



UNIONE EUROPEA  
Fondo Sociale Europeo  
Fondo Europeo di Sviluppo Regionale



## **Avviso 1735 del 13.07.2017 MIUR**

Progetti di Ricerca Industriale e Sviluppo Sperimentale nelle 12 Aree di Specializzazione individuate dal PNR 2015-2020

---

# Progettazione dell'architettura hardware e software della piattaforma ComESTo

---

*Rapporto Tecnico di Ricerca Industriale D5.7*



<b>Avviso</b>	Avviso 1735 del 13.07.2017 MIUR
<b>Codice progetto</b>	ARS01_01259
<b>Nome del progetto</b>	Community Energy Storage Gestione Aggregata di Sistemi di Accumulo dell'Energia in Power Cloud
<b>Acronimo</b>	ComESTo
<b>Documento</b>	D5.7
<b>Tipologia</b>	Rapporto Tecnico
<b>Data di Rilascio</b>	04/11/2020
<b>Obiettivo Realizzativo</b>	OR5
<b>Attività Realizzativa</b>	A5.7
<b>Soggetti Beneficiari Proponenti</b>	UNICAL, EVOLVERE, FBK, TIM, EDIS
<b>Elaborato (Nome, Cognome – Soggetto Beneficiario)</b>	Matteo Baù - FBK
<b>Verificato (Nome, Cognome – Soggetto Beneficiario)</b>	Edoardo Gino Macchi - FBK
<b>Approvato (Nome, Cognome – Soggetto Beneficiario)</b>	Membri del PEB

## Sommario

1. Executive summary	10
2. Piattaforme di mercato: Stato dell'arte	11
2.1. Introduzione al modello "Smart Grid Architecture Model"	11
2.2. Classificazione delle piattaforme di mercato	13
3. Introduzione alla piattaforma ComESTo	15
4. Il mercato della Comunità ComESTo	16
4.1. Funzionamento del mercato nel lungo periodo	18
4.2. Funzionamento del mercato nel breve periodo	19
4.3. Funzionamento del mercato in tempo reale	20
4.4. Funzionamento del mercato in tempo reale in caso di servizi a DSO e TSO	20
5. Il cloud computing e comunità ComESTo	20
5.1. I vantaggi dell'adozione del cloud computing	22
6. Architettura a micro servizi della Piattaforma ComESTo	26
6.1. L'architettura a microservizi della Piattaforma ComESTo	26
6.2. I microservizi della Piattaforma ComESTo	27
6.2.1. MQTT Broker per la comunicazione tra la Piattaforma e i dispositivi periferici	27
6.2.2. RabbitMQ per la gestione delle code dei messaggi	28
6.2.3. Redis per la memoria cache	28
6.2.4. Grafana per la analisi dei dati e la produzione di grafici	28
6.2.5. Kibana per la diagnostica dei servizi	28
6.2.6. Omoi per la memorizzazione delle misure e delle previsioni	29
6.2.7. Mirai per il forecasting di generazione e consumo	29
6.2.8. OFIS-DA e OFIS-TR per l'ottimizzazione della flessibilità interna e lo sbilanciamento	30
6.2.9. DPRG per la generazione di profili delle nanogrid	31
6.2.10. WS-APP per l'interfaccia tra Piattaforma ComESTo e dispositivi periferici e remoti	31
6.2.11. Backend per l'accesso alla Piattaforma ComESTo dell'utente finale	32

---

6.2.12.	Okane per la lettura dei prezzi zonal dal Mercato Elettrico	32
6.2.13.	Kai per la memorizzazione di dati specifici	32
6.2.14.	SIMU per la generazione di profili di generazione e di domanda	32
6.2.15.	I servizi aggiuntivi della Piattaforma	33
7.	Protocolli di comunicazione: stato dell'arte	34
7.1.	Protocolli e standard cross-funzionali	34
7.2.	Protocolli e standard per l'interazione DSO-Aggregatore	35
7.3.	Protocolli e standard per il controllo delle risorse flessibili	36
7.4.	I protocolli della piattaforma ComESTo	40
7.5.	Stima del volume di dati della <i>Piattaforma ComESTo</i>	40
8.	Blockchain nella piattaforma ComESTo	42
8.1.	La tecnologia Blockchain per le transazioni energetiche	43
8.2.	Protocolli di consenso	45
8.2.1.	Lo sviluppo degli Smart Contracts per la gestione delle interazioni tra I prosumers e l'aggregatore	47
9.	Cenni alle problematiche di sicurezza	48
9.1.	Incremento del livello di sicurezza delle procedure di Autenticazione	49
10.	Metodologia di Software Engineering applicata	49
10.1.	Modello a cascata (Waterfall)	52
10.2.	Modello Agile	54
10.3.	Il Modello adottato: Agile-Waterfall Hybrid	56
11.	Architettura hardware della Piattaforma ComESTo	57
11.1.	Caratteristiche principali del servizio	58
11.2.	Caratteristiche dell'infrastruttura/Conformità alle normative vigenti	60
12.	Definizione delle funzioni da implementare per i servizi al dso	61
12.1.	Il modello del Mercato dei Servizi Globali e Locali (MSGL) ComESTo	62
12.2.	Il mercato dei Servizi Locali della distribuzione	65

12.3.	Architettura delle comunicazioni e delle interazioni TSO-DSO-Utenti	67
12.3.1.	Il modello “ <i>traffic light</i> ”	67
12.3.2.	L'architettura delle interazioni tra le piattaforme	68
12.3.3.	Le interazioni in caso di semaforo VERDE	68
12.3.4.	Le interazioni in caso di semaforo GIALLO	69
12.3.5.	Le interazioni in caso di semaforo ROSSO	69
13.	La re-ingegnerizzazione dell'home gateway Eugenio	70
13.1.	Prima fase prototipale	71
13.1.1.	La scelta del System On Module	71
13.1.2.	La architettura hardware	72
13.1.1.	Test e validazione	72
13.2.	Seconda fase prototipale	73
13.2.1.	Test e validazione	74
13.3.	Validazione finale del nuovo Eugenio	75
13.4.	Sistema di build automatica	75
13.5.	Principali Risultati della re-ingegnerizzazione	76
14.	Allegato A	79
15.	Bibliografia	81

## Indice delle figure

Figura 1.1-1 Livello base del modello Smart Grid Architecture Model [1] .....	12
Figura 1.1-2 Architettura completa del modello Smart Grid Architecture Model [1] .....	12
Figura 1.2-1 – Livello funzionale secondo il modello SGAM per il caso d'uso NL 1 .....	15
Figura 3-1 Schema Logico del modello di mercato ComESTo .....	17
Figura 3-2 Modello del mercato con riferimento alla flessibilità interna .....	17
Figura 3-3 Modello del mercato con riferimento alla flessibilità esterna .....	18
Figura 4-1 Cloud Computing Pyramid (fonte interna) .....	21
Figura 4-2 Schema data-centric e topic-based .....	<b>Errore. Il segnalibro non è definito.</b>
Figura 4.1-1 Struttura del middleware .....	25
Figura 5.2.7-1 Sequence Diagram del servizio Mirai per il caso SSSD .....	30
Figura 5.2.15-1 Identificazione del servizio client-side e server-side .....	34
Figura 6.1-1 – Modello logico della struttura dello standard IEC 61850 [3] .....	35
Figura 6.2-1 Modello di interazione di mercato USEF [3] .....	36
Figura 6.3-1 Schema della Energy Flexibility Interface [3] .....	37
Figura 6.3-2 Livello di comunicazione secondo modello SGAM per NL UC 1-2 [] .....	39
Figura 6.3-3 Livello di comunicazione secondo modello SGAM per FR UC 3 [3] .....	40
Figura 9-1 Fasi del SDLC .....	49
Figura 9.1-1 Modello SDLC - Waterfall .....	53
Figura 9.2-1 Modello SDLC - Agile .....	54
Figura 9.3-1 Modello Ibrido Waterfall-Agile .....	56
Figura 10.1-1 blocchi di risorse configurabili per l'utente nel servizio Self Data Center .....	58
Figura 10.2-1 Sicurezza, Privacy e Conformità alle normative. ....	60
Figura 11.1-1 - Mercato comune TSO-DSO .....	63
Figura 11.3.2-1 - Architettura logica delle comunicazioni e interazioni tra le piattaforme .....	68
Figura 12-1 Eugenio prima del processo di re-ingegnerizzazione .....	70
Figura 12.1.1-1 Architettura hardware del nuovo Eugenio .....	72
Figura 12.1.2-1 Uno dei primi prototipi del nuovo Eugenio .....	73
Figura 12.2-1 Output finale della a) prima e b) seconda fase prototipale .....	74
Figura 12.2-2 Il nuovo Eugenio .....	74
Figura 12.2.1-1 Caduta di tensione per spike di corrente assorbita dal modem GSM .....	75
Figura 12.4-1 Interfaccia web per pacchetti .....	76

## Indice delle tabelle

Tabella 1.2-1 Classificazione delle piattaforme di mercato per risorse distribuite secondo [1]. .....	13
Tabella 6.3-1 Protocolli di comunicazione utilizzati nei casi d'uso NL US 1-2 e NL UC 3 del progetto InterFLEX [3]. .....	37
Tabella 11.2-1 – I servizi ancillari locali.....	66
Tabella 11.3.1-1 – Il semaforo come indicatore dello stato della rete.....	67

## Abbreviazioni ed acronimi

Abbreviazione/Acronimo	Testo Esteso
API	Application program interface
ARERA	Autorità di Regolazione per Energia, Reti e Ambiente
AT	Alta tensione
BFT	Bizantine fault tolerant
BT	Bassa tensione
BT	Bassa Tensione
CHP	Generatore/sistema combinato potenza/calore o cogeneratore
CIM	Common information model
CIM	Common Information Model
CSSD	Client-Side Service Discovery
DER	Distributed energy resources
DER	Distributed Energy Resources
DLT	Distributed ledger technology
DMS	Distribution management system
DMS	Distribution Management System
DNS	Domain name system
DSO	Distribution system operator
DSO	Distribution System Operator
EFI	Energy Flexibility Interface
EFI	<i>Energy Flexibility Interface</i>
EMA	Energy Manipulation Attack
EMS	Energy management system
ESB	Enterprise Service Bus

EV	Electric Vehicle
FAP	Flexibility aggregation platform
GID	Generic Interface Definitions
HTTP	HyperText Transfer Protocol
HW	Hardware
IaaS	Infrastructure as a Service
ICT	Information and communication technology
LIMS	Local infrastructure management system
LLD	Low-Level Design
MQTT MQ	Telemetry Transport or Message Queue Telemetry Transport
MSD	Mercato dei Servizi di Dispacciamento
MT	Media tensione
MT	Media Tensione
NCA	Node Capture Attack
P/Q	Potenza Attiva/Potenza Reattiva
P2P	Peer-to-Peer
PaaS	Platform as a Service
PDD	Punti di Dispacciamento
POF	Proof-Of-Work
POS	Proof-Of-Stack
PV	Generatore/sistema fotovoltaico
RTN	Rete di Trasmissione Nazionale
SaaS	Software as a Service
SCADA +2:43	Supervisory control and data acquisition
SDLC	System Development Life Cycle
SGAM	Smart grid architecture model
SIDMS	System Interfaces for Distribution Management
SNA	Spoof Node Attack
SOA	Service Oriented Architecture
SRS	Software Requirement Specification
SSD	Server-side Service Discovery
SW	Software
TSO	Transmission system operator
TSO	Trasmission System Operator

UAT	User Acceptance Testing
UC	Use case
USEF	Universal Smart Energy Framewor
USEF	Universal Smart Energy Framework
UVA	Unità virtuali abilitate
UVAC	Unità virtuali abilitate di consumo
UVAM	Unità virtuali abilitate miste
UVAN	Unità virtuali abilitate nodali
UVAP	Unità virtuali abilitate di produzione
XML	Linguaggio di marcatura estendibile
WG	Generatore/impianto eolico

## EXECUTIVE SUMMARY

Il presente documento, deliverable del progetto ComESTo, sintetizza i risultati delle attività condotte nell'ambito dell'attività 5.7 ("Individuazione dell'architettura hardware e software della piattaforma Community Energy Storage ") dell'Obiettivo Realizzativo 5 (OR5).

Obiettivo della attività A5.7 consiste nell'individuare l'architettura software e hardware della piattaforma ComESTo per l'implementazione

- dei modelli sviluppati nell'attività A5.1 per la pianificazione e gestione dei sistemi di accumulo sul medio lungo periodo, nell'attività A5.2 per la Demand Response in presenza di sistemi di accumulo distribuiti in ambito VPP, nell'attività A5.3 per la previsione della producibilità e di carico ai fini dell'accumulo e della Demand Response, nell'attività A5.4 per l'integrazione dello storage distribuito nei mercati energetici e di dispacciamento del DSO;
- degli standard di comunicazione e trasmissione dati per la gestione dei sistemi di accumulo distribuito sviluppati nell'attività A5.5
- delle soluzioni di cyber security e di privacy in ambito smart grid sviluppati nell'attività A5.6

A tale scopo, l'attività A5.7:

1. ha condotto uno studio circa i molteplici modelli di piattaforme cloud per la gestione di risorse di flessibilità distribuite in generale; lo studio si è poi esteso ai vantaggi nell'adozione del Cloud Computing nella gestione di risorse di flessibilità per il caso specifico di comunità energetiche come quella del progetto ComESTo.
2. ha rivisitato i modelli di mercato per l'integrazione dello storage distribuito nei mercati energetici e di dispacciamento del DSO che sono stati prodotti nell'attività A5.4 affinché questi fossero più affini al caso delle piattaforme cloud, precisando il funzionamento nel lungo e breve periodo, e nel tempo reale ed individuando i requisiti funzionali alla risoluzione di questi modelli dal punto di vista informatico.
3. ha individuato e descritto con dettaglio tutti i microservizi della Piattaforma ComESTo, inclusi i microservizi di base cioè i microservizi necessari al funzionamento stessa della piattaforma e che sono erogati da server interni, ed inclusi i microservizi specifici per il funzionamento e la gestione della Comunità ComESTo.
4. ha studiato e valutato i protocolli e gli standard esistenti per il controllo delle risorse di flessibilità nell'ambito delle comunità energetiche ed affini, incluso la tecnologia Blockchain per il tracciamento sicuro e robusto delle transizioni energetiche.
5. ha studiato le problematiche della sicurezza inerenti alle architetture IoT come quella della piattaforma ComESTo (valutando il livello di sicurezza delle procedure di autenticazione) ed ha

definito la metodologia di software engineering applicata ovvero il modello Agile-Waterfall Hybrid.

6. ha definito l'architettura hardware della piattaforma cloud ComeSTo così da garantire la disponibilità di risorse “scalabili” e “modulabili” a seconda delle mutevoli situazioni.
7. ha definito quali funzioni implementare nella piattaforma ComeSTo al fine di fornire al DSO servizi locali e globali.
8. ha mostrato e descritto la re-ingegnerizzazione del dispositivo home gateway denominato Eugenio che funge da interfaccia nella comunicazione tra le nanogrid e la piattaforma ComeSTo.

## 1. PIATTAFORME DI MERCATO: STATO DELL'ARTE

Attualmente esistono molteplici modelli di piattaforma per la gestione delle risorse di flessibilità distribuite [i, ii], che si differenziano soprattutto in termini di attori e livelli di tensione coinvolti, servizi forniti e risorse utilizzate. Il modello di riferimento *Smart Grid Architecture Model* (SGAM) permette una classificazione e confronto coerenti tra le diverse architetture ed è qui utilizzato, seguendo l'esempio dei progetti FP7 Grid4EU e H2020 InterFLEX. Il progetto InterFLEX, conclusosi a dicembre 2019, ha valutato, per mezzo di cinque dimostratori, l'utilizzo di risorse di flessibilità locali per garantire il rispetto dei limiti della rete di distribuzione sperimentando anche modelli ed architetture di interazione/comunicazione tra DSO ed Aggregatore. Di conseguenza, il progetto costituisce un'ottima base di partenza per la definizione dell'architettura della piattaforma ComESTo.

### 1.1. INTRODUZIONE AL MODELLO “SMART GRID ARCHITECTURE MODEL”

Il modello SGAM è un modello tridimensionale frutto dell'unione di un livello base, che definisce la struttura della smart grid in termini di componenti fisici e di gestione dei flussi informativi tra di essi, e di quattro livelli di interoperabilità che consentono di descrivere le altre caratteristiche dell'architettura relazionandole ai componenti fisici coinvolti.

La Figura 1.1-1 [1] mostra il livello base, che distingue tra i domini fisici della rete elettrica, ordinati secondo la classica filiera di produzione-trasmissione-distribuzione-consumo, e le zone che rappresentano i livelli gerarchici nella gestione del sistema elettrico. Quest'ultime permettono un'efficace separazione delle diverse funzioni e servizi associati alla smart grid e si dividono in:

*Processi*: fase di generazione e tutti i componenti ad essa direttamente legati (trasformatori, generatori, linee aeree o in cavo, etc.).

*Campo*: componentistica di protezione, controllo e monitoraggio del sistema elettrico. Ad esempio, eventuali funzioni di monitoraggio in tempo reale ricadono in questa zona.

*Stazione*: componenti e sistemi per l'aggregazione di misure ed informazioni (SCADA)

*Gestione*: sistemi necessari alla gestione del sistema elettrico (EMS, DMS, etc.)

*Impresa*: processi organizzativi e commerciali degli attori del sistema elettrico  
*Mercato*: operazioni di mercato lungo la filiera del sistema elettrico.

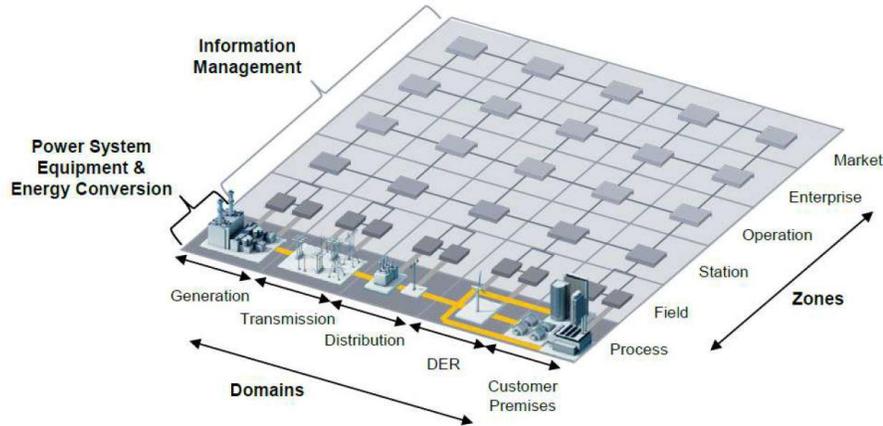


Figura 1.1-1 Livello base del modello Smart Grid Architecture Model [1]

La Figura 1.1-2 mostra invece il modello SGAM completo, comprensivo dei quattro livelli di interoperabilità oltre a quello base già definito: *comunicazione, informazione, servizi e business*. La seguente analisi presenterà la classificazione delle piattaforme di mercato esistenti per risorse distribuite sulla base di alcune classi definite in [1] e [2] in accordo con il modello SGAM.

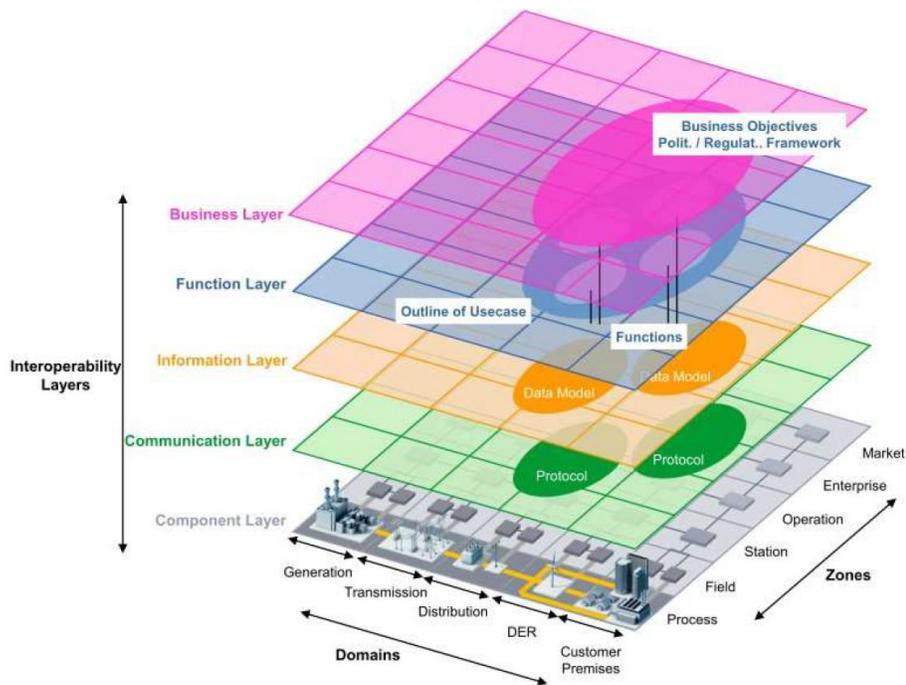


Figura 1.1-2 Architettura completa del modello Smart Grid Architecture Model [1].

## 1.2. CLASSIFICAZIONE DELLE PIATTAFORME DI MERCATO

Ciascun progetto è stato classificato in

Tabella 1.2-1 secondo cinque criteri, che permettono anche di identificare facilmente similarità con ComESTo:

- *Dominio di flessibilità*: il dominio, o livello di tensione, al quale sono riferite le risorse di flessibilità utilizzate.
- *Attori Gestione*: gli attori coinvolti nella gestione delle risorse di flessibilità.
- *Servizi*: i servizi svolti dalle risorse distribuite
- *Modello di mercato*: la presenza o meno di un mercato per le risorse distribuite

Complessivamente, la maggior parte dei progetti sviluppano logiche di controllo e comunicazione per le quali il DSO ha controllo diretto delle risorse distribuite o in alternativa si interfaccia direttamente con flexibility providers che gestiscono una o più risorse di loro proprietà. La figura dell'aggregatore compare in relazione al DSO con una definizione simile a quella applicata in ComESTo solamente nel progetto InterFLEX, casi d'uso NL UC 1-3 e FR UC 3. I casi d'uso NL 1-2 testano la fornitura di servizi locali per il DSO da parte di risorse di accumulo aggregate di diversa tipologia: batterie stazionarie nel primo caso e veicoli elettrici nel secondo. Come si può vedere in Figura 1.2-1, nella quale è mostrato il livello funzionale del modello SGAM per il caso d'uso NL UC 1, esistono due figure di aggregatore: l'aggregatore commerciale si interfaccia con il DSO tramite la Flexibility Aggregation Platform (FAP) e a sua volta si interfaccia con l'aggregatore locale che gestisce il Local Infrastructure Management System (LIMS). Analogamente, anche il caso d'uso FR 3 testa la fornitura di servizi locali da risorse distribuite, senza tuttavia limitare la tipologia di risorse utilizzabili e prevedendo un'unica figura di aggregatore. Il caso d'uso NL 3 si concentra invece sull'analisi della piattaforma di mercato in termini di meccanismi di prezzo e liquidità a fronte della fornitura di diverse tipologie di servizi da risorse differenti. La piattaforma di mercato tra aggregatore e DSO è la stessa per tutti i quattro i casi d'uso e verrà dettagliata a seguire.

Tabella 1.2-1 Classificazione delle piattaforme di mercato per risorse distribuite secondo [1].

Progetto (Nazione)	Caso d'uso	Dominio Flessibilità	Attori Gestione	Servizi	Modello di mercato
SINTEG (DE)	C/sells: ALF	AT – BT	TSO, DSO	Risoluzione delle congestioni	si
	C/sells: ReFLEX	MT - BT	DSO		
	C/sells: Comax	MT - BT	TSO, DSO		
	WindNode	Tutti	TSO, DSO		
	Enera	Tutti	TSO, DSO		

	New 4.0	AT - MT	TSO, DSO		
<u>DA/RE Platform (DE)</u>	-	MT - BT	TSO, DSO, BSP/BRP	Risoluzione delle congestioni	si
<u>NODES Market (EU)</u>	-	Tutti	TSO, DSO, BSP, BRP	Risoluzione delle congestioni	si
<u>Grid Integration (DE)</u>	-	MT - BT	DSO	Risoluzione delle congestioni	si
<u>GOPACS (NL)</u>	-	AT - MT	TSO, DSO	Risoluzione delle congestioni	si
<u>Piclo-Flex Market (UK)</u>	-	AT - BT	TSO, DSO	Risoluzione delle congestioni	si
	DE UC 1-2	BT	DSO	Risoluzione delle congestioni, power quality	no
	DE UC 3	BT	DSO	Servizi ancillari	no
	CZ UC 1-2	MT - BT	DSO	Integrazione DER	no
<u>InterFLEX (EU)</u>	CZ UC 3-4	BT	DSO	Servizi ancillari	no
	NL UC 1-3	MT - BT	DSO, Aggregatore	Servizi ancillari, power quality	si
	FR UC 1	MT - BT	DSO, Aggregatore	Isola automatica	no
	FR UC 3	MT - BT	DSO, Aggregatore	Risoluzione delle congestioni	si
<u>Equigy (IT)</u>	-	MT - BT	TSO, Aggregatore	Servizi ancillari	si

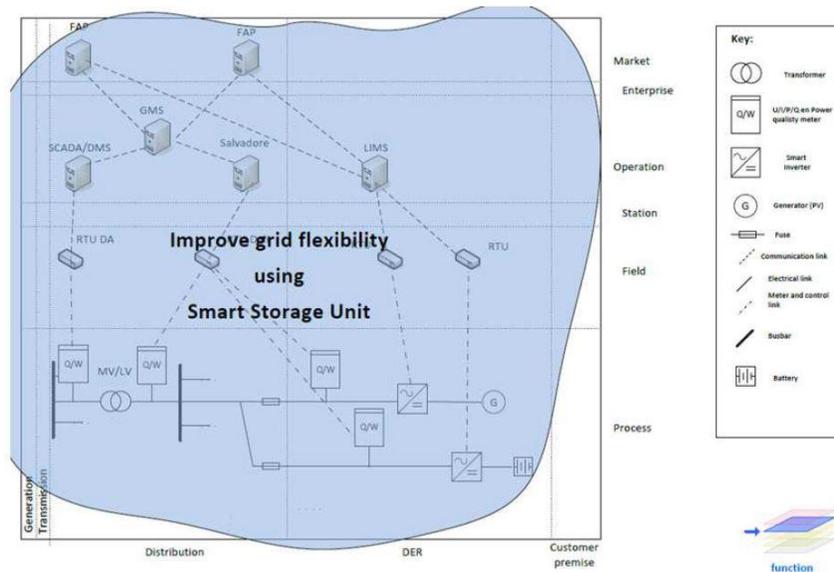


Figura 1.2-1 – Livello funzionale secondo il modello SGAM per il caso d'uso NL 1.

## 2. INTRODUZIONE ALLA PIATTAFORMA COMESTO

La comunità energetica del progetto ComESto (nel seguito anche *Comunità ComESto* o *Comunità*) è una comunità principalmente composta da unità residenziali moderne; ciascuna unità residenziale è una *nanogrid* ovvero è un sistema ibrido di alimentazione elettrica, capace di gestire contemporaneamente più tipologie di sorgenti di generazione e sistemi di accumulo di differente tecnologia. La nanogrid è capace di condizionare i flussi di potenza per generazione, consumo e stoccaggio che sono inclusi nel proprio confine. Grazie a ciò, è possibile regolare il profilo al punto di connessione alla rete pubblica di ogni unità residenziale così da inseguire un profilo desiderato; tale possibilità è anche detta flessibilità. La flessibilità di una singola unità residenziale o nanogrid è un contributo alla flessibilità della comunità nel suo complesso, pertanto, essa concorre a generare valore aggiunto, a vantaggio sia dei membri della comunità ComESto che del sistema elettrico in generale. L'Aggregatore ComESto (o *Aggregatore*) è il soggetto munito di personalità giuridica che opera per nome e per conto della *Comunità* anche al di fuori della comunità stessa; l'Aggregatore gestisce la *Comunità* sia in fase di programmazione che in tempo reale, assicurando la massimizzazione dell'efficienza energetica e l'autoconsumo collettivo nonché la corretta erogazione dei servizi ancillari venduti sui mercati elettrici. Lo strumento principale dell'aggregatore per compiere la sua missione è una piattaforma cloud-based detta Piattaforma ComESto. Scopo di questo documento è illustrare la Piattaforma ComESto, mostrando come siano stati implementati i modelli per la pianificazione e gestione dei sistemi di accumulo sul medio lungo periodo (A5.1), i modelli e gli algoritmi per la Demand Response in presenza di sistemi di accumulo distribuiti in ambito virtual power plant (A5.2), i modelli e gli algoritmi per la previsione della producibilità e di carico

ai fini dell'accumulo e della Demand Response (A5.3), i modelli di mercato per l'integrazione dello storage distribuito nei mercati energetici e di dispacciamento del DSO (A5.4).

### 3. IL MERCATO DELLA COMUNITÀ COMESTO

L'Aggregatore gestisce la *Comunità* ottimizzando le risorse energetiche distribuite mediante la regolazione dei sistemi di accumulo distribuiti; la regolazione è finalizzata a mantenere i profili di immissione e prelievo, misurati in tempo reale, quanto più prossimi a profili stabiliti in sede di programmazione di breve periodo e, ancora prima, di lungo periodo. L'insieme dei passi e delle regole che l'Aggregatore applica per gestire la *Comunità* è il modello di mercato della *Comunità ComESto*; il modello si basa su meccanismi di flessibilità che consentono ad ogni unità residenziale, e alla *Comunità* tutta, di conseguenza, di applicare profili di immissione e prelievo assegnati. Tali profili sono calcolati dall'Aggregatore utilizzando principalmente un algoritmo di ottimizzazione detto *ottimizzazione della flessibilità interna e dispacciamento* (in sigla OFIS, giusto attività A5.4, Deliverable D5.4) che restituisce il compromesso ottimale tra massimo autoconsumo collettivo, prezzo finale per l'utente consumatore e produttore, ricavi per servizi alla rete elettrica. Minore l'errore tra il profilo misurato ed il profilo assegnato, maggiore l'autoconsumo collettivo, minore il prezzo per l'acquisto dell'energia per i membri della *Comunità*, maggiore il prezzo di vendita per i produttori della *Comunità*, maggiore il ricavo della *Comunità* per servizi ancillari venduti al DSO e TSO. Nel mercato della *Comunità*, la flessibilità è distinta in interna ed esterna. Quando interna, la flessibilità migliora l'efficienza energetica e l'autoconsumo collettivo, stimolando economicamente gli utenti a modificare le proprie abitudini nella gestione energetica. Quando esterna, la flessibilità aumenta gli introiti della *Comunità* perché consente la vendita di servizi ai gestori delle reti elettriche, modificando i programmi di consumo stabiliti in fase di pianificazione. La flessibilità esterna è definita locale quando applicata per l'erogazione di servizi al DSO per congestioni e regolazione della tensione ai nodi della rete di distribuzione; la flessibilità esterna, è definita globale quando applicata per fornire servizi al TSO per la regolazione della frequenza della rete elettrica ed il bilanciamento.



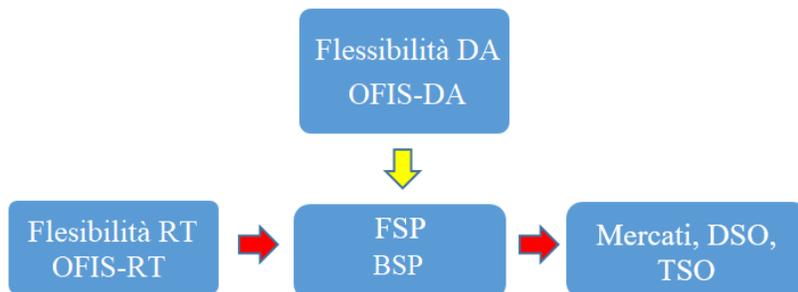


Figura 3-3 Modello del mercato con riferimento alla flessibilità esterna

Lo schema di funzionamento del mercato della Comunità, per come progettato in D5.4, è riportato in Figura 3-1. Da questo schema sono stati estrapolati i due sottoschemi dove il primo di Figura 3-2 riferisce alla sola flessibilità interna per la massimizzazione dell'efficienza energetica e dell'autoconsumo collettivo mentre il secondo di Figura 3-3 riferisce alla sola flessibilità esterna per l'erogazione di servizi a DSO e TSO. Nel caso dello schema di Fig. 3-2, la disponibilità di dati storici e di misure nel database consente l'esecuzione degli algoritmi per il calcolo dei profili aggregati e la pianificazione della gestione della Comunità nel lungo periodo. Allo stesso tempo, grazie al servizio di previsione, è anche possibile eseguire il calcolo del profilo aggregato il giorno prima per il giorno dopo (OFIS-DA). Il profilo aggregato è poi distribuito tra le varie nanogrid grazie alla tecnica di ripartizione DRPG. Gli stessi dati del database consentono il calcolo degli scostamenti e la regolazione della Comunità in tempo reale. Anche in questo caso, gli aggiustamenti calcolati a livello aggregato sono poi distribuiti tra le varie nanogrid grazie alla tecnica di ripartizione DRPG. Nel caso dello schema di Fig. 3-3, la flessibilità day-ahead e i rispettivi aggiustamenti in tempo reale sono utilizzati dal Flexibility Service Provider e dai Balance Responsible Parties per la partecipazione ai mercati elettrici per la fornitura di servizi di flessibilità alla rete elettrica.

### 3.1. FUNZIONAMENTO DEL MERCATO NEL LUNGO PERIODO

Il funzionamento del mercato nel lungo periodo (o anche pianificazione nel lungo periodo o stagionale, annuale) consiste nell'approvvigionamento delle risorse, stimando la generazione e la domanda, e scegliere le tipologie di accumulo; questa pianificazione è guidata da una funzione obiettivo economica che tiene conto della previsione dei prezzi. Output della pianificazione nel lungo periodo sono la scelta dei sistemi di accumulo al fine di aumentare l'autoconsumo collettivo e i contratti di capacità con il DSO a cui erogare dei servizi di flessibilità. La sequenza logica dei passi che porta alla pianificazione del lungo periodo è illustrata in Figura 3-2. Il trigger di colore verde punta ad un database che contiene dati storici ovvero i flussi di potenza che ciascuna nanogrid ha scambiato con la rete pubblica, i valori di

temperatura, irraggiamento e ventosità per i siti che ricadono nella Comunità, i prezzi dei mercati elettrici, etc. Oltre questi dati storici, la pianificazione nel lungo periodo utilizza i profili Aggregati di generazione e consumo per come restituiti da un apposito microservizio.

### 3.2. FUNZIONAMENTO DEL MERCATO NEL BREVE PERIODO

Il funzionamento del mercato nel breve periodo (o anche del giorno prima) consiste nel raffinare la pianificazione del lungo periodo, il giorno prima per il giorno dopo. Output del funzionamento nel breve periodo è, dunque, il profilo di generazione e di domanda della Comunità nonché i profili dei sistemi di accumulo, in forma aggregata ma distinti per tecnologia (es. un profilo per tutti i sistemi a batterie al litio, un profilo per tutti i sistemi di accumulo ad idrogeno, etc). La sovrapposizione di tutti questi profili restituisce il profilo complessivo della Comunità cioè il profilo che sarà inviato al TSO e DSO. Il funzionamento del mercato nel breve periodo è, dunque, l'ottimizzazione della flessibilità interna e dello sbilanciamento; tale ottimizzazione restituisce i valori ottimali di carica e scarica di tutti i sistemi di accumulo distribuiti, ripartendo il surplus di potenza generata all'interno della *Comunità* al fine di massimizzare l'autoconsumo collettivo, massimizzare i ricavi dei produttori della *Comunità*, minimizzare i costi di approvvigionamento di energia elettrica dei consumatori della *Comunità*. La sequenza di passi che porta alla pianificazione del breve periodo è illustrata in Figura 3-2. Il trigger2 di colore giallo punta al database che fornisce al blocco *Aggregati* i dati utili alla creazione di profili aggregati di generazione, consumo e stato dei sistemi storage. I profili così ottenuti sono input per il blocco *Forecast* che fornisce a sua volta profili aggregati di generazione e consumo di previsione per il giorno dopo. La pianificazione di lungo periodo e i profili aggregati per il giorno dopo sono input al blocco *Flessibilità day-ahead* che esegue l'ottimizzazione della flessibilità interna e sbilanciamento, il giorno prima per il giorno dopo; questo blocco restituisce anche le quantità di energia da acquistare/vendere sul mercato elettrico. Come già detto precedentemente, l'ottimizzatore day-ahead riceve e restituisce profili aggregati pertanto è ora necessario che i profili restituiti siano opportunamente divisi in tanti profili quanti sono i partecipanti alla *Comunità*. Questa missione è affidata al blocco *Ripartizione* di Figura 3-2 che riceve in input i profili aggregati orari di carica/scarica per ogni tipologia di accumulo e restituisce in output la ripartizione di tali profili aggregati tra i sistemi storage presenti in ciascuna nanogrid per tipologia di accumulo. Il blocco *Ripartizione* applica gli algoritmi progettati nella attività A5.1 ed invia a ciascuna nanogrid le quantità orarie di potenza da "inseguire".

### 3.3. FUNZIONAMENTO DEL MERCATO IN TEMPO REALE

Il funzionamento del mercato in tempo reale consiste nel raffinare la pianificazione del breve periodo, ora per ora, regolando quelle risorse flessibili che hanno opportuna velocità di risposta. In modo equivalente, è possibile affermare che il funzionamento in tempo reale consiste nel ridurre il più possibile il gap tra i profili programmati in day-ahead ed i profili misurati in tempo reale. Quest'azione correttiva è costantemente in esecuzione e compensa gli eventuali scostamenti caricando o scaricando sistemi di accumulo rapidi come, ad esempio, gli storage che impiegano batterie a litio. La sequenza di passi che porta al funzionamento in tempo reale è illustrata in Figura 3-32. Il trigger3 di colore rosso punta al database che fornisce al blocco *Scostamenti* i dati utili al calcolo degli scostamenti orari o quartorario tra i profili aggregati di generazione, consumo e scambio, misurati in tempo reale ed i profili pianificati che erano già stati comunicati agli operatori di rete. Gli scostamenti di rilievo sono input al blocco *Forecasting* che fornisce prontamente una previsione aggiornata dei profili; questi ultimi sono input per il blocco *Flessibilità Real-Time* che esegue l'ottimizzazione della flessibilità interna e sbilanciamento in tempo reale così da annullare o ridurre gli stessi scostamenti. Il blocco *Ripartizione* termina la sequenza di passi, applica gli algoritmi progettati nella attività A5.1, ripartisce la soluzione ottima di flessibilità tra le nanogrid, invia a quest'ultime le quantità orarie o quart'orarie di potenza da "inseguire".

### 3.4. FUNZIONAMENTO DEL MERCATO IN TEMPO REALE IN CASO DI SERVIZI A DSO E TSO

Il funzionamento del mercato in tempo reale include la fornitura di servizi ancillari ovvero servizi al DSO o TSO; a seconda del tempo di preavviso cioè del tempo che intercorre tra la richiesta del servizio e l'attivazione del servizio stesso, attivare un servizio ancillare può implicare la sostituzione, in tempo reale, dei profili precedentemente calcolati dal blocco *Flessibilità DA* con nuovi profili. In questo caso specifico, il processo di erogazione del servizio è gestito totalmente dall'aggregatore nella sua qualità di *flexible service provider* per i servizi al DSO e di *balancing service provider* per i servizi al TSO.

La sequenza di passi che porta al funzionamento in tempo reale in caso di servizi a DSO e TSO è illustrata in Figura 3-3; non esiste nessun trigger poiché, come già detto, il processo è gestito totalmente dall'aggregatore che si avvale dei blocchi *Flessibilità DA* e *Flessibilità RT* per definire il servizio e del blocco *Ripartitore* per distribuire il servizio tra le nanogrid.

## 4. Il cloud computing e comunità ComESTo

Il cloud computing è un modello di calcolo (modello di riferimento o anche paradigma) che fornisce servizi su richiesta, usando grandi dispositivi di archiviazione e di calcolo, distribuiti e accessibili via

Internet, sempre attivi. I servizi forniti dal cloud computing sono tre: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), per come illustrato in Figura 4-1.

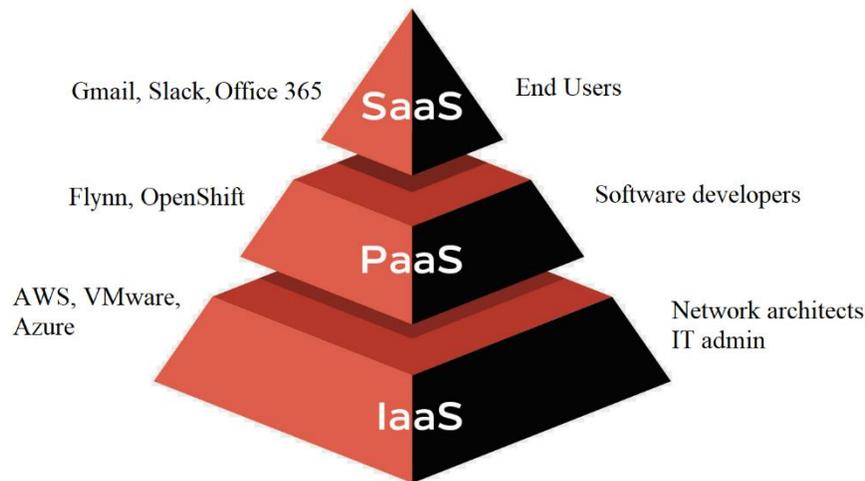


Figura 4-1 Cloud Computing Pyramid (fonte interna)

Il tipo di servizio Infrastructure as a Service offre ai clienti l'utilizzo di una infrastruttura hardware composta da dispositivi storage dati e macchine virtuali. I clienti sono amministratori di sistema o sistemisti che non gestiscono fisicamente la infrastruttura hardware ma solo le istanze e le risorse dell'infrastruttura che intendono utilizzare. I clienti installano il proprio software sull'infrastruttura; poiché il servizio supporta la verticalizzazione delle risorse, i clienti possono installare ed eseguire anche un proprio sistema operativo. Il tipo di servizio Platform as a Service offre ai clienti modelli di programmazione, i clienti sono tipicamente sviluppatori che accedono a tali modelli di programmazione tramite cloud, creano i loro programmi, li eseguono; il servizio è responsabile dell'esecuzione in runtime delle attività. Il tipo di servizio Software as a Service offre ai clienti l'utilizzo di programmi ed applicazioni tramite browser web, il servizio fornisce anche la distribuzione del software, pertanto, i clienti eseguono l'applicazione mediante lo stesso web browser, evitando l'installazione sul proprio computer. Il cloud computing può anche essere classificato come privato, pubblico, comunitario e ibrido. Il cloud privato è di proprietà di un'organizzazione privata e le informazioni sono condivise solo all'interno dell'organizzazione; lo scopo è servire le proprie applicazioni aziendali. Il cloud pubblico è di proprietà di un fornitore di servizi ed è utilizzato dal pubblico per i suoi scopi. Il cloud comunitario è simile ad un cloud privato ma ha funzionalità aggiuntive per fornire servizi non ad una singola azienda ma ad un gruppo di organizzazioni/aziende che hanno requisiti simili. Il cloud ibrido integra i tre precedenti per soddisfare requisiti trasversali. I vantaggi dell'utilizzo di un modello di cloud computing sono molteplici

e riflettono la sua stessa natura; in primis, il cloud computing è fortemente flessibile perché ha una natura elastica, pertanto, la capacità di archiviazione e di calcolo può espandersi o contrarsi in base alle necessità che si manifestano tempo per tempo. In aggiunta, il cloud computing esalta fortemente il concetto di condivisione, pertanto, previa autorizzazione e verifica dei criteri di privacy, i clienti (sistemisti, sviluppatori e utenti finali) possono scambiare e condividere mezzi, risorse ed informazioni senza restrizioni di rilievo. Allo stesso tempo, il cloud computing offre agli utenti un'infrastruttura di misurazione cioè meccanismi di ottimizzazione dei costi grazie ai quali sono pagate solo le risorse consumate.

#### 4.1. I VANTAGGI DELL' ADOZIONE DEL CLOUD COMPUTING

L'adozione del cloud computing può dare un forte contributo alle comunità energetiche come quella del progetto ComESTo nel compimento della loro missione e si propone come valido sostituto degli approcci esistenti nella gestione delle risorse distribuite e delle reti di comunicazione. Questo perché gli approcci convenzionali, come quelli adottati nelle smart grid e diversi dal cloud computing, sollevano importanti questioni, molte delle quali sono tuttora aperte. Il punto che solleva il maggior numero di questioni aperte è l'architettura relazionale master-slave, tipicamente adottata per gestire la rete di comunicazione tra le sottostazioni elettriche, i clienti finali e i loro dispositivi behind-the-meter. Tale architettura master-slave, anche quando applicata nel contesto di comunità energetiche, sarebbe costantemente sottoposto al rischio di un attacco informatico del tipo Distributed Denial of Service che consiste nell'esaurire deliberatamente tutte le risorse del sistema informatico che fornisce servizi, fino a rendere il sistema informatico stesso incapace di erogare servizi.

Altra questione aperta, e riferita all'architettura master-slave, è numero massimo di clienti che può essere contemporaneamente servito da un fornitore di servizi. Si consideri, ad esempio, il caso di una comunità energetica con una moltitudine di utenti che necessitano dello scheduling ottimizzato di carichi elettrici oppure l'ottimizzazione dei profili di immissione e prelievo per il trading dell'energia elettrica peer-to-peer. In una architettura master-slave, il numero massimo di utenti che ottiene una risposta è pari alla capacità nominale del master nel fornire il servizio; sfortunatamente, tale numero non può essere convenientemente incrementato al crescere della moltitudine degli utenti che, partecipando alla comunità, richiedono quel servizio. Ulteriore questione aperta, speculare alla precedente, riferisce alle risorse di calcolo necessarie a ciascun cliente per risolvere processi e task necessari alla partecipazione alla comunità energetica. Ad esempio, i sistemi residenziali di gestione dell'energia situati presso gli utenti finali che partecipano a programmi di demand response devono eseguire calcoli e misurazioni nonché applicare algoritmi di pianificazione e controllo, anche in tempo reale. Poiché tali sistemi locali sono spesso dotati di processori, memorie e spazi di archiviazione ridotti, la tendenza è limitare il carico di lavoro dei sistemi residenziali quanto più possibile. Ciò aumenta la

quantità di dati scambiati tra master e slaves, fino alla saturazione della capacità del sistema di gestire la stessa comunicazione, visto che la complessità gestionale cresce molto più rapidamente del crescere del numero di utenti. Infine, resta ancora aperta una questione circa la larghezza di banda del canale di comunicazione tra master e slaves nella erogazione di servizi fondamentali. Si considerino, ad esempio, i servizi ancillari per gli operatori delle reti elettriche di distribuzione e trasmissione; molti di questi servizi sono fondamentali per la sicurezza delle reti elettriche e richiedono adeguate tecniche e strumenti di gestione e controllo. In accordo ad una architettura, è sicuramente possibile adottare sensori e dispositivi intelligenti così da ottenere una allerta precoce contro guasti e contingenza ma è assai difficile implementare un conveniente controllo in tempo reale a causa della limitata larghezza di banda. Il paradigma cloud computing si dimostra essere una alternativa più che valida alla convenzionale architettura master-slave poiché risolve o mitiga fortemente le questioni aperte sopra elencate. Questo perché il paradigma cloud computing offre infrastrutture, piattaforme e software fortemente scalabili, capaci di maneggiare ed analizzare rapidamente streaming di informazioni che provengono da una vasta ed eterogenea moltitudine di dispositivi; tutto ciò è confacente alle necessità delle comunità energetiche per bilanciare la domanda e la generazione di energia elettrica, anche in tempo reale, ed in modo sicuro, economico e sostenibile. Per problemi che crescono continuamente nel tempo o per problemi che crescono velocemente ma solo per un ristretto intervallo temporale, il paradigma cloud computing offre servizi che si adeguano perfino autonomamente. Ad esempio, grazie alla natura elastica del cloud computing, è possibile aumentare automaticamente il numero di applicazioni sempre attive oppure instaurare interazioni tra piattaforme, migliorando così anche la redditività degli investimenti. L'interazione tra piattaforme si applica facilmente mediante l'uso di agenti mobili cioè una combinazione di software e dati in grado di interagire tra loro o di migrare da una macchina virtuale ad un'altra, anche se quest'ultima appartiene ad una diversa infrastruttura cloud. La migrazione può avvenire in modo autonomo e senza pregiudizi circa la esecuzione del processo sulla macchina di destinazione. Ed ancora, nell'ambito delle comunità energetiche, un agente può fungere da dispositivo di interfaccia tra i diversi livelli della stessa comunità; tale missione risulta di particolare valore qualora l'architettura di comunicazione della rete della comunità sia molto eterogenea. Analogamente, la collaborazione tra agenti consente alle comunità energetiche grids di applicare agevolmente approcci decentralizzati al fine di migliorare l'esecuzione di processi per la raccolta di enormi quantità di dati. E' questo il caso della lettura di una vasta moltitudine di contatori intelligenti. In tal caso, l'agente preposto al monitoraggio dei contatori crea un piano di download dei dati e contatta altri agenti che hanno accesso ai contatori; in base al tempo di risposta, il piano di download può essere ottimizzato nel tempo. In aggiunta, operare con gli agents facilita la condivisione dei compiti e aumenta la resilienza del sistema al guasto degli stessi agenti. Ulteriori punti forza del cloud computing sono la

comunicazione incentrata sui dati o data-centric e la comunicazione di gruppo basata su argomenti o topic-based (vedi Figura 4.1-1).

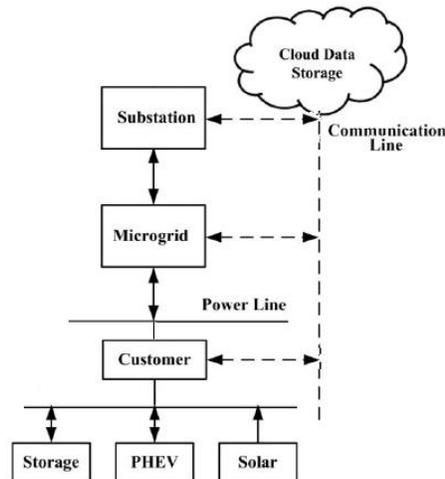


Figura 4.1-1 Schema data-centric e topic-based

Questo tipo di comunicazione si contrappone alla comunicazione incentrata sull'IP (IP-centric) e consente alle piattaforme cloud di ottimizzare la larghezza di banda del canale di comunicazione; di conseguenza, problemi che si manifestano in modo rilevante ma solo in un ristretto intervallo di tempo (es. ore di punta) possono essere risolti senza dover necessariamente investire ulteriori capitali con il solo scopo di risolvere “il picco”. Si consideri, ad esempio, la gestione dei picchi di domanda; uno noto strumento per la corretta gestione di questi picchi è l'applicazione di prezzi dinamici. Come conseguenza, nelle sole ore di punta della domanda, il flusso bidirezionale di dati (misure, prezzi, schedulazioni, etc) tra gli utenti e il sistema centralizzato di gestione aggregata dell'energia cresce in modo assai rilevante. Le informazioni ricevute da ciascun utente sono processate dal sistema centralizzato in base a criteri di priorità, alle risorse disponibili e alla larghezza di banda del canale di comunicazione. Grazie all'integrazione del cloud computing è possibile adottare meccanismi di assegnazione dinamica della larghezza di banda affinché, durante le ore di punta della domanda, la larghezza di banda assegnata sia maggiore di quella assegnata durante le ore non di punta; così facendo, è possibile processare contemporaneamente tutte le risposte di una vasta moltitudine di utenti finali senza rinforzare il canale di comunicazione. Inoltre, la comunicazione data-centric e topic-based consente alle piattaforme basate su cloud computing di condividere quantità di informazioni maggiori rispetto ad altri paradigmi, soprattutto in tempo reale. Questa peculiarità risulta di enorme vantaggio nel power management delle comunità energetiche soprattutto quando le comunità sono distribuite su una vasta area. Altro tema è l'ottimizzazione delle risorse energetiche distribuite (incluso lo

stoccaggio) nella gestione di comunità energetiche anche in tempo reale; in tale ambito, il cloud computing offre la possibilità di applicare con successo modelli di gestione e controllo, basati su una struttura di comando gerarchica e centralizzata. L'efficienza di tali modelli in un contesto caratterizzato da una pronunciata complessità è da imputare principalmente alla creazione di un ambiente virtuale che si dimostra anche essere è una delle tecniche più efficienti per la riduzione dei costi e l'ottimizzazione delle risorse. Nell'ambiente virtuale creato dal cloud computing, la gestione della comunità può essere eseguita mediante la creazione di semplici moduli, costituiti da un certo numero di risorse fisiche che includono la potenza di calcolo, lo storage ed i calcolatori; tali moduli utilizzano tecnologie di clustering e di calcolo parallelo per condividere la potenza di calcolo con altri calcolatori, in base alla necessità che si manifestano tempo per tempo. Esisterà, ad esempio, un modulo rappresentativo della infrastruttura della comunità (reti elettriche), un modulo che sovrintende al monitoraggio della comunità (sensori), un modulo che è responsabile di attività come l'allocazione e l'ottimizzazione delle risorse di flessibilità, un modulo che è responsabile della integrazione dei servizi offerti tramite Internet da operatori esterni alla comunità. La gestione tramite moduli consente un elevato grado di sicurezza e robustezza poiché i processi di calcolo ed i processi decisionali non sono a carico di una sola macchina virtuale ma di più macchine virtuali; inoltre, nuove macchine virtuali possono essere aggiunte ad un set pre-esistente a costi assai contenuti (o nulli) e senza alcuna modifica alle applicazioni o alle librerie.

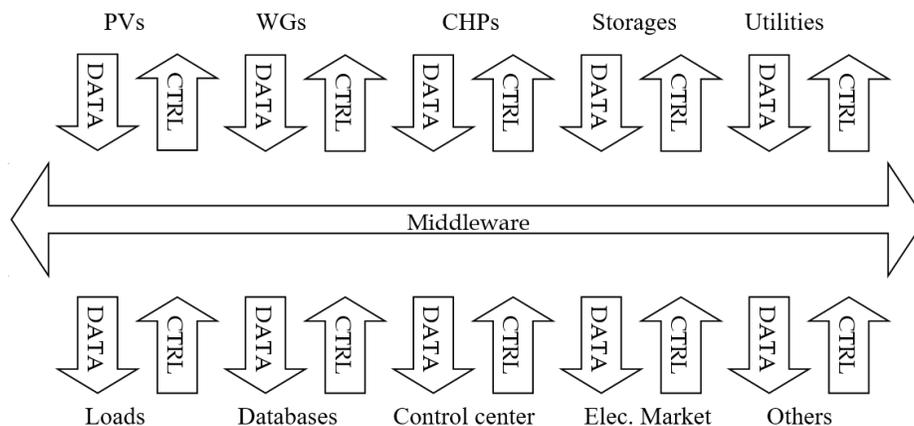


Figura 4.1 Struttura del middleware

Per ultimo, è importante sottolineare il supporto del cloud computing alla gestione di una comunità energetica mediante l'utilizzo simultaneo di molteplici servizi (Service Oriented Architecture, SOA), spesso disomogenei ed erogati da sistemi e tecnologie eterogenee. Il cloud computing consente l'interoperabilità, l'integrazione ed il coordinamento di questi servizi, assicurando lo scambio reciproco

di dati e la sicurezza informatica mediante un middleware (vedi Figura 4.1) cioè un insieme di programmi per la integrazione di servizi software secondo il classico modello hub-and-spoke o a stella. Il centro stella rappresenta l'interfaccia unica dell'architettura orientata ai servizi; un convenzionale integration broker o un più moderno enterprise service bus (ESB) possono essere implementati a questo scopo.

## 5. ARCHITETTURA A MICRO SERVIZI DELLA PIATTAFORMA COMESTO

La Piattaforma ComESTo è il punto di coordinamento e controllo della Comunità. Come illustrato in Figura 3-1 Schema Logico del modello di mercato, la Piattaforma ComESTo implementa modelli, algoritmi e protocolli di comunicazione così da fornire servizi in un ampio range che si estende dalla raccolta di dati di misura fino alla gestione in tempo reale dei sistemi di generazione, consumo ed accumulo per la massimizzazione dell'autoconsumo collettivo e l'erogazione di servizi alle reti elettriche. La Piattaforma ComESTo è stata progettata servendosi del paradigma cloud computing e, in particolare, è stata implementata utilizzando la tecnologia di virtualizzazione di tipo IaaS fornita da VMware; tale tecnologia, come altre simili e aderenti allo standard Open Virtualization Format [iii], si basa su un componente detto hypervisor che gestisce la creazione e la manutenzione delle macchine virtuali. Al fine di identificare il software e l'hardware necessario alla implementazione della Piattaforma ComESTo, l'approccio utilizzato per la modellizzazione dei sistemi è il UML [iv]. La Piattaforma ComESTo è quindi costituita da un insieme - flessibile, riconfigurabile ed economico - di elaboratori elettronici configurati in assetto di virtualizzazione delle risorse.

### 5.1. L'ARCHITETTURA A MICROSERVIZI DELLA PIATTAFORMA COMESTO

La Piattaforma ComESTo ha una architettura a micro-servizi; è questo un design pattern, originariamente adottato su larga scala da solution provider quali Netflix e Amazon, che consente l'erogazione di servizi che risultano critici sia per continuità, sia per volumi di transazioni attuate. La parola "micro" indica l'elevata specificità di ogni servizio che può essere sviluppato autonomamente sia dal punto di vista del linguaggio sia dei componenti necessari; ogni microservizio ha una propria interfaccia, che è tipicamente esposta tramite metodi REST, e può essere fisicamente dislocato su una qualsiasi macchina virtuale idonea allo scopo. Le architetture a microservizi è la controparte delle architetture monolitiche, dove il codice che implementa il software è unico e condiviso. I vantaggi delle architetture monolitiche sono la semplicità poiché l'intero progetto è di solito ospitato su una sola macchina (per cui anche il test e il rilascio sono relativamente semplici) e la velocità di intervento per la manutenzione. I difetti sono la non scalabilità proprio perché la architettura è monolitica, la non riusabilità dei pezzi di codice a causa della loro dipendenza con l'intero codice, il testing perché

penalizzato dalla lunghezza del codice. Per questo motivo, il software è frammentato in unità atomiche e separate dette microservizi, dando così vita all'omonima architettura. I vantaggi di una architettura a microservizi sono la modularità poiché il software è componibile e ogni pezzo resta di "proprietà" al proprio team di sviluppo, la scalabilità poiché è facile aggiungere più istanze di uno stesso servizio, la separazione totale dei codici di ogni servizio, la libertà di sviluppo poiché ogni microservizio può essere sviluppato autonomamente con la tecnologia più consona, lo sviluppo distribuito poiché sia i team che i microservizi possono essere fisicamente distribuiti, la facilità del testing poiché ogni servizio è testato in autonomia. Gli svantaggi dell'architettura a microservizi sono la latenza poiché i microservizi possono essere fisicamente lontani gli uni dagli altri pertanto può esistere una latenza nel processo di recupero dei dati necessari ad un microservizio, la comunicazione tra team poiché ogni microservizio potrebbe avere un team proprietario diverso dagli altri, la conoscenza cross-domain perché ogni team deve comunque essere a conoscenza dei processi dell'intero progetto, la maggiore conoscenza che ciascuno sviluppatore deve acquisire quando deve adattarsi alla logica e agli specifici strumenti utilizzati per quel particolare servizio. Dati pregi e difetti, la Piattaforma ComESTo è progettata e realizzata mediante architettura a microservizi così da dare giusta rilevanza alla elevata sensibilità dei dati e delle informazioni trattate dalla Piattaforma ComESTo e alla strategicità delle elaborazioni e delle decisioni che la stessa Piattaforma adotta. Allo stesso tempo, la Piattaforma ComESTo agevola il coordinamento del lavoro dei vari gruppi del progetto ComESTo e la loro produttività.

## 5.2. I MICROSERVIZI DELLA PIATTAFORMA COMESTO

I paragrafi che seguono descrivono i microservizi della Piattaforma ComESTo con dettaglio; dapprima sono presentati cinque microservizi di base cioè necessari al funzionamento stessa della piattaforma e sono erogati da server interni. Successivamente sono presentati i microservizi specifici per la Comunità.

### 5.2.1. MQTT BROKER PER LA COMUNICAZIONE TRA LA PIATTAFORMA E I DISPOSITIVI PERIFERICI

Il Broker MQTT [v] è un microservizio di base per la Piattaforma ComESTo ed è erogato da un server di broadcast che segue il paradigma "fire-and-forget"; il compito del Broker MQTT è tenere traccia delle connessioni tra la Piattaforma ComESTo e dispositivi periferici o sottoscrittori. I dispositivi periferici sono due: Eugenio e smart meter. Eugenio è il gateway installato presso le unità residenziali facenti parte della Comunità che è in comunicazione con una nanogrid mentre uno smart meter è un power meter evoluto che misura i parametri elettrici come tensioni nodali e correnti di linea per questi dispositivi di generazione, accumulo e storage che non sono governati da una nanogrid. Esistono sia broker MQTT open source che proprietari; la scelta è principalmente guidata dal livello di robustezza

che si desidera conseguire ovvero la maggiore o minore probabilità di eventi dannosi come il fuori servizio. Sebbene molti broker MQTT open source come ApacheMQ, VerneMQ, HiveMQ siano altamente robusti, la Piattaforma ComESto adotta un broker proprietario, progettato per le specifiche esigenze del progetto ComESto, interamente scritto in nodejs [vi], capace di gestire migliaia di connessioni concorrenti e che utilizza il core del broker open source mosca.js. La scelta di adottare tale broker è principalmente motivata dal know how in possesso del team di sviluppo sulla tecnologia node.js e dalla presenza di una florida community di sviluppatori che curano il progetto *Mosca*.

### 5.2.2. RABBITMQ PER LA GESTIONE DELLE CODE DEI MESSAGGI

RabbitMQ è un microservizio di base della Piattaforma ComESto ed è erogato da un server di livello industriale; RabbitMQ è scritto in Erlang ed ha lo scopo di gestire code dei messaggi; è utilizzato per gestire lo scambio di messaggi asincroni tra i microservizi sviluppati all'interno della Piattaforma ComESto

### 5.2.3. REDIS PER LA MEMORIA CACHE

Redis è un microservizio di base della Piattaforma ComESto ed è erogato da un server di cache del tipo key-value, notoriamente molto veloce e altamente configurabile; Redis è utilizzato come punto comune fra servizi per la memorizzazione dati; ad esempio, il servizio broker MQTT interroga il servizio Redis per conoscere i dispositivi quali Eugenio o smart meters al momento attivi.

### 5.2.4. GRAFANA PER LA ANALISI DEI DATI E LA PRODUZIONE DI GRAFICI

Grafana è un microservizio di base della Piattaforma ComESto ed è erogato da un server gestionale di analisi dati e produzione di grafici. Grafana consente la costruzione di widget o pannelli, ognuno dei quali fornisce una rappresentazione grafica di dati memorizzati o forniti da una sorgente.

### 5.2.5. KIBANA PER LA DIAGNOSTICA DEI SERVIZI

Kibana è un microservizio di base della Piattaforma ComESto il cui scopo è facilitare le operazioni di diagnostica dei microservizi. Ogni microservizio, mediante l'utilizzo di un forwarder LogStash, scrive la registrazione sequenziale e cronologica delle operazioni effettuate (il proprio log) su un server centrale che contiene, quindi, una registrazione completa di tutti gli eventi che accadono nella Piattaforma ComESto. Kibana è il microservizio che gestisce l'interrogazione del database del server centrale ed impiega un motore di ricerca basato su tecnologia Elastic search. In aggiunta Kibana consente anche di costruire procedure che restituiscono un alert o una semplice notifica quando una condizione diventa True.

### 5.2.6. OMOI PER LA MEMORIZZAZIONE DELLE MISURE E DELLE PREVISIONI

OMOI è un microservizio della Piattaforma ComESTo che gestisce la lettura e la scrittura di dati su due database, timescaleDB e mongoDB, progettati appositamente per il progetto ComESTo. La tecnologia utilizzata per comunicare con i due database è nodejs. OMOI che può essere richiesto tramite interfaccia REST all'indirizzo <https://cows.evolvere.io/public/comesto-schema/omoi>. Di seguito si elencano alcuni esempi di richieste e risposte del servizio OMOI. Richiesta: domanda dei carichi elettrici e/o dei veicoli elettrici per un desiderato intervallo di tempo; risposta: somma delle serie temporali contrassegnate come "carico". Richiesta: generazione di tutti gli impianti di generazione da sole, vento, biodiesel, per un desiderato intervallo di tempo; risposta: somma delle serie temporali contrassegnate come "produzione". Richiesta: stato di carica di sistemi di accumulo a batterie per un desiderato intervallo di tempo; risposta: somma delle serie temporali contrassegnate come "accumulo". Oltre i dati di misura e di previsione, OMOI fornisce dati circa il Mercato del Giorno Prima (MGP) e il Mercato dei Servizi di Dispacciamento (MSD); ad esempio, OMOI consente di memorizzare le quantità di energia aggregata e scambiata con il mercato, valorizzata ai prezzi zonali, così come le quantità di energia offerte e contrattualizzate nel MSD.

### 5.2.7. MIRAI PER IL FORECASTING DI GENERAZIONE E CONSUMO

Mirai è un microservizio di Forecasting e fornisce una previsione del profilo della domanda di una utenza elettrica o del profilo di generazione di un impianto a fonte rinnovabile. Il servizio di previsione è stato sviluppato con tecnologia nodejs ed è erogato da un apposito software di calcolo implementato sulla stessa Piattaforma ComESTo oppure su server di terze parti; in questo ultimo caso, lo scambio di dati tra la Piattaforma ComESTo ed il server di terze parti utilizza il protocollo M2M per come descritto in [vii]. Il servizio Mirai è raggiungibile mediante API all'indirizzo <https://cows.evolvere.io/public/comesto-schema/mirai>; l'URL è il seguente: mirai/v1/<method>. Il diagramma sequenziale del servizio Mirai è illustrato in Figura 5.2.7-1; il Backend avvia l'interrogazione per l'accesso alle risorse fornite da Mirai utilizzando il paradigma server-side service discovery (SSSD) (vedi Figura 5.2.7-1); secondo tale paradigma, un microservizio A (nel caso in oggetto Backend) non conosce l'indirizzo del microservizio B a cui richiedere le risorse (nel caso in oggetto Mirai), bensì conosce l'indirizzo di un microservizio intermedio C (conosciuto con il nome di router o API Gateway) che, a sua volta, è capace di inoltrare la richiesta di A a B e restituire la risposta.

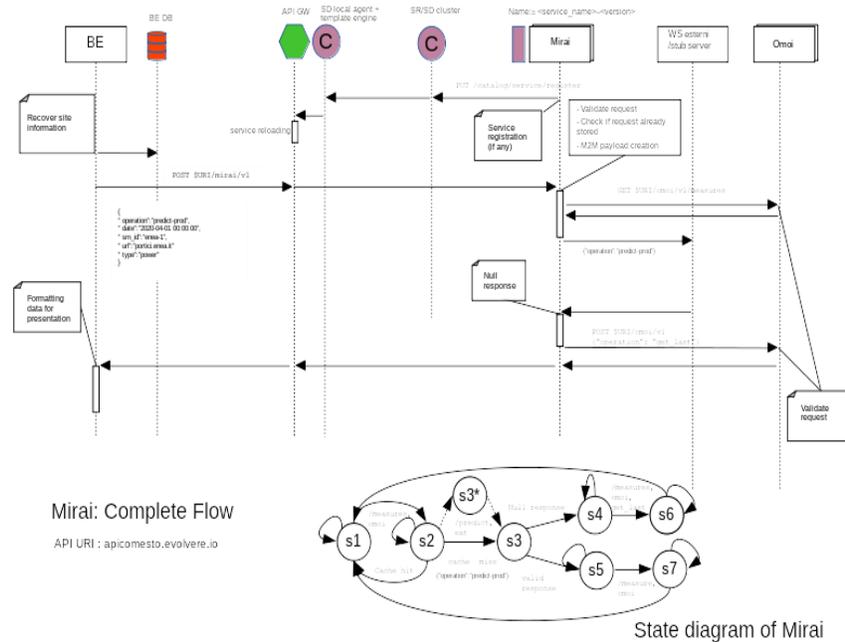


Figura 5.2.7-1 Sequence Diagram del servizio Mirai per il caso SSSD

Le previsioni restituite da Mirai sono consegnate al microservizio che ne ha fatto richiesta ma sono anche memorizzate in un database interno per il tramite del servizio Omoi discusso nel precedente paragrafo. Conservare localmente le previsioni è una scelta appositamente adottata così da sopperire, in qualche misura, alla eventuale e momentanea indisponibilità di dati come soleggiamento e ventosità, utili al microservizio Mirai o ai servizi di previsione resi da server di terze parti. In questi casi, Mirai recupera i dati di previsione memorizzati per l'ora prima o per il giorno prima o per la settimana prima, etc.

### 5.2.8. OFIS-DA E OFIS-TR PER L'OTTIMIZZAZIONE DELLA FLESSIBILITÀ INTERNA E LO SBILANCIAMENTO

OFIS-DA è il servizio sviluppato con tecnologia nodejs che ottimizza la flessibilità interna e lo sbilanciamento della Comunità; per fare ciò, il servizio OFIS esegue gli algoritmi di ottimizzazione per la massimizzazione dell'autoconsumo collettivo sia in fase di pianificazione che in tempo reale. Nel primo caso, OFIS calcola profili di immissione e prelievo per tutti i siti di generazione, consumo e stoccaggio; questo calcolo è eseguito il giorno prima per il giorno dopo, pertanto, è adottata la notazione di OFIS-Day Ahead (OFIS-DA). Nel secondo caso, OFIS corregge i profili di immissione e prelievo

precedentemente calcolati; questa correzione è eseguita in tempo reale (es. ogni ora) pertanto è adottata la notazione di OFIS-Real Time (OFIS-RT). Nel caso di ottimizzazione del giorno prima, OFID-DA necessita delle previsioni del profilo della domanda di una o più utenze elettriche o del profilo di generazione di uno o più impianti di generazione a fonte rinnovabile; tali previsioni sono fornite dal servizio MIRAI. Poiché i tempi di calcolo necessari all'erogazione del servizio di previsione potrebbero non essere compatibili coi vincoli di una sessione HTTP di tipo REST, tra OFIS-A e MIRAI è stata implementata una comunicazione asincrona, utilizzando il server di code RabbitMQ per come illustrato in Fig. X. In questa figura, il servizio OFIS genera un messaggio sulla coda PREDICT gestita dal server di code RabbitMQ per chiedere al servizio MIRAI le previsioni. Il servizio MIRAI, consumatore della coda PREDICT, prende in carico la richiesta ed invia messaggi utili a recuperare le informazioni necessarie alla previsione. Terminata la raccolta delle informazioni, il servizio MIRAI calcola le previsioni e le salva sul database locale grazie al servizio OMOI; fatto ciò, MIRAI pubblica sulla coda PREDICT-END il cui consumatore della coda è il servizio OFIS-DA che, da questo momento in avanti, potrà leggere la previsione sul database locale grazie ancora al servizio OMOI.

#### 5.2.9. DPRG PER LA GENERAZIONE DI PROFILI DELLE NANOGRID

Il Demand Response Profile Generator si occupa di “spedire” i nuovi profili a partecipanti, solitamente rappresentati da NanoGrid collegate a un gateway Eugenio. Il DPRG è a tutti gli effetti un driver: recupera i profili aggregati generati da OFIS, li ripartisce tramite un criterio di ottimizzazione e, insieme alle informazioni di ogni singolo sito, costruisce i pacchetti da inviare tramite MQTT su un particolare canale o stanza. I profili, dunque, vengono generati a partire dalle baseline aggregate prodotte da OFIS.

#### 5.2.10. WS-APP PER L'INTERFACCIA TRA PIATTAFORMA COMESTO E DISPOSITIVI PERIFERICI E REMOTI

Il servizio fornisce dati a dispositivi periferici, quali Eugenio ed eventuali App installate su dispositivi mobili, mediante web service remoti. Questo servizio è offerto con API di tipo RESTful (acronimo di Representational State Transfer). Quando una richiesta viene inviata tramite un'API RESTful, questa trasferisce al richiedente uno stato rappresentativo della risorsa. L'informazione, o rappresentazione, viene consegnata in uno dei diversi formati tramite protocollo HTTP: JSON (Javascript Object Notation), è il formato utilizzando dal servizio WS-APP.

### 5.2.11. BACKEND PER L'ACCESSO ALLA PIATTAFORMA COMESTO DELL'UTENTE FINALE

Backend è il microservizio della Piattaforma ComESto che fornisce all'utente finale accesso alla stessa Piattaforma tramite interfaccia Web di BackOffice. Il servizio fornisce anche metodi interni di supporto a tutti i microservizi della Piattaforma ComESto senza autenticazione, tranne l'autenticazione di base tramite token che è richiesta a tutti i microservizi.

### 5.2.12. OKANE PER LA LETTURA DEI PREZZI ZONALI DAL MERCATO ELETTRICO

Okane è il microservizio che quotidianamente, alle 23:59, si collega via FTP al server del Gestore del Mercato Elettrico e recupera i prezzi zonali dell'energia elettrica. Il servizio Okane, disponibile al link <https://cows.evolvere.io/public/comesto-schema/okane>, salva i prezzi sul locale database mongoDb per il tramite del servizio OMOI.

### 5.2.13. KAI PER LA MEMORIZZAZIONE DI DATI SPECIFICI

Kai è il microservizio che, come il microservizio OMOI, scrive dati sul database locale timescaleDB; in particolare, Kai scrive solo i dati che provengono da dispositivi come Eugenio e smart meters che comunicano attraverso due specifici canali (#ComESto/ng e #ComESto/sm, rispettivamente) mediante il protocollo P-NG per come progettato e testato in altra attività di progetto (si veda il deliverable D4.1)

### 5.2.14. SIMU PER LA GENERAZIONE DI PROFILI DI GENERAZIONE E DI DOMANDA

SIMU è il microservizio che produce profili di generazione e di domanda a partire da un profilo reale; questo microservizio è particolarmente utile per il testing ed il collaudo della Piattaforma ComESto. Il microservizi SIMU è raggiungibile al seguente link <https://cows.evolvere.io/public/comesto-schema/simu>. Questo servizio permette di generare serie temporali che modellano produzione fotovoltaica o assorbimento di carico. La generazione può avvenire in due modi distinti: 1) tramite seme 2) tramite modello matematico.

Generazione tramite seme: è selezionata una serie temporale ottenuta da misure reali della grandezza in questione (potenza di produzione o carico). Attraverso un campionamento poissoniano della serie, si stabiliscono dei punti all'interno della stessa (detti anche punti di difetto) dove applicare un fattore di scaling <sup>[viii]</sup>. Il valore dello scaling nei punti è casuale e parametrizzato. In questo modo, si possono generare profili "simili" al seme.

Generazione tramite modello: la serie è generata a partire da un modello matematico del fenomeno in questione. Ad esempio, per la previsione del carico elettrico residenziale è utilizzato un modello ottenuto empiricamente che mette in relazione la potenza oraria con il numero di stanze e il numero di utenti dell'abitazione [ix]. Inviando questi parametri a SIMU è possibile simulare il carico tipico di un'utenza residenziale. Per la previsione della generazione fotovoltaica, è stato utilizzato i modelli di pvlib [x], una libreria sviluppata in python dai laboratori Sandia USA. Tale libreria rappresenta il riferimento de-facto dei modelli di radiazione solare e di conversione fotovoltaica attualmente usati in tutto il mondo. Inoltre, per la generazione da modello, è possibile utilizzare anche la radiazione prevista stimata da servizi on-line, come SodaPro, molto utile per la stima della previsione futura (un giorno). Tramite SIMU, dunque, è possibile anche simulare degli impianti fittizi, una funzionalità molto utile soprattutto in fase di collaudo di tutte l'architettura.

#### 5.2.15. I SERVIZI AGGIUNTIVI DELLA PIATTAFORMA

Al fine di coordinare i microservizi, la Piattaforma dispone dei seguenti tre servizi aggiuntivi: Service Registrar, Service Discovery e Load Balancer. Il Service Registrar è il servizio aggiuntivo che gestisce il registro di tutti i microservizi della Piattaforma; tale registro contiene dati ed informazioni come DNS, IP e porta di ciascun microservizio. Il Load Balancer è il servizio aggiuntivo che gestisce il bilanciamento del carico computazionale tra istanze multiple di un microservizio. La necessità di questo servizio è perché si prevede che un elevato numero di microservizi siano contemporaneamente attivi nella Piattaforma ComESTo; pertanto, al fine di scongiurare congestione e fuori servizio, i microservizi della Piattaforma sono clonati (o replicati su più istanze) mediante una operazione di scaling orizzontale. Il Load Balancer è il servizio aggiuntivo che instrada le richieste alle specifiche istanze di un medesimo servizio. Dati tutti i microservizi e le loro molteplici istanze, il servizio aggiuntivo Service Discovery è capace di individuare ciascuna istanza. Due sono le modalità di implementazione del Service Discovery: Client-Side Service Discovery (CSSD) e Server-side Service Discovery (SSSD). Nel caso di CSSD, il client contatta il Service Registrar e accede al registro, recupera le informazioni necessarie ad individuare un microservizio, contatta il microservizio cercato; la Fig. 5.2.15-1 illustra graficamente quanto appena detto. Nel caso di SSSD, il client non contatta il Service Registrar ma una interfaccia del servizio cioè una API gateway del Service Register. È questa stessa API gateway che, per conto del client, contatta il microservizio cercato e pone ad esso una richiesta per conto del client.

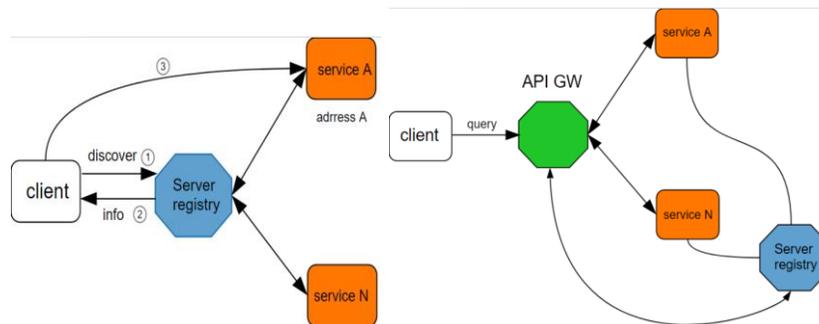


Figura 5.2.15-1 Identificazione del servizio client-side e server-side.

In particolare, Consul [xi] è stato scelto come strumento per il service orchestration e mentre nginx [xii] è stato scelto come Api gateway; è questa una soluzione semplice ed espandibile per gestire i servizi di 1) registrazione distribuita dei servizi, 2) discovery con tecniche Rest e/o DNS e 3) configurazione automatica. Consul fornisce anche meccanismi per autenticazione e protezione dei singoli servizi. La configurazione automatica dell'Api gateway avviene tramite un software aggiuntivo, consul-template, che interagisce con l'API gateway.

## 6. PROTOCOLLI DI COMUNICAZIONE: STATO DELL'ARTE

In generale, la gestione di risorse di flessibilità distribuite MT e BT richiede sistemi con tecnologie dell'informazione e della comunicazione addizionali, che devono garantire elevati livelli di interoperabilità tra i molteplici componenti della rete di distribuzione. A seguire sono dunque riportati i protocolli più recenti che permettono di gestire le risorse distribuite minimizzando la distanza di integrazione, secondo la trattazione proposta in [xiii-xiv]. Inizialmente sono presentati i protocolli o standard cross-funzionali o che trovano applicazione in più domini, seguono i protocolli utili per la gestione DSO-Aggregatore ed infine i protocolli tra sistema di controllo (DSO/Aggregatore) e dispositivo finale.

### 6.1. PROTOCOLLI E STANDARD CROSS-FUNZIONALI

Il Common Information Model (CIM), su cui sono basati gli standard IEC 61968-11 e IEC 61970-301, costituisce un riferimento sia nell'area della standardizzazione delle interfacce di sistema e modelli dati che nell'integrazione di applicazioni nei sistemi SW e HW di fornitori di energia, aggregatori e DSOs. Di conseguenza è ampiamente utilizzato nella gestione del sistema elettrico. Oltre agli standard sopracitati, sono stati sviluppati anche due standard di interfaccia:

- GID (Generic Interface Definitions), che rappresenta un'interfaccia indipendente dalla tecnologia ed è utilizzata per determinate classi e dati.

- SIDMS (System Interfaces for Distribution Management) che si basa su blocchi funzione dello standard IEC 61968 e messaggi basati su XML.

La famiglia di standard della serie IEC 61850 ha invece lo scopo di garantire interoperabilità nelle comunicazioni tra strumentazione di misura e di controllo, utilizzando una modalità di comunicazione unificata tra tutte le zone del modello SGAM (tipicamente campo, stazione e gestione). Figura 6.1-1 mostra le aree coperte dalla famiglia di standard IEC 61850. Infine, lo standard IEC TS 62361-102:2018 fornisce linee guida per aumentare l'interoperabilità tra CIM e IEC 61850 e di conseguenza anche l'efficienza nello scambio dati tra i due sistemi.

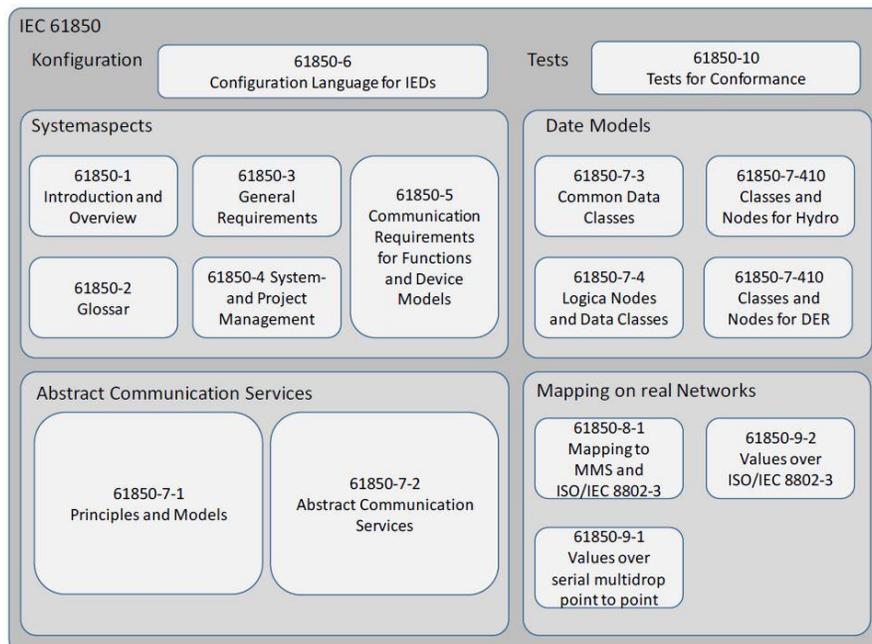


Figura 6.1-1 – Modello logico della struttura dello standard IEC 61850 [3].

## 6.2. PROTOCOLLI E STANDARD PER L'INTERAZIONE DSO-AGGREGATORE

Il modello Universal Smart Energy Framework (USEF) definisce la struttura e le interazioni di un mercato per risorse di flessibilità in cui l'aggregatore, attore principale del mercato, è in grado di offrire servizi di flessibilità al DSO. USEF distingue le interazioni tra tre differenti livelli, mostrati in Figura 6.2-1:

- Trasmissione e distribuzione dell'energia.
- Filiera della fornitura di energia, secondo il modello europeo di mercato liberalizzato.
- Filiera della fornitura di flessibilità, cui partecipano Aggregatori, BRPs, DSOs e TSOs.
- USEF è applicato a tutti e quattro i casi d'uso del progetto InterFlex in cui è prevista la figura dell'aggregatore.

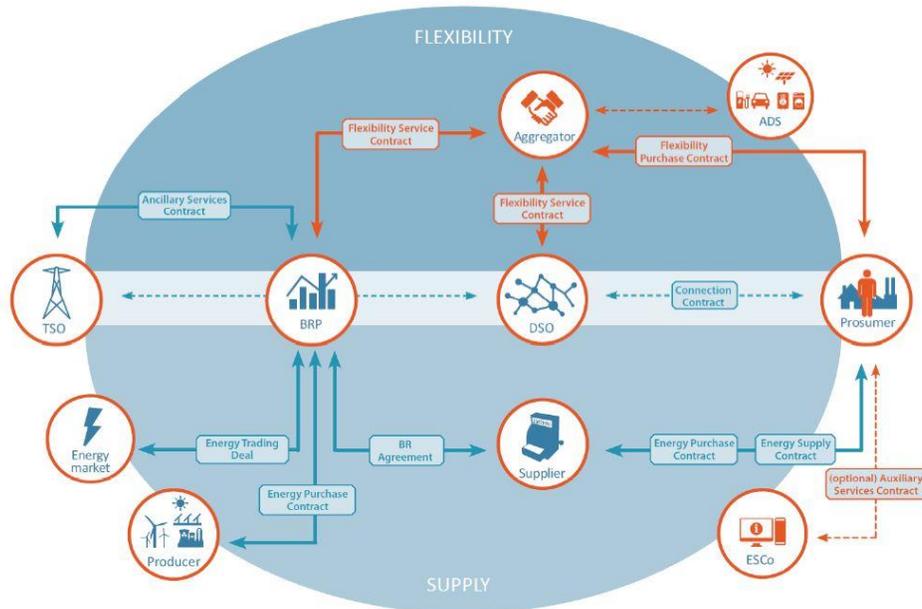


Figura 6.2-1 Modello di interazione di mercato USEF [3].

### 6.3. PROTOCOLLI E STANDARD PER IL CONTROLLO DELLE RISORSE FLESSIBILI

Il protocollo *Energy Flexibility Interface* (EFI) è un protocollo di comunicazione che permette ad applicazioni per la gestione di smart grid (EMS, DMS, etc.) di comunicare con un'ampia varietà di dispositivi che possono offrire servizi di flessibilità (es. pannelli solari, pompe di calore, punti di ricarica per EVs, etc.). EFI, infatti, permette di astrarre la disponibilità di flessibilità di un dispositivo, le cui caratteristiche non sono standardizzate, tramite un driver messo a disposizione dal fornitore stesso (si veda Figura 6.2-1). In questo modo l'aggregatore (o l'attore responsabile per la gestione della smart grid) non deve conoscere le caratteristiche tecniche del dispositivo per poter utilizzare le risorse di flessibilità messe a disposizione da esso. Esistono quattro categorie astratte di dispositivi in EFI:

- Dispositivi non controllabili
- Dispositivi a controllo temporale
- Dispositivi flessibili limitatamente a un buffer predefinito (sistemi di accumulo)
- Dispositivi flessibili non vincolati

Qualora sia richiesto dall'Aggregatore/DSO una modifica del comportamento del dispositivo astratto, EFI converte l'informazione secondo il formato specifico del venditore al fine di applicare il segnale corretto al dispositivo.

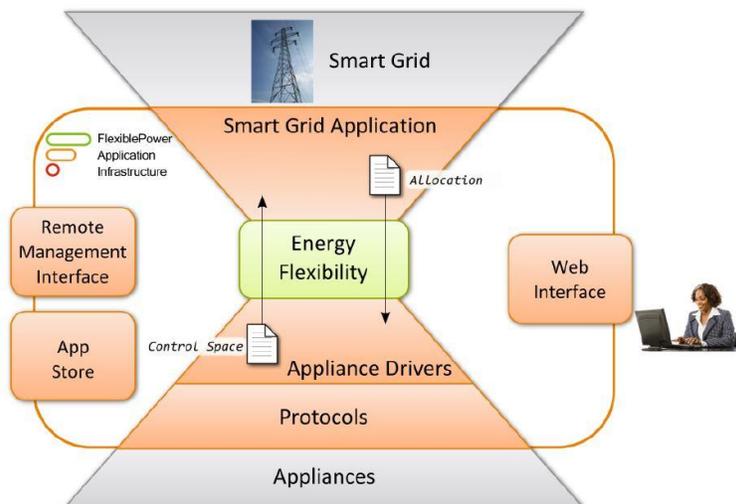


Figura 6.3-1 Schema della Energy Flexibility Interface [3].

Altri protocolli di comunicazione utilizzati per il controllo di singole risorse distribuite sono:

- *Open Charge Point Protocol (OCPP)*, protocollo di comunicazione tra il punto di ricarica per veicoli elettrici e il *Charging Station Management System (CSMS)*.
- *SunSpec*, protocollo di comunicazione e scambio dati per impianti fotovoltaici.
- *OpenADR*, standard di comunicazione per l'utilizzo e il controllo di risorse di *demand response*.
- *Smart Appliance REference Ontology (SAREF)*, interfaccia che garantisce una maggiore interoperabilità tra dispositivi IoT.

A titolo esemplificativo sono riportati in Tabella 6.3-1 e Figura 6.3-2 e Figura 6.3-3 i protocolli di comunicazione utilizzati tra ogni dispositivo nell'implementazione dei dimostratori NL 1-2 e FR 3 del progetto H2020 InterFLEX.

Tabella 6.3-1 Protocolli di comunicazione utilizzati nei casi d'uso NL US 1-2 e NL UC 3 del progetto InterFLEX [3].

ID	Da dispositivo	A dispositivo	Soluzione pianificata	Soluzione adottata	Criticità
NL 1.1a	RTU SSU	SSU	SUNSpec 800	Modbus	Interoperabilità
NL 1.1b	RTU PV	PV	SUNSpec 800	Modbus	Interoperabilità
NL 1.4	FAP DER	GMS	Definire standard di connessione tra DSO e aggregatore (basato su USEF)	USEF + (USEF modificato)	Interoperabilità
NL 2.1	Operatore CP	Punto di Ricarica EV		OCPP o GPRS/LTE	Cyber security
NL 2.2	CPMS	Operatore CP		OCPP o GPRS/LTE	Interoperabilità

<b>NL 2.4</b>	CPMS	FAP		OCPI	Cyber security
<b>NL 2.4</b>	FAP EV	GMS	Definire standard di connessione tra DSO e aggregatore (basato su USEF)	USEF + (USEF modificato)	Interoperabilità
<b>NL 3.2</b>	FAP	GMS	Definire standard di connessione tra DSO e aggregatore (basato su USEF)	USEF + (USEF modificato)	Interoperabilità
<b>FR 1.6</b>	RTU SSU	SSU	SUNSpec 800	Modbus TCP	Interoperabilità
<b>FR 3.3</b>	Piattaforma di mercato per flessibilità	Aggregatore	Definire standard di connessione tra DSO e aggregatore (basato su USEF o CIM)	XMPP, CIM Market	Interoperabilità
<b>FR 3.4</b>	RTU SSU	SSU	SUNSpec 800	Modbus TCP	Interoperabilità
<b>FR 3.5</b>	Aggregatore (Engie)	Carico	Connessione cablata o SAREF	KIWI/TOPKAPI HTTPS	Interoperabilità
<b>FR 3.7</b>	Server IT Gas	Carico ibrido (Boiler)	Connessione cablata o SAREF	3G	Interoperabilità
<b>FR 3.8</b>	Aggregatore (EDF)	Carico	Connessione cablata o SAREF	TCP/IP, SMS o SMTP	Interoperabilità

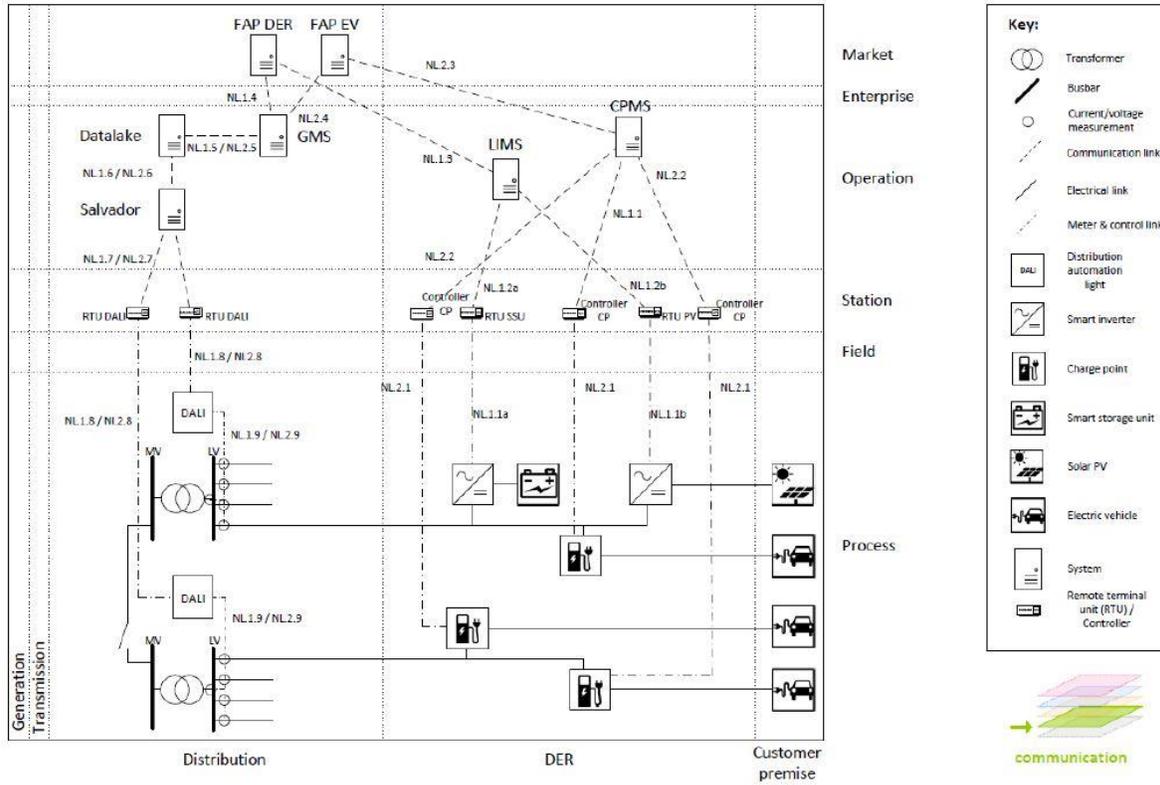


Figura 6.3-2 Livello di comunicazione secondo modello SGAM per NL UC 1-2 [xv]

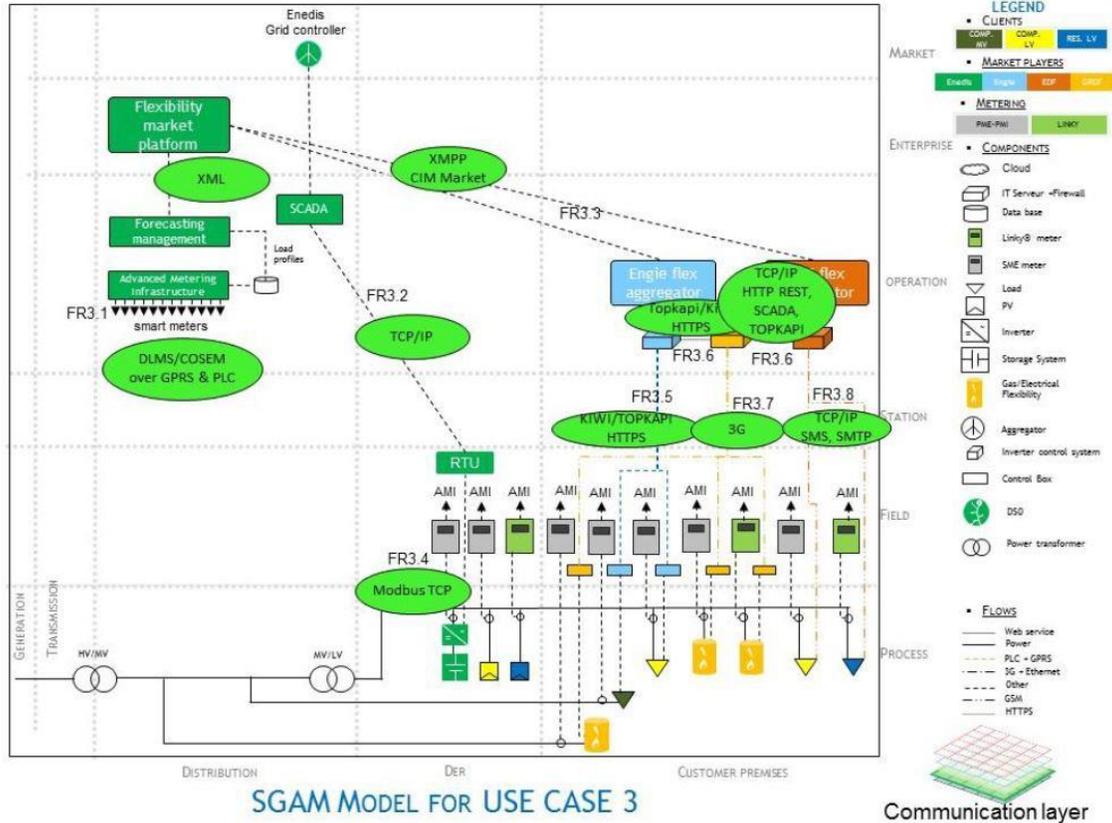


Figura 6.3-3 Livello di comunicazione secondo modello SGAM per FR UC 3 [3]

#### 6.4. I PROTOCOLLI DELLA PIATTAFORMA COMESTO

I protocolli impiegati internamente alla Piattaforma ComESto tra microsistemi sono i protocolli HTTP e WebSocket; questi protocolli sono utilizzati mediante un socket sicure del livello di trasporto per mezzo del protocollo TLS 1.2, basato su crittografia asimmetrica e generazione dinamica delle chiavi di cifratura [xvi]. Di conseguenza, il canale di comunicazione fra end-points garantisce la segretezza, la non forgiabilità, l'autenticità dell'informazione.

#### 6.5. STIMA DEL VOLUME DI DATI DELLA PIATTAFORMA COMESTO

Al fine di eseguire una stima del traffico dati e l'analisi prestazionale teorica dei canali di telecomunicazione in termini di Packet Loss e Mean Delay, il traffico medio dei dati di misura che arriva al microsistema KAI è così calcolato:

$$G = D \mathbb{E} \left[ \sum_{r=1}^M \sum_{j=1}^{N_r} \sum_{i=1}^{S_j} \tau_{ijr} \right] \quad (1)$$

dove:

$M$	Numero di impianti
$N$	Numero di dispositivi di misura per ogni impianto
$S$	Numero di grandezze misurate da ogni dispositivo
$\tau_s$	Tasso di invio medio delle misure di un dispositivo di misura (messaggi/sec)
$D$	Dimensione media di un pacchetto MQTT contenente dati di misura (bytes)
$P_l$	Probabilità di perdita di un pacchetto MQTT
$T_q$	Timeout di ritrasmissione per MQTT in QoS 1 (sec)
$N_r$	Numero di dispositivi di misura per ciascun impianto
$S_j$	Grandezze misurate per ciascun impianto

Se si ignorano i dettagli del protocollo e si considerano impianti simili tra loro, la (1) assume una forma semplificata:

$$G = DMNS\tau_s \quad (2)$$

che però non permette di valutare altre grandezze come il ritardo medio di consegna di un pacchetto MQTT in funzione del numero di trasmettitori (o sottoscrittori) MQTT, tenendo conto dei meccanismi di QoS e dei parametri del protocollo stesso, come i timeout, le ritrasmissioni e la capacità di memorizzazione dei messaggi sul broker. Un'analisi più dettagliata richiederebbe il ricorso alla teoria delle code ma tale analisi va oltre gli obiettivi dell'attività del progetto ComESTo. Esistono, tuttavia, studi empirici simulativi o emulativi [xvii-xviii], che possono dare un contributo al calcolo del traffico medio, tenendo conto delle sole variabili che maggiormente influenzano il ritardo nella trasmissione di dati. Una stima lineare del ritardo è, dunque, la seguente:

$$\text{ritardo} \sim \text{Msg MQTT} + \text{Msg attivi} + \text{Totale Sottoscrittori} + \text{SottoScrittori Messaggio PUB} \quad (3)$$

dove tutte le variabili sono intese come medie di una distribuzione gaussiana associata, il cui valore è stimato tramite regressione lineare. Per quanto in merito alla scalabilità, è indispensabile prevedere strategie di scalabilità all'aumentare del traffico MQTT. Una possibile strategia consiste nell'aggiungere istanze dello stesso broker in un cluster di container, eventualmente gestito da orchestratori tipo kubernetes [xix]. In questo modo, si realizza la strategia del bilanciamento o traffic shaping di modo che ogni istanza del broker riceva una quota di traffico consona alle sue capacità. RabbitMQ, essendo scritto in Erlang (linguaggio basato sul message-passing per la comunicazione inter-processo) ha già in "corpo"

la capacità di scalare orizzontalmente pertanto è possibile creare cluster di molteplici istanze di RabbitMQ e distribuire il traffico in maniera uniforme. Oltre al traffico, è importante sapere quale sarà l'impegno delle risorse della Piattaforma ComESTo. La parte più "critica" è quella relativa al database delle grandezze di misura. Anche qui, un calcolo molto approssimativo parte dalla Eq. (1). Considerando un periodo di 1 anno, otteniamo che un lower bound per l'impegno di memoria è:

$$R \geq DMNS\tau_S \cdot 31.53 \cdot 10^6 \text{ bytes} \quad (4)$$

## 7. BLOCKCHAIN NELLA PIATTAFORMA COMESTO

La forte penetrazione delle fonti energetiche rinnovabili nella rete elettrica, che raggiungerà una quota intorno al 30% entro il 2022 e supererà anche il 60% entro il 2050, rende necessaria la validazione di nuovi modelli di comunità energetica in grado di fornire servizi alla rete. Negli ultimi anni, i concetti di demand side management e demand response sono stati valorizzati e, sempre più frequentemente, applicati al fine di bilanciare la generazione di energia con il consumo di energia stessa e per aiutare a prevenire problemi di congestione delle reti. Applicando un approccio DR, gli utenti finali con programmi basati su incentivi, consentono ai fornitori di controllare tutti o alcuni dei loro carichi mediante specifica tecnologia. Ad esempio, durante le ore di punta, gli utenti possono ricevere incentivi per ridurre i carichi. Per fare ciò, una tecnologia di aggregazione che includa degli smart meter per inviare in tempo reale dati tra tutti gli utenti. Pertanto, la rete generale deve, sempre più, diventare una rete intelligente, in quanto ormai ci si trova a dover fare i conti non soltanto con flussi di energia, ma anche con un flusso di dati che, gestito correttamente, determinerà una distribuzione efficiente dell'energia attraverso l'intero sistema. La figura del soggetto terzo aggregatore (balance service provider) riveste un ruolo di interfaccia tra gli utenti attivi o passivi connessi alla rete di bassa, media e alta tensione e il c.d. TSO. L'aggregatore deve, quindi, conoscere le caratteristiche tecniche delle risorse aggregate (vincoli tecnici, potenza massima in prelievo e in immissione, codice POD, piani di lavoro, manutenzione e fuori servizio, etc.) e deve, altresì, disporre di idonei apparati per la raccolta di informazioni e l'invio e la ricezione di segnali in tempo reale, sia con il TSO che con gli utenti. Il paradigma in questione si basa sull'architettura tecnologica impiegata dall'aggregatore e si fonda su due concetti chiave, vale a dire: energia transattiva e peer-to-peer, per mezzo della tecnologia blockchain. In Italia, la delibera n. 300/2017 dell'Autorità di Regolazione per Energia Reti e Ambiente ha previsto l'apertura del mercato dei servizi di dispacciamento alla domanda e alle unità di produzione ad oggi non abilitate (inclusi i sistemi di accumulo). Secondo la citata delibera, le unità di produzione e di consumo non rilevanti possono essere abilitate alla partecipazione al mercato dei servizi di dispacciamento su base aggregata, nel rispetto di opportuni criteri di localizzazione geografica, con- correndo a formare dei punti di dispacciamento per unità virtuali abilitate (UVA), distinti per unità virtuali abilitate di

produzione (UVAP), unità virtuali abilitate di consumo (UVAC), unità virtuale abilitate miste (UVAM) e unità abilitate virtuali nodali (UVAN). Per raggiungere questo obiettivo, l'approccio centralizzato utilizzato nei sistemi di alimentazione tradizionali non risulta applicabile perché, nella nuova prospettiva, entrambi, prosumer ed utenti attivi, vorrebbero prendere parte alle transazioni energetiche e, pertanto, dovrebbe essere adottato un approccio decentralizzato basato sui sistemi transattivi e transazioni energetiche peer-to-peer. In questo contesto, la tecnologia distributed ledger (DLT), basata sul concetto di blockchain, si presenta come la soluzione più promettente per consentire smart-contract tra prosumer e utenti attivi protetti in modo sicuro in blocchi con crittografia hash.

### 7.1. LA TECNOLOGIA BLOCKCHAIN PER LE TRANSAZIONI ENERGETICHE

In un contesto di comunità energetica basata sul modello peer-to-peer, la gestione è decentrata e regolata tra i "pari" che partecipano alla rete energetica, che diventa una rete elettrica di energia virtuale. In un'architettura centralizzata, la comunicazione tra prosumer dovrebbe, invece, essere autorizzata da server centralizzati ed il set di requisiti per la loro corretta identificazione/interazione, aumenta con l'aumentare del numero dei nodi. Pertanto, le architetture tecnologiche centralizzate non sono facilmente scalabili (incrementabili, al crescere della popolosità) per gestire un aumento esponenziale dei prosumer, che invece produce alti volumi di dati ad alta frequenza di trasmissione. Di conseguenza, i costi di integrazione per rendere conto dell'entità di tale aumento renderebbe un sistema centralizzato per la gestione delle transazioni non economicamente sostenibile. La valutazione delle prestazioni di un sistema centralizzato per la gestione delle transazioni appare complessa, potendo considerare caratteristiche diverse, tra le quali:

- sicurezza dei dati, poiché vengono scambiati anche dati finanziari;
- riservatezza dei dati, perché lo scambio di dati sull'energia può "profilare" l'utente, ovvero, identificarlo;
- velocità delle transazioni finanziarie o inserimento veloce delle transazioni;
- resilienza ai guasti e integrità dei dati;
- impronta di energia ridotta: il sistema deve, cioè, consumare poca energia.

Le transazioni tra utenti peer to peer sono decisamente più economiche rispetto a quelle dei sistemi tradizionali basati su un sistema centralizzato. A voler instaurare un parallelismo con le reti di computer in cui il valore di scambio è un file (o un database), P2P riduce i costi del sistema ed evita l'installazione di un hardware centralizzato. Inoltre, i server centralizzati rappresentano singoli punti di errore (aree di criticità o "colli di bottiglia") e possono, quindi, rappresentare facili bersagli per attacchi da parte di hacker. I carichi e i ritardi dei server possono, però, essere ridotti sfruttando le capacità di reti P2P che si adattano con il numero di dispositivi collegati. Per sintetizzare, i principali vantaggi di un P2P

decentralizzato, rispetto a uno centralizzato, possono, dunque, apprezzarsi in termini di scalabilità, resilienza, adattabilità, tolleranza agli errori, sicurezza e fiducia. Immaginare micro-grid totalmente disconnesse dal DSO appare ancora prematuro; uno sviluppo più realistico del sistema transattivo potrebbe essere, piuttosto, costituito da una soluzione ibrida, dove la regolamentazione e l'accesso alle informazioni richieste per l'implementazione di energia P2P basati su transazioni energetiche vengano gestiti da un fornitore di servizi tecnologici alla comunità energetica. Tale soggetto, che potrebbe corrispondere alla figura dell'aggregatore, potrebbe gestire centralmente le transazioni energetiche, garantendo un'interazione con DSO e TSO. Ciò significa che:

- le transazioni relative ai dati energetici (flussi energetici scambiati) devono essere confermate dai "pari", utilizzando una sorta di protocollo di consenso incorporato in una routine ad esecuzione condivisa, conosciuta come smart contract;
- le transazioni devono essere memorizzate in modo sicuro all'interno di peer (nodi) che partecipano al programma.

Il P2P può fare riferimento, sia alle modalità in base alle quali si svolgono le transazioni energetiche (procedure informatiche che definiscono l'interazione tra i nodi), che all'architettura informatica (infrastruttura tecnologica impiegata dai nodi) che le supporta. Di conseguenza, dal punto di vista dell'architettura dell'informazione, DLT è anche una architettura basata su P2P e, quindi, può sembrare una tecnologia ideale per implementare un sistema TE basato su scambi di energia P2P (P2PTE). Per quanto riguarda il tipo di accesso alla rete P2P, si usa distinguere tra un accesso "senza permesso" e un accesso "con permesso". Ebbene, un'architettura senza permesso è una rete pubblica in cui tutti possono partecipare senza alcun meccanismo speciale di autenticazione/autorizzazione. Tutti i pari sono, dunque, anonimi e potenzialmente non affidabili. In un'architettura autorizzata, l'accesso alla rete deve, invece, essere concesso (ad esempio mediante la registrazione della propria identità a un centro dati centrale), anche se il centro non interferisce durante lo scambio di informazioni tra pari (ad esempio, Skype potrebbe essere pensato come una rete P2P autorizzata). In un'architettura autorizzata, i pari condividono un certo tipo di fiducia e l'identità non è completamente anonima. Per abilitare le transazioni di energia P2P tra prosumer e utenti attivi all'interno di un contesto di mercato locale, la Blockchain sembra essere una soluzione promettente, in quanto può fare a meno di un intermediario e garantire transazioni quasi in tempo reale. Tali transazioni sono memorizzate in blocchi di dati concatenati uno dopo l'altro per rendere la falsificazione dei dati pressoché impossibile. Lo stoccaggio dei dati delle transazioni nei blocchi è protetto da funzioni crittografiche e metodologie di sicurezza a livello industriale (ad esempio firma basata sul paradigma a chiave pubblica). La blockchain è, pertanto, memorizzata globalmente nella rete di "partecipanti pari" e costituisce un'applicazione virtuosa per il decentramento di algoritmi. Nell'universo di Internet, applicazioni piuttosto popolari basate sulla comunicazione decentrata di informazioni sono Bittorrent (per condividere file), Skype e

Whatsapp (per condividere comunicazioni con i propri contatti). Nell'universo blockchain, l'oggetto condiviso è, essenzialmente, un qualche tipo di "valore": una moneta virtuale o cripto-valuta, contratti intelligenti, beni virtuali, etc. Mentre nell'approccio centralizzato, le parti partecipanti si rapportano con l'autorità centrale (ad esempio, nel caso di un istituto bancario), nel P2P viene utilizzato un meccanismo funzionale a raggiungere una sorta di fiducia. Questo trust deve almeno garantire: 1) anonimato; 2) impossibilità di ripudiare una transazione una volta che è stata salvata dal sistema; 3) probabilità molto bassa di falsificazione dei dati salvati; 4) resilienza a possibili attacchi. I profili sub 2) e 3) sono generalmente garantiti usando hash crittografici che definiscono ogni blocco e il concatenamento di tutti i blocchi tra loro. In questo modo, l'operazione di contraffazione dovrebbe essere in grado di modificare tutta la catena. Questo non è un nuovo concetto, essendo noto dal 2009, da quando, cioè, la tecnologia blockchain è stata utilizzata dalla piattaforma Bitcoin per cryptocurrency, per abilitare transazioni virtuali sicure. Un DLT basato su blockchain potrebbe gestire centinaia o addirittura migliaia di contratti intelligenti in tempo quasi reale e senza ostacoli dovuti alla progettazione e alla manutenzione del data center centralizzato. Però, l'algoritmo originale definito in DLT come bitcoin è in grado di far fronte ad un massimo di sette transazioni al secondo. Perciò sono state concepiti alcuni sistemi alternativi, come Ethereum, che può gestire decine di transazioni energetiche al secondo, o Hyperledger, che può far fronte a centinaia di transazioni al secondo e presenta l'ulteriore vantaggio di costituire una soluzione utilizzabile per volumi sempre crescenti di transazioni e/o nodi, molto adatta per gli smart contract. Un esempio di un progetto in corso finalizzato alla sperimentazione del trading di energia P2P basato sulla blockchain è "Transactive grid" di L03 Energy, dove prosumers inseriti all'interno di una piccola comunità energetica, quella di Brooklyn - USA, possono comprare e vendere energia l'uno con l'altro usando la piattaforma Ethereum per i contratti intelligenti. Un altro esempio è quello della società britannica Electron, che ha usato la tecnologia blockchain per creare una piattaforma open source, al fine di fornire misurazioni veritiere.

## 7.2. PROTOCOLLI DI CONSENSO

In un mercato locale basato su blockchain, poiché non esiste un'autorità centrale che gestisca le transazioni energetiche, tutti i prosumer (o nodi dal punto di vista di un sistema) devono accordarsi su una transazione prima di memorizzarla nella blockchain. La transazione (o blocco, cioè gruppo di transazioni) è valida se, e solo se, viene raggiunto un consenso tra tutti i nodi. I protocolli di consenso sono un insieme di algoritmi e dati strutturati ben noto in molti campi dell'ingegneria, come l'informatica e l'elaborazione del segnale. Le proprietà chiave o requisiti di un protocollo di consenso sono:

- sicurezza: i nodi che prendono parte ad una procedura di consenso producono lo stesso output secondo le regole del protocollo.

- vitalità: tutti i nodi sani che prendono parte al consenso producono un valore.
- tolleranza ai guasti: se un nodo che prende parte al consenso fallisce, il protocollo di consenso può continuare a funzionare.

I protocolli di consenso più comuni in DLT basati su blockchain sono: proof-of-work (PoW), proof-of-stack (PoS) e bizantine Fault Tolerant (BFT). Il protocollo PoW è utilizzato da piattaforme non autorizzate (come il già citato Bitcoin o piattaforme Ethereum) in cui un gran numero di nodi non fidati cercano il consenso per approvare una transazione energetica. L'algoritmo PoW sembra essere l'opzione migliore per quanto riguarda la sicurezza dei dati privati, perché tutti i nodi devono risolvere un difficile puzzle crittografico prima di aggiungere un blocco alla catena; ciò rende il sistema impermeabile. Il processo per trovare una soluzione a questo puzzle è chiamato mining. Per ottenere il diritto di approvare il nuovo blocco (e quindi per ottenere anche un profitto economico), è necessario investire in hardware: più l'hardware è potente, più alta è la probabilità di risolvere rapidamente il puzzle crittografico. Una volta che una soluzione viene raggiunta, gli altri peer possono, dunque, confermare la soluzione. Anche se è stato sottolineato che i DLT basati su blockchain potrebbero consentire scambi TE tra molteplici parti DER, occorre prestare attenzione all'impronta energetica di (qualsiasi) tecnologia DLT. In effetti, uno studio recente rivela che l'"impronta energetica" della blockchain del Bitcoin è simile al consumo medio irlandese di energia elettrica [72], a causa della fame di energia di algoritmi PoW. Inoltre, il Bitcoin richiede oltre 3 GB di dati compressi per contenere l'intera blockchain, ovviamente superando le capacità di inverter intelligenti o controllori transattivi. Al fine di ridurre l'impatto energetico nel Proof of Work implementato da Bitcoin, il protocollo Proof of stake sostituisce il processo di mining effettuato dal nodo che funge da valutatore. In altre parole, viene concesso il diritto di convalidare e inserire un nuovo blocco a quel nodo (peer) che può dimostrare la proprietà di una certa somma di una variabile chiamata stake (nella cripto-valuta lo stake può essere anche la valuta stessa). La selezione basata solo su stake presenta, tuttavia, alcuni problemi. Come si dirà, PoS appare, comunque, un ottimo sistema da utilizzare nel contesto energetico. Pertanto, nel prosieguo, verrà formulato un esempio di architettura basata su PoS, mostrando come i problemi di base del PoS possono essere facilmente evitati usando un'architettura con permesso e valori stake difficili da falsificare. Una versione modificata molto interessante di PoS viene, ad esempio, utilizzata in Solar-Coin, una interessante piattaforma per la vendita di energia solare attraverso impianti di produzione certificati. In Solar Coin, il proprietario di ogni impianto fotovoltaico registra, infatti, le proprie installazioni fotovoltaiche, diventando così un prosumer, ovvero, un produttore/consumatore di energia. Successivamente, dopo la verifica da parte della piattaforma Solar-Coin dell'identità e dei dettagli dei componenti dell'installazione, il proprietario ottiene l'accesso alla piattaforma e riceve un portafoglio digitale. In Solar Coin, il software installato lato utente sullo smart inverter comunica la produzione di energia soltanto se un blocco di transazioni "solari" è inserito nel DLT. Per ogni MWh

prodotto, la piattaforma restituisce alcune “monete solari” (virtual token) e le transazioni vengono archiviate nel portafoglio digitale per mezzo della blockchain. Non c'è registro centralizzato per le transazioni. Viene, piuttosto, utilizzato il protocollo Byzantine Fault Tolerant per rilevare disallineamenti tra le informazioni condivise tra tutti i nodi, evitando così il malfunzionamento dell'intero sistema. Recentemente, con la visione di un DLT altamente scalabile per il mondo IoT, sono emersi altri concetti. Ad esempio, nell'algoritmo Tangle, la validazione e l'inserimento delle transazioni sono basati su grafi aciclici, e non su catene di blocchi. Tangle è il nucleo algoritmo del popolare token crittografico “IOTA”.

### 7.2.1. LO SVILUPPO DEGLI SMART CONTRACTS PER LA GESTIONE DELLE INTERAZIONI TRA I PROSUMERS E L'AGGREGATORE

Si presume che un aggregatore energetico del tipo virtual power plant possa fornire servizi al gestore del sistema di trasmissione ai fini della partecipazione al mercato dei servizi di bilanciamento, come nel caso della sperimentazione UVAM di Terna. L'architettura tecnologica in carico all'aggregatore energetico può, infatti, essere utilizzata per fornire una affidabile rapida comunicazione a due vie, consentendo all'aggregatore di interfacciarsi con i prosumer inclusi nel proprio aggregato e con parti esterne come il DSO, il TSO o il gestore del mercato. Durante il giorno prima (o ora prima) l'aggregatore fornisce anche un programma di generazione/carico per i prosumer aggregati, considerando i vincoli tecnici della rete di trasmissione. Il programma del giorno prima viene fatto per consentire all'aggregatore di partecipare efficacemente al mercato dei servizi di dispacciamento. Il programma si basa, infatti, su dati storici e sulla previsione del carico elettrico per ogni prosumer, oltre alla previsione della produzione. Inoltre, il programma dovrebbe dipendere dalla strategia che l'aggregatore adotta per partecipare al mercato dei servizi ancillari. Questa scelta dipende, quindi, dal prezzo e dalle previsioni nel mercato dell'elettricità e sulle fasce orarie ottimali per la vendita/acquisto di energia nel mercato dei servizi ancillari. I prosumer associati al VPP dovrebbero essere connessi alla stessa rete di distribuzione elettrica, così che lo scambio energetico fisico determinato dall'aggregatore possa avere luogo nel rispetto dei vincoli tecnici di rete. La fattibilità tecnica dello scambio dovrebbe essere stata precedentemente approvata dal DSO, e, se si presentano problemi tecnici nelle reti di distribuzione, dovrebbe poter chiedere all'aggregatore di modificare il proprio programma. Se l'aggregatore non può seguire il programma del giorno prima in tempo reale, a causa di una deviazione di energia causata da errori nella previsione di energia elettrica generata da RES o assorbita da carichi, allora, potrà formulare un'offerta a tutti i suoi aggregati prosumer per vendere o acquistare, rispettivamente, la necessaria quantità di energia elettrica ad un prezzo determinato. A loro volta, i prosumer potranno reagire all'offerta dell'aggregatore formulando una propria offerta. Infatti, quando un'offerta viene

fatta dall'aggregatore, uno smart contract viene distribuito alla blockchain e viene avviata un'asta consentendo ai prosumer di presentare offerte. Il programma dello smart contract, che è un insieme di regole codificate nella blockchain, consente, quindi, di effettuare un'asta tra prosumer che consente di selezionare le offerte più vantaggiose economicamente che potranno dare luogo a transazioni energetiche nel mercato locale dell'aggregatore. Il tipo di asta codificato all'interno del contratto intelligente è concordato tra l'aggregatore e i prosumer. Ad esempio, quando l'aggregatore fa un'offerta, le offerte dei prosumer che hanno prezzi inferiori all'offerta dell'aggregatore vengono selezionate in ordine crescente di prezzo fino al raggiungimento della quantità di energia richiesta dall'offerta (merit order). Dopo di ciò le transazioni sono completate e verificate dai dispositivi locali intelligenti (ovvero dispositivi tecnologici in grado di comunicare con smart meters, inverters fotovoltaici e storage elettrici) del prosumer e, infine, vengono autorizzati gli scambi di cripto-valute all'interno della blockchain, secondo quanto definito negli smart contract.

## 8. CENNI ALLE PROBLEMATICHE DI SICUREZZA

Un'analisi dettagliata ed esaustiva delle problematiche di sicurezza inerenti ad architetture IoT come quella della Piattaforma ComESTo è evidentemente una attività che va oltre le finalità del presente documento. Per tale motivo, verranno forniti brevi cenni al tema della sicurezza delle architetture IoT con riferimento a quanto in letteratura [xx]. Le problematiche di sicurezza sono classificate in livelli. Il livello "perception" o sensoristico dipende dalle tecnologie di accesso dei singoli dispositivi di misura. Ad esempio, Eugenio può raccogliere dati sia da sensori Z-Wave che WiFi. Il protocollo Z-Wave S2 utilizza una crittografia asimmetrica basata su scambio di chiavi dinamico. Il Node Capture Attack (NCA) e lo Spoof Node Attack (SNA) sono alquanto improbabili in un contesto Z-Wave, visto che il protocollo fisico e il protocollo applicativo per lo scambio delle chiavi è stato progettato tenendo presente proprio i suddetti attacchi. In Z-Wave, ad esempio, ogni dispositivo di tipo S2 è dotato di una chiave DSK che deve essere inserita all'atto dell'associazione del dispositivo nella rete gestita dal gateway, in questo caso Eugenio. Tutto il traffico è cifrato con le chiavi scambiate. Anche l'attacco Energy Manipulation Attack (EMA) è improbabile. Esso si riferisce alla possibilità di depauperare i singoli nodi della loro fonte di energia (batterie) e renderli perciò inutilizzabili. In un contesto di energy, difficilmente troveremo dispositivi di misura funzionanti a batteria! Per inciso, gli smart meter di Eugenio sono tutti alimentati dalla rete, così come anche le nanogrid. Gli attacchi NCA e SNA vanno invece esaminati nel dettaglio per tutti gli altri componenti, come le nanogrid. Dal nostro punto di vista, essendo i protocolli di trasporto basati tutti su TLS 1.2, o qualche variante come nel caso di Z-Wave, è molto improbabile che si riesca a perpetrare un attacco a livello di sensore o di rete. L'impersonificazione è possibile, ma siamo già nel livello Application. Qui, effettivamente, possono accadere tutti i tipi di attacchi menzionati nella Figura 19. Il phishing è relativamente semplice: un attaccante potrebbe forgiare un Access Point fasullo

per ridirigere le applicazioni verso un ambiente simile all'originale. L'ambiente fasullo funge da assorbitore delle informazioni sensibili, come le credenziali di accesso. Una volta in possesso delle credenziali per l'accesso alla piattaforma, l'attaccante può eseguire ogni sorta di danno. Sempre al livello delle applicazioni, e quindi gestionale della Piattaforma ComESTo, sono possibili attacchi legati al cosiddetto privacy o data leakage. I pattern energetici potrebbero rivelare dati o comportamenti sensibili degli utenti finali. Così come, la mancanza di regole di ACL piramidali potrebbe portare, volontariamente o involontariamente, a interventi non permessi. Ad esempio, accensione o spegnimento di apparecchiature, forgiature di dati di misura, eccetera. In questi ed altri casi esistono delle soluzioni.

### 8.1. INCREMENTO DEL LIVELLO DI SICUREZZA DELLE PROCEDURE DI AUTENTICAZIONE

Per proteggere l'accesso ai gestionali della *Piattaforma ComESTo* tramite spoofing dell'impersonificazione nel layer di accesso, si potrebbe ricorrere alla doppia autenticazione o autenticazione a due fattori. In altri termini, oltre a username e password occorre un secondo valore da presentare alla *Piattaforma ComESTo* per usare il gestionale. Questo secondo fattore potrebbe essere un codice usa e getta comunicato tramite canali paralleli. Oppure, si potrebbe proteggere l'accesso tramite una sentinella che determina se l'accesso è anomalo. Tramite semplici tecniche di change detection è possibile, infatti, stabilire se un accesso è anomalo oppure no. Questa tecnica consiste nel raccogliere nel tempo i parametri di ogni accesso (indirizzo IP, tipo di dispositivo, coordinate geografiche, eccetera) e costruirvi un profilo di riferimento, tramite considerazioni statistiche. L'algoritmo di change detection potrebbe bloccare ogni accesso marcato come anomalo.

## 9. METODOLOGIA DI SOFTWARE ENGINEERING APPLICATA

Il software da implementare nella Piattaforma ComESTo sarà sviluppato secondo le fasi del System Development Life Cycle (SDLC). Un SDLC mira a produrre sistemi di alta qualità che soddisfino o superino le aspettative dei committenti e che si muovono attraverso ogni fase entro tempi programmati. I sistemi informatici sono complessi e spesso (soprattutto con il recente aumento delle architetture orientate ai servizi ) collegano fra loro sistemi tradizionali potenzialmente forniti da diversi fornitori di software.



Figura 9-1 Fasi del SDLC

Le fasi di un SDLC, rappresentate in Figura 9-1, sono le seguenti:

- **Raccolta e definizione dei requisiti.** E' la prima fase del processo, fornisce una rappresentazione più chiara dell'intero progetto, definisce gli obiettivi ed individua potenziali problematiche. E' fortemente necessario che i gruppi di lavoro producano requisiti dettagliati e puntuali in modo da finalizzare la timeline di completamento dei lavori
- **Studio di fattibilità.** In questa fase, si mettono le specifiche del software da realizzare sono messe per iscritto all'interno di un documento conosciuto con il nome di "Software Requirement Specification" (SRS). L'SRS contiene anche requisiti non funzionali e può includere un insieme dei casi di utilizzo che descrivono le interazioni che gli utenti devono avere con il software. Le verifiche di fattibilità possono essere raggruppate in cinque tipologie:
  - **Economica:** stabilisce se il progetto può essere completato nel rispetto del budget previsto
  - **Legale:** verifica che il progetto sia conforme al diritto informatico e rispetti i quadri normativi previsti dal dominio applicativo in cui il sistema software si colloca
  - **Fattibilità operativa:** determina se è possibile creare operazioni informatiche che soddisfino le aspettative dei clienti
  - **Tecnica:** consolida che l'architettura informatica scelta sia in grado di supportare il software
  - **Rispetto delle scadenze:** si accerta che il progetto può essere completato o meno nei tempi stabiliti

- **Progettazione del sistema**

Sulla base del documento di specifica dei requisiti, si producono i documenti di progetto del software che aiutano a definire l'architettura generale del sistema. Tali documenti appartengono a due categorie:

**High-Level Design (HLD)** in cui si rappresentano:

- il nome dei moduli ed una breve descrizione di ciascuno di essi
- una panoramica delle funzionalità di ciascuno modulo
- relazioni e dipendenze fra i moduli
- le tabelle della base dati con i relativi elementi chiave
- i diagrammi completi dell'architettura corredati di dettagli sulla tecnologia scelta

**Low-Level Design (LLD)** in cui si definiscono:

- la logica funzionale dei moduli
- tabelle della base dati che devono includere tipi e dimensioni di ciascun campo
- dettagli completi delle interfacce
- la risoluzione di tutti i problemi di dipendenza

- l'elenco dei messaggi di errore
- ingressi e uscite di ciascun modulo
- **Implementazione o coding**

In questa fase, le attività sono suddivise in unità o moduli ed assegnate agli sviluppatori che iniziano a costruire il sistema scrivendo codice nel/nei linguaggio/i di programmazione scelto/i ed utilizzando software per l'ausilio alla programmazione come compilatori, interpreti e debugger. In letteratura, l'implementazione è conosciuta come la fase del progetto caratterizzata dalla maggiore durata.
- **Testing**

Terminato lo sviluppo del software, questo viene rilasciato in ambiente di test e si avvia il processo di verifica con l'obiettivo che le funzionalità dell'intera applicazione rispettino i requisiti individuati. Durante la fase di testing, gruppi di lavoro dedicati a tale scopo (detti di testing e quality assurance), hanno il compito di identificare errori e comportamenti anomali e segnalare gli stessi agli sviluppatori che dovranno intervenire per correggerli. Si tratta di un processo ciclico che continua finché il software sarà stabile e privo di anomalie. In letteratura, per ambiente di test si intende l'insieme di macchine e storage configurato in maniera speculare allo stesso insieme presente in ambiente di produzione
- **Rilascio**

Il sistema, ormai privo di errori e stabile, viene portato (rilasciato) in ambiente di produzione ed è pronto per essere utilizzato dagli utenti finali
- **Manutenzione**

L'obiettivo principale di questa fase è garantire che i requisiti continuino ad essere soddisfatti e che il sistema continui a funzionare secondo le specifiche menzionate nella prima fase. Questo viene garantito dalle seguenti tre attività:

  - **Bug Fixing:** si identificano e correggono errori generati da scenari non testati in maniera opportuna
  - **Aggiornamento (o Upgrade):** l'applicazione viene aggiornata a nuove versioni di software
  - **Miglioramento (o Enhancement):** il software viene arricchito di nuove funzionalità

Nell'ambito del progetto ComESTo, è fondamentale seguire un approccio fortemente strutturato allo sviluppo della Piattaforma ComESTo come l'SDLC per due motivi principali:

1. La complessità del sistema informatico da implementare
2. L'eterogeneità e l'elevato numero di fornitori che compongono il partenariato di progetto

Per gestire sistemi informatici complessi, categoria a cui la *Piattaforma ComESTo* appartiene, esistono in letteratura diversi modelli o metodologie SDLC suddivisi in due categorie:

- **Predittive:** presuppone che sia possibile pianificare ed organizzare lo sviluppo di un progetto. Implica la piena comprensione del prodotto finale e la determinazione del processo per la sua consegna. In questa forma di ciclo di vita del progetto, si determina il costo, l'ambito e la tempistica nelle prime fasi del progetto.
- **Adattive:** è un mix di sviluppo incrementale e iterativo. Si tratta di aggiungere funzionalità in modo incrementale e apportare modifiche e perfezionamenti in base al feedback. In altre parole, il lavoro può adattarsi facilmente alle mutevoli esigenze in base ai feedback ricevuti dal committente.

Il modello predittivo più noto in letteratura è sicuramente il waterfall (a cascata) mentre il modello Agile rappresenta il caposaldo dei modelli adattivi. I paragrafi seguenti descrivono, sinteticamente, pro e contro delle due metodologie mentre nel paragrafo “**Modello adottato per lo sviluppo del ComESTo: Agile-Waterfall Hybrid**” si motiva la sua aderenza al progetto ComESTo

### 9.1. MODELLO A CASCATA (WATERFALL)

Si tratta del primo modello SDLC predittivo ad essere ampiamente utilizzato nell'Ingegneria del Software per garantire il successo di un progetto. L'intero processo di sviluppo del software prevede la produzione di documentazione ad ogni step, segue fedelmente le fasi SDLC sopra descritte ed il risultato di ciascuna fase agisce da input per la successiva. La Figura 9.1-1 mostra una rappresentazione grafica del modello Waterfall

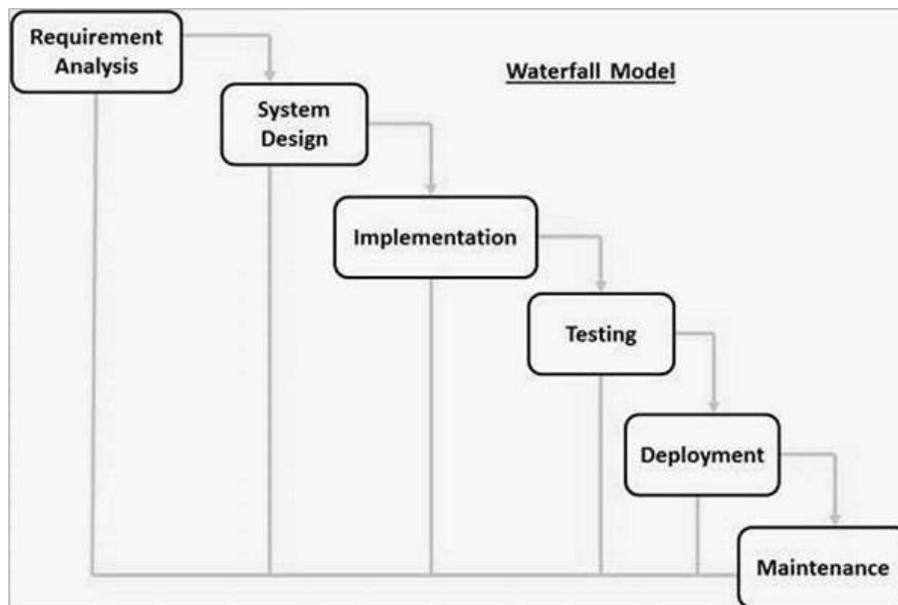


Figura 9.1-1 Modello SDLC - Waterfall

### Vantaggi

Se correttamente applicato, lo sviluppo a cascata presenta molti benefici nella gestione di un progetto, poiché:

- fornisce un elevato grado di controllo a livello manageriale poiché ogni fase di sviluppo ha punti di inizio e fine chiaramente definiti
- struttura in maniera puntuale le attività da eseguire in quanto sono stati preventivamente definiti obiettivi, tempistiche e risultati di progetto
- la presenza di una solida documentazione riduce al minimo il verificarsi di errori legati ad incomprensioni o malintesi

### Svantaggi

La rigidità che contraddistingue tale approccio rappresenta il limite più forte di tale modello, poiché:

- il software funzionante è disponibile solo nelle ultime fasi
- E' complesso misurare gli avanzamenti di progetto al susseguirsi delle fasi
- Non è consentita la modifica dei requisiti in corsa
- i test sul software sono eseguiti soltanto alla fine: la rilevazione di errori nelle ultime fasi del processo, potrebbe comportare rischi per il prodotto in quanto una rilavorazione completa è costosa e spesso non fattibile

### Applicabilità in ComESTo

Tale modello non è applicabile, in senso stretto, nel contesto di ComESTo poiché, dovendo affrontare tematiche innovative e sperimentali, occorre essere molto flessibili nel modificare in corso d'opera i

requisiti definiti in fase iniziale soprattutto in seguito all'analisi dei risultati forniti dall'implementazione dei servizi software che compongono la Piattaforma ComESto.

## 9.2. MODELLO AGILE

Il modello Agile è una combinazione di modelli di processo iterativi e incrementali con particolare attenzione all'adattabilità del processo e alla soddisfazione del committente grazie alla rapida consegna del prodotto software funzionante. I metodi agili suddividono il prodotto in piccole build incrementali che sono fornite in iterazioni successive. Ogni iterazione dura tipicamente da una a tre settimane circa. La Figura 9.2-1 mostra una rappresentazione grafica del modello Agile

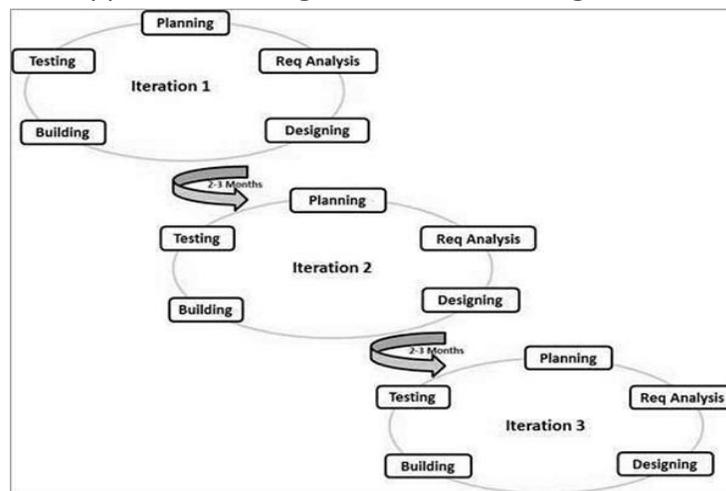


Figura 9.2-1 Modello SDLC - Agile

In Agile, le attività sono suddivise in time box (piccoli intervalli di tempo), detti anche sprint, per fornire funzionalità specifiche per un rilascio. Si segue un approccio iterativo e la build del software funzionante viene consegnata dopo ogni iterazione. Ogni build è incrementale in termini di funzionalità; la build finale contiene tutte le funzionalità richieste dal cliente. L'Agile si basa su metodi di sviluppo software adattivi, mentre i modelli SDLC tradizionali come il modello a cascata si basano su un approccio predittivo. I team predittivi nei modelli SDLC tradizionali, di solito, lavorano con una pianificazione dettagliata e hanno una previsione completa delle attività e delle funzionalità esatte da fornire nei prossimi mesi o durante il ciclo di vita del prodotto. I metodi predittivi dipendono interamente dall'analisi dei requisiti e dalla pianificazione eseguita all'inizio del ciclo. Qualsiasi modifica da incorporare passa attraverso una rigorosa gestione del controllo delle modifiche e assegnazione delle priorità. Agile utilizza un approccio adattivo in cui non esiste una pianificazione dettagliata e vi è chiarezza sui compiti futuri solo rispetto a quali caratteristiche devono essere sviluppate. È previsto uno sviluppo basato sulle funzionalità e il team si adatta dinamicamente ai requisiti del prodotto in

evoluzione. Il prodotto viene testato molto frequentemente, attraverso le iterazioni di rilascio, riducendo al minimo il rischio di eventuali guasti importanti in futuro. L'interazione con il committente è la spina dorsale di questa metodologia Agile e la comunicazione aperta con documentazione minima sono le caratteristiche tipiche dell'ambiente di sviluppo Agile.

### **Vantaggi**

I metodi agili sono stati ampiamente accettati nel mondo del software di recente. Tuttavia, questa metodologia potrebbe non essere sempre adatta a tutti i prodotti. I vantaggi riconosciuti di tale modello sono i seguenti:

- È un approccio molto realistico allo sviluppo del software.
- Promuove il lavoro di squadra e il cross training.
- Le funzionalità possono essere sviluppate rapidamente e dimostrate.
- I requisiti di risorse sono minimi.
- Fornisce soluzioni di lavoro parziali iniziali.
- Buon modello per ambienti che cambiano costantemente.
- Regole minime, documentazione facilmente utilizzabile.
- Facile da gestire.

### **Svantaggi**

La metodologia Agile potrebbe non essere sempre adatta a tutti i prodotti poiché:

- Non è adatta per la gestione di dipendenze complesse.
- Maggiori rischi di sostenibilità, manutenibilità ed estensibilità.
- Un piano generale, un leader agile e una pratica di PM agile è un must senza il quale non funzionerà.
- Una rigorosa gestione delle consegne determina l'ambito, le funzionalità da fornire e gli adeguamenti per rispettare le scadenze.
- Dipende fortemente dall'interazione del cliente, quindi se il cliente non è chiaro, il team può essere guidato nella direzione sbagliata.
- Esiste una dipendenza individuale molto elevata, poiché viene generata una documentazione minima.
- Il trasferimento di tecnologia a nuovi membri del team può essere piuttosto impegnativo a causa della mancanza di documentazione.

### **Applicabilità in ComESTo**

Il motivo principale che rende tale modello non applicabile in senso stretto è dovuto, ancora una volta, dalla natura sperimentale e fortemente innovativa delle tematiche affrontate nel contesto del progetto ComESTo. Procedendo senza una documentazione sufficiente o una visione chiara di come deve presentarsi il prodotto finale si corre il rischio che, da un lato, i teams coinvolti non riescano a

raggiungere gli obiettivi, dall'altro, i manager di progetto non abbiano strumenti per misurare gli avanzamenti

### 9.3. IL MODELLO ADOTTATO: AGILE-WATERFALL HYBRID

I progetti spesso optano per un approccio ibrido, combinando i tratti preferiti delle metodologie di sviluppo a cascata e agile per ridurre il rischio e aumentare il feedback e l'impatto delle parti interessate durante il progetto. Il modello sfrutta cinque fasi: pianificazione, requisiti iniziali e progettazione, sprint agili iterativi, controllo della qualità e distribuzione. La Figura 9.3-1 offre una rappresentazione di tale modello ibrido.

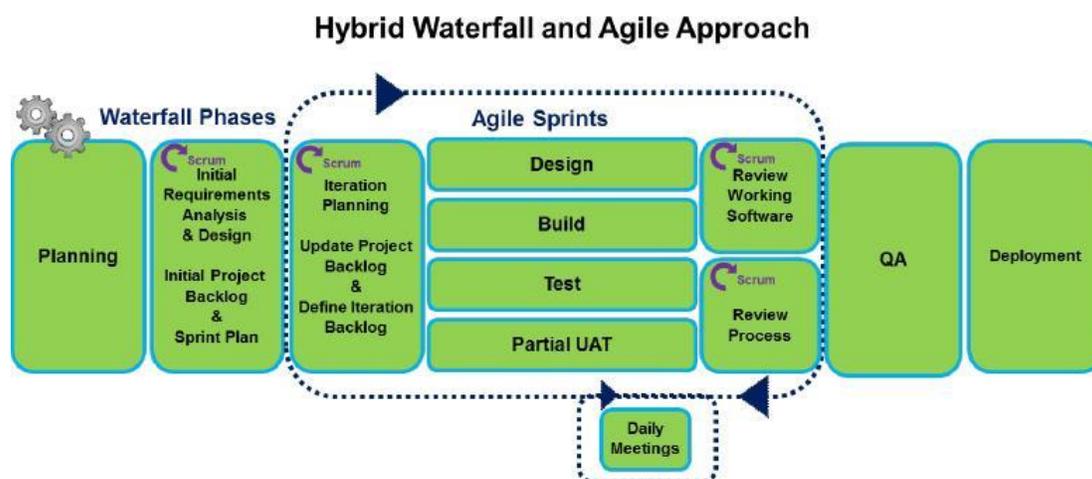


Figura 9.3-1 Modello Ibrido Waterfall-Agile

Si utilizzano sprint agili al posto delle tradizionali fasi di sviluppo e test tipiche del modello a cascata. Ogni sprint affronta il design della funzionalità da implementare, lo sviluppo e il test, inclusi gli UAT<sup>1</sup> (**User Acceptance Testing**). L'utilizzo di un approccio ibrido consente al progetto di compartimentalizzare le diverse aree della soluzione, migliorando al contempo la collaborazione tra il team di consegna e il committente, specialmente durante gli sprint di sviluppo in cui il feedback precoce e frequente del cliente è fondamentale per confermare la soluzione in tutto il progetto, in attesa della fase finale di QA.

#### Applicabilità in ComESto

<sup>1</sup> L'acronimo UAT (User Acceptance Testing) descrive la parte finale dello sviluppo di un software. In questa fase si sottopone il prodotto a un campione di utenti finali, con lo scopo di simulare un utilizzo reale e valutare il suo funzionamento in uno scenario più realistico possibile.

L'adozione di tale modello ibrido è fortemente calzante per le esigenze del progetto ComESTo poiché:

- Segue la fase di pianificazione, budget e analisi dei requisiti iniziale del modello waterfall con la produzione di documentazione
- Il ciclo di sviluppo procede in maniera rapida ed è scandito da cicli di lavorazione di durata contenuta e caratterizzati dal rilascio incrementale di funzionalità software oggetto di valutazione
- consente di essere flessibili nel rivedere, in corso d'opera, i requisiti di progetto soprattutto alla luce delle valutazioni al software espresse nel punto precedente

## 10. ARCHITETTURA HARDWARE DELLA PIATTAFORMA COMESTO

Nell'ambito del dimensionamento delle risorse CLOUD a supporto della piattaforma ComESTo, a causa dell'alto numero di variabili che entrano in gioco nella previsione dei dati di traffico, è emersa forte la necessità di avere disponibilità di risorse "scalabili" e "modulabili" a seconda delle mutevoli situazioni. A tale proposito Telecom Italia S.p.A. ha strutturato e reso disponibile il servizio cloud denominato Self Data Center vCloud. Il servizio TIM *Self Data Center* rientra nella categoria delle piattaforme IaaS per l'erogazione di capacità elaborativa come servizio e consente di:

- fornire un accesso rapido e sicuro a risorse computazionali con la massima flessibilità di configurazione delle stesse;
- rendere disponibili risorse effettivamente utilizzate con la possibilità di aumentare o diminuire le risorse necessarie;
- configurare con estrema facilità i server virtuali grazie ad una console web-based gestita in autonomia dal Cliente;
- ridurre il Time to Market attraverso il lancio di un servizio con investimenti ridotti e rapido self-provisioning;

Self Data Center è predisposto per gestire in modo facile e completo ogni percorso di IT e Cloud Trasformazione, mettendo a disposizione gli strumenti e le risorse informatiche necessarie per adeguare nel tempo e in autonomia, alle eventuali necessità, il Data Center Virtuale. La soluzione è caratterizzata infatti da:

- **SELF-SERVICE** Possibilità di istanziare in autonomia il proprio Virtual Data Center con tutti i server e le configurazioni di rete necessarie;
- **ACCESSIBILITÀ** Risorse disponibili e configurabili in qualsiasi momento e da qualsiasi device tramite la console Web;
- **SCALABILITÀ** Possibilità di aumentare o diminuire autonomamente le risorse in base alle proprie esigenze;

La soluzione Self Data Center, data l'alta affidabilità, può essere utilizzata in ogni fase del ciclo di vita di un'applicazione: sviluppo, test, pre-produzione, produzione; in particolare con la soluzione adottata per l'implementazione della Piattaforma ComESto consente di attivare e disattivare pool di risorse e, in caso di superamento di specifiche soglie, usufruire di ulteriori risorse condivise per gestire picchi di attività. Di seguito sono illustrate le caratteristiche principali del servizio e della Piattaforma ComESto.

### 10.1. CARATTERISTICHE PRINCIPALI DEL SERVIZIO

Il servizio Self Data Center vCloud Allocation Pool prevede che in autonomia si determinino le risorse necessarie per la Piattaforma ComESto. I blocchi di risorse configurabili, a seconda delle esigenze, sono riassunti in Figura 10.1-1.



Figura 10.1-1 blocchi di risorse configurabili per l'utente nel servizio Self Data Center

A supporto dell'infrastruttura vi sono poi le componenti di servizio Self Data Center vCloud:

- **Network**

L'utilizzo di dell'hypervisor VMware garantisce completa ridondanza fisica e possibilità di 'fail-over', eliminando di fatto la possibilità di 'down-time' per fault hardware. Il bilanciamento automatico dei carichi di lavoro (VMware DRS) garantisce, inoltre, le performance necessarie anche durante gli eventuali picchi di utilizzo delle applicazioni. Per la connettività dall'esterno è possibile prevedere accessi internet e/o MPLS.

- **Assistenza e SLA di disponibilità**

La disponibilità dell'infrastruttura virtuale è misurabile sulla base dei seguenti parametri di funzionalità operativa:

A) Connettività di Data Center: Up-time del 99,99% su base mensile, di accessibilità tramite rete Internet o MPLS alla Infrastruttura Virtuale.

B) Nodi fisici: Up-time del 99,99% su base mensile, per la disponibilità dei nodi fisici (server) che ospitano l'Infrastruttura Virtuale.

- **Snapshot, Image Level Backup e Antivirus**

Per le macchine virtuali costituenti l'architettura, è messa a disposizione la funzionalità per il salvataggio ed il ripristino dei dati, che utilizza una infrastruttura tecnologica centralizzata basata sulla suite software Veeam Backup&Replication v9. La funzionalità di Image Level Backup consente di effettuare una copia di tutti i Server del Data Center Virtuale, utilizzabile poi successivamente per ripristinare sia il singolo file che l'intera immagine del Server. La soluzione scelta per il servizio di antivirus è basata sulla piattaforma Deep Security di Trendmicro. Il prodotto è caratterizzato da un'alta profilabilità degli Utenti, che pur avendo accesso alla console di gestione, possono essere autorizzati (vincolati) ad operare solo su un sottoinsieme dell'intera infrastruttura ospite, che nel nostro caso coincide con la organization e con le macchine virtuali della Piattaforma ComESTo.

- **Cataloghi**

TIM Self Data Center vCloud fornisce un catalogo standard di vApp Template utilizzabili (Catalogo Servizi Pubblico), è inoltre possibile creare in autonomia cataloghi all'interno delle proprie Organization così come eseguire upload di proprie immagini .iso ed installare macchine virtuali completamente nuove con sistemi operativi propri. I cataloghi rappresentano un elemento per la standardizzazione delle distribuzioni.

- **La console di gestione del servizio**

È a disposizione la *Web User Interface*, l'interfaccia grafica attraverso la quale il gestire in autonomia il servizio.

- **Servizi Professionali**

È previsto, nel caso dovesse essere necessario, la possibilità di richiedere dalla Console di Gestione Servizi, sezione Servizi Professionali una serie di servizi utili a installare middleware, richiedere pacchetti di formazione e consulenza ed effettuare operazioni di partizionamento dischi sui Server Virtuali. Tutti i servizi sono erogati previo accordo telefonico.

- **Formazione**

Possibilità di usufruire dalla propria sede in Web Conference di servizi professionali di formazione da parte della Control Room di TIM.

- **Portale vRealizeOperations**

VMware vRealize Operations automatizza la gestione delle operations IT, consente di gestire in modo proattivo le prestazioni e aumenta la visibilità nell'infrastruttura fisica e virtuale. Mette in correlazione i dati delle applicazioni con lo storage in uno strumento di gestione unificato, facile da utilizzare, che fornisce il controllo di prestazioni, capacità e configurazione, con funzionalità di analisi predittive che consentono azioni proattive e automazione basata su policy.

## 10.2. CARATTERISTICHE DELL'INFRASTRUTTURA/CONFORMITÀ ALLE NORMATIVE VIGENTI

L'infrastruttura, vista come un unico pool logico indipendentemente dalla tipologia di architettura implementata, potrà essere gestita mediante un portale di servizio. Tale componente invia inoltre ai sistemi Billing di TIM i dati relativi al consumo delle risorse ai fini della loro fatturazione.

La piattaforma TIM Self Data Center garantisce in oltre la conformità alle principali normative vigenti come esplicitato in Figura 10.2-1.

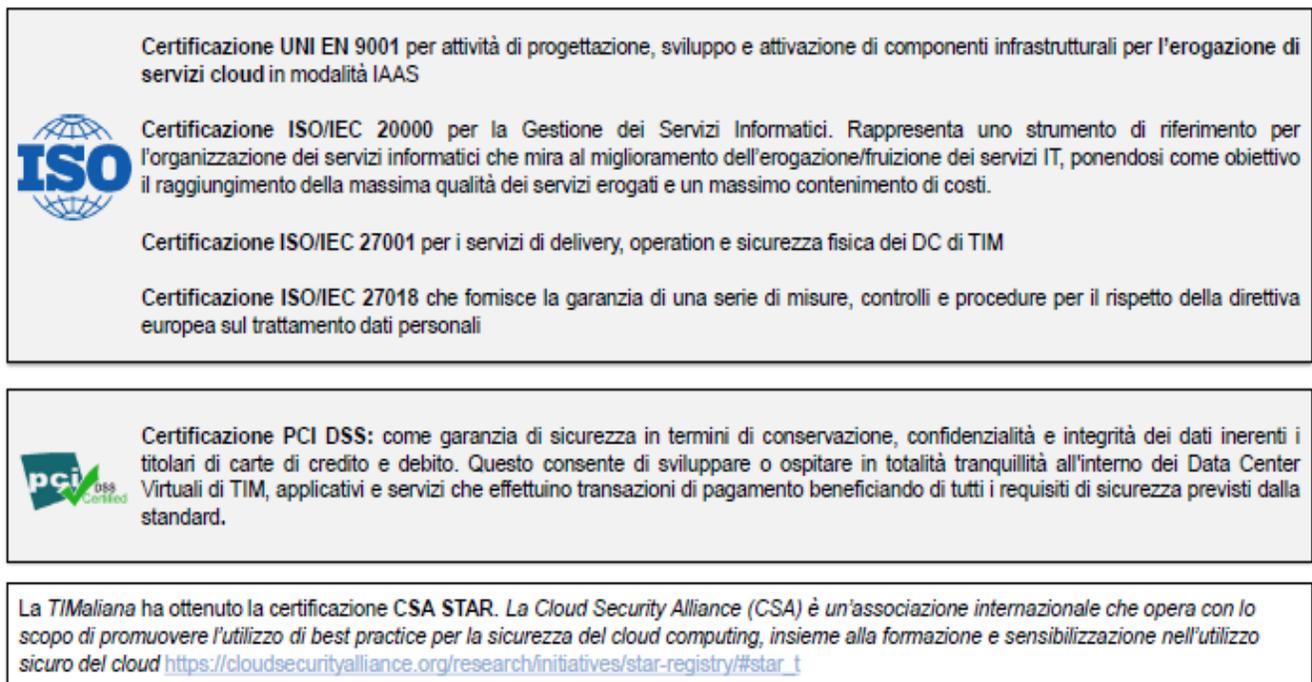


Figura 10.2-1 Sicurezza, Privacy e Conformità alle normative.

### Autonomia di Gestione e Responsabilità

Self Data Center prevede l'utilizzo di infrastrutture tecnologiche centralizzate tramite la quali TIM eroga ed è responsabile dei servizi operativi "core" (es. creazione e gestione dell'utenza amministrativa, aggiornamento impronte virali, monitoraggio delle componenti tecnologiche HD/SW, patch e fixing tecnologici, ecc.), mentre alla Piattaforma viene delegata la componente di gestione dei due servizi, ovvero la possibilità di decidere se, come, quando e per quali macchine virtuali utilizzare le funzionalità di Backup e Antivirus rese disponibili nell'ambito del servizio. La funzionalità viene fornita attraverso l'accesso a strumenti (console) per tale gestione; su questi strumenti sono definite idonee permission che, pur consentendo autonomia gestionale, limiteranno l'operatività solo agli host e alla

organizzazione di pertinenza. Il servizio fornisce funzionalità di Image Level Backup con possibilità di restore autonomo sia dell'intero Server che del singolo file.

## 11. DEFINIZIONE DELLE FUNZIONI DA IMPLEMENTARE PER I SERVIZI AL DSO

Lo scenario del sistema elettrico che si delinea nel prossimo futuro prevede uno sfruttamento delle fonti di energia rinnovabile in continua crescita. Allo stesso tempo la domanda di energia elettrica continua a crescere, data il sempre maggiore sfruttamento del vettore elettrico anche in settori tradizionalmente serviti da fonti energetiche di origine fossile (per es. sistemi di riscaldamento elettrici o la mobilità elettrica). Al crescere dei fenomeni sopra citati, l'adeguatezza del sistema elettrico rischia di essere considerevolmente ridotta a causa di una sempre maggiore difficoltà nel mantenere il bilanciamento fra generazione e carico e delle crescenti sollecitazioni sulle reti.

Allo scopo di raggiungere gli obiettivi di decarbonizzazione posti a livello europeo, è necessario che l'evoluzione nell'utilizzo e nella produzione di energia elettrica coinvolga in maniera consistente il maggior numero possibile di utenti, quindi anche gli impianti domestici di più piccola taglia. Questo delinea una nuova tipologia di utenza connessa sulle reti, non più un'utenza prevalentemente consumatrice di energia ma anche produttrice. Un'utenza 'attiva' quindi, e non solo perché capace di immettere energia in rete, ma anche perché capace di sfruttare la possibilità di modulare il profilo di prelievo/immissione dei propri dispositivi secondo logiche di ottimizzazione interna (fabbisogno energetico/ricavi) ed esterna (nei confronti della rete).

In questo contesto la nanogrid, la nano-rete tipicamente domestica protagonista del progetto ComESTo, costituisce una potente risorsa tecnologica. Essa permette infatti di coordinare ed integrare in un'unica piattaforma tutte le risorse energetiche di una generica utenza favorendone l'esercizio ottimale. Tramite questo strumento le risorse energetiche diventano delle vere e proprie **risorse di flessibilità**, sia per l'utente che per il sistema elettrico in generale. Inoltre, la flessibilità viene notevolmente aumentata nel caso in cui si integrino nelle nanogrid dei sistemi di accumulo, capaci di assorbire l'intermittenza di produzione degli impianti FER e di aumentare le opportunità di modulazione dei profili di scambio energetico interni ed esterni alla nano-rete. Le risorse energetiche appena citate, dato il carattere fortemente decentralizzato, vengono collettivamente denominate risorse energetiche distribuite (*Distributed Energy Resources – DER*).

Il sistema di distribuzione risulta essere quello maggiormente interessato dall'evoluzione dell'utenza, dato che le connessioni delle DER si svilupperanno prevalentemente in media e bassa tensione. Le esigenze di regolazione rendono quindi indispensabile lo sfruttamento della flessibilità offerta dalla nuova tipologia di utenza per la gestione e l'ottimizzazione del sistema di distribuzione. Lo sfruttamento della flessibilità può essere conseguito in diversi modi, tipicamente attraverso l'adozione di programmi di Demand Response. Nell'ambito dei modelli proposti nel progetto ComESTo lo sfruttamento da parte

dei System Operator (SO) della flessibilità offerta dagli utenti, gestiti in forma aggregata per tramite di un Aggregatore, avviene attraverso la partecipazione degli utenti ad un mercato dei servizi di regolazione. Il modello di quest'ultimo è stato elaborato sulla base di un attento studio dei modelli proposti in letteratura per quanto riguarda la partecipazione al Mercato dei Servizi di Dispacciamento (MSD) da parte delle risorse di flessibilità connesse sulla rete di distribuzione, in particolare per quanto riguarda l'architettura e le modalità di coordinamento tra gli attori principali coinvolti: il DSO, l'Aggregatore e il TSO. Lo scopo del presente paragrafo è quello di descrivere i servizi di regolazione previsti per il DSO nel modello di mercato ComESTo, sia dal punto di vista delle caratteristiche tecniche dei prodotti che dal punto di vista delle modalità di approvvigionamento e coordinamento tra i vari attori.

### 11.1. IL MODELLO DEL MERCATO DEI SERVIZI GLOBALI E LOCALI (MSGL) COMESTO

Per meglio contestualizzare la descrizione dei servizi di regolazione che si prevede vengano offerti al distributore da parte delle DER è opportuno richiamare il modello di MSD proposto nel progetto, studiato in seno alle attività di AR 5.4. Come già accennato, la proposta elaborata trae ispirazione dagli studi presenti in letteratura circa i migliori modelli di coordinamento DSO-TSO da adottare per abilitare la partecipazione a MSD delle risorse connesse sulle reti di distribuzione. Difatti aprire il mercato (o i mercati) dei servizi ancillari anche alle risorse allacciate alle reti di distribuzione rischia di generare conflitti di competenza o sovrapposizioni di accesso alle risorse tra TSO e DSO.

Preliminarmente è opportuno riportare la classificazione dei servizi ancillari in *servizi globali*, quindi necessari per l'esercizio in sicurezza del sistema elettrico nazionale, e *servizi locali* necessari per l'esercizio in sicurezza delle reti di distribuzione. Per l'implementazione efficiente del relativo mercato l'Autorità propone, col DCO 322/19, di rivedere il ruolo delle imprese distributrici prevedendo che il DSO svolga:

- il ruolo di **facilitatore** ai fini dell'approvvigionamento dei servizi globali da parte del TSO;
- il ruolo di **acquirente** di servizi locali per le regolazioni da effettuare sulle proprie reti.

L'evoluzione del mercato dei servizi ancillari implica quindi contestualmente un'evoluzione dei ruoli e delle responsabilità degli operatori di rete, quindi anche del modello di coordinamento da adottare tra questi. Sempre nel DCO 322/19 l'Autorità prende a riferimento i modelli di coordinamento sviluppati dal progetto SmartNet. Tra questi, secondo i risultati di SmartNet, gli schemi di coordinamento preferibili sarebbero:

- lo *schema centralizzato* nel caso in cui non si riscontrino particolari condizioni di congestione sulla rete di distribuzione;
- il *modello di mercato comune TSO-DSO* nel caso in cui, al contrario, si prevedano congestioni di

entità non trascurabile.

Prendendo a riferimento lo scenario ComESTo, dove è prevista una massiccia diffusione di DER e di nanogrid abbinata a sistemi di accumulo nelle reti di distribuzione, è facile immaginare un alto rischio di congestioni sulle reti, il che ha portato alla scelta di prendere spunto dal modello di mercato comune SmartNet, in particolare nella sua variabile decentralizzata. Secondo questo modello TSO e DSO, per l'approvvigionamento di risorse di flessibilità connesse sulle reti di distribuzione, gestiscono assieme un mercato comune che si articola in due fasi; nella prima (mercato locale) il DSO si approvvigiona di risorse per esigenze locali. I risultati di questa fase – che non costituiscono ancora un vincolo per i partecipanti – sono integrati nella seconda fase (mercato globale) dove l'esito complessivo della sessione di mercato viene ottimizzato considerando le esigenze di regolazione globali del sistema. In ultimo vengono quindi comunicate le offerte accettate per il DSO e per il TSO. La flessibilità viene in questo modo assegnata all'operatore di sistema che ha priorità maggiore, senza nessuna priorità, nell'ottica di minimizzare i costi complessivi di sistema e quindi di massimizzare il benessere sociale.

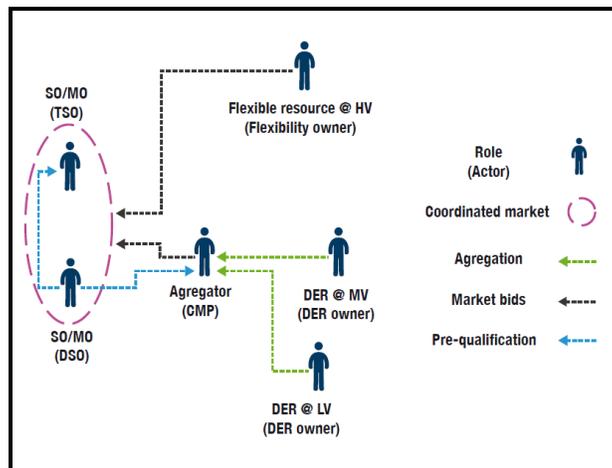


Figura 11.1-1 - Mercato comune TSO-DSO

Le caratteristiche peculiari del modello di Mercato dei Servizi Globali e Locali ComESTo (MSGL), rispetto al modello di mercato comune Smartnet, possono essere sintetizzate nei seguenti punti:

- i servizi di regolazione possibili per il DSO saranno di tipo *non-frequency*, in linea con i contenuti del DCO 322/19 ARERA;
- non esistendo la necessità di approvvigionarsi ex ante di servizi per la regolazione della frequenza, i prodotti da approvvigionare sul MSD locale saranno in una prima fase dei contratti di tipo *capacity* di lungo termine (di orizzonte almeno annuale). Non esiste una vera e propria sessione di mercato per queste contrattazioni, le quali possono avere luogo in un momento qualsiasi dell'anno. Attraverso questi contratti il DSO si riserverà una capacità di regolazione che potrà attivare all'occorrenza, secondo le modalità prestabilite, durante il periodo di validità del contratto;

- nella prima sessione di mercato (MSD locale) il gestore selezionerà le offerte di flessibilità per soddisfare le richieste di regolazione del DSO secondo criteri di merito economico. A parità di prezzo delle offerte presentate dalle risorse di flessibilità, sarà privilegiata l'attivazione delle risorse precedentemente approvvigionate in capacity;
- vengono considerate di norma prioritarie le movimentazioni di risorse in risposta alle esigenze di regolazione del DSO, a meno di esigenze di sicurezza più critiche sulla RTN. Dato lo scenario di utenza ComESTo sopra citato, lo sfruttamento della flessibilità rappresenta persino un importante prerequisito per abilitare la connessione di DER sulla rete di distribuzione. L'esigenza diventa ancora più stringente per le risorse approvvigionate in capacity, che costituiscono il margine di regolazione sul quale il DSO potrebbe fare affidamento nel decidere di rinviare/annullare un intervento di adeguamento strutturale;
- i risultati della prima sessione di MSD possono essere modificati dall'ottimizzazione della seconda sessione (MSD globale) se:
  - sono rilevate condizioni di emergenza nella RTN e vi è una significativa aggregazione che può contribuire efficacemente a risolverla senza mettere in pericolo l'operatività della rete di distribuzione;
  - la redistribuzione della flessibilità non incide sulla programmazione in esito alla prima ottimizzazione per più di una predeterminata soglia percentuale in termini di capacità regolante o valore economico;
- il DSO avrà il ruolo di validatore ex-ante rispetto alle movimentazioni richieste dal TSO.

Tali scelte trovano giustificazione nel particolare scenario di riferimento del progetto ComESTo, dove la massiccia e capillare presenza di risorse di flessibilità sulla rete di distribuzione ne costituisce di fatto una minaccia più diretta per il rispetto dei vincoli di esercizio rispetto agli impatti previsti sulla RTN.

Di seguito sono riepilogati i ruoli per i principali operatori coinvolti nel modello MSGL:

- il **TSO** è il gestore del MSD globale e responsabile del bilanciamento del sistema elettrico in tempo reale. Esso può approvvigionarsi di risorse di flessibilità connesse sulla rete di distribuzione previa validazione della movimentazione da parte del DSO. In caso di emergenza sulla RTN, il TSO può interagire direttamente con le suddette risorse;
- il **DSO** si approvvigiona sul MSD locale di servizi *non-frequency*. Esso è anche facilitatore di mercato e validatore delle movimentazioni rispetto ai servizi di regolazione richiesti dal TSO da risorse connesse sulla rete di distribuzione;
- l'**Aggregatore** offre servizi di rete in forma aggregata sia su MSD locale che globale. Agisce coordinandosi in prossimità del tempo reale con il DSO per l'attivazione di servizi offerti da risorse connesse sulla rete di distribuzione;
- il **Gestore del MSGL** gestisce la piattaforma di mercato, dove raccoglie e abbina le offerte di vendita e di acquisto di servizi di regolazione.

I servizi previsti nel MSGL si possono raggruppare in due macro-tipologie:

- 1) Servizi Locali a lungo termine: sono i servizi di **capacity** già citati, formulati secondo contratti di durata pluriennale stipulati tra il DSO e gli aggregatori. Si possono definire come i **servizi della**

**pianificazione della rete di distribuzione**, in quanto sono approvvigionati dal DSO per far fronte alle possibili esigenze di regolazione previste per la rete in alternativa alla realizzazione delle soluzioni infrastrutturali tipiche della pianificazione. Tale sessione del mercato può avvenire in una sezione apposita del MSGL.

- 2) Servizi Globali e Locali in 'tempo reale': sono i servizi per la gestione real-time della rete, che gli aggregatori possono offrire nel mercato infragiornaliero sia al DSO (servizi non-frequency) che al TSO. Le compravendite di questo tipo avverranno nella piattaforma MSGL tramite le seguenti modalità:
- Servizi locali al DSO → in base a criteri di convenienza tecnico-economica i servizi saranno selezionati tra le offerte di **attivazione dei servizi precedentemente approvvigionati in capacity** (punto 1) e le offerte da aggregatori e utenti abilitati (di taglia sufficiente) senza contratto di capacity. A parità di condizioni i primi saranno privilegiati;
  - Servizi globali al TSO → in questo caso preliminarmente **le offerte vengono pubblicate sul MSGL per la validazione del DSO**. Qualora questa abbia esito positivo, sulla piattaforma MSGL verranno pubblicati entrambi gli esiti della validazione del DSO e, eventualmente, dell'accettazione dell'offerta da parte del TSO.

## 11.2. IL MERCATO DEI SERVIZI LOCALI DELLA DISTRIBUZIONE

Secondo il CEER (*Council of European Energy Regulators*), per l'approvvigionamento di risorse di regolazione il ricorso a meccanismi di mercato, ove sia presente sufficiente liquidità, rappresenta l'approccio migliore in quanto consente al DSO di acquisire risorse di flessibilità su base competitiva [xxi]. Teoricamente la soluzione migliore sarebbe quella di un mercato "real-time" dove il DSO possa acquistare flessibilità quando serve e nella quantità necessaria. Tuttavia, per ragioni di liquidità, è preferibile adottare un approccio basato su una contrattazione di lungo termine, anche rischiando di sovrabbondare nelle quantità approvvigionate. Inoltre, data la natura radiale della rete di distribuzione, i contratti a lungo termine limitano le possibili speculazioni legate a eventuali problematiche di natura tecnica della rete di distribuzione. La definizione delle caratteristiche tecniche del modello di Mercato dei Servizi Locali riflette le effettive esigenze del sistema di distribuzione proiettato nello scenario ComESTo, in relazione alle caratteristiche fisiche della rete.

Di seguito si elencano le principali peculiarità:

- i **Punti di dispacciamento (PDD)** sono identificati esclusivamente **nei nodi MT** (Cabine Secondarie). Rispetto a tali PDD il DSO si approvvigionerà di servizi ancillari presso gli **Aggregatori** sottesi ai PDD stessi in base alle specifiche esigenze:
  - o Controllo profili di tensione sulla sbarra MT delle cabine secondarie;
  - o Controllo profili di tensione sulla sbarra BT delle cabine secondarie;

- Controllo sfruttamento in corrente delle linee MT e BT.

Le operazioni di controllo saranno realizzate dal DSO per mezzo di sensoristica in Cabina Secondaria.

- per quanto riguarda l'entità e il perimetro dell'aggregazione, le unità minime che potranno offrire flessibilità in forma aggregata saranno le 'nanogrid', ossia nano-sistemi elettrici domestici di potenza non superiore ai 5 kW. Non si esclude comunque la partecipazione all'aggregato di unità di produzione/consumo/storage di taglia superiore. Ad ogni modo, a prescindere dal posizionamento nella rete delle unità gestite da un *Aggregatore*, quest'ultimo potrà presentare offerte sul mercato locale in riferimento ai singoli PDD, e quindi considerando le sole unità sottese al PDD di riferimento. Tale scelta si è rivelata conforme all'attuale Delibera 318/2020/R/eel ARERA, secondo la quale i membri di una comunità di energia rinnovabile, ai fini dell'accesso alla valorizzazione e incentivazione dell'energia elettrica condivisa, devono essere titolari di punti di connessione su reti BT sottese alla medesima cabina secondaria.
- i **Servizi Ancillari** previsti sono 5, descritti in Tabella 11.2-1. Il quinto servizio prevede, in caso di emergenza, la possibilità per il DSO di comandare la modifica del programma di immissione/prelievo di una singola risorsa, senza passare dall'Aggregatore.

Tabella 11.2-1 – I servizi ancillari locali

SERVIZIO		TIPOLOGIA CONTRATTO	DESCRIZIONE
<b>CONGESTION MANAGEMENT</b>	<b>1. Congestion Management Capacity.</b>	(Pluri)Annuale	<i>In fase di programmazione il DSO si approvvigiona di capacità regolante al fine di poter gestire potenziali congestioni, garantendo il rispetto dei vincoli di massima corrente sui rami e massima variazione di tensione ai nodi. Il servizio è acquistato a un prezzo di 'capacity'.</i>
	<b>2. Congestion Management Real Time</b> (attivazione del relativo servizio contrattato in Capacity).	Spot	<i>È acquistato al prezzo offerto (pay as bid) per tutti gli intervalli di tempo di esecuzione, per ridurre lo sfruttamento dei rami e/o la variazione di tensione ai nodi. A parità di condizioni tecniche ed economiche sono selezionate prioritariamente le risorse precedentemente riservate in capacity.</i>
<b>VOLTAGE CONTROL</b>	<b>3. Voltage Control - Capacity Reactive Power.</b>	(Pluri)Annuale	<i>In fase di programmazione il DSO si approvvigiona di capacità regolante di immissione/assorbimento di potenza reattiva per mantenere il set-point della tensione di rete in condizioni di esercizio normali. Il servizio è acquistato a un prezzo di 'capacity'.</i>
	<b>4. Voltage Control - Real Time</b> (attivazione del relativo servizio contrattato in Capacity).	Spot	<i>È acquistato al prezzo offerto (pay as bid) per tutti gli intervalli di tempo di esecuzione per mantenere il set-point della tensione di rete in condizioni di esercizio normali. A parità di condizioni tecniche ed economiche sono selezionate prioritariamente le risorse precedentemente riservate in capacity.</i>

	<b>5. Voltage Control Active Power - in emergenza.</b>	<i>Spot</i>	<i>Immissione/assorbimento di potenza attiva per mantenere il set point della tensione di rete in condizioni di esercizio di emergenza.</i>
--	--	-------------	---

Per la descrizione completa dei prodotti e dei servizi di flessibilità per il mercato dei servizi di regolazione locale ComESto si rimanda all'allegato A.

### 11.3. ARCHITETTURA DELLE COMUNICAZIONI E DELLE INTERAZIONI TSO-DSO-UTENTI

Il coinvolgimento attivo delle risorse connesse alla rete di distribuzione nello svolgimento del Mercato dei Servizi Globali e Locali (MSGSL) prevede un insieme di interazioni tra gli attori del sistema che si svolgono su diverse scale temporali e attraverso specifici canali di comunicazione. In questo paragrafo si analizzano le interazioni previste tra la piattaforma MSGSL e le piattaforme di TSO, DSO e Aggregatori, descrivendo la tipologia di dati scambiati e citando i protocolli di comunicazione adottati.

#### 11.3.1. IL MODELLO "TRAFFIC LIGHT"

Le tipologie di interazione previste nel MSGSL dipendono dallo stato del sistema rilevato in uno specifico segmento di rete e in un determinato intervallo di tempo. Per tali motivi è stato scelto un approccio di tipo 'traffic light' dove lo stato della rete è rappresentato dal colore del semaforo in base al livello di criticità rilevato. La stima sullo stato della rete è affidata al gestore della rete di distribuzione, che ha il compito di mantenerlo aggiornato e a disposizione di tutti i partecipanti al mercato sulla piattaforma MSGSL. La Tabella 11.3.1-1 riassume le condizioni che determinano lo stato della rete nei tre diversi casi.

Tabella 11.3.1-1 – Il semaforo come indicatore dello stato della rete

	Non esistono condizioni di rete critiche. La flessibilità è sfruttata solamente per il mercato energetico interno delle risorse aggregate. Il DSO monitora lo stato della rete e non interviene.
	Il DSO rileva congestioni effettive o potenziali sulla propria rete.
	Uno tra il TSO e il DSO (o entrambi) rileva condizioni di criticità pericolose per il sistema.

### 11.3.2. L'ARCHITETTURA DELLE INTERAZIONI TRA LE PIATTAFORME

Una panoramica dell'architettura logica delle comunicazioni e delle interazioni è riportata in Figura 11.3.2-1.

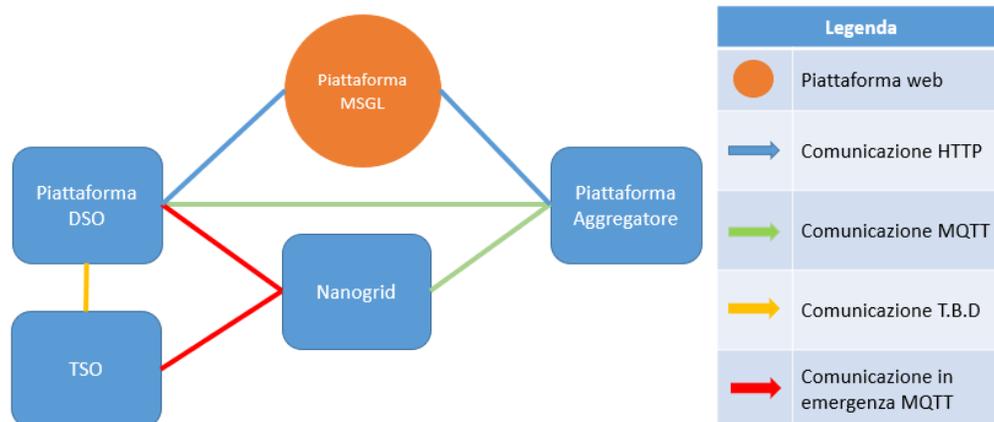


Figura 11.3.2-1 - Architettura logica delle comunicazioni e interazioni tra le piattaforme

La piattaforma di mercato MSGL si configura come un servizio web. In questa verranno pubblicate le offerte di vendita e di acquisto di servizi ancillari, oltre all'aggiornamento degli esiti delle contrattazioni a carico del gestore del mercato. Sempre su tale piattaforma il **DSO si occuperà di mantenere aggiornato in tempo reale lo stato del semaforo**, per tutte le porzioni di rete dove è attivo il MSGL. Gli aggregatori invece manterranno aggiornati i dati sui **margini di regolazione a scendere/salire residui al punto di connessione** delle risorse gestite, al netto di quanto già impegnato per il mercato energetico e per il MSGL stesso. La Piattaforma ComESto riceverà i dati di misura dagli Smart Meter presenti nelle nanogrid e scambierà con queste i dati per l'ottimizzazione dei loro programmi di carico/produzione/accumulo. Le piattaforme dei System Operator comunicheranno direttamente con le nanogrid in caso di emergenza per richieste urgenti di modifica dei programmi di immissione/prelievo. TSO e DSO comunicheranno anche per il processo di proposta e validazione di risorse di regolazione richieste dal TSO da attivare sulla rete di distribuzione.

### 11.3.3. LE INTERAZIONI IN CASO DI SEMAFORO VERDE

In caso di semaforo verde sono previste le interazioni di seguito riportate:

- **La Piattaforma Aggregatore si interfaccia col DSO**, utilizzando un protocollo di comunicazione MQTT, per comunicare i seguenti dati:
  - Profili di immissione/prelievo di tutti i punti di connessione delle risorse gestite (nanogrid) alla rete di distribuzione in esito al mercato energetico (**curva P/Q vs tempo**, al quarto d'ora);
  - Stato degli impianti gestiti (in esercizio, dismessi, in manutenzione, **margini di**

regolazione residua a salire/scendere).

- **La Piattaforma Aggregatore comunica con la piattaforma MSGL**, utilizzando un protocollo http, per pubblicare le proposte di servizi che vorrebbe offrire sul mercato al giorno prima e nel mercato infragiornaliero.
- Il DSO si interfaccia con la piattaforma MSGL, ancora tramite protocollo http, per:
  - Pubblicare lo stato di validazione dei servizi richiesti dal TSO o offerti al TSO da risorse connesse alle reti di distribuzione;
  - Aggiornare lo stato semaforo.
- **Il TSO pubblicherà sulla piattaforma MSGL**, al giorno N-1 e al giorno N, le richieste di servizi da far validare al DSO.
- **Il gestore del MSGL**, tramite la relativa piattaforma, metterà a disposizione di tutti i dati su:
  - Stato del semaforo;
  - Esiti del mercato del giorno prima e infragiornaliero, e cioè tutte le offerte al DSO e al TSO che hanno trovato riscontro positivo sul mercato;
  - Portfolio delle offerte di acquisto e vendita di servizi di regolazione.

#### 11.3.4. LE INTERAZIONI IN CASO DI SEMAFORO GIALLO

Oltre a quelle già previste nelle condizioni di semaforo verde, in caso di semaforo giallo si aggiungono le seguenti interazioni:

- **Il DSO pubblicherà sulla piattaforma MSGL** (protocollo http) le **richieste di servizi** che verranno prese in carico dal gestore del mercato;
- **Il DSO comunicherà al TSO** la previsione del profilo di scambio di potenza con la RTN a livello delle CP.

#### 11.3.5. LE INTERAZIONI IN CASO DI SEMAFORO ROSSO

In caso di semaforo rosso, alle interazioni previste nei due stati precedenti si aggiungono le seguenti:

- **Il DSO raggiungerà direttamente le nanogrid** di cui ha necessità per risolvere la criticità, utilizzando un protocollo MQTT, ordinando la modifica del profilo di immissione/prelievo al punto di connessione.
- A loro volta le nanogrid restituiranno al DSO i seguenti dati:
  - Stato degli impianti;
  - Notifica modifica profilo di scambio potenza con la rete in seguito ad un'eventuale richiesta in emergenza del TSO.
- Il DSO si interfacerà col TSO per fornire i dati su:
  - Aggiornamento stato semaforo, specie in relazione alle eventuali movimentazioni operate dal TSO sulla rete di distribuzione;
  - Previsione scambio di potenza nelle CP (profilo al quarto d'ora) che alimentano gli impianti dell'utenza attivati per la gestione dell'emergenza sulla RTN.

- **Il TSO si interfacerà col DSO** per notificare l'eventuale attivazione di movimentazioni su impianti connessi alla rete di distribuzione, con dettaglio della modifica del profilo di immissione/prelievo (al quarto d'ora) previsto in tutti i punti di connessione dell'utenza interessata.

## 12. LA RE-INGEGNERIZZAZIONE DELL'HOME GATEWAY EUGENIO

Eugenio è il dispositivo home gateway basato su tecnologia Z-Wave che opera da interfaccia per una corretta e sicura comunicazione tra ciascuna nanogrid della comunità ComESto e la piattaforma della stessa Comunità. Eugenio consente anche di monitorare e controllare altri dispositivi domestici, sensori e attuatori presenti nell'abitazione dell'utente finale.



Figura 12-1 Eugenio prima del processo di re-ingegnerizzazione

La Figura 12-1 Eugenio prima del processo di re-ingegnerizzazione mostra il dispositivo Eugenio prima del processo di re-ingegnerizzazione. E' facile notare che tale versione del dispositivo si basava principalmente sull'impiego di un Raspberry Pi 3B e di una board di espansione. Tale scelta era risultata conveniente per diversi motivi e, tra questi, la minimizzazione dei tempi di sviluppo.

Con l'avvento del progetto ComESto, si sono manifestate nuove e rilevanti esigenze; ciò ha reso necessario la re-ingegnerizzazione del dispositivo Eugenio. Ad esempio, l'impiego di un Raspberry Pi 3B e di una board di espansione – sebbene velocizzi la fase di sviluppo - non è idoneo alla produzione di dispositivi in quantità elevate, principalmente perché non permette di abbattere i costi di produzione. Tale impiego, inoltre, conserva una forte dipendenza dalle scelte del produttore del Raspberry Pi 3B e richiede l'uso di memorie micro-SD esterne. Questo tipo di memorie possono corrompersi facilmente e, poiché rappresentano la memoria dello stesso Eugenio, pregiudicano la comunicazione tra la Piattaforma ComESto e le nanogrid nonché l'esecuzione dei comandi per la gestione della Comunità sia in fase di pianificazione che in tempo reale, in accordo alla risoluzione degli algoritmi dei modelli di ottimizzazione OFI-DA e OFIS-TR.

Di conseguenza, si è mostrata forte ed evidente la necessità di adeguare il dispositivo Eugenio alle nuove esigenze maturate in seno al progetto ComESto; pertanto, il dispositivo Eugenio è stato re-

ingegnerizzato per come illustrato nei successivi paragrafi. È importante precisare che uno dei requisiti principali del passaggio dall'attuale al nuovo Eugenio è che il case plastico non subisse variazioni.

## 12.1. PRIMA FASE PROTOTIPALE

In una prima fase delle attività di re-ingegnerizzazione è stata studiata la possibilità di realizzare un nuovo ed unico circuito stampato che integrasse un system on chip (SoC), la memoria RAM, la memoria di archiviazione flash e tutti i relativi circuiti. La successiva valutazione di tale scelta ha concluso che la realizzazione di un nuovo ed unico circuito stampato avrebbe richiesto costi elevati e lunghi tempi di validazione, non compatibili con quelli del progetto ComESTo. Per tale motivo, è stata studiata una possibile alternativa cioè l'impiego di un System On Module (SOM) all'interno di una carrier board che ospiti sia le periferiche di espansione che i connettori di interfaccia. La valutazione di questa alternativa ha dato esito positivo, pertanto, il nuovo Eugenio è munito di un nuovo circuito stampato si basa sull'utilizzo di un Compute Module 3, dotata di SoC Broadcom BCM2837 e architettura Raspberry. Inoltre, il nuovo Eugenio impiega memorie eMMC 8GB, più affidabili delle precedenti micro-SD, ed un chip detto uSonic capace di rilevare onde sonore in spettro subsonico. Come già illustrato in D4.1, l'analisi delle onde subsoniche rilevate dal chip uSonic restituisce una stima del numero delle persone presenti nell'ambiente chiuso e contribuisce al miglioramento delle soluzioni degli algoritmi di demand response.

### 12.1.1. LA SCELTA DEL SYSTEM ON MODULE

Per la scelta del System On Module sono state considerati diversi prodotti presenti sul mercato; tra questi: i moduli Compute Module 3 (CM3) con Chip Broadcom, gli analoghi CM3 lite, i moduli Variscite con Chip Frescale IM6 Quad, etc. Al termine di riflessioni, considerazioni e comparazioni, la scelta è stata il system on chip Compute Module 3. Tra i vantaggi di questa scelta, è doveroso citare: i costi di produzione sono molto bassi e le necessarie modifiche al software già implementato nella attuale versione di Eugenio sono minime. La Compute Module 3 ha le seguenti caratteristiche: Chip Broadcom BCM2837, Cortex-A53 64-bit SoC @ 1.2GHz, RAM 1GB LPDDR2 SDRAM, FLASH Memoria 4GB eMMC Flash (versione Lite senza eMMC Flash). La memoria flash è collegata direttamente al processore, mentre tutte le altre interfacce sono disponibili direttamente sui pin del connettore. Il modulo Wi-Fi è realizzato mediante chip Mediatek RT5370, ha interfaccia USB, è disponibile in due versioni con tensioni di alimentazione differenti: 3.3V e 5.0V, è pienamente supportato dal kernel Linux, supporta gli standard 802.11 b/g/n a 150Mbit/s, è in grado di funzionare come access point.

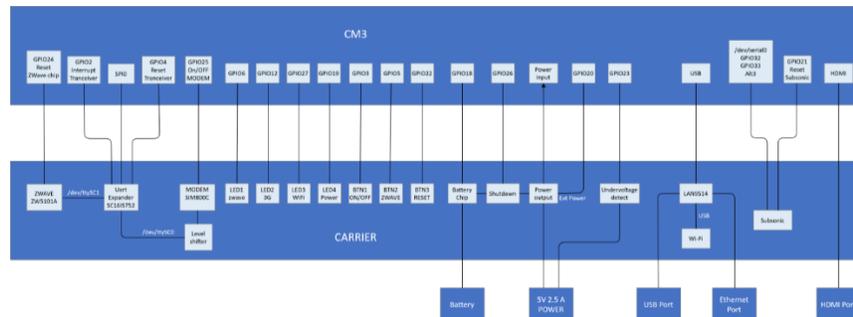


Figura 12.1.1-1 Architettura hardware del nuovo Eugenio

## 12.1.2. LA ARCHITETTURA HARDWARE

L'architettura hardware del nuovo Eugenio è illustrata in Figura 12.1.1-1; la parte superiore illustra la Compute Module 3 mentre la parte inferiore illustra la carrier board. Su quest'ultima board è disponibile un serial expander SC16IS762, con due porte seriali sulle quali sono collegati rispettivamente un modem SIM800C del tipo GSM dual band ed un trasmettitore Z-Wave ZM5101A. Quest'ultimo non è disponibile come modulo a sé stante pertanto il chip, l'antenna, la memoria flash da 2Mbit e tutti i componenti necessari sono stati integrati sulla carrier board. Sulla stessa carrier board è inoltre presente un modulo WiFi Mediatek, un circuito integrato per lo switch-over fra le fonti di alimentazione (alimentatore da rete 230V-50Hz o batterie), un hub USB, un controller ethernet, un chip subsonico e un connettore HDMI standard (utile solo in fase di sviluppo e per il debug). La Figura 12.1.1-1 mostra uno dei primi prototipi del nuovo Eugenio secondo la nuova architettura hardware.

### 12.1.1. TEST E VALIDAZIONE

Il primo prototipo del nuovo Eugenio di Figura 12.1.1-1 è stato sottoposto ad una campagna di test, soprattutto software, atti a verificare il corretto funzionamento delle periferiche e dei moduli installati. I test sono stati condotti per seguenti dispositivi e periferiche: porta HDMI per le fasi di debug e sviluppo, collegando il nuovo Eugenio ad un monitor; controller LAN9514 e controller ethernet, collegando il nuovo Eugenio a periferiche come mouse, tastiera, pen-drive; pulsanti accensione, Z-Wave e reset; modulo Wi-Fi anche in modalità Access Point, collegando il nuovo Eugenio a router domestico; modulo subsonico, generando rumore artificiale; modem 3G; modulo Z-Wave; abbassamento della tensione di alimentazione esterna, riducendo la tensione in uscita di un alimentato programmabile da bando; switch automatico tra l'alimentazione esterna e batteria interna; spegnimento del sistema (shutdown); battery status (carica/scarica).



Figura 12.1.1-1 Uno dei primi prototipi del nuovo Eugenio

## 12.2. SECONDA FASE PROTOTIPALE

La prima fase prototipale del nuovo Eugenio terminava con la realizzazione di più prototipi; ciascuno di essi implementava le scelte progettuali che si susseguivano. Il risultato finale era più che soddisfacente, tuttavia, si ravvisava l'opportunità di ottimizzare il processo produttivo e ridurre i costi di produzione. Una seconda fase prototipale aveva così inizio.

In questa seconda fase, tutti i componenti sono stati spostati sulla faccia superiore del circuito stampato ad eccezione di componenti opzionali che non verranno montati nella versione finale di produzione. Lo spostamento dei componenti sulla faccia superiore del circuito stampato ha comportato una modifica al layout e un riposizionamento di diversi componenti come le porte USB, il modulo Wi-Fi, il modem GSM, lo slot per la nano SIM. Inoltre, è stata aggiunta una antenna ceramica al modem GSM, il chip subsonico è stato sostituito con una versione più compatta, è stata aggiunta una matching network all'antenna Z-Wave, alcuni condensatori elettrolitici sono stati sostituiti con condensatori equivalenti ma meno costosi, è stata anche introdotta la possibilità di monitorare lo stato del modem GSM e del chip subsonico.



a)



b)

Figura 12.2-1 Output finale della a) prima e b) seconda fase prototipale



Figura 12.2-2 Il nuovo Eugenio

La Figura 12.2-1 mostra il circuito stampato del nuovo Eugenio per come progettato e realizzato al termine della prima e della seconda fase di prototipazione. La Figura 12.2-2 mostra il nuovo Eugenio quando montato all'interno del proprio case.

### 12.2.1. TEST E VALIDAZIONE

I test per la validazione del prototipo realizzato nella seconda fase prototipale sono identici a quelli eseguiti nella prima fase prototipale, fatta eccezione per i test circa la corretta integrazione delle funzioni di monitoraggio del modem GSM e del chip subsonico. Quest'ultimo, diversamente dal modem GSM, non emette un segnale fisso quando è in funzione bensì intermittente e ciò rende difficile intercettare il funzionamento del dispositivo stesso. Per ovviare a ciò, è stata sviluppata una utility ad-hoc, pertanto, anche il monitoraggio del funzionamento del chip subsonico del nuovo Eugenio è eseguito senza incertezze.

Inoltre, sono stati eseguiti stress test con lo scopo di verificare il corretto funzionamento del nuovo Eugenio anche per casi limite, ben oltre i casi d'uso quotidiano. A tal fine, è stata calcolata la somma delle potenze nominali riportate nei datasheet dei principali componenti e delle principali periferiche montate sul nuovo Eugenio (es. CM3, modulo Wi-Fi, chip uSonic, SIM card, LAN Ethernet, Z-WAVE, altro); tale somma è pari a circa 16000mW o 3200mA@5V. E' facile intuire che questo valore non corrisponda ad alcun punto di funzionamento reale anche perché né il convertitore analogico/digitale TV né l'unità di elaborazione grafica sono utilizzati nel nuovo Eugenio, la corrente assorbita dalle GPIO è pochi mA, la scheda di memoria non è in uso, il modem GSM correnti fino a 2A ma per tempi inferiori al millisecondo (i cosiddetti picchi o spike, soppressi per la maggiore dai condensatori presenti sul

circuito stampato). Detto ciò, gli stress test sono stati eseguiti per valori prossimi, ma inferiori, alla somma delle potenze nominali dei principali componenti e delle principali periferiche montate sul nuovo Eugenio.

Particolare attenzione è stata posta a fenomeno degli spike di corrente del modem GSM. La Figura 12.2.1-1 riporta la cattura schermo di un oscilloscopio usato per monitorare la tensione di alimentazione principale del circuito stampato durante uno degli stress test di laboratorio. La figura mostra ripetute cadute di tensione dovute all'attivazione del modem; le cadute di tensione sono pari a 860mV circa. Poiché la tensione nominale dell'alimentatore principale è circa 5.5V, la tensione di alimentazione del modem non è mai inferiore a 4.54V. E' questo un ottimo risultato infatti il datasheet del modem GSM indica una tensione inferiore limite pari a 3.0V pertanto il nuovo Eugenio è protetto contro lo spegnimento improprio del modem GSM.

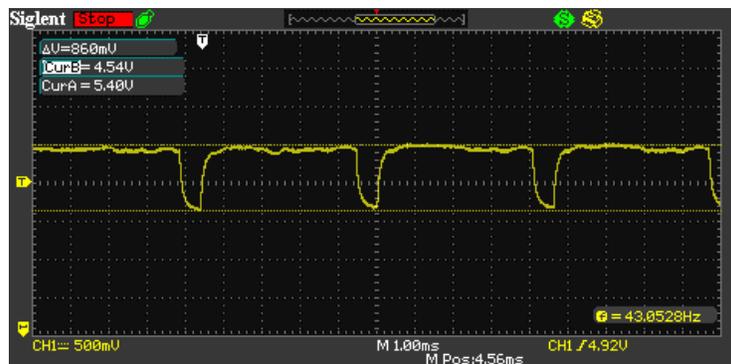


Figura 12.2.1-1 Caduta di tensione per spike di corrente assorbita dal modem GSM

### 12.3. VALIDAZIONE FINALE DEL NUOVO EUGENIO

Il prototipo finale del nuovo Eugenio per come restituito dalla seconda fase di prototipazione è stato assemblato in tutte le sue parti e sottoposto a test hardware; sono stati verificati i meccanismi del funzionamento del circuito di alimentazione, le tensioni elettriche, l'avvio corretto del sistema, le porte USB, il controller Ethernet, la porta HDMI, il modulo Wi-Fi, il modulo Z-Wave e del modem GSM, il modulo uSonic. Tutti i test hardware sono stati conclusi con successo.

### 12.4. SISTEMA DI BUILD AUTOMATICA

Terminata la prima e la seconda fase di prototipazione del nuovo Eugenio ed i relativi test, è stata avviata una terza fase di attività finalizzata alla generazione del software da installare sullo stesso prototipo. A tal fine, è stato realizzato un sistema di build automatica che fornisce sia gli archivi necessari per il processo di flashing del nuovo Eugenio sia i pacchetti utili per l'aggiornamento del software già installato. Tutti gli archivi e i pacchetti sono salvati su un server FTP. La generazione degli

archivi e dei pacchetti può essere personalizzata. Ad esempio, è possibile scegliere l'etichetta che identifica la build generata, è possibile scegliere i file da compilare ed installare tramite build, è possibile scegliere le informazioni relative alla destinazione del server FTP, è possibile scegliere un indirizzo email per ricevere la notifica del termine del processo di build. La Figura 12.4-1 mostra l'interfaccia web per la personalizzazione della generazione sia degli archivi che dei pacchetti.

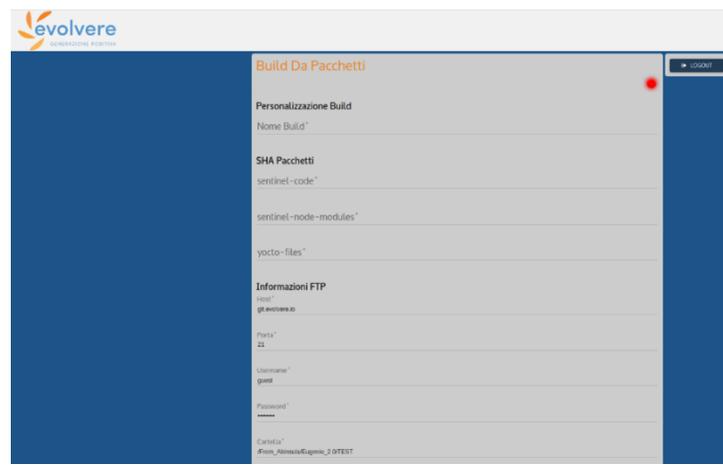


Figura 12.4-1 Interfaccia web per pacchetti

## 12.5. PRINCIPALI RISULTATI DELLA RE-INGEGNERIZZAZIONE

La re-ingegnerizzazione di Eugenio è stata una attività necessaria poiché la esistente versione non si dimostrava idonea ad attuare le specifiche richieste maturate nell'ambito delle attività di ricerca industriale e che avrebbero poi trovato applicazione nelle attività di sviluppo sperimentale (obiettivo realizzativo OR7). Per tale motivo, la re-ingegnerizzazione di Eugenio è un processo che inizia con gli obiettivi realizzativi OR1, OR4 e OR5 e prosegue in OR7. In particolare, in OR1 è stata richiesta la installazione di 100 smart meter per la popolazione di un cloud-database con dati e misure della domanda e della generazione di prosumer reali ad elevata risoluzione temporale. Il nuovo Eugenio funge da tramite tra gli smart meter ed il cloud-database ovvero la piattaforma ComESTo. Questo perché la migliore e diffusa conoscenza della domanda e della generazione presso prosumer reali aiutano l'intero partenariato nel migliorare le analisi del contesto locale ed il coinvolgimento dei cittadini e delle PMI (A1.1), nel migliorare le analisi delle potenzialità di mercato (A1.2), nel migliorare la definizione di modelli gestionali di un aggregato dal punto di vista di un soggetto provato (A1.6). La nuova versione di Eugenio possiede le adeguate risorse di calcolo per gestire il flusso dei dati ad alta risoluzione temporale (medie aritmetiche temporali ogni 5 secondi), generato dai suddetti smart meter. Inoltre, sebbene questo primo set di smart meter utilizzi la convenzionale rete domestica Wi-Fi, non è possibile escludere che i successivi smart meter impieghino la più affidabile tecnologia Z-Wave visto

che la precedente versione di Eugenio già montava un modulo per tale tecnologia. A tal proposito, è opportuno ricordare che il modulo Z-Wave montato sul precedente Eugenio è oggi fuori produzione; ne consegue che il mal funzionamento di tale modulo Z-Wave nega la possibilità di aggiornamenti del software e, nel caso estremo di rottura, nega la possibilità di riparare il dispositivo Eugenio, costringendo alla sostituzione dell'intero dispositivo. Per tale motivo, per il nuovo Eugenio, l'esistente modulo Z-Wave è stato sostituito con un nuovo e più affidabile modello ZM5101A; inoltre, l'antenna Z-Wave è ora stampata direttamente sul circuito stampato anziché esterna come nel caso precedente. Anche il modem GSM montato sulla versione precedente di Eugenio è oggi fuori produzione; per tale motivo, il precedente modem è stato sostituito con un nuovo modem il cui package è più compatto e ciò ha permesso di incrementare la libertà di posizionamento sul circuito stampato. Ed ancora, in OR4 è stata richiesta una funzionalità che stimasse il numero di persone nell'ambiente domestico al fine di migliorare la soluzione restituita dagli algoritmi di demand response; la nuova versione di Eugenio rende possibile tale funzionalità grazie all'utilizzo di un chip subsonico non presente nella precedente versione di Eugenio. Dalla analisi delle onde subsoniche, è possibile stimare con precisione il numero presunto di persone presenti nell'ambiente chiuso dove è posizionato il nuovo Eugenio. Sempre in OR4, è stata richiesta una funzionalità che consentisse la carica/scarica di sistemi di accumulo a batterie per applicazioni domestiche e di tipo commerciale; questo perché anche i prosumer esistenti e/o senza nanogrid potranno partecipare alla comunità ComESTo, mettendo al servizio dell'aggregatore il proprio sistema di accumulo e la corrispondente "flessibilità". Il nuovo Eugenio possiede le adeguate risorse di calcolo per gestire anche il flusso di dati verso il dispositivo di interfaccia – detto EVOBESS – collegato ai sistemi di accumulo commerciali. Più sofisticata è la funzionalità richiesta al nuovo Eugenio dalle attività di progetto nell'ambito dell'OR5 dal titolo "Progettazione della piattaforma per la gestione della Community Energy Storage". La piattaforma ComESTo implementa tutti i modelli e gli algoritmi necessari per la ottimizzazione della flessibilità interna ed il dispacciamento, sia in pianificazione che in tempo reale, ad eccezione del forecasting della domanda e della generazione i cui software sono eseguiti su calcolatori elettronici esteri alla Piattaforma ComESTo. Questi software prelevano dalla Piattaforma ComESTo i dati di misura per come ricevuti dagli smart meter, eseguono le previsioni, restituiscono alla Piattaforma ComESTo il risultato. Al termine del processo computazionale complessivo, la Piattaforma ComESTo invia a ciascuna nanogrid - per il tramite di Eugenio - un profilo di riferimento cioè il profilo di potenza al meter che la nanogrid è tenuta ad "inseguire". Sebbene la Piattaforma ComESTo sia di tipo cloud cioè distribuita su molteplici calcolatori, il processo decisionale generale può classificato come centralizzato. In riferimento a ciò, sono state avanzate alcune ipotesi per una parziale decentralizzazione del processo decisionale generale. Tale decentralizzazione non è una attività prevista nel progetto ComESTo ma potrebbero comunque consentire interessanti ed ulteriori vantaggi; pertanto tale decentralizzazione troverà attuazione in OR7 solo se l'avanzamento

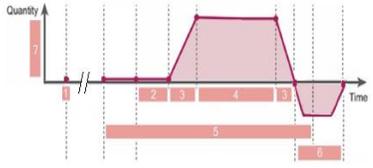
delle attività già previste dal progetto ComESTo lo consentirà. Un esempio di decentralizzazione è la seguente: il nuovo Eugenio calcola localmente le previsioni della domanda e della generazione per la specifica nanogrid. Lo stesso dicasi per l'esecuzione locale di algoritmi di optimal scheduling dei carichi elettrici domestici controllabili, se non alimentati da una nanogrid, al fine di aumentare la flessibilità della domanda della comunità ComESTo. Tale decentralizzazione ha diverse implicazioni; tra queste se ne evidenziano tre: la necessità che Eugenio abbia una adeguata capacità computazionale (target già conseguito grazie al system on module CM3 previsto a seguito della re-ingegnerizzazione); la necessità che la piattaforma ComESTo adegui il processo decisionale e di calcolo; la necessità che la piattaforma ComESTo possa modificare/aggiornare, tempo per tempo, il software di calcolo in esecuzione su ciascun Eugenio. La nuova versione di Eugenio possiede i requisiti hardware e software per ricevere e processare nuovi software per come richiesto dalla Piattaforma ComESTo, come attuazione di un processo decisionale generale non più centralizzato ma parzialmente decentralizzato.

Non per ultimo, nel nuovo Eugenio sono stati aggiunti dei transistor per il funzionamento dei led; così facendo è stata ridotta la potenza fornita ai pin del system on module CM3 e ciò consente di ridurre la probabilità di guasto o di invecchiamento precoce dello stesso CM3. Per concludere, il nuovo Eugenio consente una maggiore e generale integrazione dei componenti, più consona alle esigenze maturate nel progetto ComESTo; inoltre, tutti i componenti sono stati riposizionati su un solo layer così da ridurre il costo di produzione e agevolare la diffusione di questo dispositivo anche nell'ambito delle comunità energetiche e oltre.

13. ALLEGATO A

Nome prodotto		CONGESTION MANAGEMENT		VOLTAGE CONTROL		
		Congestion Management Capacity	Congestion Management Real Time (attivazione del relativo servizio contrattato in Capacity)	Voltage Control - Capacity Reactive Power	Voltage Control - Real Time (attivazione del relativo servizio contrattato in Capacity)	Voltage Control Active Power
Caratteristiche generali	Tipologia	Capacity - Based	Energy - Based	Capacity - Based	Energy based	Energy - based
	Entità fisica	W per h	W per h	Var per h	Var h	W per h
	Modalità di fornitura	Programmata (SRP)	Condizionale (CRP)	Programmata (SRP)	Condizionale (CRP)	Condizionale (CRP)
	Stato funzionamento sistema	Normale/Allerta	Normale/Alerta	Normale/Allerta	Normale/Alerta	Emergenza
	Periodo di validità contratto	(Pluri)Annuale	(Pluri)Annuale	(Pluri)Annuale	(Pluri)Annuale	(Pluri)Annuale
	Remunerazione	È acquistato ad un 'prezzo di capacity' in fase di programmazione	È acquistato al prezzo offerto (pay as bid) per tutti il tempo di esecuzione	È acquistato ad un 'prezzo di capacity' in fase di programmazione	È acquistato al prezzo offerto (pay as bid) per tutti il tempo di esecuzione	È acquistato al prezzo offerto (pay as bid) per tutti il tempo di esecuzione
	Perimetro di esecuzione	Cabine secondarie (nodi MT)	Cabine secondarie (nodi MT)	Cabine secondarie (nodi MT)	Cabine secondarie (nodi MT)	Cabine secondarie (nodi MT)
	Direzionalità	Bidirezionale	Unidirezionale	Bidirezionale	Unidirezionale	Unidirezionale
	Margine di fornitura	Non previsto	Non previsto	Non previsto	Non previsto	Non previsto
	Modalità di attivazione	Attraverso l'aggregatore	Attraverso l'aggregatore	Attraverso l'aggregatore	Attraverso l'aggregatore	Direttamente dal DSO
	Modalità di approvvigionamento	Asta nel caso di più aggregatori al nodo. È previsto un prezzo massimo (cap)	Asta nel caso di più aggregatori al nodo. È previsto un prezzo massimo (cap)	Asta nel caso di più aggregatori al nodo. È previsto un prezzo massimo (cap)	Asta nel caso di più aggregatori al nodo. È previsto un prezzo massimo (cap)	Direttamente dal DSO
	Aggregabilità	SI	SI	SI	SI	SI

Nome prodotto	Curva Potenza-Tempo	Id	Descrizione	CONGESTION MANAGEMENT		VOLTAGE CONTROL		
				Congestion Management Capacity	Congestion Management Real Time	Voltage Control - Capacity Reactive Power	Voltage Control - Real Time (attivazione del	Voltage Control Active Power

				(attivazione del relativo servizio contrattato o in Capacity)		relativo servizio contrattato o in Capacity)		
<b>Caratteristiche della curva Potenza-Tempo</b>		1	Termine delle contrattazioni	Abbondante prima della (probabile) fornitura	(Near) Real - Time, contestuale alle sessioni del MB	Abbondantemente prima della (probabile) fornitura	(Near) Real - Time, contestuale alle sessioni del MB	
		2	Periodo di preparazione	ore	pochi minuti: intervallo 2 + intervallo 3 <15 min	N.A.	pochi minuti: intervallo 2 + intervallo 3 <15 min	pochi minuti: intervallo 2 + intervallo 3 <15 min
		3	Attivazione/Disattivazione	N.A.	pochi minuti: intervallo 2 + intervallo 3 <15 min	N.A.	pochi minuti: intervallo 2 + intervallo 3 <15 min	pochi minuti: intervallo 2 + intervallo 3 <15 min
		4	Periodo di erogazione	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dall'esigenza di sicurezza
		5	Intervallo di servizio	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dall'esigenza di sicurezza
		6	Payback	N.A.	Non previsto	N.A.	Non previsto	N.A.
		7	Volume del prodotto	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dal contratto	Dipende dall'esigenza di sicurezza

## 14. BIBLIOGRAFIA

---

- <sup>i</sup> CEZ Distribuce a. s., “D2.1 Use case detailed definitions and specifications of joint activities in the Demonstrators,” 2017.
- <sup>ii</sup> J. Radecke, J. Hefele and L. Hirth, “Markets for Local Flexibility in Distribution Networks,” ZBW – Leibniz Information Centre for Economics, Kiel, Hamburg, 2019.
- <sup>iii</sup> ‘ISO/IEC 17203:2011’, ISO. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/93/59388.html> (accessed Aug. 28, 2020).
- <sup>iv</sup> M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition*, Third. Addison-Wesley Professional, 2003
- <sup>v</sup> ‘Message Queuing Telemetry Transport (MQTT)’, ISO. <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/94/69466.html> (accessed Aug. 05, 2020).
- <sup>vi</sup> Node.js, ‘Node.js’, *Node.js*. <https://nodejs.org/en/> (accessed Sep. 08, 2020).
- <sup>vii</sup> AR. 4.1.3 - P-NG: Protocollo di comunicazione M2M per la Nano Grid. Progetto COMESTO, 26 febbraio 2020 .
- <sup>viii</sup> ‘Time-series data simplified’. <https://www.timescale.com/> (accessed Sep. 21, 2020).
- <sup>ix</sup> Y. Ge, C. Zhou, and D. M. Hepburn, ‘Domestic electricity load modelling by multiple Gaussian functions’, *Energy Build.*, vol. 126, pp. 455–462, Aug. 2016, doi: 10.1016/j.enbuild.2016.05.060
- <sup>x</sup> W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, ‘pvlb python: a python package for modeling solar energy systems’, *J. Open Source Softw.*, vol. 3, no. 29, p. 884, Sep. 2018, doi: 10.21105/joss.00884
- <sup>xi</sup> ‘Consul by HashiCorp’, *Consul by HashiCorp*. <https://www.consul.io/> (accessed Sep. 11, 2020).
- <sup>xii</sup> ‘NGINX | High Performance Load Balancer, Web Server, & Reverse Proxy’, *NGINX*. <https://www.nginx.com/> (accessed Sep. 15, 2020).
- <sup>xiii</sup> The Internet of Things is powered by Z-Wave, *Z-Wave Alliance*. <https://z-wavealliance.org/> (accessed Aug. 08, 2020).
- <sup>xiv</sup> M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language, Third Edition*, Third. Addison-Wesley Professional, 2003
- <sup>xv</sup> Enedis, Avacon, CEZ Distribuce, E.ON, Enexis, RWTH, «D3.4: Interoperable APIs Specification,» 2019.
- <sup>xvi</sup> E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, <https://tools.ietf.org/html/rfc5246> (accessed Nov. 03, 2020).
- <sup>xvii</sup> G. Bouloukakis, A. Kattepur, N. Georgantas, and V. Issarny, Queueing Network Modeling Patterns for Reliable and Unreliable Publish/Subscribe Protocols, in *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, New York NY USA, Nov. 2018, pp. 176–186, doi: 10.1145/3286978.3287002.

<sup>xviii</sup> B. K. Aichernig and R. Schumi, How Fast Is MQTT?: Statistical Model Checking and Testing of IoT Protocols, in *Quantitative Evaluation of Systems*, vol. 11024, A. McIver and A. Horvath, Eds. Cham: Springer International Publishing, 2018, pp. 36–52.

<sup>xix</sup> Kubernetes, <https://kubernetes.io/it/> (accessed Nov. 04, 2020).

<sup>xx</sup> P. Mishra, A. Biswal, S. Garg, R. Lu, M. Tiwary, and D. Puthal, Software Defined Internet of Things Security: Properties, State of the Art, and Future Research, *IEEE Wirel. Commun.*, vol. 27, no. 3, pp. 10–16, Jun. 2020, doi: 10.1109/MWC.001.1900318.

<sup>xxi</sup> CEER. [Online]. Available: <https://www.ceer.eu/documents/104400/-/-/e5186abe-67eb-4bb5-1eb2-2237e1997bbc>.