If STG-Report Conf = True→ System State = Complete
- If STG-Report Conf = False→ System State = Error

If the confirmation has been successfully resolved, the *STG-DC Telemetry Job* intervenes*,* updates the corresponding logs and saves the report in its corresponding log.

If, on the other hand, the confirmation has been resolved with error, simply indicate in the logs the reason why the communication could not be successfully completed and end the interaction.

4.2.3. MODEL 3

Model 3 is the set of Asynchronous tasks, which use WS for sending and receiving information and which use the STG-Report as communication WS.

Table 6 contains a list of the infomes that have been constructed pertaining to this model:

Model 3	
Report	**Content**
S13	Meter Spontaneous Events
S15	Concentrator Spontaneous Events
S19/S22 3	Meter Firmware Update Confirmation

Table 6: Model 3 Reports

This model is responsible for generating the spontaneous event logs and for confirming the counter firmware update.

The WS involved in the communications process is the *STG-Report,* which has been previously defined in the previous section.

The main difference between Model 3 and Model 2 is that in Model 3 the exchange of information is no longer done by means of the FTP protocol, but by means of a WS in which a payload is received with the required information written in such a way that it can be interpreted by the PRIME protocol. This WS must be properly referenced with a label or an identifier so that the system can distinguish which order the payload corresponds to. Furthermore, it is stored in an *interface_task, contrary to* FTP which is dumped directly into a common directory.

For this model it has been decided to use the reports S13 and S15 (in charge of generating spontaneous event reports) to describe the sharing of this model.

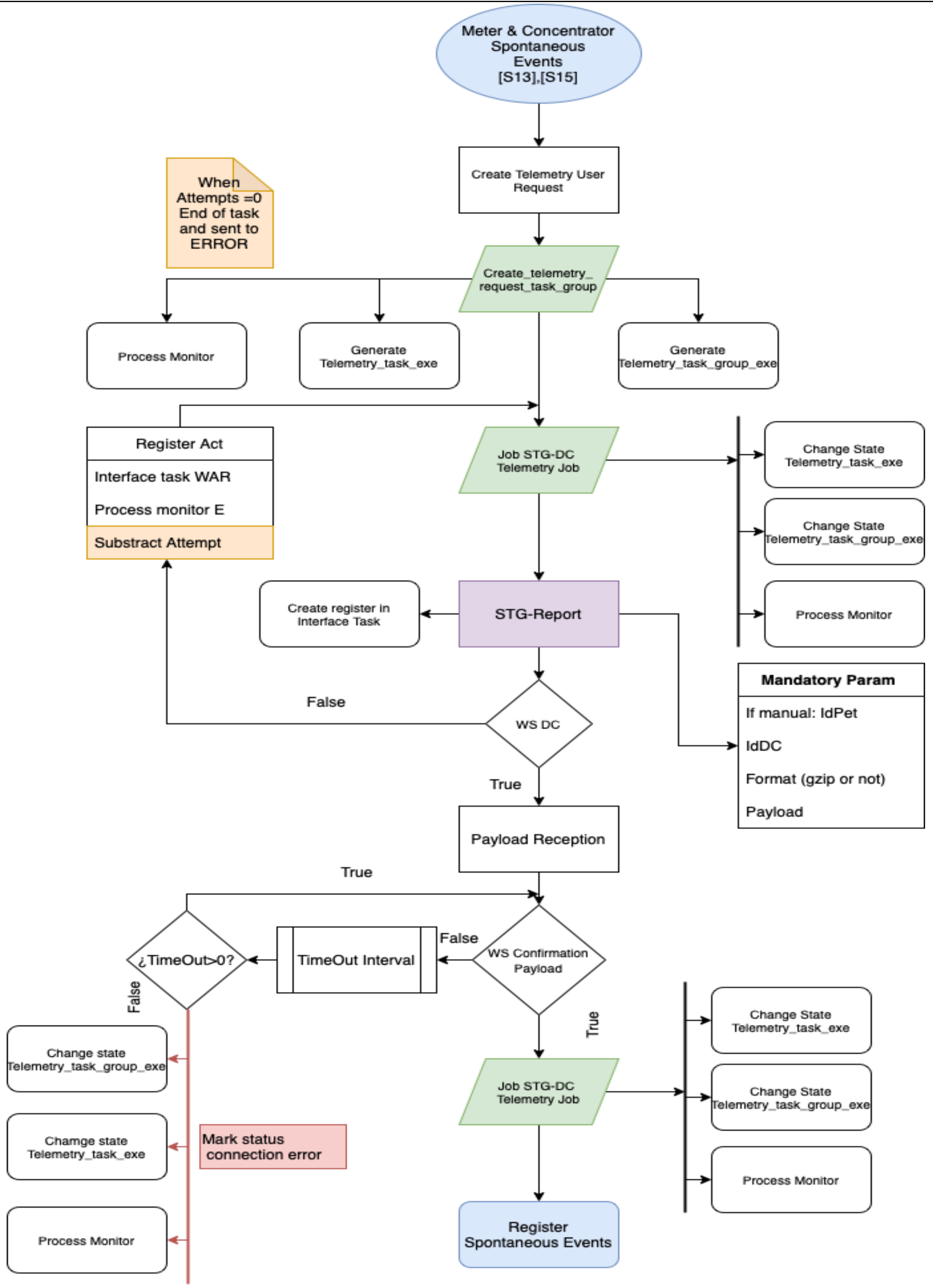

Meter & Concentrator Spontaneous Events [S13],[S15]
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
STG-Report
False
WS DC
True
Mandatory Param
If manual: IdPet
IdDC
Format (gzip or not)
Payload
Payload Reception
True
¿TimeOut>0?
TimeOut Interval
False
WS Confirmation Payload
False
True
Change state Telemetry_task_group_exe
Chamge state Telemetry_task_exe
Process Monitor
Mark status connection error
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Register Spontaneous Events

Figure 32: Meter and DC Spontanous Events FlowChart

In the same way as in the other cases, when the user executes the request, the first thing that is created is the *"Create_Telemetry_User_Request"* record, the Job *"Create_telemetry_request _task"* intervenes and the records mentioned in the two previous sections are generated in READY status. Note that if it were a planned execution, this step would not exist.

As in model 2, at this point in the communication, the "Job *STG-DC Telemetry-Job"* intervenes to manage the set of parameters received and generated in the request. It is then that the process marked by the *STG-Report* intervenes, which, depending on these parameters, assigns one state or another to the OUM tool registers.

Once the receipt of the confirmation WS has been confirmed, the system receives the WS containing the payload with the requested report and its corresponding identifying attributes.

Finally, the system must confirm that it has received the payload, update the records of the OUM tool and save the report in the corresponding interface.

Note that there is a timeout time in which the system must confirm receipt of the payload. If this timeout is exceeded, the OUM system shall be informed that an error has occurred in the communications process.

It is important to note that it is common that this request is made at the initiative of the DC and the first level of the flowchart does not exist.

4.2.4. MODEL 4

Model 4 is the set of Asynchronous tasks, which use WS for sending and receiving information and which use the DC-Order as communication WS.

Table 7 contains a list of the infomes that have been constructed pertaining to this model:

Model 4			
Report	**Content**	**Report**	**Content**
B02	Power Modification	B07	Modification of the Conncentrator Configuration
B03	Cut-Off / Reconnection	B09	Meter Parameters Modification
B04	Contract Modification	B9B	Meter Event Handling Modification
B05 B08	Meter Firmware Update		

Table 7: Model 4 Reports

This model is in charge of managing the direct action orders from the STG to the concentrator, i.e. the orders marked with Bxx. Like the report requests, these orders use messages in XML format sent through a WS in the form of a payload.

This model uses the DC-Order type as confirmation WS. It is a protocol in which an action is prepared in a specific measuring equipment, the necessary information is extracted by means of report requests, and finally the action is executed.

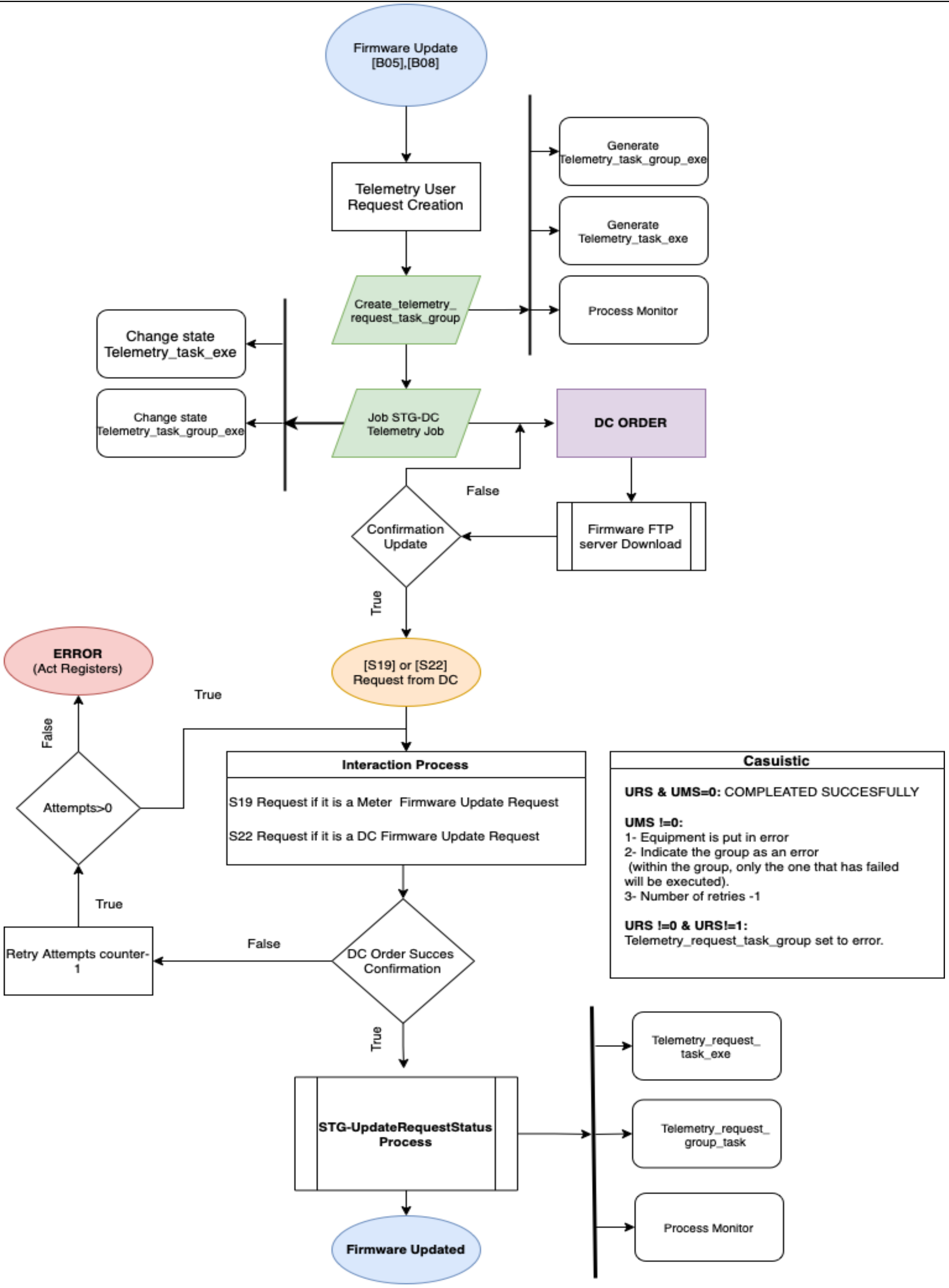
Once the DC-Order has been successfully executed, the concentrator must explicitly verify the requested order to each counter by means of a request from the STG.

The sending of an Order Request from the GTS to a counter may result in the execution of different phases or steps (download file update, pre-reporting, execution phase, post-reporting, etc.). These phases shall be executed sequentially in the CC, so that a phase cannot be started until the previous one is completed.

In case of failure in the execution of a phase, the execution sequence shall be stopped and the status of the order shall be reported to the GTS via UMS and URS with the appropriate category and error code.

To describe Model 4, we will use the flowchart corresponding to the *Firmware Update* command*,* contained in requests B05 and B08.

It is important to note that during the execution of this command, the system prompts the GTS to generate reports S19 or S22 (depending on whether the firmware upgrade is being performed on a counter or on the concentrator). The process of collecting and executing these reports is performed according to Model 3.


Firmware Update [B05],[B08]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_request_task_group
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC ORDER
False
Confirmation Update
Firmware FTP server Download
True
ERROR (Act Registers)
True
[S19] or [S22] Request from DC
False
Interaction Process
S19 Request if it is a Meter Firmware Update Request
S22 Request if it is a DC Firmware Update Request
Attempts>0
True
Retry Attempts counter-1
False
DC Order Succes Confirmation
True
STG-UpdateRequestStatus Process
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
Firmware Updated

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error (within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Figure 33: Firmware Update FlowChart

As in all previous cases, the first step in executing the command is to generate the three main OUM system registers through the *Telemetry_User_Request* script.
The *STG-DC Telemetry Job* is executed in exactly the same way as in the case of Model 2 and 3 only this time it precedes a WS of type *DC-Order*. This WS, once it is correctly confirmed, requests an S19 or S22 report to save a previous record (a backup) in case something goes wrong in the process of executing the job.

Once the process of generating the above reports is finished, the system must confirm the WS. To do so, it has a limited number of retries and takes the system to error status in case they are completed.

Finally, the *STG-Service Job* intervenes and by processing URS and UMS following exactly the same process described in Model 1, the states of the registers are updated and the communication is finalised by carrying out the update action in the requested equipment.

4.2.5. MODEL 5

Model 5 is the set of Synchronous tasks, which use WS for sending and receiving information and which use DC-Request / STG-Report as communication WS.

Table 5 contains a list of the infomes that have been constructed pertaining to this model:

Model 5	
Report	**Content**
S01	Instant Data Values
S21	Advanced Instant Data Values
S27	Current Billing Values on Demand

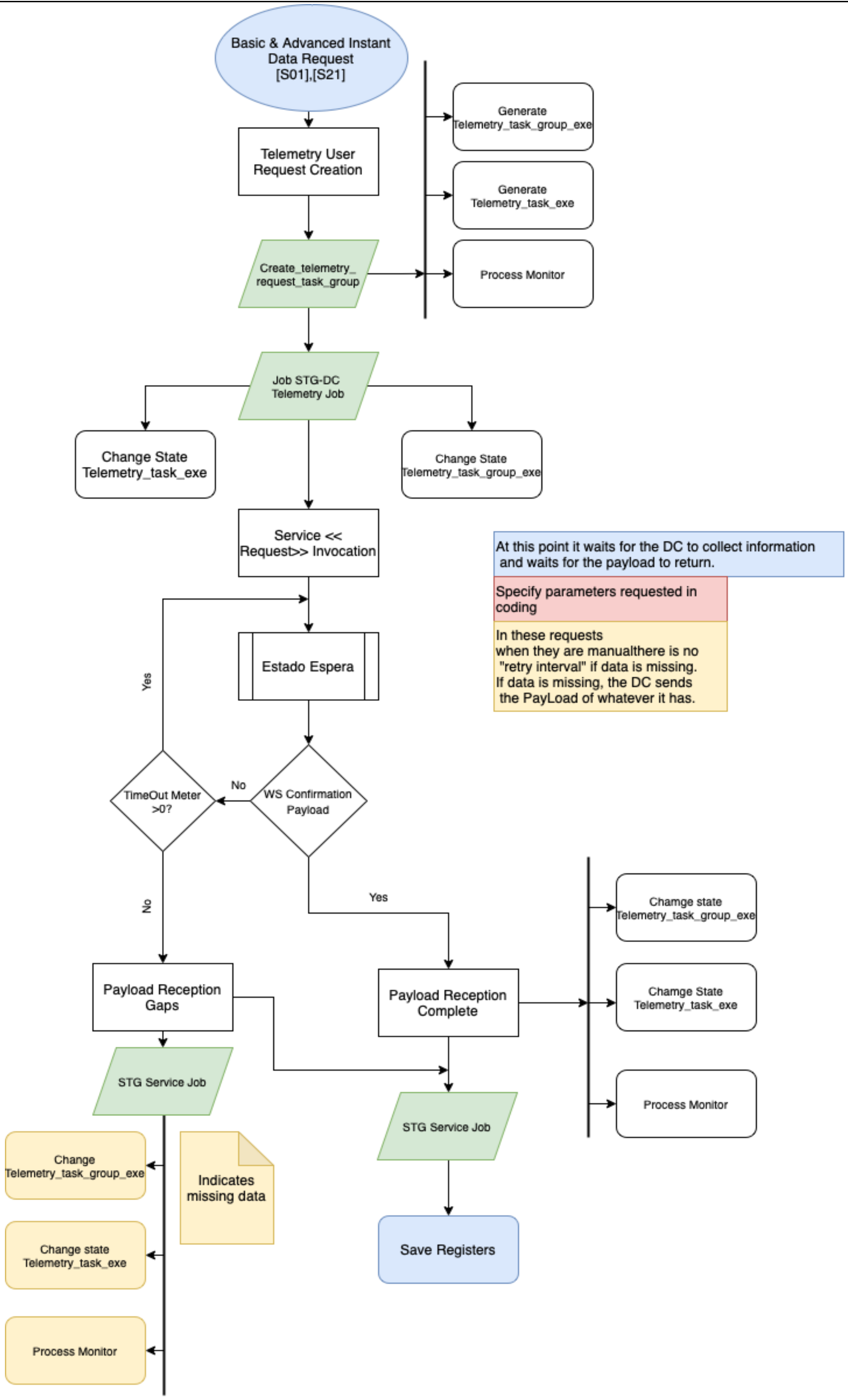
Table 8: Model 5 Reports

This model is composed of instantaneous value requests. As the word "instantaneous" itself indicates, the communication between STG and DC must be just that, instantaneous. For this reason, this last model deals with requests executed synchronously.

Synchronous request means that during the communication between the different entities there is synchronism, i.e. the communication channel is neither closed nor changed from the beginning of the process until the process is finished and the records are saved.

For this section it has been decided to show the request for instantaneous data requests, both basic and advanced S01 and S21.

The WS communications involved such as the payload confirmation process, and the updates of the registers work in the same way as for the previous sections.


Basic & Advanced Instant Data Request [S01],[S21]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_request_task_group
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Service << Request>> Invocation
Estado Espera
Yes
TimeOut Meter >0?
No
WS Confirmation Payload
No
Yes
Payload Reception Gaps
Payload Reception Complete
Chamge state Telemetry_task_group_exe
Chamge State Telemetry_task_exe
Process Monitor
STG Service Job
STG Service Job
Change Telemetry_task_group_exe
Change state Telemetry_task_exe
Process Monitor
Indicates missing data
Save Registers

At this point it waits for the DC to collect information and waits for the payload to return.

Specify parameters requested in coding

In these requests when they are manualthere is no "retry interval" if data is missing. If data is missing, the DC sends the PayLoad of whatever it has.

Figure 34: Basic and Advanced Instant Data FlowChart

CHAPTER 5. CASE STUDY

When drawing the flowcharts necessary to characterise the OUM integration with the STG-DC specification, a pattern is generated when executing the commands. This pattern consists of a series of database records that allow the traceability of the message in order to correctly communicate the head-end system with the MDM. They contain the necessary information to identify the request, the destination, and the format of the request for its correct processing and to allow the user to monitor and act accordingly.

It can be seen in the diagrams shown in Annex X how three registers appear in the first level and how they are updated in the following levels. These three registers are:

- Process Monitor
- Telemetry_Request_Task_Execution_Group
- Telemetry_Request_Task_Execution

However, although it does not appear directly, intrinsically before generating these three main records, when the MDM executes a manual or scheduled task, it generates another record:

- Telemetry_User_Request: This register is in charge of creating the requests in the database. These requests are created in "Prepared" status so that the corresponding Job can later update the records by changing their status for each stage and case that the communication protocol presents.

Knowing these processes at the database level is fundamental to understand the programming logic and the creation of the flowcharts proposed in the project. The generic interaction scheme that these registers follow is as follows:

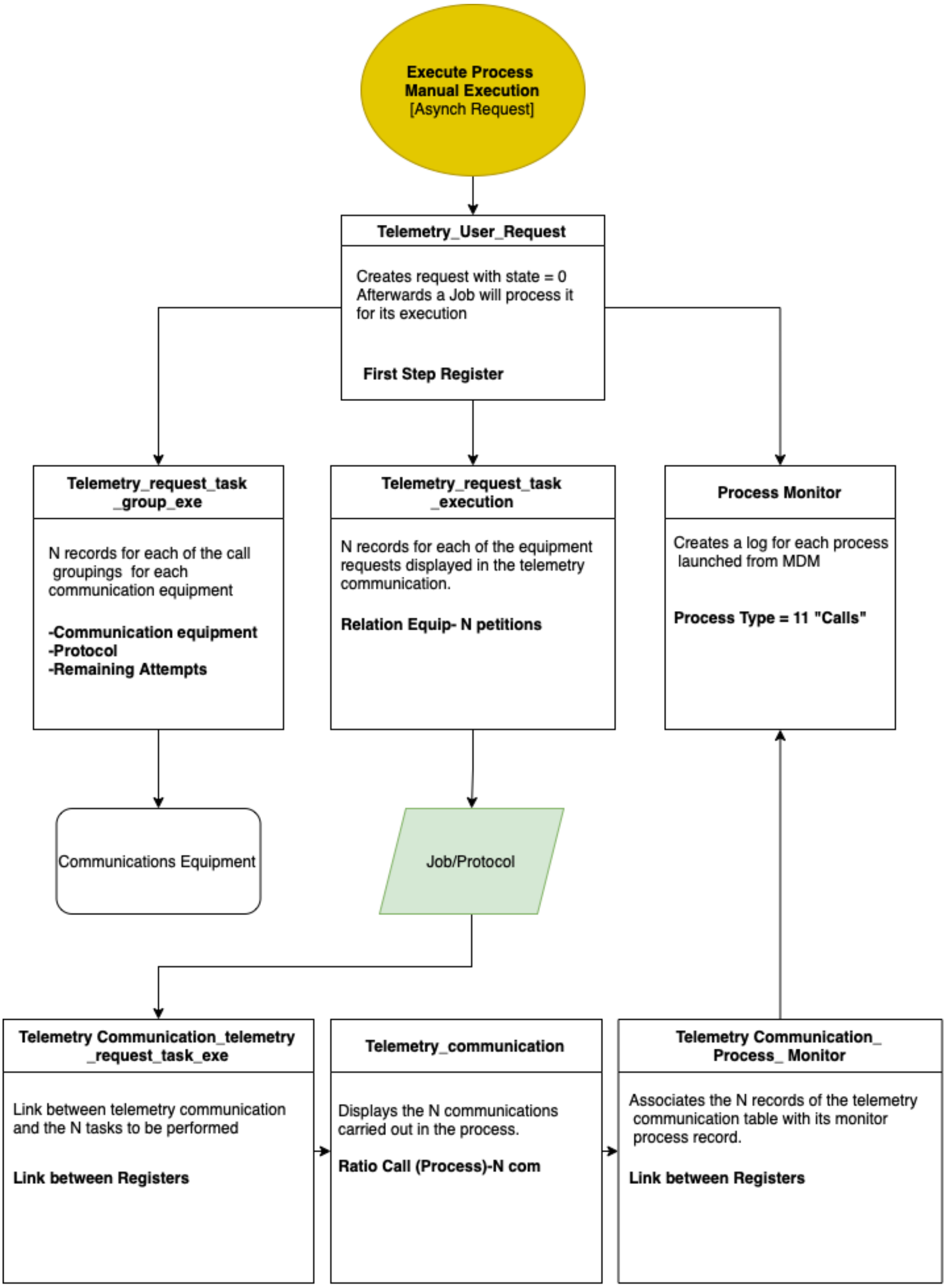

Execute Process
Manual Execution
[Asynch Request]
Telemetry_User_Request
Creates request with state = 0
Afterwards a Job will process it for its execution
First Step Register
Telemetry_request_task _group_exe
N records for each of the call groupings for each communication equipment
-Communication equipment
-Protocol
-Remaining Attempts
Telemetry_request_task _execution
N records for each of the equipment requests displayed in the telemetry communication.
Relation Equip- N petitions
Process Monitor
Creates a log for each process launched from MDM
Process Type = 11 "Calls"
Communications Equipment
Job/Protocol
Telemetry Communication_telemetry _request_task_exe
Link between telemetry communication and the N tasks to be performed
Link between Registers
Telemetry_communication
Displays the N communications carried out in the process.
Ratio Call (Process)-N com
Telemetry Communication_ Process_ Monitor
Associates the N records of the telemetry communication table with its monitor process record.
Link between Registers

Figure 35: Registers Interactions FlowChart

5.1. REGISTERS AND TABLE INTERACTION

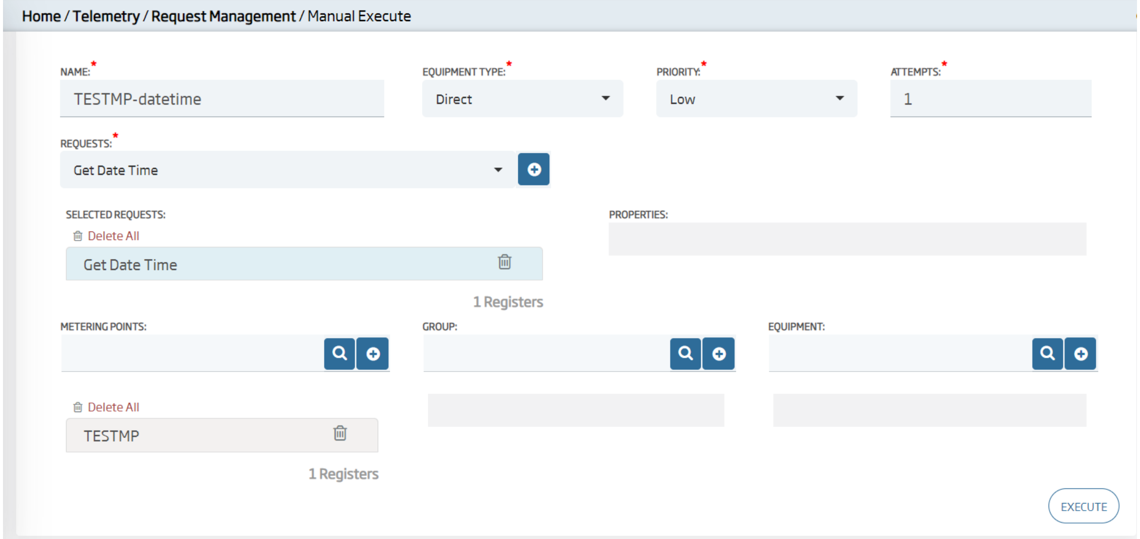
The purpose of this chapter is to describe in detail the registers and the interactions between them that are created at the database level in order to represent in OUM the communications process of interest to the user when making a request to the concentrator in manual execution topology (push). These interactions are represented in Figure 35 of the previous chapter.

It should be noted that for planned pull-type tasks, the structure of the process is almost the same, with two main differences:

- Telemetry_User_Request record is not created beforehand
- The IdPet (Request Identifier) is equal to zero.

For the description of these interactions, a practical example has been applied to facilitate the reader's traceability and understanding of the different identifiers, codes, points, equipment, and other relevant data.

A simple command has been executed, with a single request addressed to a single measuring point with one measuring equipment. The order is called ***"TESTMP-datetime"***, and the request is a *"Get Date Time"* requested to the *"TESTMP"* point. In addition, direct type equipment, low priority and a single connection attempt have been selected and this execution is represented in Figure 36.



Home / Telemetry / Request Management / Manual Execute

NAME:*	EQUIPMENT TYPE:*	PRIORITY:*	ATTEMPTS:*
TESTMP-datetime	Direct	Low	1

REQUESTS:* Get Date Time

SELECTED REQUESTS: Delete All

Get Date Time

1 Registers

PROPERTIES:

METERING POINTS: Delete All

TESTMP

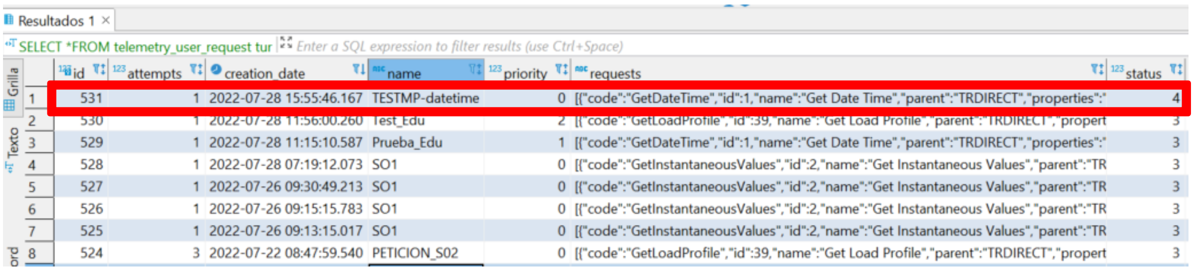
1 Registers

GROUP:

EQUIPMENT:

Figure 36: Telemetry Request Manually Executed

As mentioned above, when the command is executed, the first record that appears is the "*Telemetry_User_Request",* which is responsible for creating the request in the database so that the Job can subsequently process the request. As can be seen in Table 9, the request is created with state READY (state=4) and is assigned an ID=531.



Resultados 1 ×

SELECT *FROM telemetry_user_request tur | Enter a SQL expression to filter results (use Ctrl+Space)

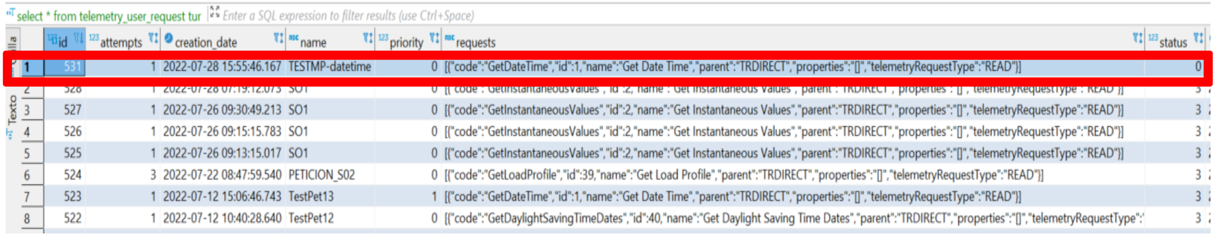
	id	attempts	creation_date	name	priority	requests	status
1	531	1	2022-07-28 15:55:46.167	TESTMP-datetime	0	[{"code":"GetDateTime","id":1,"name":"Get Date Time","parent":"TRDIRECT","properties":"	4
2	530	1	2022-07-28 11:56:00.260	Test_Edu	2	[{"code":"GetLoadProfile","id":39,"name":"Get Load Profile","parent":"TRDIRECT","propert	3
3	529	1	2022-07-28 11:15:10.587	Prueba_Edu	1	[{"code":"GetDateTime","id":1,"name":"Get Date Time","parent":"TRDIRECT","properties":"	3
4	528	1	2022-07-28 07:19:12.073	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TR	3
5	527	1	2022-07-26 09:30:49.213	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TR	3
6	526	1	2022-07-26 09:15:15.783	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TR	3
7	525	1	2022-07-26 09:13:15.017	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TR	3
8	524	3	2022-07-22 08:47:59.540	PETICION_S02	0	[{"code":"GetLoadProfile","id":39,"name":"Get Load Profile","parent":"TRDIRECT","propert	3

Table 9: Telemetry User Request

Additionally, in this register, the "Jason" of the request is represented by all the parameters involved in the order:

[{"code": "GetDateTime", "id":1, "name": "Get Date Time", "parent": "TRDIRECT", "properties":"[]", "telemetryRequestType": "READ"}]]

Once the relevant Job (it will be a different Job depending on the type of request) has processed this record, the first thing you notice is that it changes the status of the "*Telemetry_User_Request"* to IN PROCESS (state=1) and when the task is finished it sets it to FINISHED (state=0) if the task has been completed satisfactorily, or to ERROR (state=3) if a problem has arisen in the course of the communication. Table 10 shows the status of the successfully completed task.



select * from telemetry_user_request tur | Enter a SQL expression to filter results (use Ctrl+Space)

	id	attempts	creation_date	name	priority	requests	status
1	531	1	2022-07-28 15:55:46.167	TESTMP-datetime	0	[{"code":"GetDateTime","id":1,"name":"Get Date Time","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	0
2	528	1	2022-07-28 07:19:12.073	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
3	527	1	2022-07-26 09:30:49.213	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
4	526	1	2022-07-26 09:15:15.783	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
5	525	1	2022-07-26 09:13:15.017	SO1	0	[{"code":"GetInstantaneousValues","id":2,"name":"Get Instantaneous Values","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
6	524	3	2022-07-22 08:47:59.540	PETICION_S02	0	[{"code":"GetLoadProfile","id":39,"name":"Get Load Profile","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
7	523	1	2022-07-12 15:06:46.743	TestPet13	1	[{"code":"GetDateTime","id":1,"name":"Get Date Time","parent":"TRDIRECT","properties":"[]","telemetryRequestType":"READ"}]	3
8	522	1	2022-07-12 10:40:28.640	TestPet12	0	[{"code":"GetDaylightSavingTimeDates","id":40,"name":"Get Daylight Saving Time Dates","parent":"TRDIRECT","properties":"[]","telemetryRequestType":'	3

Table 10: Telemetry User Request Status

In order to generate the next three records that make up the next level (step two of the scheme) in the traceability of the records, the system needs to locate "to whom" the information has been requested. For this purpose, it has been decided to create three additional tables in the database according to the existing request possibilities (request to a measuring point, request to a group of points, request to a specific piece of equipment). Figure 37 shows the relationship between these records and "*Telemetry_User_Request"*.

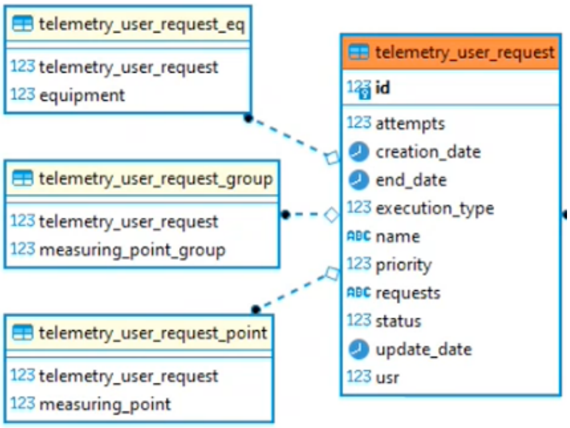
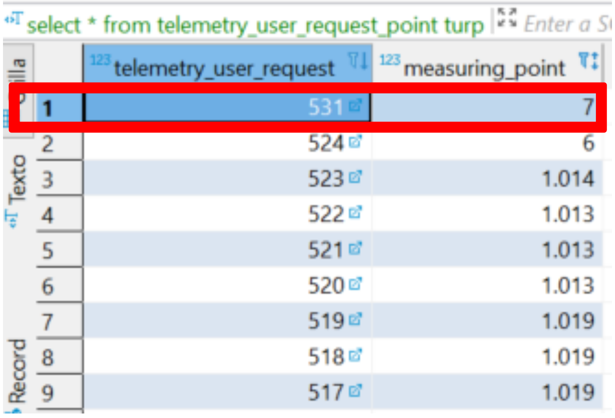

telemetry_user_request_eq
123 telemetry_user_request
123 equipment
telemetry_user_request_group
123 telemetry_user_request
123 measuring_point_group
telemetry_user_request_point
123 telemetry_user_request
123 measuring_point
telemetry_user_request
123 id
123 attempts
creation_date
end_date
123 execution_type
ABC name
123 priority
ABC requests
123 status
update_date
123 usr

Figure 37: Telemetry User Request Interactions

In this Case Study, the request has only been executed to one measurement point (TESTMP), which means that only the *"Telemetry_User_Request_Point"* register will be involved, leaving the other two empty. As can be seen in the table of this register, the ID of the *"Telemetry_User_Request"* is associated with the measurement point, which in this case is number 7.



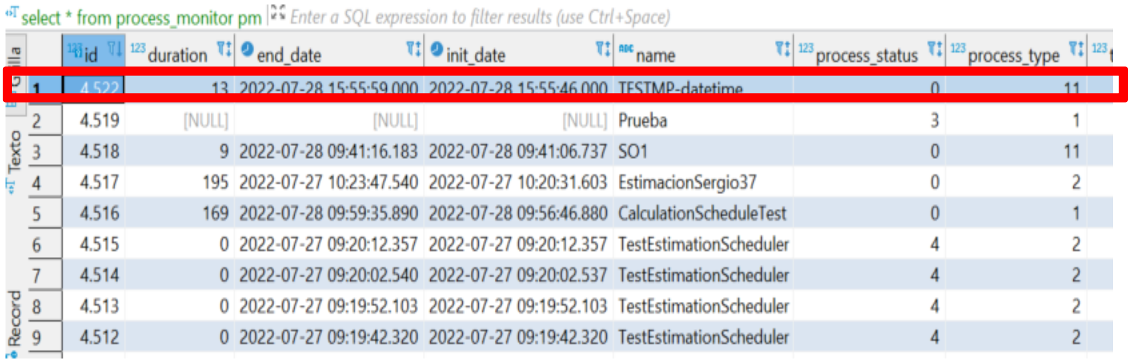
select * from telemetry_user_request_point turp

	telemetry_user_request	measuring_point
1	531	7
2	524	6
3	523	1.014
4	522	1.013
5	521	1.013
6	520	1.013
7	519	1.019
8	518	1.019
9	517	1.019

Table 11: Telemetry User Request Point

Once the registration is ready and the STG-DC communication channel is free, the next level is reached.

First, the *"Process_Monitor"* record is generated and a new identifier is associated to the Case Study request in order to keep track of the records that appear in the process monitor for each event occurring as a result of the request. In this case the *ID=4522.* It should be noted that this table shows that the total duration of the communication was 13 seconds and that *"process_type=11"* indicates that it belongs to the telemetry interface.

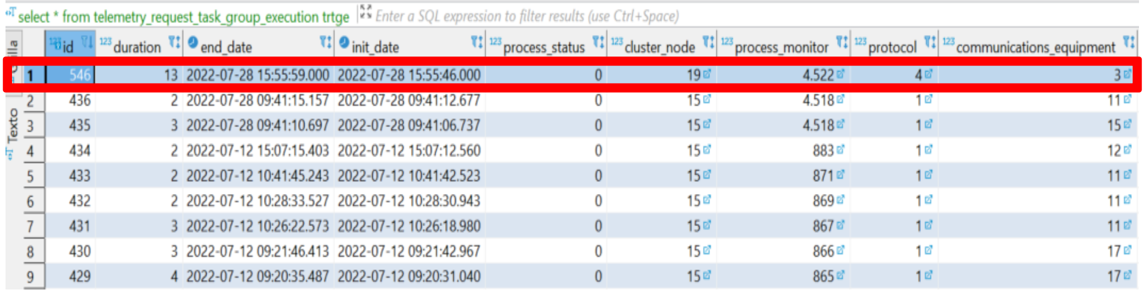


select * from process_monitor pm

	id	duration	end_date	init_date	name	process_status	process_type
1	4.522	13	2022-07-28 15:55:59.000	2022-07-28 15:55:46.000	TESTMP-datetime	0	11
2	4.519	[NULL]	[NULL]	[NULL]	Prueba	3	1
3	4.518	9	2022-07-28 09:41:16.183	2022-07-28 09:41:06.737	SO1	0	11
4	4.517	195	2022-07-27 10:23:47.540	2022-07-27 10:20:31.603	EstimacionSergio37	0	2
5	4.516	169	2022-07-28 09:59:35.890	2022-07-28 09:56:46.880	CalculationScheduleTest	0	1
6	4.515	0	2022-07-27 09:20:12.357	2022-07-27 09:20:12.357	TestEstimationScheduler	4	2
7	4.514	0	2022-07-27 09:20:02.540	2022-07-27 09:20:02.537	TestEstimationScheduler	4	2
8	4.513	0	2022-07-27 09:19:52.103	2022-07-27 09:19:52.103	TestEstimationScheduler	4	2
9	4.512	0	2022-07-27 09:19:42.320	2022-07-27 09:19:42.320	TestEstimationScheduler	4	2

Table 12: Process Monitor

On the other hand, the *"Telemtry_request_task_group_execution"* is generated*, which is* in charge of gathering the requests in a minimum number of possible communication addresses and protocols (it organises the IP addresses of each *"communication_equipment"* and its protocols) to optimise the efficiency of the channel and to be able to execute all the orders that go to the same communications equipment under the same protocol at the same time. It saves communication transactions, since it is possible that orders executed or planned at different times require the same information or simply execute all orders at the same time.



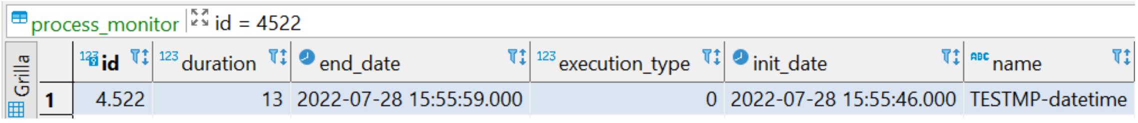
select * from telemetry_request_task_group_execution trtge | Enter a SQL expression to filter results (use Ctrl+Space)

	id	duration	end_date	init_date	process_status	cluster_node	process_monitor	protocol	communications_equipment
1	546	13	2022-07-28 15:55:59.000	2022-07-28 15:55:46.000	0	19	4.522	4	3
2	436	2	2022-07-28 09:41:15.157	2022-07-28 09:41:12.677	0	15	4.518	1	11
3	435	3	2022-07-28 09:41:10.697	2022-07-28 09:41:06.737	0	15	4.518	1	15
4	434	2	2022-07-12 15:07:15.403	2022-07-12 15:07:12.560	0	15	883	1	12
5	433	2	2022-07-12 10:41:45.243	2022-07-12 10:41:42.523	0	15	871	1	11
6	432	2	2022-07-12 10:28:33.527	2022-07-12 10:28:30.943	0	15	869	1	11
7	431	3	2022-07-12 10:26:22.573	2022-07-12 10:26:18.980	0	15	867	1	11
8	430	3	2022-07-12 09:21:46.413	2022-07-12 09:21:42.967	0	15	866	1	17
9	429	4	2022-07-12 09:20:35.487	2022-07-12 09:20:31.040	0	15	865	1	17

Table 13: Telemetry Request Task Group

As can be seen in Table 13, a cluster is generated with *ID=546* that references the Process_Monitor ID generated earlier. It indicates *cluster_node=19* for this request and selects protocol number 4 corresponding to the STG-DC, finally it links this cluster to an address of *communications_equipment=3.*

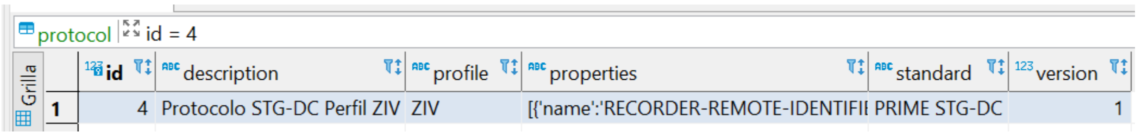
Details of each of the registers discussed above are shown below.



process_monitor | id = 4522

	id	duration	end_date	execution_type	init_date	name
1	4.522	13	2022-07-28 15:55:59.000	0	2022-07-28 15:55:46.000	TESTMP-datetime

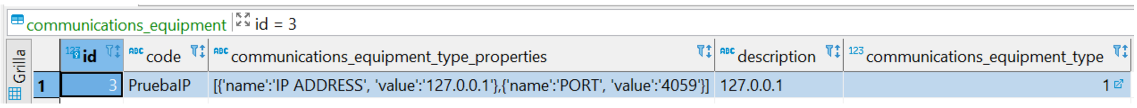
Table 14: Process Monitor Detail



protocol | id = 4

	id	description	profile	properties	standard	version
1	4	Protocolo STG-DC Perfil ZIV	ZIV	[{'name':'RECORDER-REMOTE-IDENTIFI	PRIME STG-DC	1

Table 15: Protocol Interface Detail

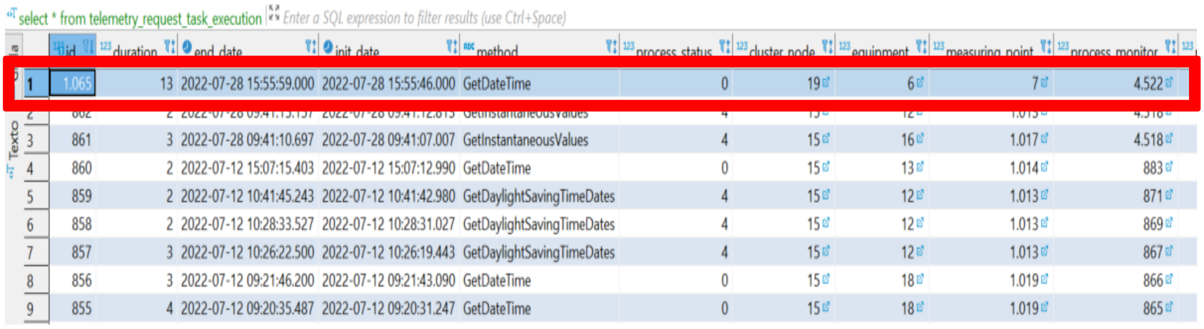


communications_equipment | id = 3

	id	code	communications_equipment_type_properties	description	communications_equipment_type
1	3	PruebaIP	[{'name':'IP ADDRESS', 'value':'127.0.0.1'},{'name':'PORT', 'value':'4059'}]	127.0.0.1	1

Table 16: Communications Equipment Detail

Finally, there is the "*telemetry_request_task_execution"* which can be understood by definition as the table representing the maximum granularity of the records. It is the Cartesian product between teams and tasks. That is, it represents a single request for each row, together with a particular piece of equipment for a particular measuring point or measuring equipment. The equipment and the point, as mentioned in previous sections, appear related for the first time in this record.



select * from telemetry_request_task_execution | Enter a SQL expression to filter results (use Ctrl+Space)

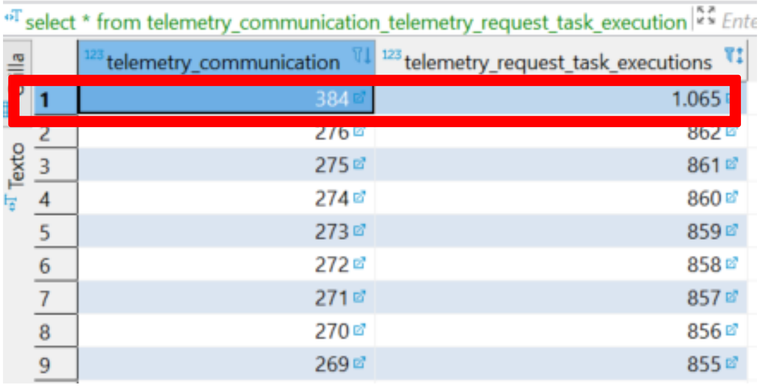
	id	duration	end_date	init_date	method	process_status	cluster_node	equipment	measuring_point	process_monitor
1	1.065	13	2022-07-28 15:55:59.000	2022-07-28 15:55:46.000	GetDateTime	0	19	6	7	4.522
2	862	2	2022-07-28 09:41:13.137	2022-07-28 09:41:12.813	GetInstantaneousValues	4	15	12	1.015	4.516
3	861	3	2022-07-28 09:41:10.697	2022-07-28 09:41:07.007	GetInstantaneousValues	4	15	16	1.017	4.518
4	860	2	2022-07-12 15:07:15.403	2022-07-12 15:07:12.990	GetDateTime	0	15	13	1.014	883
5	859	2	2022-07-12 10:41:45.243	2022-07-12 10:41:42.980	GetDaylightSavingTimeDates	4	15	12	1.013	871
6	858	2	2022-07-12 10:28:33.527	2022-07-12 10:28:31.027	GetDaylightSavingTimeDates	4	15	12	1.013	869
7	857	3	2022-07-12 10:26:22.500	2022-07-12 10:26:19.443	GetDaylightSavingTimeDates	4	15	12	1.013	867
8	856	3	2022-07-12 09:21:46.200	2022-07-12 09:21:43.090	GetDateTime	0	15	18	1.019	866
9	855	4	2022-07-12 09:20:35.487	2022-07-12 09:20:31.247	GetDateTime	0	15	18	1.019	865

Table 17: Telemetry Request Task Execution

In Table 17, corresponding to *"telemetry_request_task_execution",* it is important to highlight that a new identifier is generated to be able to trace the communications process in the following registers *ID=1065*, it represents the *GateDateTime* field*, it* is associated to the *cluster_node* and the corresponding *process_monitor* and finally it relates measurement point number 7 with equipment number 6.

At this point, everything is perfectly prepared and ordered so that the call can be processed by the corresponding Job. To do this, the Job in charge of processing the call will use the information stored in these registers and we move on to the third and final level of the registration process.

It is necessary to create a table that relates each *"Task_Execution"* with its call. This table is the *"Telemetry_Communication_Telemetry_Request_Task_Execution" shown in* Table 18.

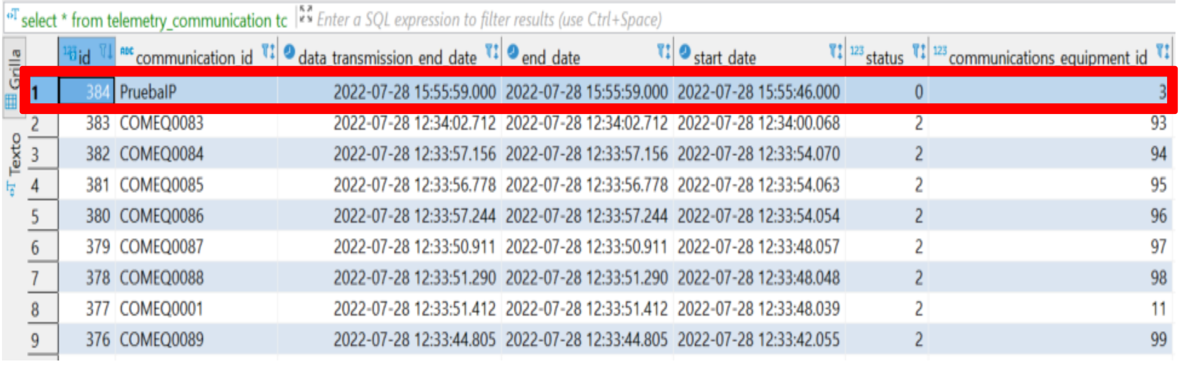


select * from telemetry_communication_telemetry_request_task_execution

	telemetry_communication	telemetry_request_task_executions
1	384	1.065
2	276	862
3	275	861
4	274	860
5	273	859
6	272	858
7	271	857
8	270	856
9	269	855

Table 18: Telemetry Communication & Telemetry Request Task Execution

It can be seen that this table links the identifier of the *"telemetry_request_task_execution" (ID=1605),* with a new identifier of its own for *"Telemetry_Communication" (ID=384).* From this new identifier, the *"Telemetry_Communication"* table is generated, which is in charge of representing the N communications that are carried out during the whole process. In other words, it relates each process with its N communications. This table is shown in Table 19.



select * from telemetry_communication tc

Enter a SQL expression to filter results (use Ctrl+Space)

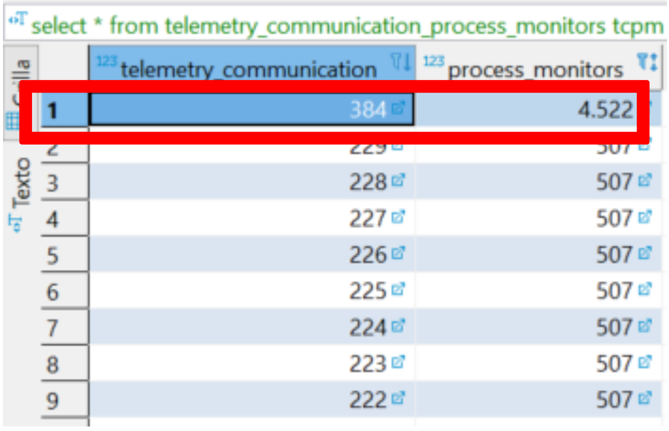
	id	communication id	data transmission end date	end date	start date	status	communications equipment id
1	384	PruebaIP	2022-07-28 15:55:59.000	2022-07-28 15:55:59.000	2022-07-28 15:55:46.000	0	3
2	383	COMEQ0083	2022-07-28 12:34:02.712	2022-07-28 12:34:02.712	2022-07-28 12:34:00.068	2	93
3	382	COMEQ0084	2022-07-28 12:33:57.156	2022-07-28 12:33:57.156	2022-07-28 12:33:54.070	2	94
4	381	COMEQ0085	2022-07-28 12:33:56.778	2022-07-28 12:33:56.778	2022-07-28 12:33:54.063	2	95
5	380	COMEQ0086	2022-07-28 12:33:57.244	2022-07-28 12:33:57.244	2022-07-28 12:33:54.054	2	96
6	379	COMEQ0087	2022-07-28 12:33:50.911	2022-07-28 12:33:50.911	2022-07-28 12:33:48.057	2	97
7	378	COMEQ0088	2022-07-28 12:33:51.290	2022-07-28 12:33:51.290	2022-07-28 12:33:48.048	2	98
8	377	COMEQ0001	2022-07-28 12:33:51.412	2022-07-28 12:33:51.412	2022-07-28 12:33:48.039	2	11
9	376	COMEQ0089	2022-07-28 12:33:44.805	2022-07-28 12:33:44.805	2022-07-28 12:33:42.055	2	99

Table 19: Telemetry Communication

It can be seen that I linked the identifier *"Telemetry_Communication" (ID=384)* with the identifier *"communications_equipment_id" (ID=3)* that first appeared in the table *"Telemtry_request_task_group_execution"* . It also indicates the date and time of the start and end of the call.

It should be noted that the status is shown as *state=0 as* the process has finished capturing these records and has resulted in a successful operation.

Finally, in order to be able to represent this whole process in the user interface of the OUM application, it is necessary to link these communication registers of this last level with the *"Process_Monitor"* register and interface. For this purpose, the *"Telemetry_Communications_Process_Monitor"* table is created, which is responsible for associating the N registers of the communications table with its process monitoring register.



```
select * from telemetry_communication_process_monitors tcpm
```

	telemetry_communication	process_monitors
1	384	4.522
2	229	507
3	228	507
4	227	507
5	226	507
6	225	507
7	224	507
8	223	507
9	222	507

Table 20: Telemetry Communication Process Monitor

Table 20, links the *"Telemetry_Communication" (ID=384) with the "process_monitor_id" (ID=4522). At this point, all data are internally linked and any case can be considered for each manual execution or for each planned execution.*

Once this whole process has been represented, two fundamental entities remain to be described. On the first hand, the call monitor, a key element in the monitoring of the system, and on the other hand, the description of the measurement point and the equipment used in this case study.

5.2. MONITORING OF CALLS AND REQUESTS:

The first of the entities belonging to the OUM system to be described is the call monitor, which, although it is not an active part of the decision-making process, is essential for monitoring the status of calls.

Whether the triggering of the different requests is done manually or automatically, their execution can be monitored in the Call Monitor tool.

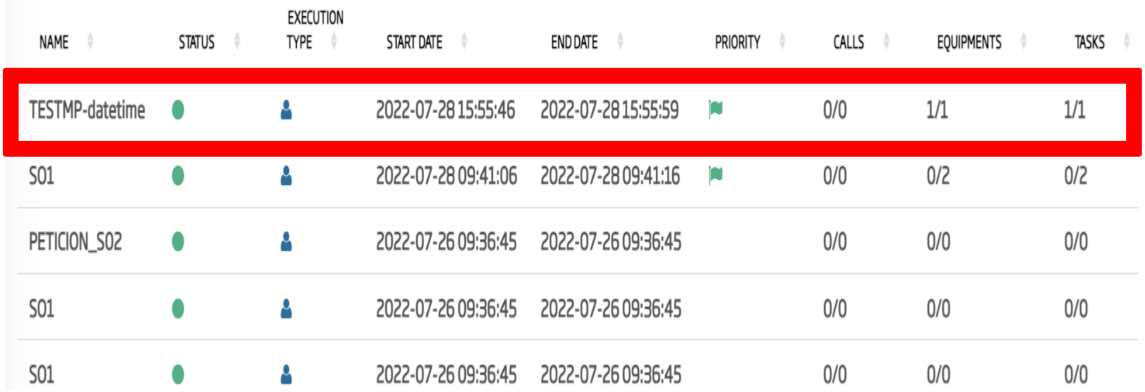
This tool monitors all calls and requests executed in the system regardless of whether they are: synchronous or asynchronous, manual or automatic, any type of request on any type of equipment.

This monitoring is carried out in a view with different levels of detail. In the following we define the details of this view according to its different levels:

- **Level 1:** Process or Call Group (Corresponding to manual or automatic settings).
- **Level 2:** Communication or Call: Breakdown by call or communication direction, of those corresponding to that group or process (depending on the configuration of the equipment).
- **Level 3:** Team: Breakdown by team, of those dependent on each communications directorate.
- **Level 4:** Request or task: Breakdown by each of the above requests.
- **Level 5:** Traces or logs: In this last level, the different traces or logs of the communications and recovery of each task can be displayed, in order to inform the user of the result of this operation and to indicate possible errors or warnings.

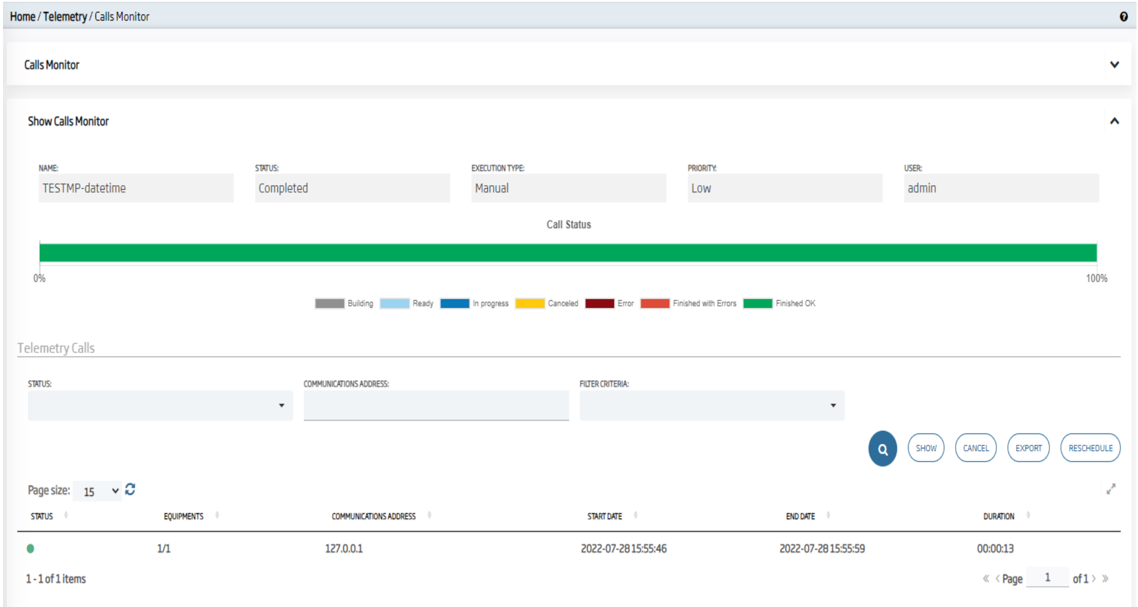
At each of the levels mentioned above, the corresponding statistics, status, traceability and audit information will be provided. Therefore, by means of the call monitor, the user will be able to monitor the status, results and traceability of the different tasks of interaction with the measuring equipment, providing timely information in real time.

These levels are represented in the following figures:



NAME	STATUS	EXECUTION TYPE	START DATE	END DATE	PRIORITY	CALLS	EQUIPMENTS	TASKS
TESTMP-datetime	●	👤	2022-07-28 15:55:46	2022-07-28 15:55:59	⚑	0/0	1/1	1/1
S01	●	👤	2022-07-28 09:41:06	2022-07-28 09:41:16	⚑	0/0	0/2	0/2
PETICION_S02	●	👤	2022-07-26 09:36:45	2022-07-26 09:36:45		0/0	0/0	0/0
S01	●	👤	2022-07-26 09:36:45	2022-07-26 09:36:45		0/0	0/0	0/0
S01	●	👤	2022-07-26 09:36:45	2022-07-26 09:36:45		0/0	0/0	0/0

Figure 38: Level 1 - Calls Monitor and Processes



Home / Telemetry / Calls Monitor

Calls Monitor

Show Calls Monitor

NAME: TESTMP-datetime

STATUS: Completed

EXECUTION TYPE: Manual

PRIORITY: Low

USER: admin

Telemetry Calls

STATUS:

COMMUNICATIONS ADDRESS:

FILTER CRITERIA:

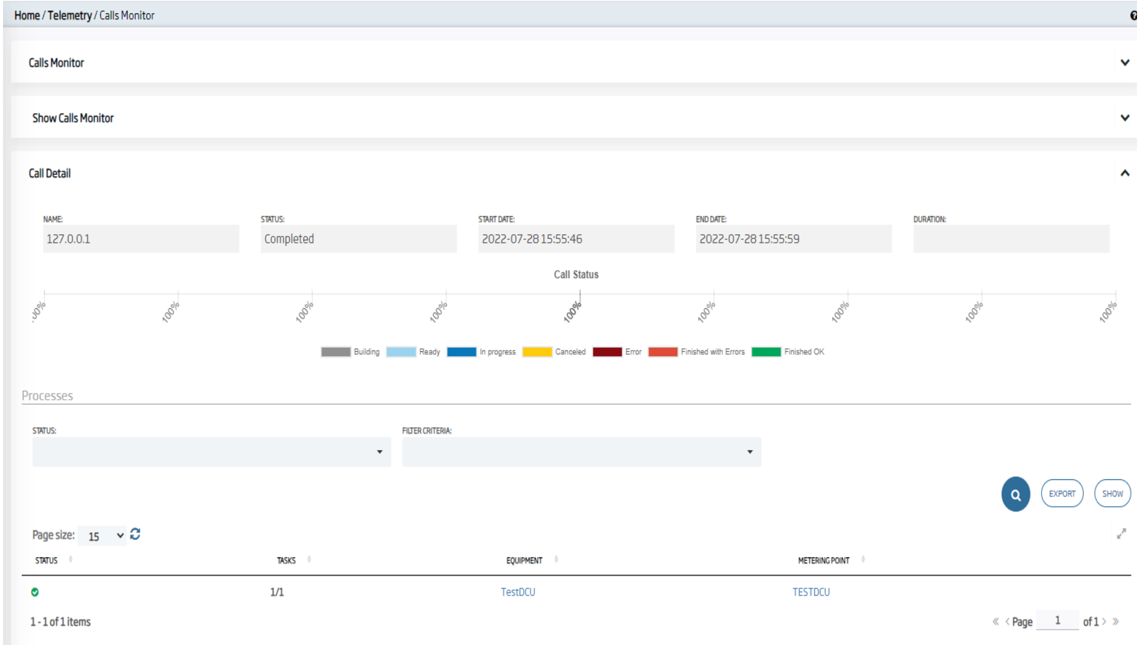
Page size: 15

STATUS	EQUIPMENTS	COMMUNICATIONS ADDRESS	START DATE	END DATE	DURATION
●	1/1	127.0.0.1	2022-07-28 15:55:46	2022-07-28 15:55:59	00:00:13

1 - 1 of 1 items

Page 1 of 1

Figure 39: Level 2, Communications and calls



Home / Telemetry / Calls Monitor

Calls Monitor

Show Calls Monitor

Call Detail

NAME:	STATUS:	START DATE:	END DATE:	DURATION:
127.0.0.1	Completed	2022-07-28 15:55:46	2022-07-28 15:55:59	

Processes

STATUS:

FILTER CRITERIA:

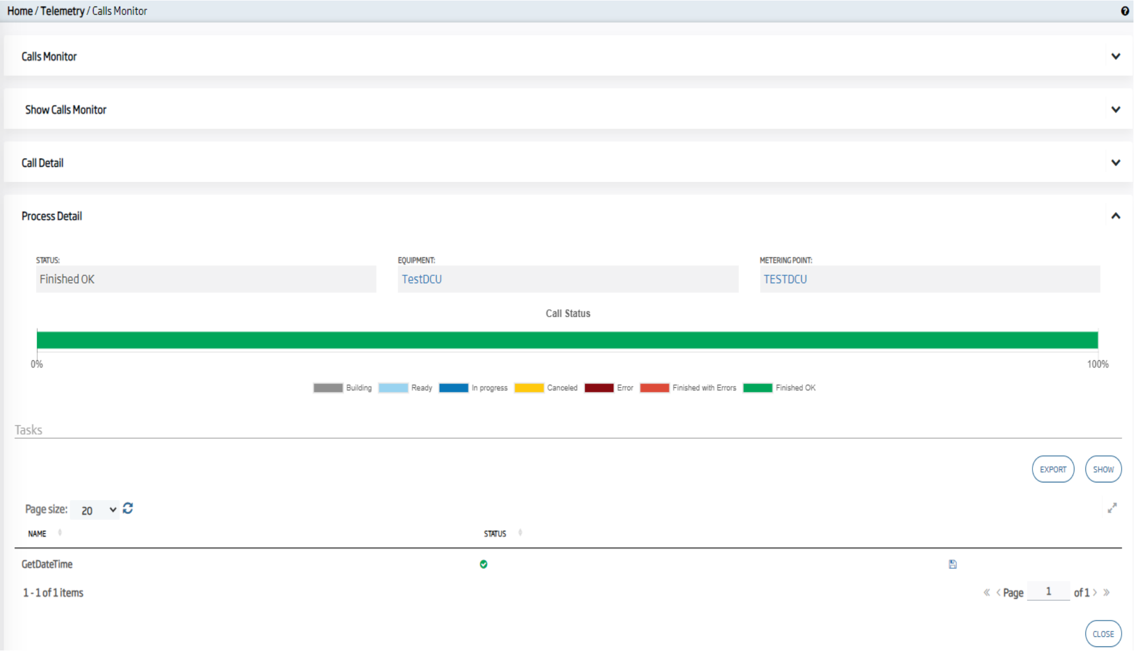
Page size: 15

STATUS	TASKS	EQUIPMENT	METERING POINT
✓	1/1	TestDCU	TESTDCU

1 - 1 of 1 items

Page 1 of 1

Figure 40: Level 3 Equipment



Home / Telemetry / Calls Monitor

Calls Monitor

Show Calls Monitor

Call Detail

Process Detail

STATUS:	EQUIPMENT:	METERING POINT:
Finished OK	TestDCU	TESTDCU

Tasks

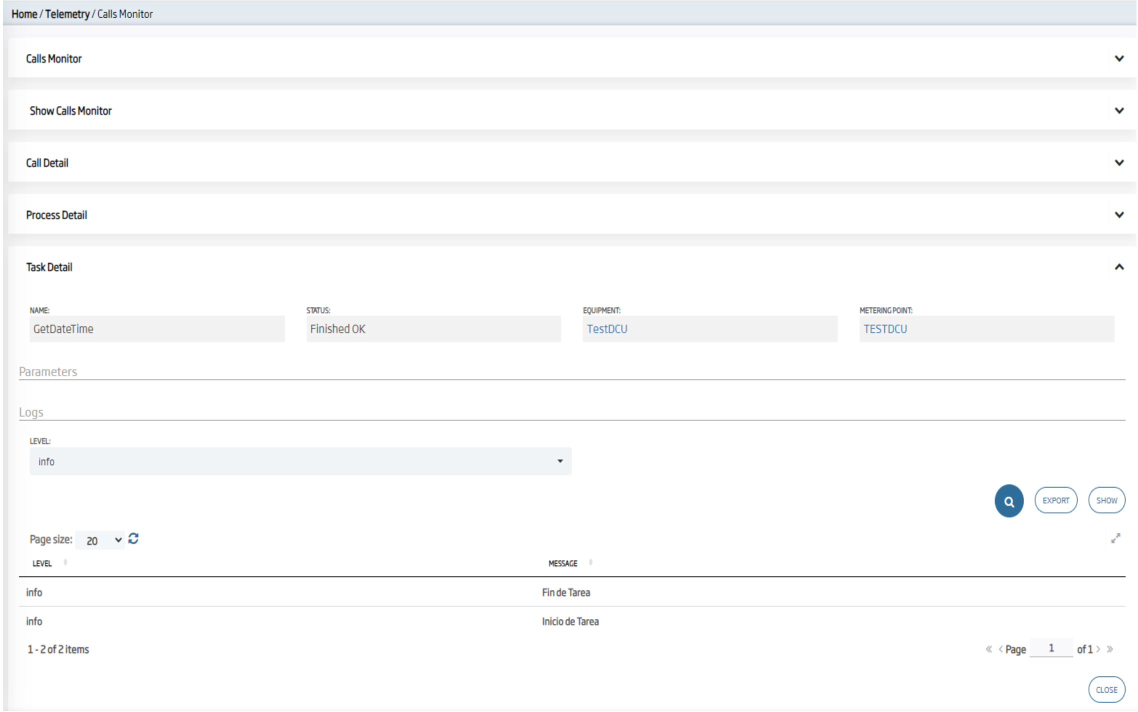
Page size: 20

NAME	STATUS
GetDateTime	✓

1 - 1 of 1 items

Page 1 of 1

Figure 41: Level 4, Petitions



NAME: GetDateTime

STATUS: Finished OK

EQUIPMENT: TestDCU

METERING POINT: TESTDCU

Parameters

Logs

LEVEL: info

Page size: 20

LEVEL	MESSAGE
info	Fin de Tarea
info	Inicio de Tarea

1 - 2 of 2 items

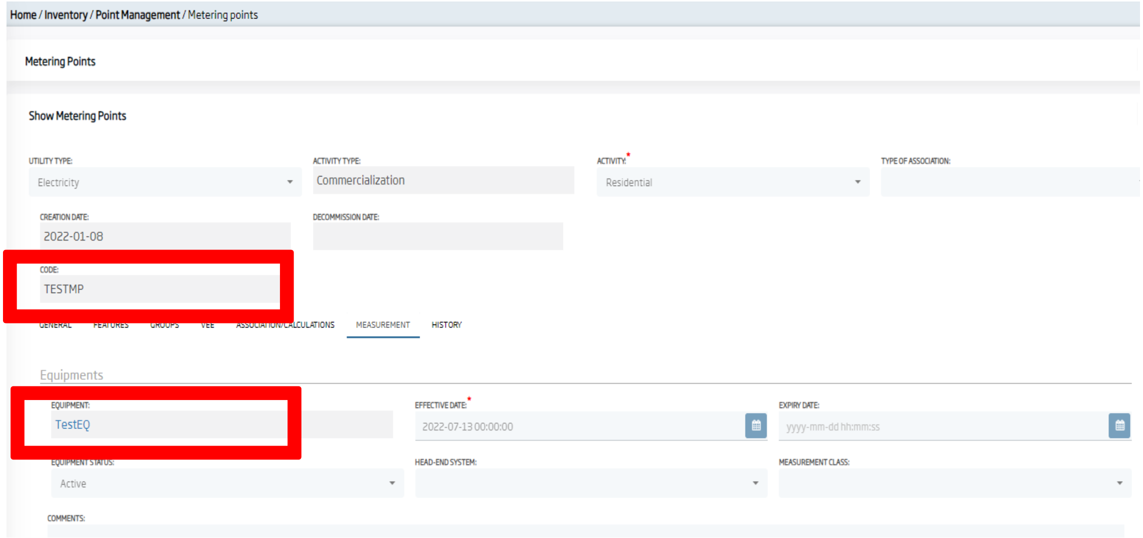
Page 1 of 1

Figure 42: Level 5, Traces

5.3. POINT OF MEASUREMENT AND EQUIPMENT

This section will describe in detail the configuration of the measurement point that has been selected for the Case Study discussed above.

The selected measurement point is a test point, designed to perform simulations to verify new functionalities that may be introduced in the tool. This point is called *"TESTMP"*.



UTILITY TYPE: Electricity

ACTIVITY TYPE: Commercialization

ACTIVITY: Residential

TYPE OF ASSOCIATION:

CREATION DATE: 2022-01-08

DECOMMISSION DATE:

CODE: TESTMP

EQUIPMENT: TestEQ

EFFECTIVE DATE: 2022-07-13 00:00:00

EXPIRY DATE: yyyy-mm-dd hh:mm:ss

EQUIPMENT STATUS: Active

HEAD-END SYSTEM:

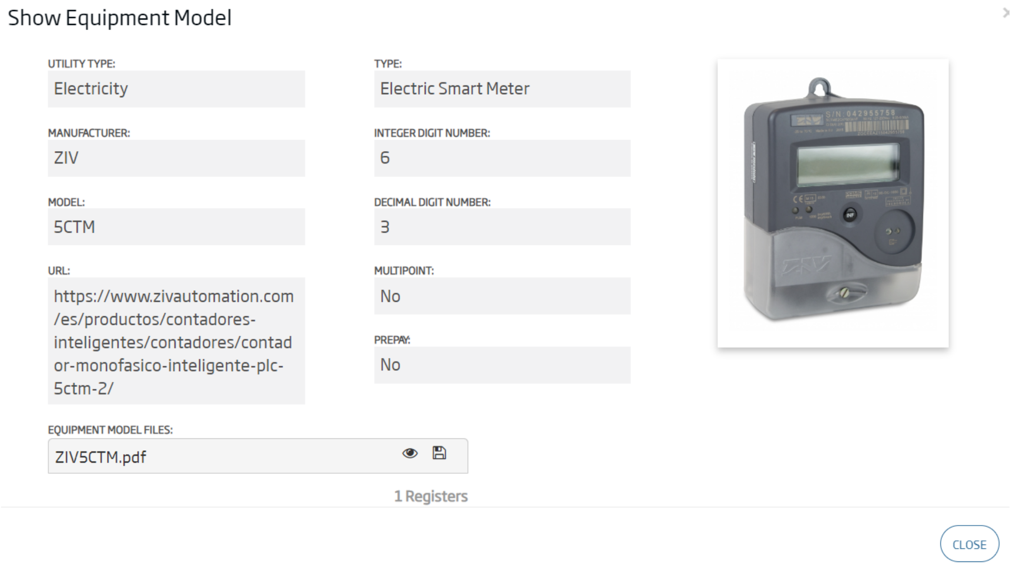
MEASUREMENT CLASS:

COMMENTS:

Figure 43: Measuring Point Selection

As can be seen in Figure 43, this measurement point is directly associated with the *"TESTEQ"* measurement equipment. This equipment is an Electric Smart Meter model 5CTM produced by the manufacturer ZIV and specifically designed to support the Prime protocol. Figure 43 shows its main characteristics and additionally it can be seen that the equipment has been located on the border between Spain and Andorra.

Furthermore, as can be seen in Figure 44, the "*TESTEQ"* unit has a Parent Unit associated with it, which is in charge of managing communications with the STG. This Parent Unit is called *"TESTDCU"*, and it is also designed to operate under the STG-DC protocol compatible with PRIME units.



Show Equipment Model

UTILITY TYPE: Electricity

TYPE: Electric Smart Meter

MANUFACTURER: ZIV

INTEGER DIGIT NUMBER: 6

MODEL: 5CTM

DECIMAL DIGIT NUMBER: 3

URL: https://www.zivautomation.com/es/productos/contadores-inteligentes/contadores/contador-monofasico-inteligente-plc-5ctm-2/

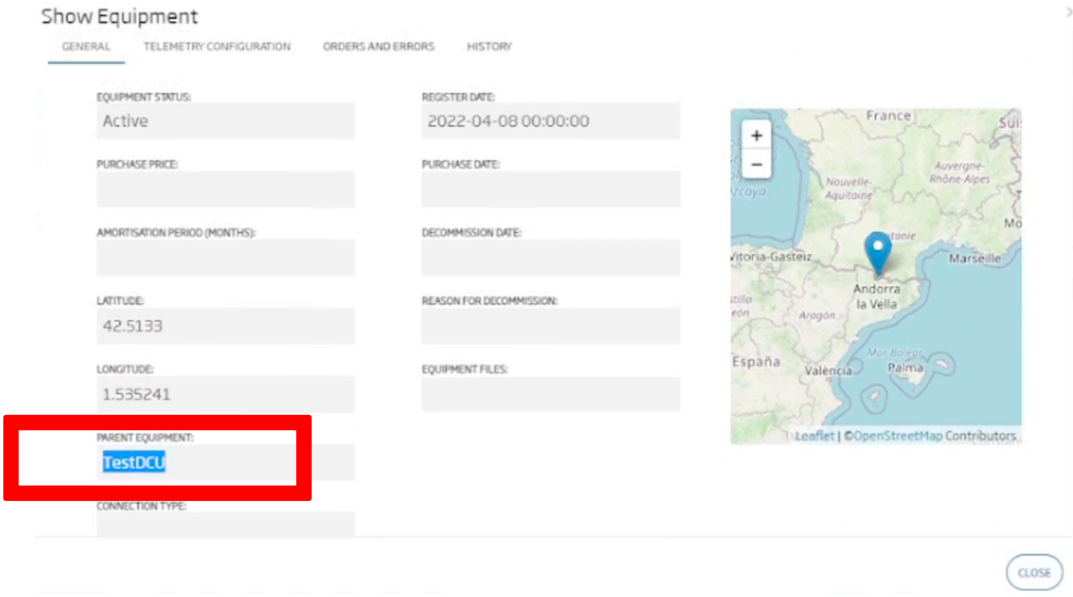
MULTIPOINT: No

PREPAY: No

EQUIPMENT MODEL FILES: ZIV5CTM.pdf

1 Registers

Figure 44: Equipment Features



Show Equipment

EQUIPMENT STATUS: Active

REGISTER DATE: 2022-04-08 00:00:00

PURCHASE PRICE:

PURCHASE DATE:

AMORTISATION PERIOD (MONTHS):

DECOMMISSION DATE:

LATITUDE: 42.5133

REASON FOR DECOMMISSION:

LONGITUDE: 1.535241

EQUIPMENT FILES:

PARENT EQUIPMENT: TestDCU

CONNECTION TYPE:

Figure 45: Equipment Communication Features

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

In this master's thesis, a functional analysis of the integration of PRIME's STG-DC protocol with Minsait's Onesait Utilities Metring technology has been developed. This functional analysis is featured by its technical component that describes in detail the structure of the communication processes and the main interactions between metering equipment in order to carry out an efficient and organised low-level programming of the different OUM modules.

This project has been approached and developed from an academic perspective within a work environment, so that the practical manual developed in it can be used for the training of professionals within the business environment. To synthesise this vision, it has been decided to schematise the communication flows in flowcharts that allow the reader to follow step by step the communication interactions in a visual and orderly manner. Therefore, it can be concluded that all the project objectives listed in chapter 1 have been covered, from the study of the STG-DC specification and the Onesait tool to the development of these diagrams.

Additionally, a case study of the execution of a "Manual Request" has been developed to provide the reader with greater visibility of both the user experience and the communication interactions between the different entities involved in the communication process.

Under my point of view, as we move forward, the processing of massive data is becoming more and more important, and the electricity sector is no stranger to this. For this reason, integrating tools such as Onesait Utilities Metering plays an important role due to the value they add to the business. OUM acquires measurements from multiple sources and protocols, incorporates them into a unified repository and processes them, increasing energy efficiency, reducing operating costs and the risk of fraud, and locates anomalous behaviour in the network, reducing the risk of outages.

Finally, the next step to take as Future Works would be to put this manual into practice and start programming requests in OUM under the STG-DC protocol. It is proposed to carry out a modular work in which the scripts of each request can be used to programme the adjoining requests in the same way as has been done with the flowcharts, making a coherent classification.

CHAPTER 7: BIBLIOGRAPHY

[1] C. R. Bayliss and B. J. Hardy, “Smart Grids,” in *Transmission and Distribution Electrical Engineering*, Elsevier, 2012, pp. 1059–1074.

[2] T. Ribera, “Hacia un cambio de paradigma del sector eléctrico,” *Cinco Días*, 16-Mar-2020. [Online]. Available: https://cincodias.elpais.com/cincodias/2020/03/13/opinion/1584118203_706646.html.

[3] hdplc_alliance, “What is Power Line Communication (PLC)?,” *HD-PLC Alliance*, 16-Dec-2021. [Online]. Available: https://hd-plc.org/what-is-power-line-communication-plc/.

[4] *Biblus.us.es*. [Online]. Available: https://biblus.us.es/bibing/proyectos/abreproy/11138/fichero/memoria%252FCap%C3%ADtulo+5.pdf.

[5] “PRIME alliance,” *PRIME Alliance | Advanced Meter Management & Smart Grid*, 05-Aug-2016. [Online]. Available: https://www.prime-alliance.org. [Accessed: 18-Aug-2022].

[6] A. Haidine, A. Tabone and J. Müller, "Deployment of power line communication by European utilities in advanced metering infrastructure," *2013 IEEE 17th International Symposium on Power Line Communications and Its Applications*, 2013, pp. 126-130, doi: 10.1109/ISPLC.2013.6525837.

[7] O. D. Javier, “Análisis del potencial de una estructura ciberfísica para telegestión de contadores inteligentes basada en software comercial” Leganes, Febrero 2017.

[8] A. Martínez Sánchez, «Desarrollo de un módulo de análisis de mapas Shapefile para un simulador de redes PRIME,» Proyecto Fin de Carrera, Leganés, 2016.

[9] ITU-T Standard G.9903, «Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks», 2014.

[10] C. M. Jesús Miguel, “PFC- Physical Layer PRIME Chapter 2”

[11] Villasante M. Borja, “Analysis and preparation of the deployment of Prime 1.4 Smart Meters in an Interoperable Field Enviroment for 15000 Meters” Madrid, Septiembre 2021

[12] Iberdrola Distribución Eléctrica SA, “STG-DC Interface Specification v3.4”, Oct 2015

[13] A. Sanz, "Evolution of PRIME to PLC-RF hybrid systems," *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, 2019, pp. 74-79, doi: 10.1109/GPECOM.2019.8778506.

[14] “Onesait Utilities Metering,” *Onesait Utilities Metering*. [Online]. Available: https://www.onesait.com/products/metering.

[15] M. J. Gamez, “Objetivos y metas de desarrollo sostenible,” *Desarrollo Sostenible*, 17-Sep-2015. [Online]. Available: https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/.

[16] *Indracompany.com*. [Online]. Available: https://www.indracompany.com/es/sostenibilidad/conoce-ods.

[17] Stamatis Karnouskos, Orestis Terzidis, Panagiotis Karnouskos, “ An Advanced Metering Infrastructure for Future Energy Networks" 2007

ANNEX I: SUSTAINABLE DEVELOPMENT GOALS

The Sustainable Development Goals (SDGs) are a list of 17 goals defined in the 2030 Agenda adopted by the United Nations in 2015. They are a universal call to end poverty, protect the planet and improve the living conditions of people around the world.

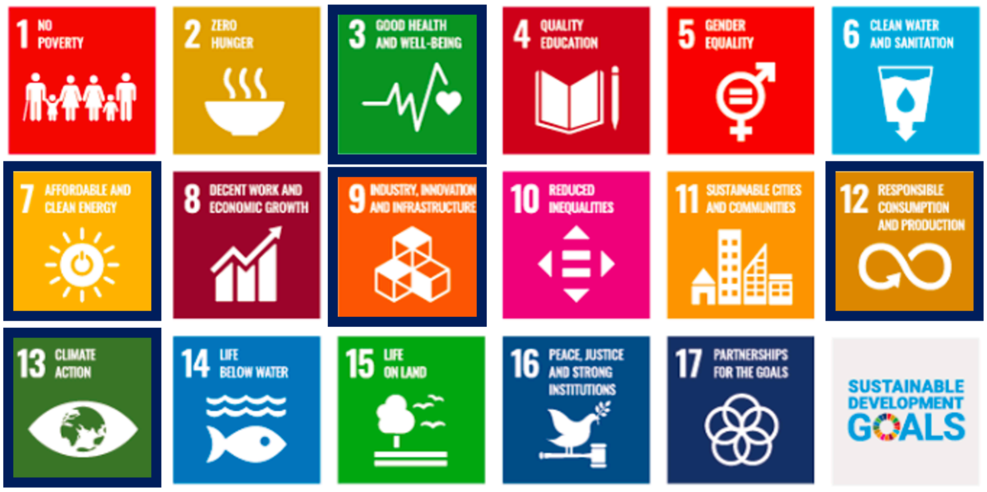

1 NO POVERTY
2 ZERO HUNGER
3 GOOD HEALTH AND WELL-BEING
4 QUALITY EDUCATION
5 GENDER EQUALITY
6 CLEAN WATER AND SANITATION
7 AFFORDABLE AND CLEAN ENERGY
8 DECENT WORK AND ECONOMIC GROWTH
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE
10 REDUCED INEQUALITIES
11 SUSTAINABLE CITIES AND COMMUNITIES
12 RESPONSIBLE CONSUMPTION AND PRODUCTION
13 CLIMATE ACTION
14 LIFE BELOW WATER
15 LIFE ON LAND
16 PEACE, JUSTICE AND STRONG INSTITUTIONS
17 PARTNERSHIPS FOR THE GOALS
SUSTAINABLE DEVELOPMENT GOALS

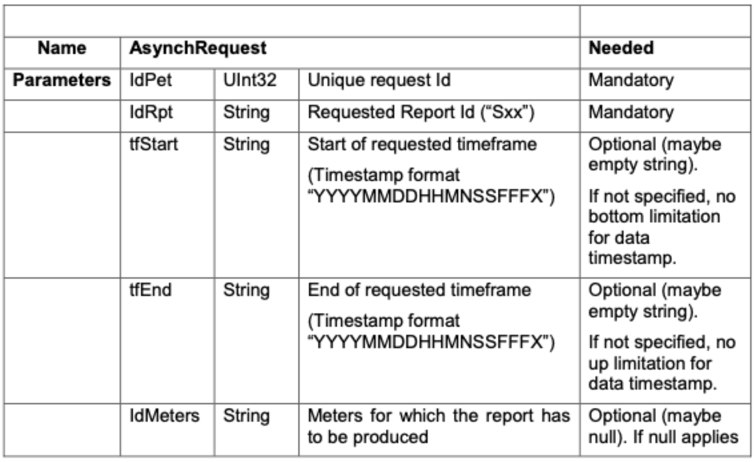
Figure 46: Sustainable Development Goals [15].

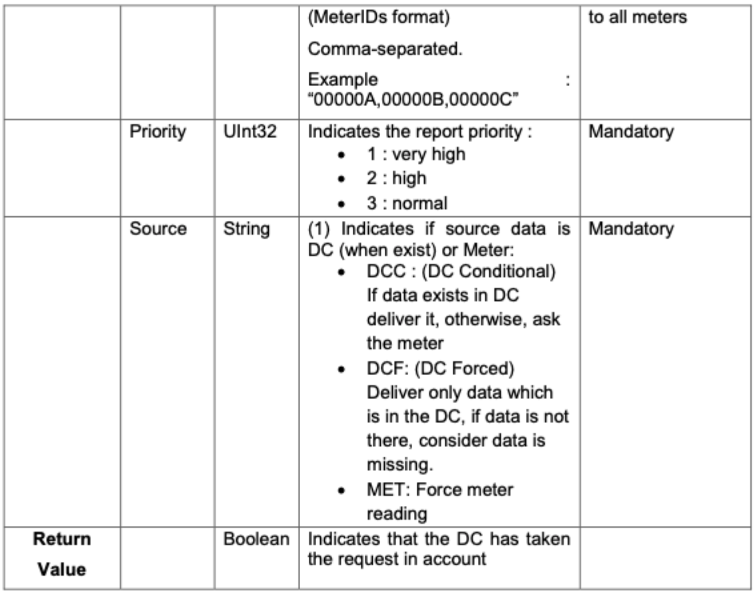
At Indra, the Sustainable Development Goals have been incorporated into its business strategy. "Technology is key to achieving the SDGs. That is why we develop technological solutions to address the social and environmental challenges facing our society. We are committed to developing technology that puts people at the center, respects their rights and is environmentally friendly."[16].

With regard to this project, it contributes to accelerating the energy transition, promoting the decarbonisation of the economy by developing the technologies necessary for utilities to meet this challenge. It is also working on the digitalization of the energy sector, which will result in a service of greater value for the consumer. The objectives that are most identified with this project are marked.

ANNEX II: WEB & STG SERVICES SPECIFICATIONS

- **AsynchReport Request**



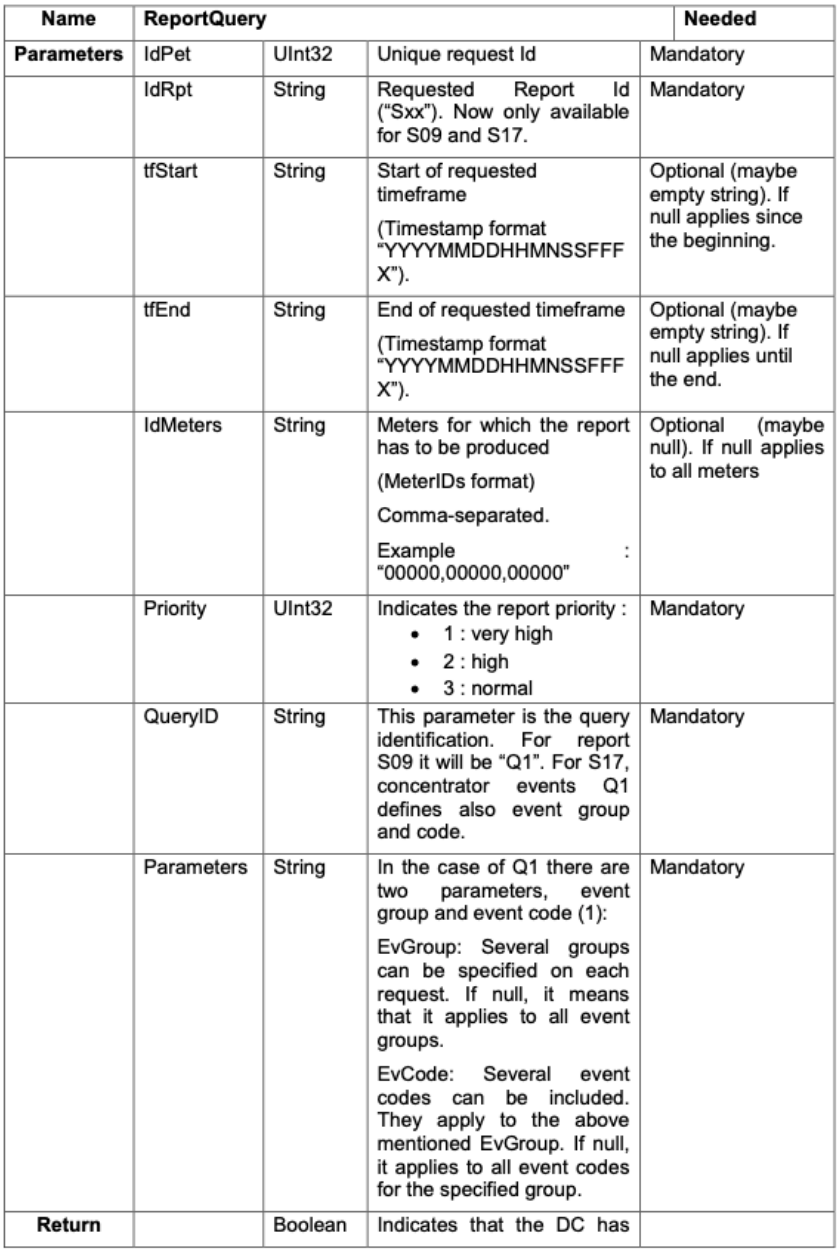


Name	**AsynchRequest**			**Needed**
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	IdRpt	String	Requested Report Id (“Sxx”)	Mandatory
	tfStart	String	Start of requested timeframe (Timestamp format “YYYYMMDDHHMNSSFFFX”)	Optional (maybe empty string). If not specified, no bottom limitation for data timestamp.
	tfEnd	String	End of requested timeframe (Timestamp format “YYYYMMDDHHMNSSFFFX”)	Optional (maybe empty string). If not specified, no up limitation for data timestamp.
	IdMeters	String	Meters for which the report has to be produced (MeterIDs format) Comma-separated. Example : “00000A,00000B,00000C”	Optional (maybe null). If null applies to all meters
	Priority	UInt32	Indicates the report priority : • 1 : very high • 2 : high • 3 : normal	Mandatory
	Source	String	(1) Indicates if source data is DC (when exist) or Meter: • DCC : (DC Conditional) If data exists in DC deliver it, otherwise, ask the meter • DCF: (DC Forced) Deliver only data which is in the DC, if data is not there, consider data is missing. • MET: Force meter reading	Mandatory
Return Value		Boolean	Indicates that the DC has taken the request in account	

(1) A source="MET" goes directly to the meter. While the report will contain the meter data, the DC does not store the data when source="MET". It is effectively an STG to meter pass-through. A source="DCC" however will go to the meter for any data not found on the DC and then store the results prior to sending the report. When the DC receives any errors from the meters) in previous DCC request (or scheduled tasks), in the next DCC request, the DC

must go to the meter for request all information needed to build the requested report correctly.

- **Report Query**

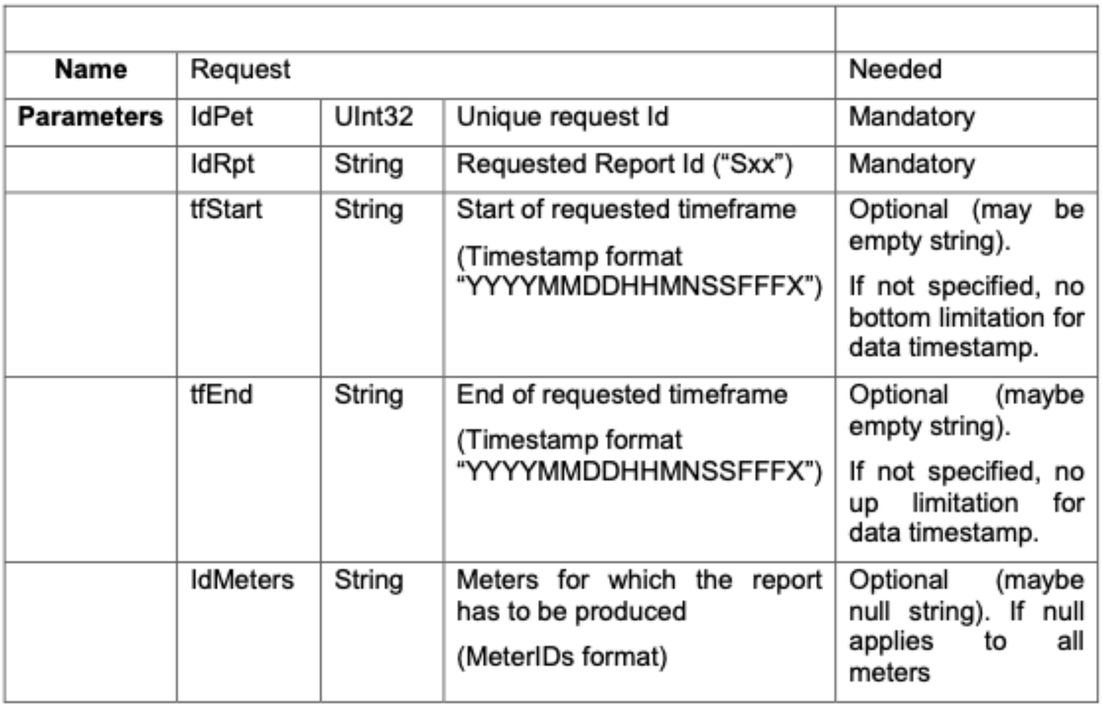
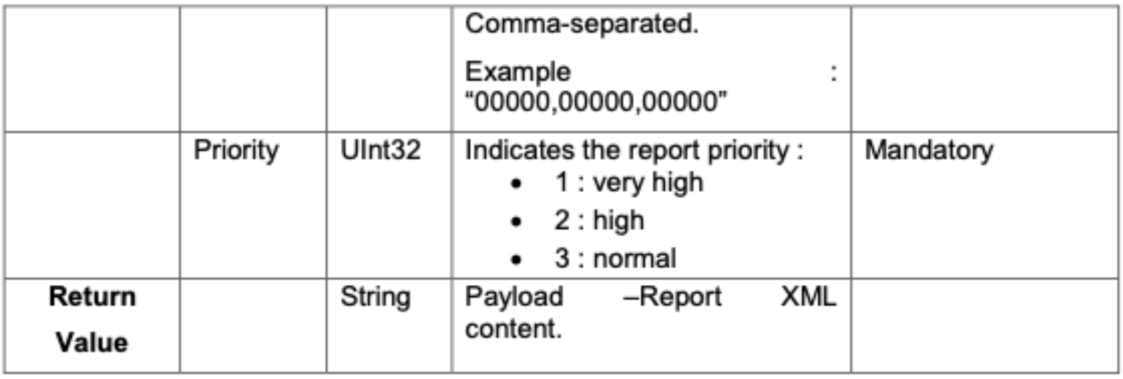


Name	ReportQuery			Needed
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	IdRpt	String	Requested Report Id ("Sxx"). Now only available for S09 and S17.	Mandatory
	tfStart	String	Start of requested timeframe (Timestamp format "YYYYMMDDHHMNSSFFFX").	Optional (maybe empty string). If null applies since the beginning.
	tfEnd	String	End of requested timeframe (Timestamp format "YYYYMMDDHHMNSSFFFX").	Optional (maybe empty string). If null applies until the end.
	IdMeters	String	Meters for which the report has to be produced (MeterIDs format) Comma-separated. Example : "00000,00000,00000"	Optional (maybe null). If null applies to all meters
	Priority	UInt32	Indicates the report priority : • 1 : very high • 2 : high • 3 : normal	Mandatory
	QueryID	String	This parameter is the query identification. For report S09 it will be "Q1". For S17, concentrator events Q1 defines also event group and code.	Mandatory
	Parameters	String	In the case of Q1 there are two parameters, event group and event code (1): EvGroup: Several groups can be specified on each request. If null, it means that it applies to all event groups. EvCode: Several event codes can be included. They apply to the above mentioned EvGroup. If null, it applies to all event codes for the specified group.	Mandatory
Return		Boolean	Indicates that the DC has	

(1) In case of Q1, the ";" is the separator for the two entries, the ":" is a separator between code and value. The "," is the separator for specify possible several groups/codes inside

EvGroup/EvCode parameter. The "+" is the separator for adding more than one filtering criteria.

- **SynchReport Request**

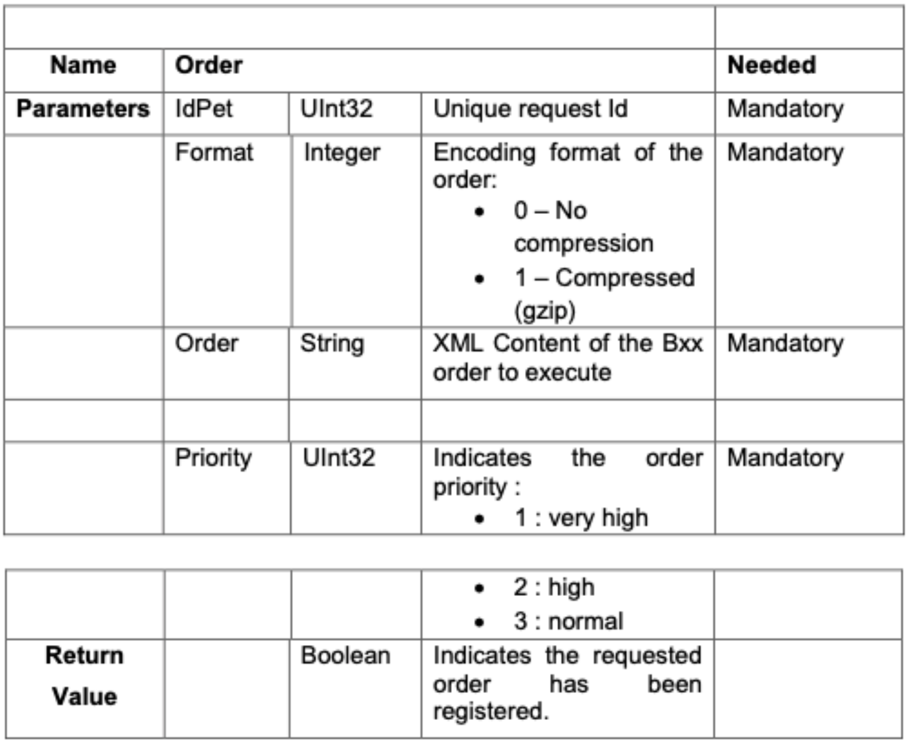



Name	Request			Needed
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	IdRpt	String	Requested Report Id ("Sxx")	Mandatory
	tfStart	String	Start of requested timeframe (Timestamp format "YYYYMMDDHHMNSSFFFX")	Optional (may be empty string). If not specified, no bottom limitation for data timestamp.
	tfEnd	String	End of requested timeframe (Timestamp format "YYYYMMDDHHMNSSFFFX")	Optional (maybe empty string). If not specified, no up limitation for data timestamp.
	IdMeters	String	Meters for which the report has to be produced (MeterIDs format) Comma-separated. Example : "00000,00000,00000"	Optional (maybe null string). If null applies to all meters
	Priority	UInt32	Indicates the report priority : • 1 : very high • 2 : high • 3 : normal	Mandatory
Return Value		String	Payload –Report XML content.	

Reports are considered as any request for information of either a meter or the concentrator.

The report has to be generated as the result of the service invocation: the information is given on synchronous mode.

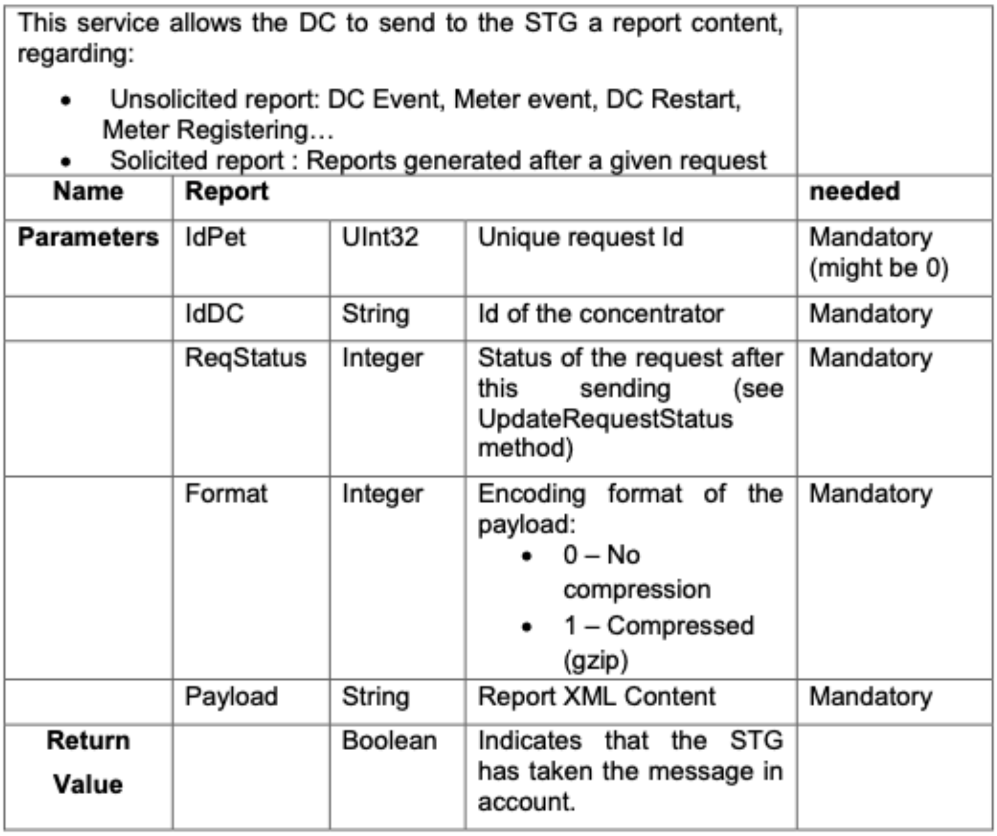
- **Order Request**



Name	**Order**			**Needed**
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	Format	Integer	Encoding format of the order: • 0 – No compression • 1 – Compressed (gzip)	Mandatory
	Order	String	XML Content of the Bxx order to execute	Mandatory
	Priority	UInt32	Indicates the order priority : • 1 : very high • 2 : high • 3 : normal	Mandatory
Return Value		Boolean	Indicates the requested order has been registered.	

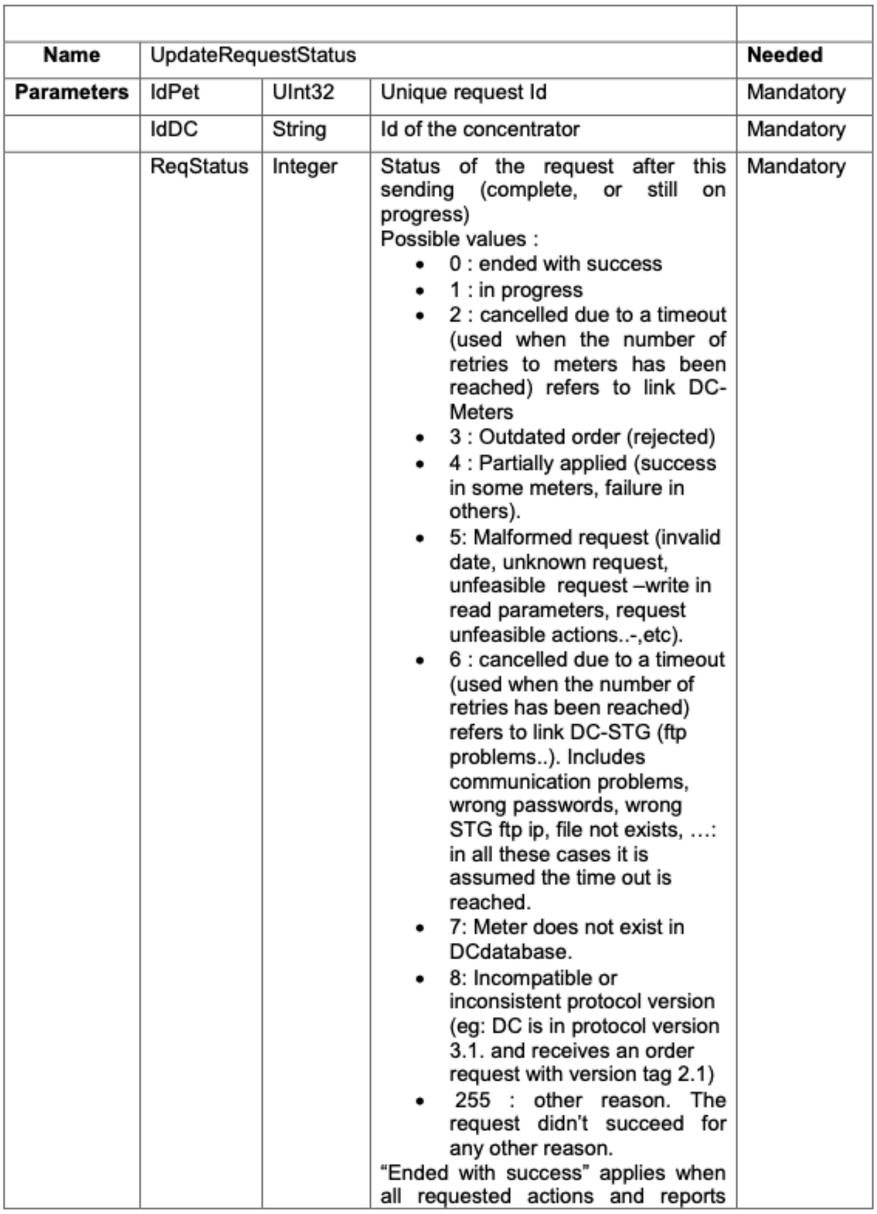
Orders are considered as those messages which imply a change (parameter, command execution...) either a meter or the concentrator.

- **STG Service: Report**

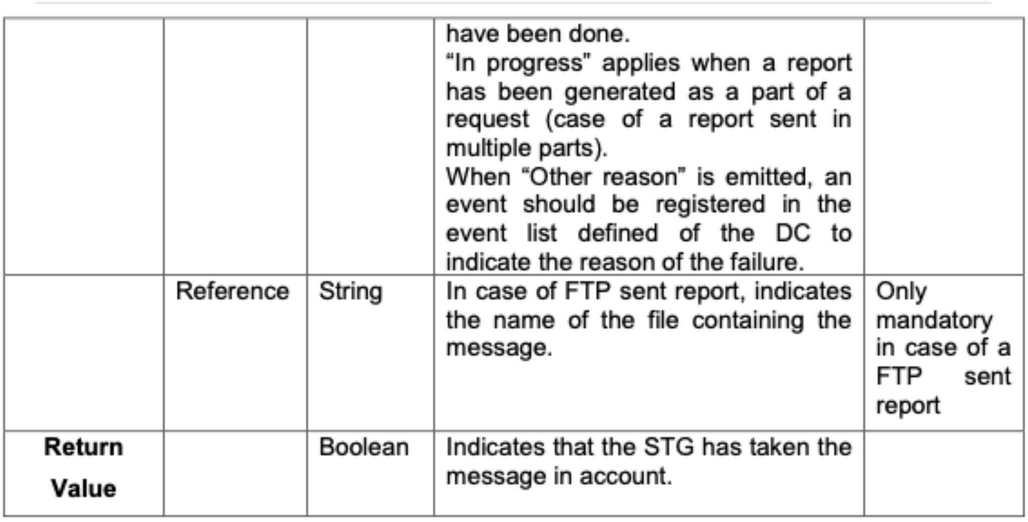


This service allows the DC to send to the STG a report content, regarding: • Unsolicited report: DC Event, Meter event, DC Restart, Meter Registering… • Solicited report : Reports generated after a given request				
Name	**Report**			**needed**
Parameters	IdPet	UInt32	Unique request Id	Mandatory (might be 0)
	IdDC	String	Id of the concentrator	Mandatory
	ReqStatus	Integer	Status of the request after this sending (see UpdateRequestStatus method)	Mandatory
	Format	Integer	Encoding format of the payload: • 0 – No compression • 1 – Compressed (gzip)	Mandatory
	Payload	String	Report XML Content	Mandatory
Return Value		Boolean	Indicates that the STG has taken the message in account.	

- **STG Services: UpdateRequestStatus**

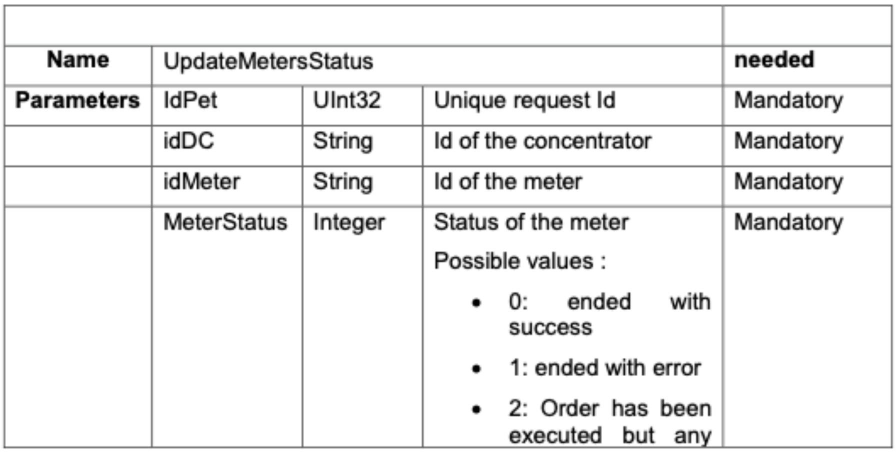


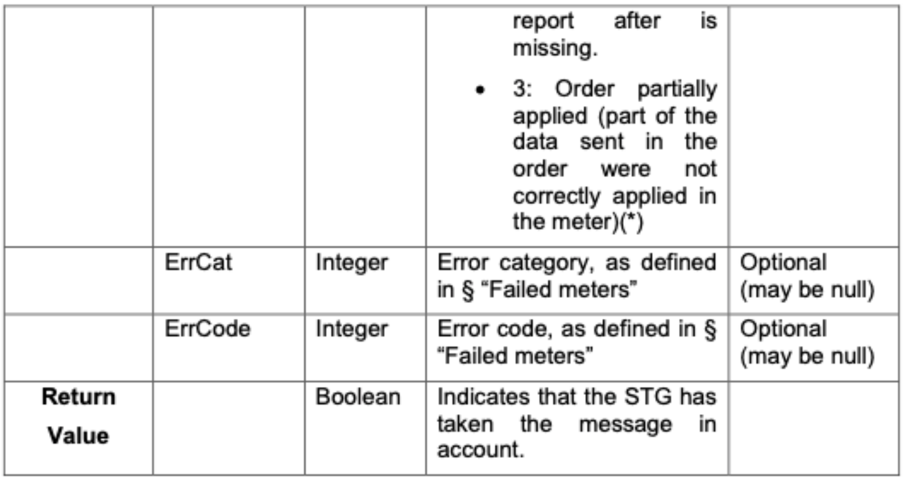
Name	UpdateRequestStatus			**Needed**
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	IdDC	String	Id of the concentrator	Mandatory
	ReqStatus	Integer	Status of the request after this sending (complete, or still on progress) Possible values : • 0 : ended with success • 1 : in progress • 2 : cancelled due to a timeout (used when the number of retries to meters has been reached) refers to link DC-Meters • 3 : Outdated order (rejected) • 4 : Partially applied (success in some meters, failure in others). • 5: Malformed request (invalid date, unknown request, unfeasible request –write in read parameters, request unfeasible actions..-,etc). • 6 : cancelled due to a timeout (used when the number of retries has been reached) refers to link DC-STG (ftp problems..). Includes communication problems, wrong passwords, wrong STG ftp ip, file not exists, …: in all these cases it is assumed the time out is reached. • 7: Meter does not exist in DCdatabase. • 8: Incompatible or inconsistent protocol version (eg: DC is in protocol version 3.1. and receives an order request with version tag 2.1) • 255 : other reason. The request didn't succeed for any other reason. "Ended with success" applies when all requested actions and reports	Mandatory



			have been done. "In progress" applies when a report has been generated as a part of a request (case of a report sent in multiple parts). When "Other reason" is emitted, an event should be registered in the event list defined of the DC to indicate the reason of the failure.	
	Reference	String	In case of FTP sent report, indicates the name of the file containing the message.	Only mandatory in case of a FTP sent report
Return Value		Boolean	Indicates that the STG has taken the message in account.	

- **STG Services: UpdateMetersStatus**





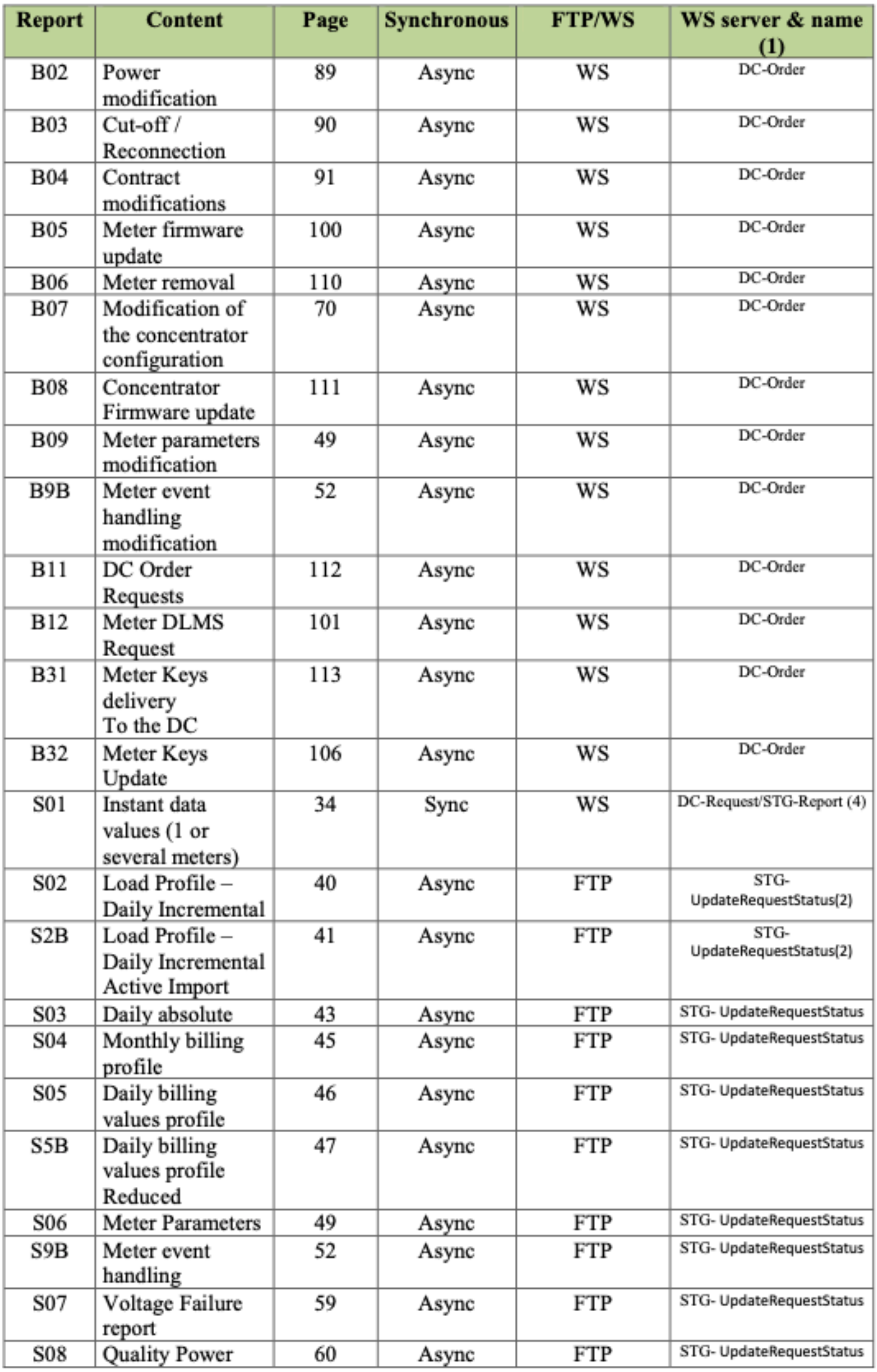
Name	UpdateMetersStatus			**needed**
Parameters	IdPet	UInt32	Unique request Id	Mandatory
	idDC	String	Id of the concentrator	Mandatory
	idMeter	String	Id of the meter	Mandatory
	MeterStatus	Integer	Status of the meter Possible values : • 0: ended with success • 1: ended with error • 2: Order has been executed but any report after is missing. • 3: Order partially applied (part of the data sent in the order were not correctly applied in the meter)(*)	Mandatory
	ErrCat	Integer	Error category, as defined in § "Failed meters"	Optional (may be null)
	ErrCode	Integer	Error code, as defined in § "Failed meters"	Optional (may be null)
Return Value		Boolean	Indicates that the STG has taken the message in account.	

(*)Covers the case of sending several parameters change the meter, when some parameter are changed but for different reasons another not.

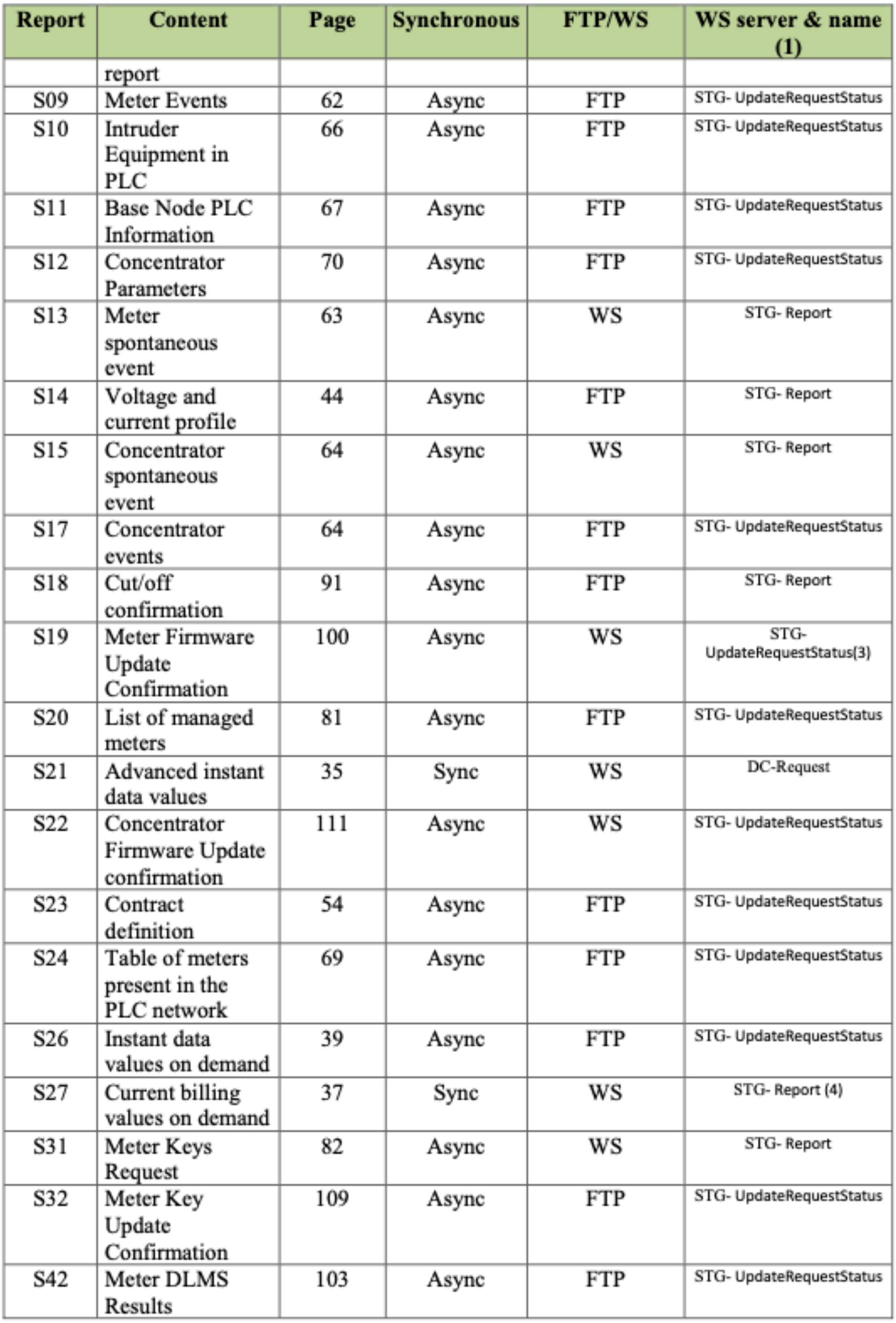
ANNEX III: REPORT INDEX

The following table represents for each order and report:

- The Data flow model:
  - "Asynch" for the asynchronous data flow
  - "Synch" for the synchronous data flow
- The underlying protocol to be used:
  - "WS" for WebServices
  - "FTP" for File Transfer Protocol



Report	Content	Page	Synchronous	FTP/WS	WS server & name (1)
B02	Power modification	89	Async	WS	DC-Order
B03	Cut-off / Reconnection	90	Async	WS	DC-Order
B04	Contract modifications	91	Async	WS	DC-Order
B05	Meter firmware update	100	Async	WS	DC-Order
B06	Meter removal	110	Async	WS	DC-Order
B07	Modification of the concentrator configuration	70	Async	WS	DC-Order
B08	Concentrator Firmware update	111	Async	WS	DC-Order
B09	Meter parameters modification	49	Async	WS	DC-Order
B9B	Meter event handling modification	52	Async	WS	DC-Order
B11	DC Order Requests	112	Async	WS	DC-Order
B12	Meter DLMS Request	101	Async	WS	DC-Order
B31	Meter Keys delivery To the DC	113	Async	WS	DC-Order
B32	Meter Keys Update	106	Async	WS	DC-Order
S01	Instant data values (1 or several meters)	34	Sync	WS	DC-Request/STG-Report (4)
S02	Load Profile – Daily Incremental	40	Async	FTP	STG-UpdateRequestStatus(2)
S2B	Load Profile – Daily Incremental Active Import	41	Async	FTP	STG-UpdateRequestStatus(2)
S03	Daily absolute	43	Async	FTP	STG- UpdateRequestStatus
S04	Monthly billing profile	45	Async	FTP	STG- UpdateRequestStatus
S05	Daily billing values profile	46	Async	FTP	STG- UpdateRequestStatus
S5B	Daily billing values profile Reduced	47	Async	FTP	STG- UpdateRequestStatus
S06	Meter Parameters	49	Async	FTP	STG- UpdateRequestStatus
S9B	Meter event handling	52	Async	FTP	STG- UpdateRequestStatus
S07	Voltage Failure report	59	Async	FTP	STG- UpdateRequestStatus
S08	Quality Power	60	Async	FTP	STG- UpdateRequestStatus



Report	Content	Page	Synchronous	FTP/WS	WS server & name (1)
	report				
S09	Meter Events	62	Async	FTP	STG- UpdateRequestStatus
S10	Intruder Equipment in PLC	66	Async	FTP	STG- UpdateRequestStatus
S11	Base Node PLC Information	67	Async	FTP	STG- UpdateRequestStatus
S12	Concentrator Parameters	70	Async	FTP	STG- UpdateRequestStatus
S13	Meter spontaneous event	63	Async	WS	STG- Report
S14	Voltage and current profile	44	Async	FTP	STG- Report
S15	Concentrator spontaneous event	64	Async	WS	STG- Report
S17	Concentrator events	64	Async	FTP	STG- UpdateRequestStatus
S18	Cut/off confirmation	91	Async	FTP	STG- Report
S19	Meter Firmware Update Confirmation	100	Async	WS	STG- UpdateRequestStatus(3)
S20	List of managed meters	81	Async	FTP	STG- UpdateRequestStatus
S21	Advanced instant data values	35	Sync	WS	DC-Request
S22	Concentrator Firmware Update confirmation	111	Async	WS	STG- UpdateRequestStatus
S23	Contract definition	54	Async	FTP	STG- UpdateRequestStatus
S24	Table of meters present in the PLC network	69	Async	FTP	STG- UpdateRequestStatus
S26	Instant data values on demand	39	Async	FTP	STG- UpdateRequestStatus
S27	Current billing values on demand	37	Sync	WS	STG- Report (4)
S31	Meter Keys Request	82	Async	WS	STG- Report
S32	Meter Key Update Confirmation	109	Async	FTP	STG- UpdateRequestStatus
S42	Meter DLMS Results	103	Async	FTP	STG- UpdateRequestStatus

(1): In the case of messages through ftp, the name of the Web Service refers to the WS which is invoked for notifying that the ftp file has been uploaded.

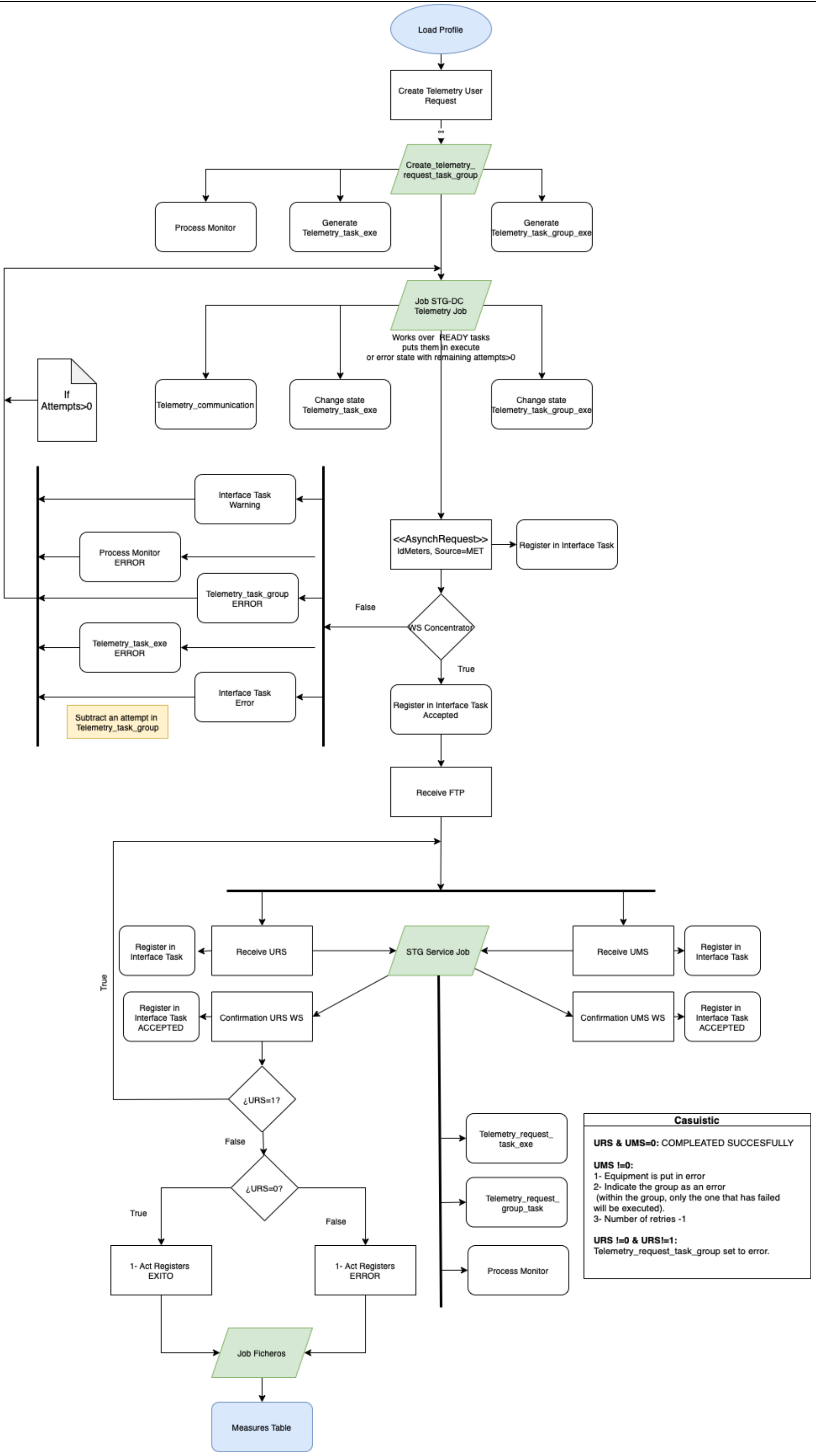
(2): This WS applies either if the report is sent as a result of a request or if the report is sent as a result of a scheduled task in the DC.

(3): UpdateMetersStatus can be called for meter during request for processing by the DC. This WS admits several meters included (for the same status reporting).

(4): When this report S01 or S27 is part of an order request answer from the DC, it will be uploaded asynchronous with WS-Report.

ANNEX IV: FLOW DIAGRAMS

Index

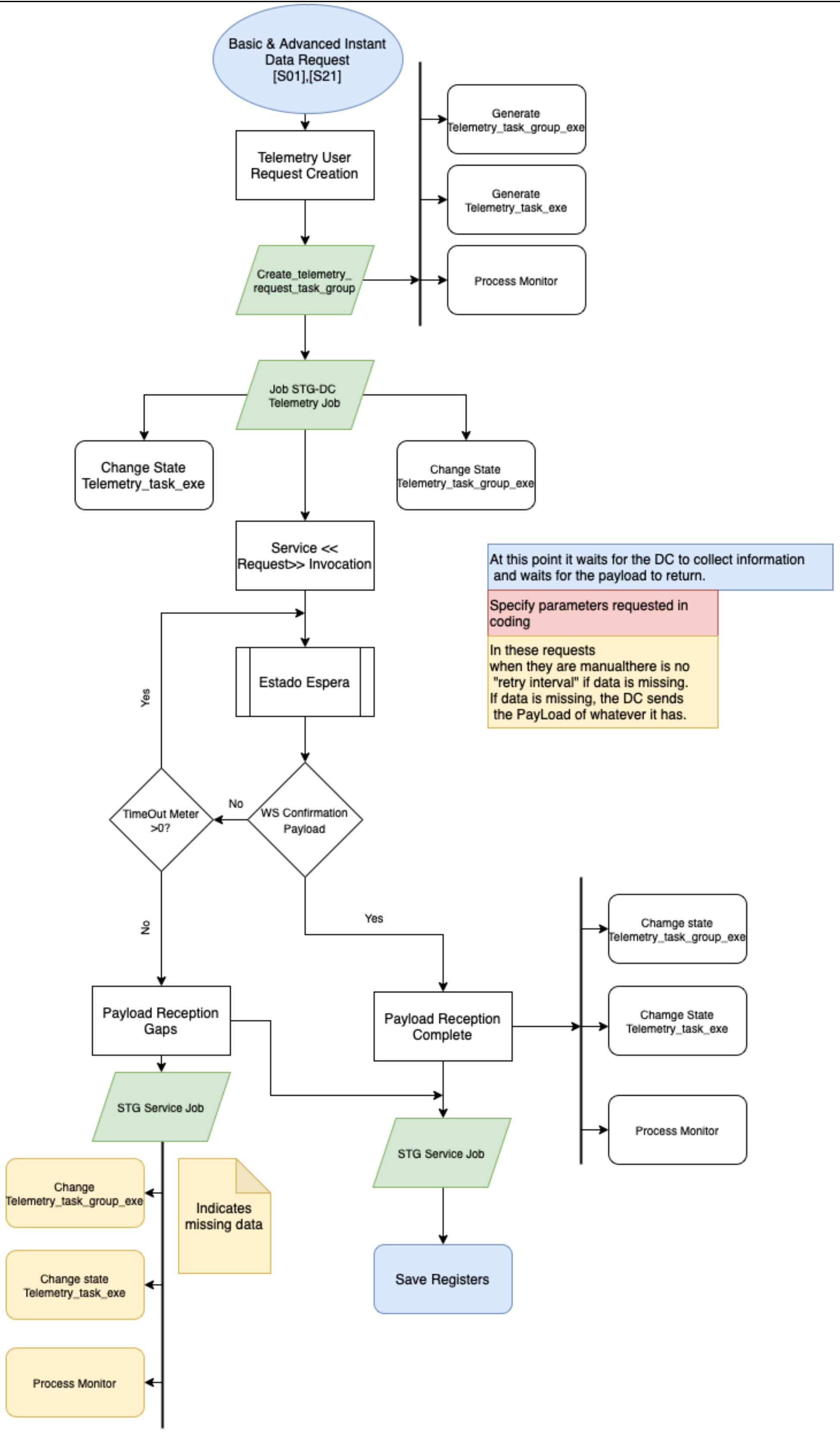

Load Profile
Create Telemetry User Request
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Job STG-DC Telemetry Job
Works over READY tasks puts them in execute or error state with remaining attempts>0
If Attempts>0
Telemetry_communication
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Interface Task Warning
Process Monitor ERROR
Telemetry_task_group ERROR
Telemetry_task_exe ERROR
Interface Task Error
Subtract an attempt in Telemetry_task_group
<<AsynchRequest>> IdMeters, Source=MET
Register in Interface Task
False
WS Concentrator
True
Register in Interface Task Accepted
Receive FTP
True
Register in Interface Task
Receive URS
STG Service Job
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
¿URS=1?
False
¿URS=0?
True
False
1- Act Registers EXITO
1- Act Registers ERROR
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
Job Ficheros
Measures Table

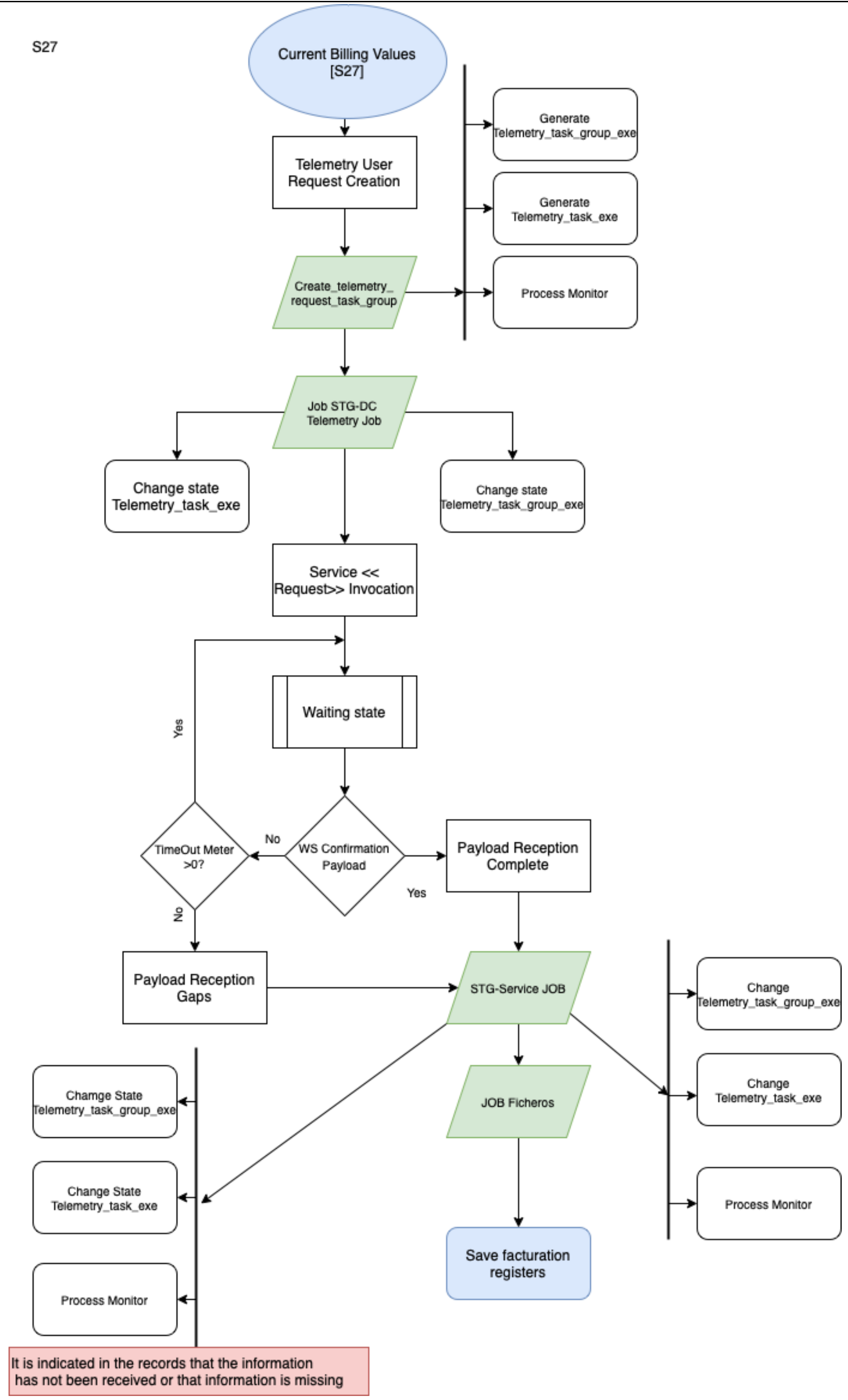
Casuistic

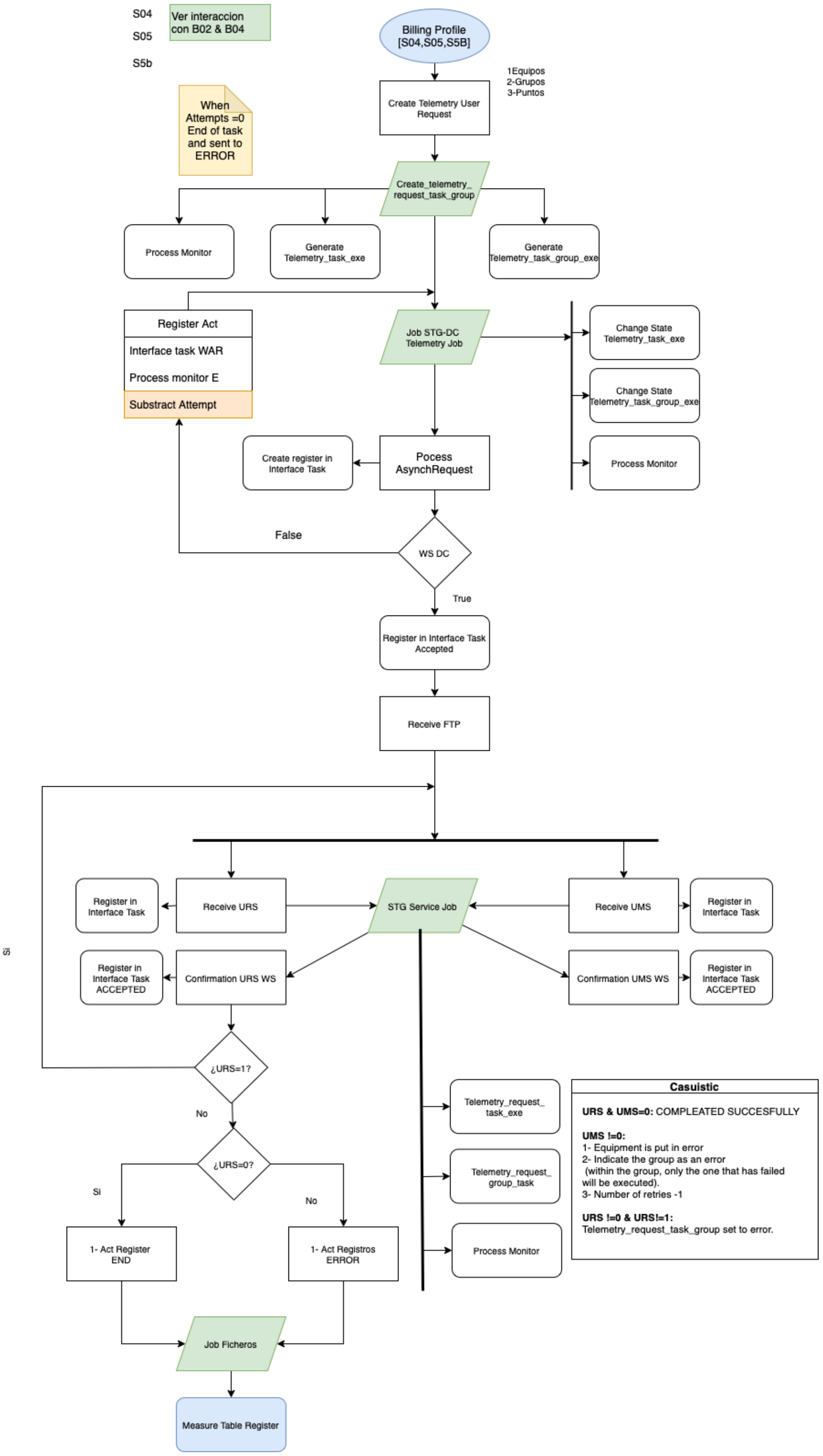
URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.


Basic & Advanced Instant Data Request [S01],[S21]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_ request_task_group
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Service << Request>> Invocation
At this point it waits for the DC to collect information and waits for the payload to return.
Specify parameters requested in coding
In these requests when they are manualthere is no "retry interval" if data is missing. If data is missing, the DC sends the PayLoad of whatever it has.
Estado Espera
Yes
No
TimeOut Meter >0?
WS Confirmation Payload
No
Yes
Chamge state Telemetry_task_group_exe
Chamge State Telemetry_task_exe
Process Monitor
Payload Reception Gaps
Payload Reception Complete
STG Service Job
STG Service Job
Change Telemetry_task_group_exe
Indicates missing data
Change state Telemetry_task_exe
Process Monitor
Save Registers


S27
Current Billing Values [S27]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_ request_task_group
Job STG-DC Telemetry Job
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Service << Request>> Invocation
Yes
Waiting state
TimeOut Meter >0?
No
WS Confirmation Payload
Yes
Payload Reception Complete
No
Payload Reception Gaps
STG-Service JOB
Change Telemetry_task_group_exe
Change Telemetry_task_exe
Process Monitor
Chamge State Telemetry_task_group_exe
Change State Telemetry_task_exe
Process Monitor
JOB Ficheros
Save facturation registers
It is indicated in the records that the information has not been received or that information is missing

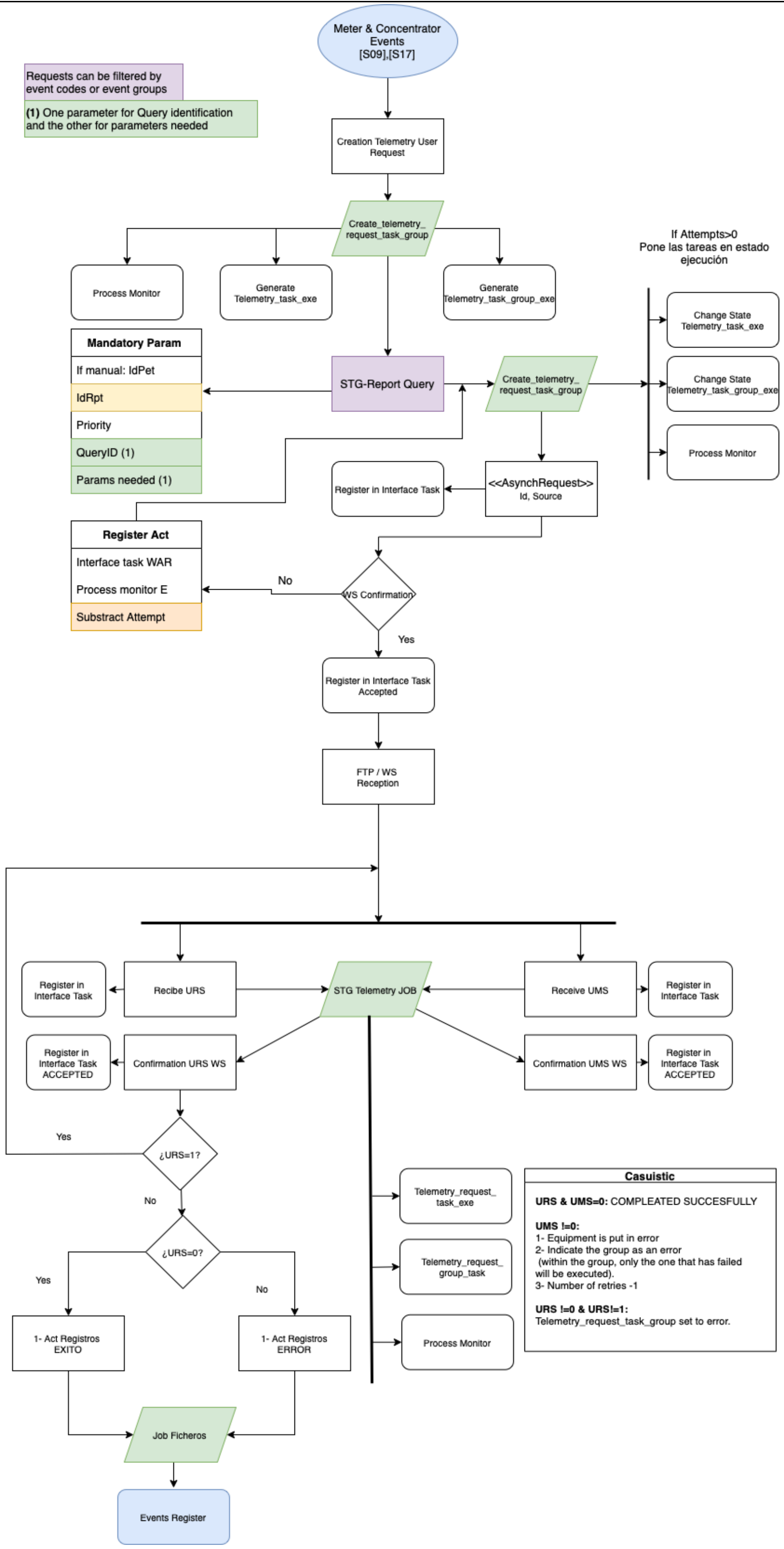

S04
S05
S5b
Ver interaccion con B02 & B04
Billing Profile [S04,S05,S5B]
1Equipos
2-Grupos
3-Puntos
When Attempts =0 End of task and sent to ERROR
Create Telemetry User Request
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
Pocess AsynchRequest
False
WS DC
True
Register in Interface Task Accepted
Receive FTP
Si
Register in Interface Task
Receive URS
STG Service Job
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
¿URS=1?
No
¿URS=0?
Si
No
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
1- Act Register END
1- Act Registros ERROR
Job Ficheros
Measure Table Register

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

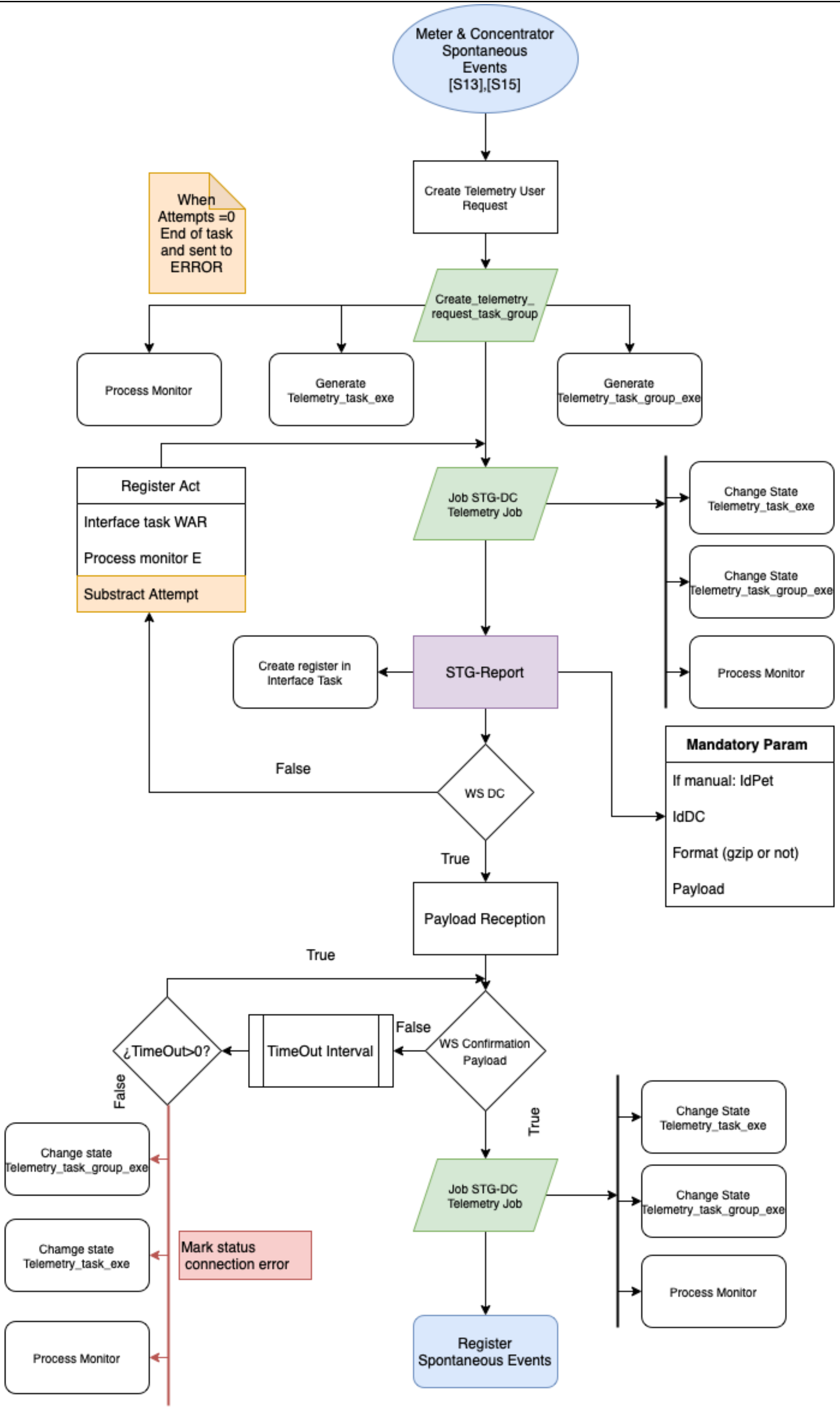

Meter & Concentrator Events [S09],[S17]
Requests can be filtered by event codes or event groups
(1) One parameter for Query identification and the other for parameters needed
Creation Telemetry User Request
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
If Attempts>0
Pone las tareas en estado ejecución
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Mandatory Param
If manual: IdPet
IdRpt
Priority
QueryID (1)
Params needed (1)
STG-Report Query
Create_telemetry_request_task_group
Register in Interface Task
<<AsynchRequest>>
Id, Source
Register Act
Interface task WAR
Process monitor E
Substract Attempt
No
WS Confirmation
Yes
Register in Interface Task Accepted
FTP / WS Reception
Register in Interface Task
Recibe URS
STG Telemetry JOB
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
Yes
¿URS=1?
No
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
Yes
¿URS=0?
No
1- Act Registros EXITO
1- Act Registros ERROR
Job Ficheros
Events Register

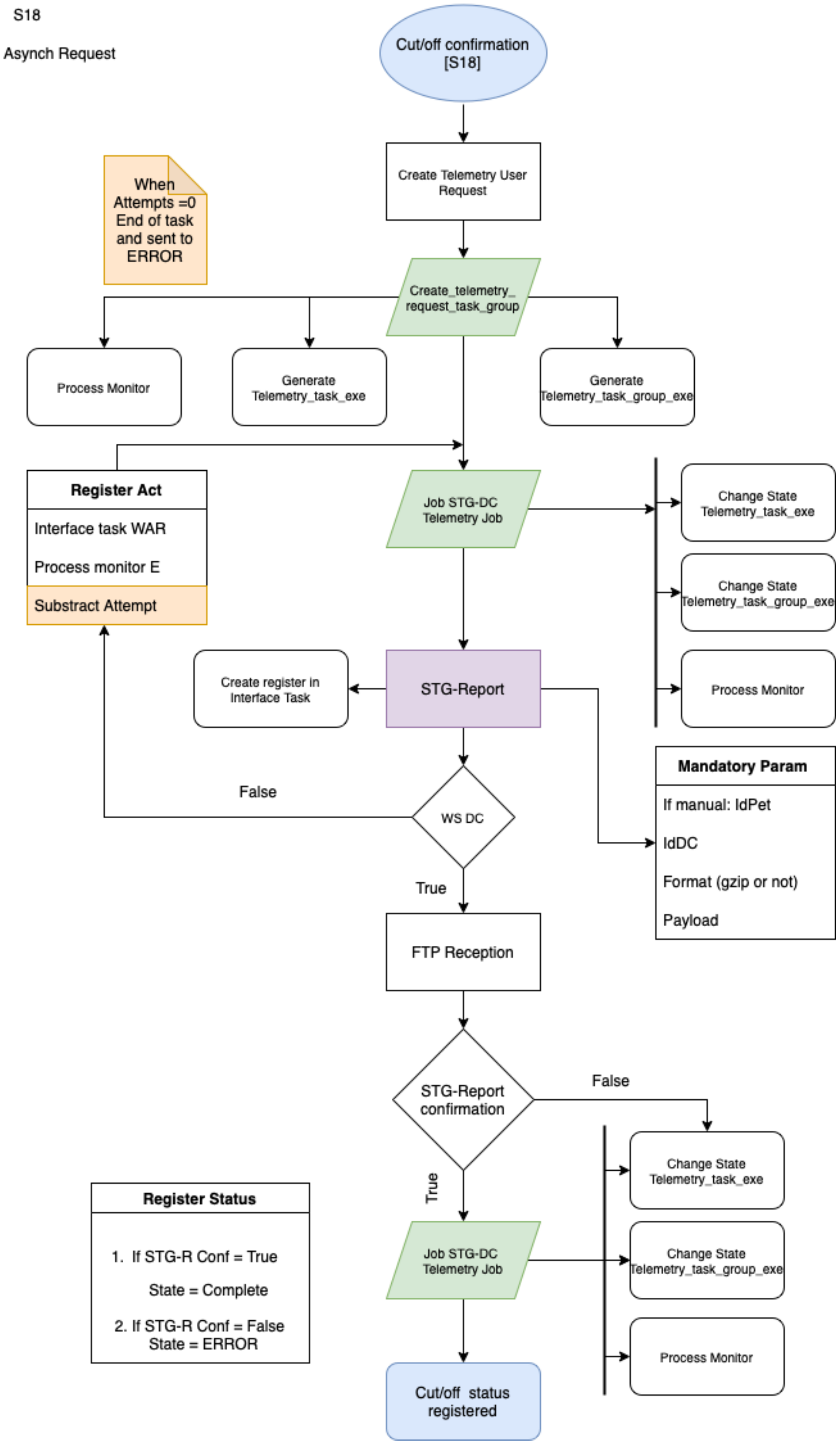
Casuistic

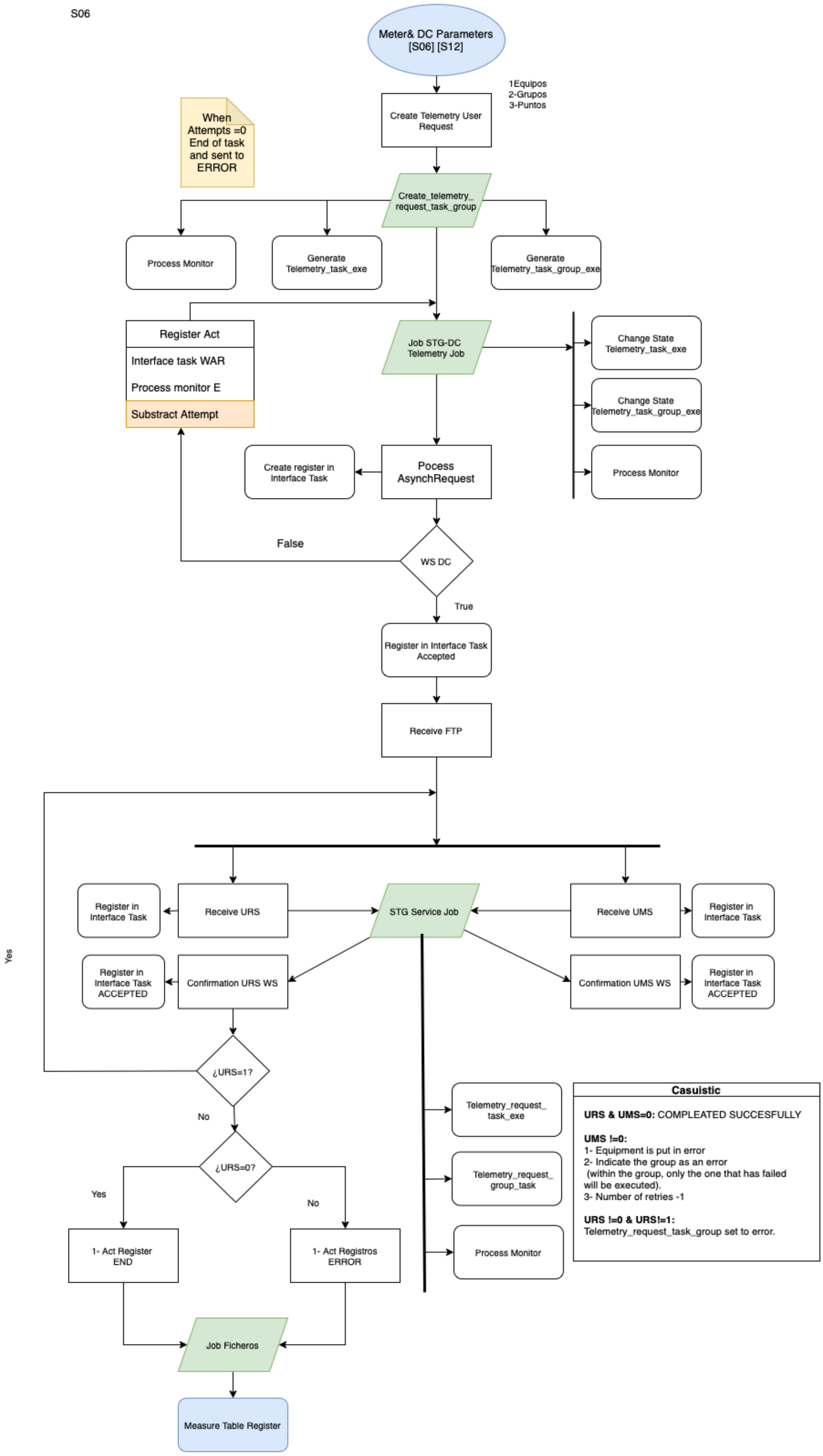
URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.


Meter & Concentrator Spontaneous Events [S13],[S15]
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
STG-Report
Mandatory Param
If manual: IdPet
IdDC
Format (gzip or not)
Payload
False
WS DC
True
Payload Reception
True
¿TimeOut>0?
TimeOut Interval
False
WS Confirmation Payload
False
True
Change state Telemetry_task_group_exe
Chamge state Telemetry_task_exe
Process Monitor
Mark status connection error
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Register Spontaneous Events


S18
Asynch Request
Cut/off confirmation [S18]
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
STG-Report
Mandatory Param
If manual: IdPet
IdDC
Format (gzip or not)
Payload
False
WS DC
True
FTP Reception
STG-Report confirmation
False
True
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Job STG-DC Telemetry Job
Register Status
1. If STG-R Conf = True
State = Complete
2. If STG-R Conf = False
State = ERROR
Cut/off status registered


S06
Meter& DC Parameters [S06] [S12]
1Equipos
2-Grupos
3-Puntos
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
Pocess AsynchRequest
False
WS DC
True
Register in Interface Task Accepted
Receive FTP
Yes
Register in Interface Task
Receive URS
STG Service Job
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
¿URS=1?
No
¿URS=0?
Yes
No
1- Act Register END
1- Act Registros ERROR
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
Job Ficheros
Measure Table Register

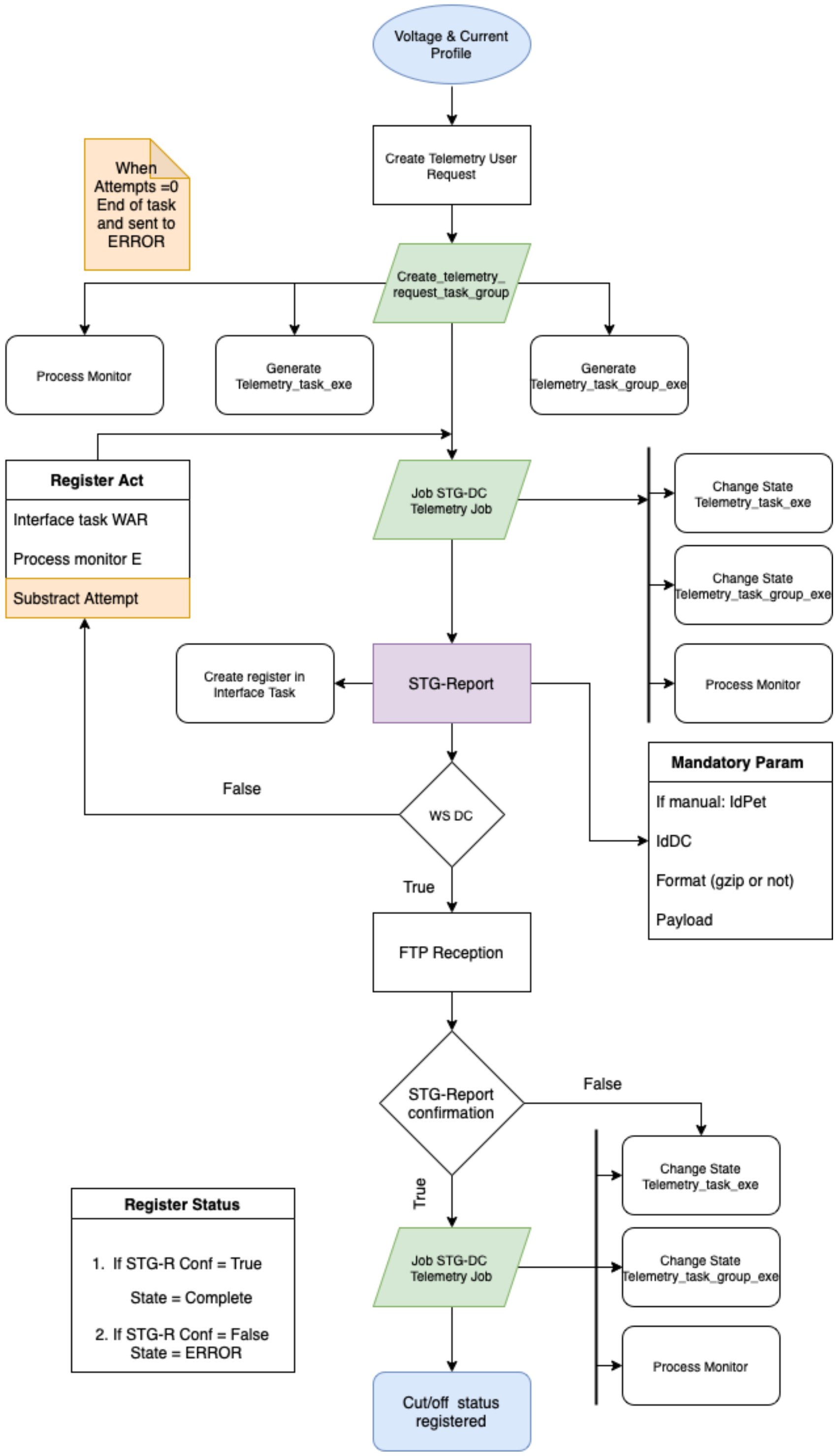
Casuistic

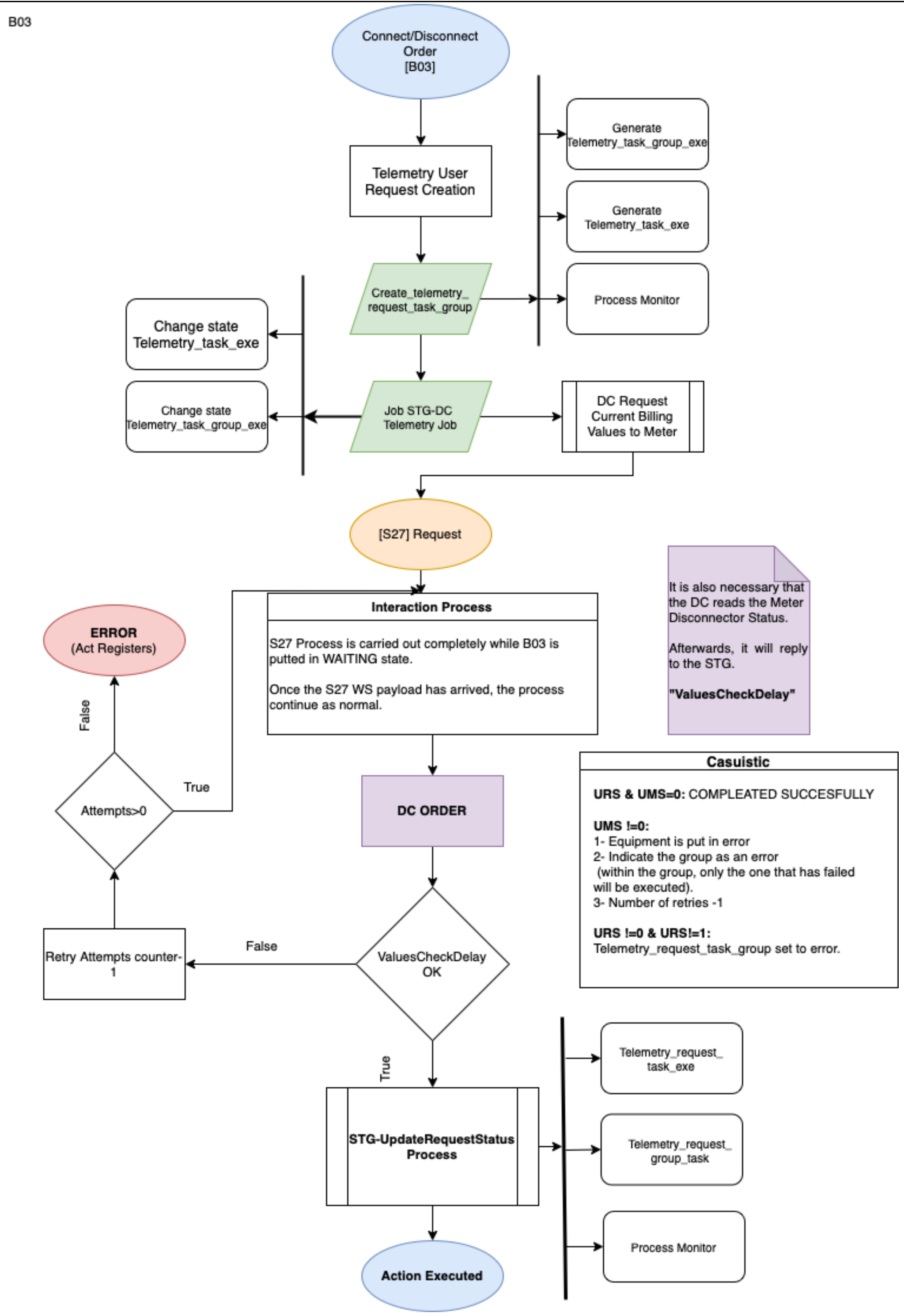
URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

S14


Voltage & Current Profile
Create Telemetry User Request
When Attempts =0 End of task and sent to ERROR
Create_telemetry_request_task_group
Process Monitor
Generate Telemetry_task_exe
Generate Telemetry_task_group_exe
Register Act
Interface task WAR
Process monitor E
Substract Attempt
Job STG-DC Telemetry Job
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Create register in Interface Task
STG-Report
Mandatory Param
If manual: IdPet
IdDC
Format (gzip or not)
Payload
False
WS DC
True
FTP Reception
STG-Report confirmation
False
True
Change State Telemetry_task_exe
Change State Telemetry_task_group_exe
Process Monitor
Register Status
1. If STG-R Conf = True
State = Complete
2. If STG-R Conf = False
State = ERROR
Job STG-DC Telemetry Job
Cut/off status registered


B03
Connect/Disconnect Order [B03]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_ request_task_group
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC Request Current Billing Values to Meter
[S27] Request
ERROR (Act Registers)
False
True
Attempts>0
DC ORDER
Retry Attempts counter-1
False
ValuesCheckDelay OK
True
STG-UpdateRequestStatus Process
Telemetry_request_ task_exe
Telemetry_request_ group_task
Process Monitor
Action Executed

Interaction Process

S27 Process is carried out completely while B03 is putted in WAITING state.

Once the S27 WS payload has arrived, the process continue as normal.

It is also necessary that the DC reads the Meter Disconnector Status.

Afterwards, it will reply to the STG.

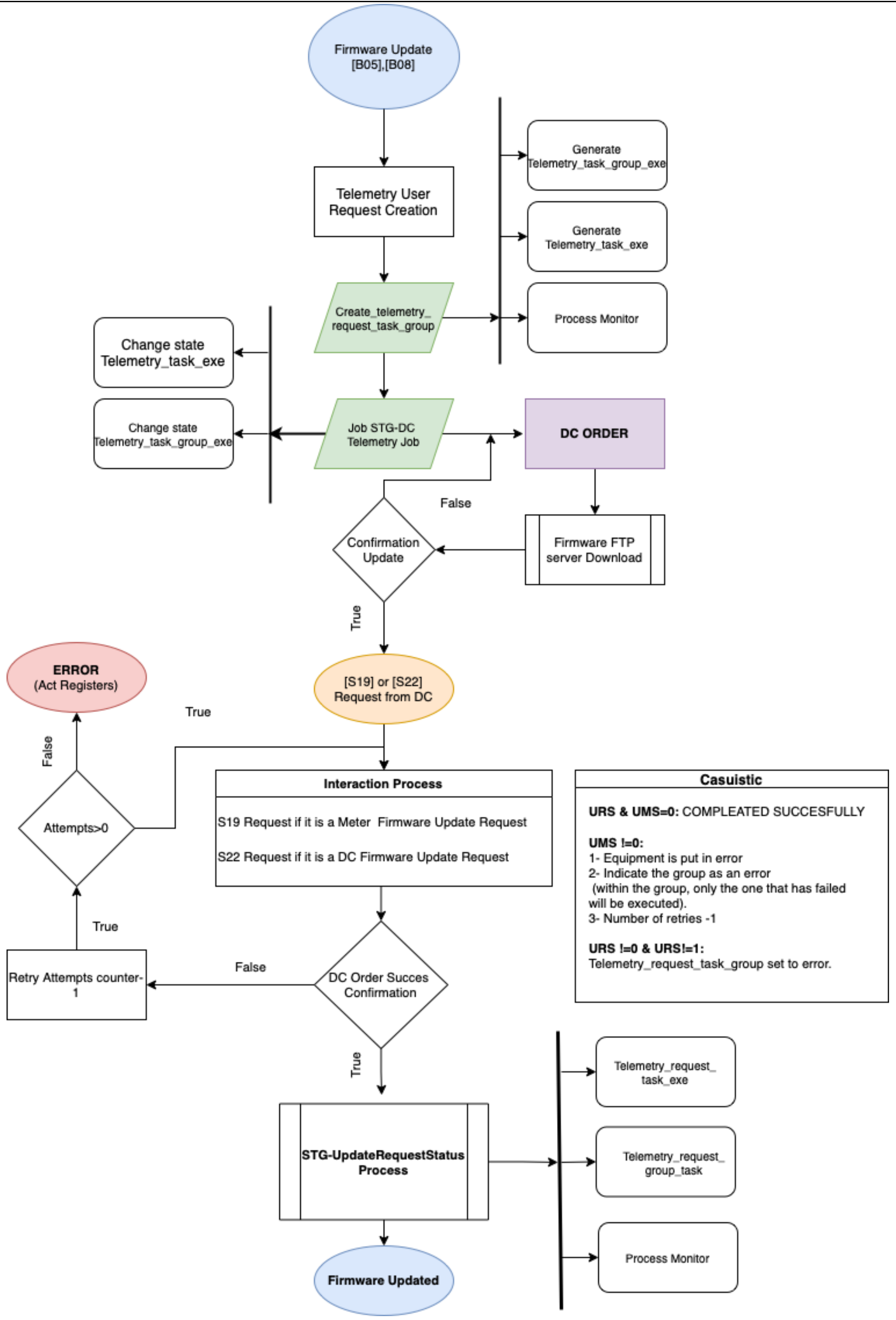
"ValuesCheckDelay"

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error (within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.


Firmware Update
[B05],[B08]
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_request_task_group
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC ORDER
False
Confirmation Update
Firmware FTP server Download
True
ERROR
(Act Registers)
[S19] or [S22] Request from DC
True
False
Attempts>0
True
Retry Attempts counter-1
False
DC Order Succes Confirmation
True
STG-UpdateRequestStatus Process
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
Firmware Updated

Interaction Process

S19 Request if it is a Meter Firmware Update Request

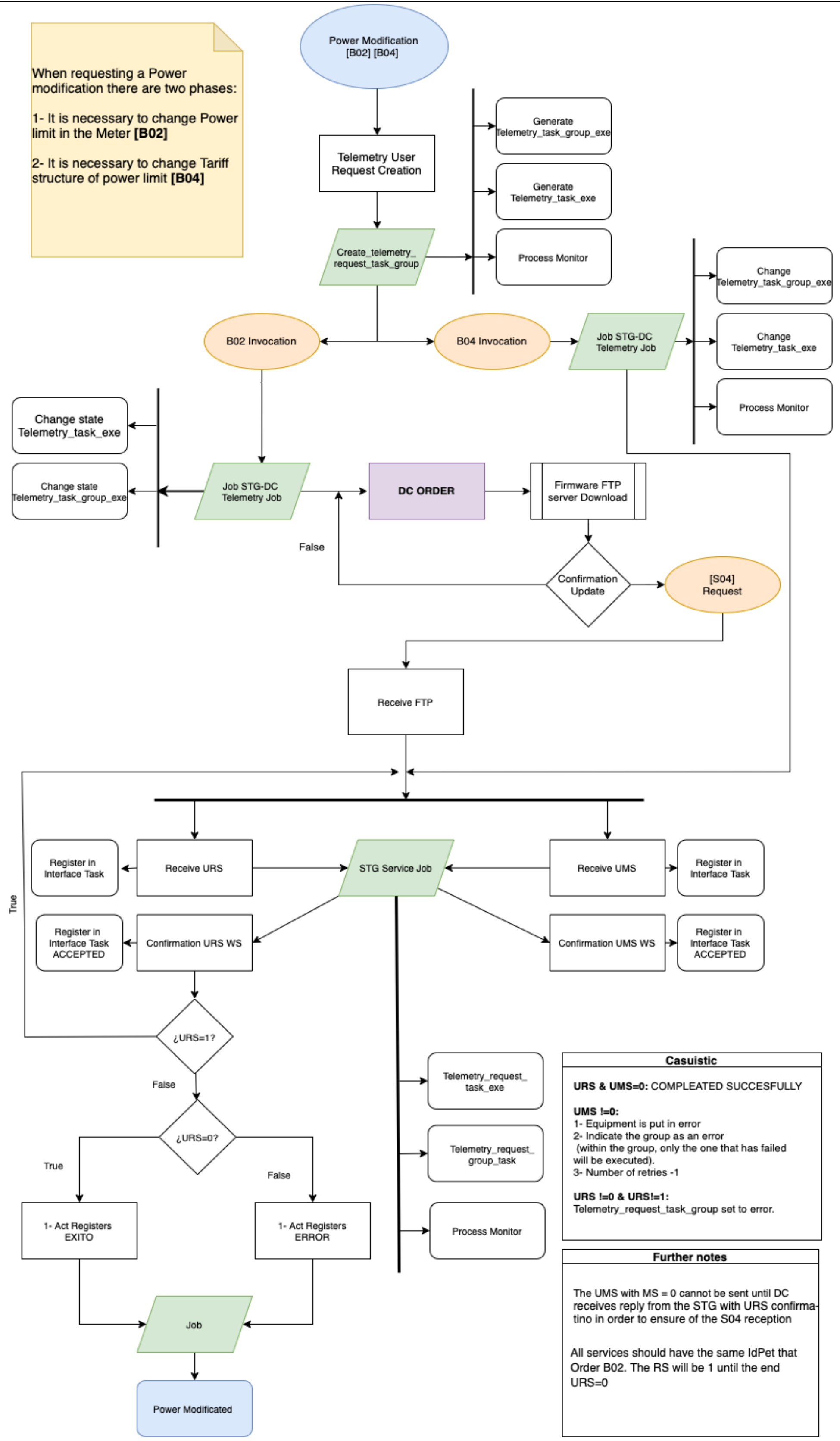
S22 Request if it is a DC Firmware Update Request

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.


When requesting a Power modification there are two phases:

1- It is necessary to change Power limit in the Meter **[B02]**

2- It is necessary to change Tariff structure of power limit **[B04]**

Power Modification
[B02] [B04]
Telemetry User Request Creation
Create_telemetry_request_task_group
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
B02 Invocation
B04 Invocation
Job STG-DC Telemetry Job
Change Telemetry_task_group_exe
Change Telemetry_task_exe
Process Monitor
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC ORDER
Firmware FTP server Download
False
Confirmation Update
[S04] Request
Receive FTP
True
Register in Interface Task
Receive URS
STG Service Job
Receive UMS
Register in Interface Task
Register in Interface Task ACCEPTED
Confirmation URS WS
Confirmation UMS WS
Register in Interface Task ACCEPTED
¿URS=1?
False
¿URS=0?
True
False
Telemetry_request_task_exe
Telemetry_request_group_task
Process Monitor
1- Act Registers EXITO
1- Act Registers ERROR
Job
Power Modificated

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

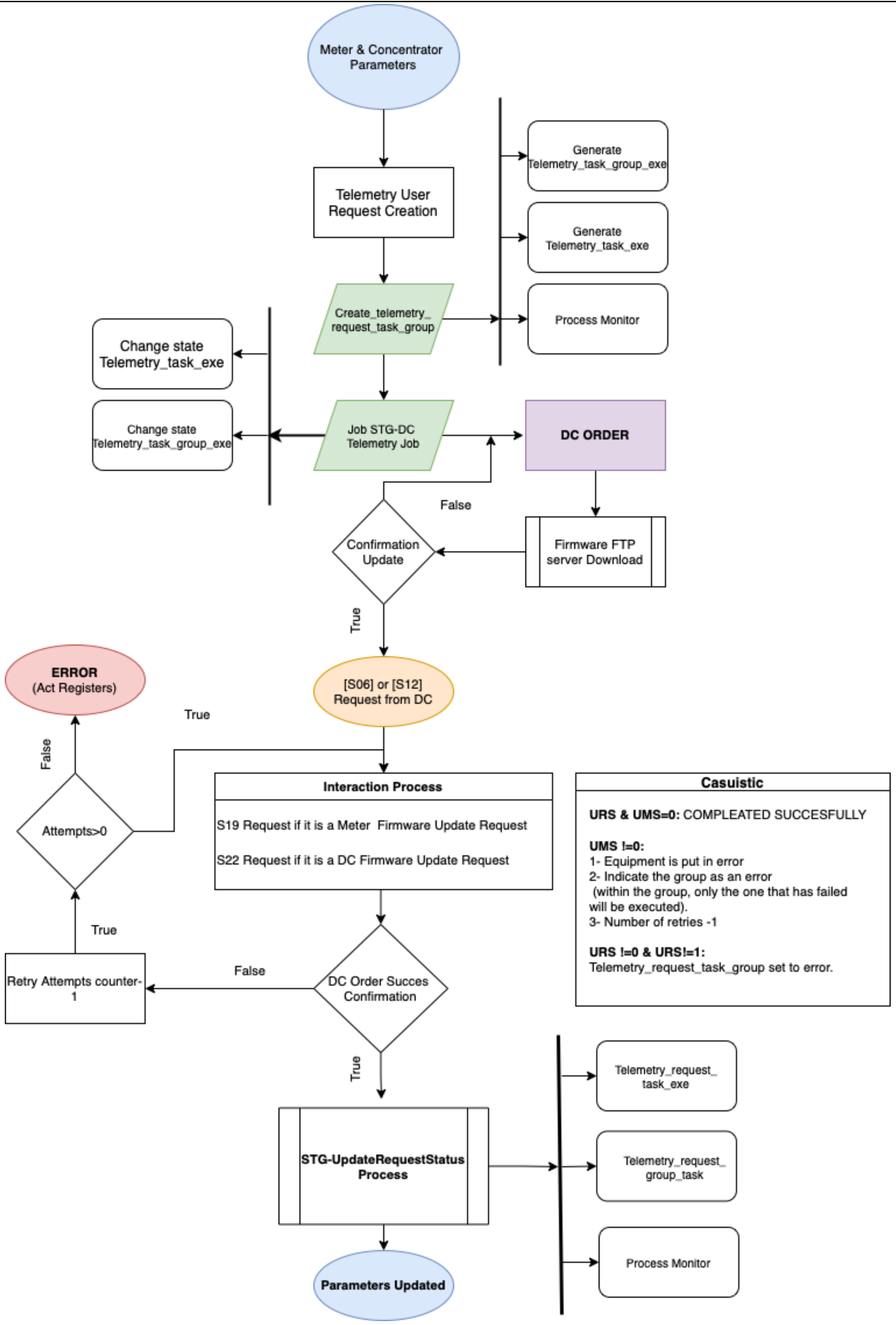
UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.

Further notes

The UMS with MS = 0 cannot be sent until DC receives reply from the STG with URS confirmatino in order to ensure of the S04 reception

All services should have the same IdPet that Order B02. The RS will be 1 until the end URS=0


Meter & Concentrator Parameters
Telemetry User Request Creation
Generate Telemetry_task_group_exe
Generate Telemetry_task_exe
Process Monitor
Create_telemetry_request_task_group
Change state Telemetry_task_exe
Change state Telemetry_task_group_exe
Job STG-DC Telemetry Job
DC ORDER
False
Confirmation Update
Firmware FTP server Download
True
ERROR (Act Registers)
[S06] or [S12] Request from DC
True
False
Attempts>0
Interaction Process
S19 Request if it is a Meter Firmware Update Request
S22 Request if it is a DC Firmware Update Request
True
False
Retry Attempts counter-1
DC Order Succes Confirmation
True
Telemetry_request_task_exe
STG-UpdateRequestStatus Process
Telemetry_request_group_task
Process Monitor
Parameters Updated

Casuistic

URS & UMS=0: COMPLEATED SUCCESFULLY

UMS !=0:
1- Equipment is put in error
2- Indicate the group as an error
(within the group, only the one that has failed will be executed).
3- Number of retries -1

URS !=0 & URS!=1:
Telemetry_request_task_group set to error.