



Ricerca di Sistema elettrico

Sperimentazione piattaforma Smart City su scala nazionale

C. Novelli, A. Frascella, A. Brutti, N. Gessa, S. Pizzuti,
F. Moretti, G. Santomauro, M. Chinnici, G. Ponti

SPERIMENTAZIONE PIATTAFORMA SMART CITY SU SCALA NAZIONALE

C. Novelli, A. Frascella, A. Brutti, N. Gessa, S. Pizzuti,
F. Moretti, G. Santomauro, M. Chinnici, G. Ponti

Dicembre 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - III annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package: Local Energy District

Linea di attività: Sperimentazione del Framework per la Governance dei Dati Urbani Energetici

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Indice

SOMMARIO.....	5
INTRODUZIONE.....	6
1.1 SOTTO-ATTIVITÀ E GANTT.....	7
1.2 ORGANIZZAZIONE DEL RAPPORTO TECNICO.....	8
1.3 GRUPPI DI LAVORO.....	9
2 SPERIMENTAZIONE PIATTAFORMA INTER-SCP.....	10
2.1 CONSOLIDAMENTO PLATFORMSTATUS.....	12
2.2 ESPORTAZIONE PLATFORMSTATUS.....	13
2.2.1 Sviluppo software Job di Export.....	13
2.2.2 Configurazione Produzione sulla SCP-Casaccia.....	14
2.2.3 Configurazione Job "Export" sulla SCP-Casaccia.....	17
2.2.4 Esecuzione del Job "Export" sulla SCP-Casaccia.....	18
2.3 TRASPORTO UD DA SCP-CASACCIA A INTER-SCP.....	19
2.3.1 Configurazione Produzione "Platform Status" (singolo) sulla inter-SCP.....	20
2.3.2 Configurazione Job "Get Remote" sulla inter-SCP.....	23
2.3.3 Esecuzione del Job "Get Remote" sulla inter-SCP.....	24
2.4 REPLICABILITÀ DEL CASO STUDIO PILOTA.....	26
2.5 DATA FUSION PLATFORMSTATUS.....	29
2.5.1 Sviluppo software Job di Data Fusion.....	29
2.5.2 Configurazione Produzione "Platform Status" (multiplo) sulla inter-SCP.....	30
2.5.3 Configurazione Job "Data Fusion" sulla inter-SCP.....	32
2.5.4 Esecuzione del Job "Data Fusion" sulla inter-SCP.....	34
2.6 DASHBOARD DELLA INTER-SCP.....	36
2.7 INDICATORI DELLA SPERIMENTAZIONE.....	38
2.7.1 Indicatori su Esportazione.....	38
2.7.2 Indicatori su Trasporto.....	41
2.7.3 Indicatori su DataFusion.....	42
3 SVILUPPO SOFTWARE INTER-SCP 1.0.....	43
3.1 EVOLUZIONE SCP-GUI 2.0.....	44
3.1.1 Sezione DEVELOPER.....	44
3.1.2 Duplice View Solution/Service.....	49
3.1.3 Configurazione UrbanDataset da Ontologia.....	50
3.1.4 Map integrata nella produzione.....	50
3.1.5 Alert via email.....	51
3.2 SVILUPPO SCP-SCHEDULER E JOB.....	52
3.3 EVOLUZIONE URBANDATASET GATEWAY.....	53
3.3.1 Implementazione 2 nuovi metodi.....	53
3.3.2 Consolidamento e correzione anomalie.....	54
3.3.3 Validatore Timestamp.....	54
3.3.4 Consolidamento validatore JSON-Schema.....	56
3.3.5 Integrazione validatore Schematron.....	56
3.3.6 Riorganizzazione ordine dei validatori.....	57
3.4 ORGANIZZAZIONE PROGETTI.....	59
3.5 INTERSEZIONI CON ALTRE LA.....	61
3.5.1 Nuovo Identity Provider.....	61
3.5.2 Web Application per modifica Ontologia.....	61
3.5.3 Connessione SCP-PELL a inter-SCP.....	61
4 GUIDA UTENTE PER L'ADESIONE.....	64

4.1	PERCORSO 1 - ADESIONE CON UNA NUOVA SCP	65
4.1.1	Passo 1 - Adesione alla specifica SCPS Functional 2.0.....	66
4.1.2	Passo 2 - Adesione alla specifica SCPS Semantic 2.0.....	66
4.1.3	Passo 3 - Adesione alla specifica SCPS Information 2.0	67
4.1.4	Passo 4 - Adesione alla specifica SCPS Collaboration 2.0.....	67
4.1.5	Passo 5 - Adesione alla specifica SCPS Communication 2.0	68
4.1.6	Conclusioni Percorso 1	68
4.2	PERCORSO 2 - ADESIONE CON UNA SOLUTION A UNA SCP.....	69
4.2.1	Passo 1 - Adesione all'Architettura di Riferimento.....	69
4.2.2	Passo 2 - Analisi e Progettazione dei flussi dati	70
4.2.3	Passo 3 - Esportazione degli UrbanDataset	70
4.2.4	Passo 4 - Configurazione della Collaborazione sulla SCP	71
4.2.5	Passo 5 - Implementazione della Comunicazione	72
4.2.6	Conclusioni Percorso 2	72
4.3	PERCORSO 3 - ADESIONE CON UNA SCP ALLA ISCP.....	73
4.3.1	Passo 1 - Adesione all'Architettura di Riferimento.....	74
4.3.2	Passo 2 - Progettazione dei flussi dati.....	74
4.3.3	Passo 3 - Esportazione degli UrbanDataset	75
4.3.4	Passo 4 - Configurazione della Collaborazione	75
4.3.5	Passo 5 - Implementazione della Comunicazione	77
4.3.6	Conclusioni Percorso 3	78
5	SVILUPPI IN CORSO E APPLICAZIONI FUTURE	79
5.1	CASO STUDIO URBANO	79
5.1.1	"Export Solutions Platform Status"	80
5.1.2	"Data Fusion Smart Building"	80
5.1.3	"Data Fusion Smart Home"	80
5.1.4	"Export Map View"	80
5.2	KPI ENERGETICI.....	81
5.3	DASHBOARD NAZIONALE E REGIONALE	83
6	CONCLUSIONI.....	84
7	RIFERIMENTI.....	85
	APPENDICE A – DEFINIZIONI URBANDATASET.....	87
	A1 URBANDATASET "PLATFORM STATUS" 2.0	87
	A2 URBANDATASET "ELECTRIC METER READING" 2.0.....	88
	A3 URBANDATASET "GAS METER READING" 2.0	91
	APPENDICE B – ANALISI REQUISITI PER WEBAPP ONTOLOGY-BASED	93

Sommario

Questo è il rapporto tecnico, parte del WP1 “Local Energy District”, per il progetto “Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali”, **Linea di Attività 21 (LA 1.21), “Sperimentazione del Framework per la Governance dei Dati Urbani Energetici”**.

La LA 1.21 fa parte di un percorso, organizzato in tre annualità, pianificato come segue:

1. LA 1.19 (anno 2019): “Progettazione Framework per la Governance dei Dati Urbani Energetici”; linea di attività descritta nel report [1];
2. LA 1.20 (anno 2020): “Sviluppo Piattaforma per la Governance dei Dati Urbani Energetici”; linea di attività descritta nel report [2];
3. LA 1.21 (anno 2021): “Sperimentazione del Framework per la Governance dei Dati Urbani Energetici” che ha previsto l’implementazione del caso studio pilota e la sperimentazione finale dello scambio dati tra piattaforma agente su scala urbana e piattaforma agente su scala nazionale; è la linea di attività descritta in questo report.

La sperimentazione del “caso studio pilota” è parte finale dell’intero percorso: utilizzando l’analisi, i requisiti e gli studi compiuti nella prima annualità (2019), ha permesso di ultimare la progettazione e avviare il primo sviluppo della piattaforma agente su scala nazionale nella seconda annualità (2020), la cui implementazione finale e sperimentazione hanno trovato compimento nella terza annualità (2021).

La realizzazione del caso studio pilota fornisce l’implementazione di una piattaforma centrale di governance in grado di recuperare i dati urbani dalle piattaforme smart city agenti su scala urbana, abilitando la comunicazione interoperabile su scala nazionale, tramite l’utilizzo di specifiche e formati condivisi.

Descriveremo il risultato della sperimentazione sottolineando che,

- da un lato, **è stato raggiunto e superato l’obiettivo previsto nel caso studio pilota**, di mettere in connessione la singola piattaforma urbana “Smart Village Casaccia”, con la “piattaforma nazionale”, implementando, in totale, la connessione di cinque piattaforme, ottenendo così un traffico dati molto interessante e una dimostrazione esauriente della possibilità di replicare l’approccio;
- dall’altro lato, **sebbene siano stati individuati dati urbani energetici interessanti, non vi è ancora un’implementazione effettiva** (p.es. i KPI energetici, di cui è ancora in corso la valutazione ai fini dell’implementazione nella inter-SCP e di cui comunque parleremo) e quindi anche la dashboard finale per visualizzarli risulta ancora poco matura; proprio sulla visualizzazione dei KPI energetici nelle Dashboard si suggerisce di investire il lavoro nel prossimo futuro per valorizzare il framework di comunicazione consolidato.

Introduzione

Questo report descrive il risultato finale del triennio perseguito nelle LA.1.19 [1], LA.1.20 [2], **LA.1.21**, in particolare quest'ultima prevede la “**Sperimentazione del Framework per la Governance dei Dati Urbani Energetici**”. Tale sperimentazione prevede il completamento dell'implementazione della “**Piattaforma per la Governance dei Dati Urbani Energetici**” e del caso studio pilota che descrive la comunicazione tra questa piattaforma agente su scala nazionale e una piattaforma agente su scala urbana.

L'attività si inserisce nell'iniziativa dello “SCP Project” [3] utilizzando due importanti risultati:

- le *Smart City Platform Specification for interoperability layer* versione 2.0 (SCPS 2.0) [4];
- il prototipo di piattaforma ICT per lo Smart District / Smart City, *Smart City Platform (SCP)* [5];

nati dal lavoro del triennio precedente della Ricerca di Sistema Elettrico [6] e che, in questo triennio, sono stati consolidati ed evoluti.

Ricordiamo che le SCPS, che richiameremo spesso nel testo, sono composte da 5 specifiche:

- *SCPS Functional 2.0* [7]: architettura della Smart City Platform;
- *SCPS Semantic 2.0* [8]: Ontologia 2.0 degli UrbanDataset;
- *SCPS Information 2.0* [9]: sintassi JSON/XML degli UrbanDataset;
- *SCPS Collaboration 2.0* [10]: definizione delle collaborazioni nel Registry;
- *SCPS Communication 2.0* [11]: interfaccia del web service *UrbanDatasetGateway*.

A corollario delle specifiche ricordiamo anche il *Glossario 2.0* [12] che si suggerisce di utilizzare per ricercare eventuali termini tecnici utilizzati nel testo.

Il prototipo di “Piattaforma Inter-City per la Governance dei Dati Urbani Energetici”,

- è denominato **inter-SmartCityPlatform (inter-SCP o iSCP)**,
- implementa le specifiche SCPS;
- è basato sul prototipo SCP, che estende per abilitare la comunicazione tra SCP;
- ha un duplice obiettivo:
 1. Il monitoraggio su scala nazionale della gestione dei dati urbani energetici;
 2. la comunicazione tra piattaforme Smart City di città diverse.

Questo report, relativo alla linea di Attività 1.21, descrive i **risultati** raggiunti in questa terza annualità:

- il prototipo di **piattaforma inter-city versione 1.0** [13], secondo l'architettura del framework per la governance dei dati urbani su scala nazionale, per l'integrazione di piattaforme cittadine;
- la **dashboard della inter-SCP** (iSCP-Dashboard) [14] per la visualizzazione di dati raccolti da piattaforme agenti su scala urbana;
- l'implementazione del **caso studio pilota** che, oltre alla connessione pianificata della SCP-Casaccia alla inter-SCP, ha effettuato la connessione di altre quattro SCP (tra cui la SCP-PELL);
- la **guida utente per l'adesione al framework, versione finale e pubblica**.

Si noti che lo sviluppo dello **SCP-Scheduler**, elemento centrale del lavoro svolto, era inizialmente previsto per la seconda annualità (LA1.20) ed è stato spostato nella terza annualità (LA1.21); ciò è da attribuirsi alla definizione del caso studio pilota, presentato nel precedente report [2], che esprimendo tutti i requisiti e le funzionalità per implementare la comunicazione tra SCP e inter-SCP è una descrizione propedeutica alla progettazione e sviluppo del componente SCP-Scheduler svolti quest'anno (descritti nel report [15]).

Descriveremo nel seguito di questa introduzione:

1. le sotto-attività organizzate nello schema Gantt;
2. l'organizzazione del report nei suoi contenuti;
3. i gruppi di lavoro che hanno partecipato alle diverse attività.

1.1 Sotto-Attività e Gantt

Le principali **sotto-attività** che hanno portato ai risultati della LA 1.21 sono le seguenti (segue GANTT con le rispettive tempistiche relative):

1. Guida Utente per l'Adesione: versione finale della guida utente, con diversi percorsi di adesione, pubblicati online sul sito web SCP Project;
2. Progettazione e Sviluppo SCP-Scheduler: realizzazione del componente SCP-Scheduler che permette di eseguire Job/Service;
3. SCP-GUI 2.0: aggiornamento del componente GUI per implementare la inter-SCP;
4. Connessione SCP – inter-SCP: sperimentazione base del caso studio pilota con prima comunicazione tra SCP-Casaccia e inter-SCP;
5. Implementazione Job: sviluppo dei job che vengono eseguiti dallo SCP-Scheduler per abilitare la comunicazione tra SCP e la datafusion;
6. Migrazione SCP: aggiornamento di 5 SCP all'ultima versione per la connessione alla inter-SCP;
7. Configurazione iSCP-Dashboard: utilizzo del componente SCP-DASH per definire la prima dashboard di monitoraggio per la inter-SCP;
8. Sperimentazione inter-SCP 1.0: prototipo di piattaforma inter-SCP finale, agente su scala nazionale, connessa a 5 SCP urbane;
9. Report LA1.21: stesura del presente report.

Si noti che le prime due righe (“definizione caso studio pilota” e “prototipo iSCP 0.1”) sono le attività finali del lavoro svolto nella LA 1.20 (annualità 2020) e sono state riportate per meglio far comprendere il senso di continuità con la fase sperimentale della LA 1.21 (annualità 2021).

Linea Attività	Sotto Attività	2021												2022	
		gennaio	febbraio	marzo	aprile	maggio	giugno	luglio	agosto	settembre	ottobre	novembre	dicembre	gennaio	febbraio
LA 1.21	Definizione Caso Studio Pilota su scala nazionale														
	Prototipo inter-SCP 0.1														
	Guida Utente per l'Adesione														
	Progettazione e Sviluppo SCP-Scheduler														
	SCP-GUI 2.0														
	Connessione SCP - interSCP														
	Implementazione Jobs														
	Migrazione SCPs														
	Configurazione iSCP-DASHBOARD														
	Sperimentazione inter-SCP 1.0														
	Report LA1.21														

Figura 1. Gantt relativo a Linea di Attività 1.21

1.2 Organizzazione del rapporto tecnico

Questo rapporto tecnico è organizzato come segue:

Il **Capitolo 1** è questa **Introduzione** al lavoro svolto, dove si tenta di fornire una descrizione generale dei principali risultati, delle principali sotto-attività e dei gruppi di lavoro, anche mediante grafico Gantt.

Nel **Capitolo 2**, che possiamo ritenere il capitolo più importante, riportata la **Sperimentazione della piattaforma di governance** e, quindi, l'**implementazione del Caso Studio Pilota**, che ben rappresenta l'obiettivo finale del triennio: in esso sono evidenziati tutti i concetti chiave e componenti necessari per abilitare la comunicazione interoperabile da una piattaforma smart city urbana (la SCP-Casaccia) verso la piattaforma agente su scala nazionale (inter-SCP), per ottenere una visualizzazione dati finale. Vedremo come, in questa sperimentazione, **si sono connesse alla inter-SCP un totale di 5 SCP**. Il capitolo termina con l'**analisi di indicatori** relativi alle diverse fasi della sperimentazione.

Nel **Capitolo 3**, scende nel dettaglio dello sviluppo software avvenuto per ottenere la **inter-SCP 1.0** agente su scala nazionale. In particolare, è stato fatto uno sforzo consistente per consolidare ed evolvere il componente **SCP-GUI** alla versione 2.0, è stato sviluppato il componente **SCP-Scheduler** che permette di eseguire **JOB/Service** (che, tra le altre cose, rendono possibile la comunicazione tra SCP), è stato consolidato ed evoluto il web service **UrbanDatasetGateway** per l'invio/recupero di UrbanDataset e, infine, tra la moltitudine di progetti (di cui si offre una panoramica) si descrivono anche alcuni interventi compiuti in altre linee di attività (la più importante è senz'altro quella relativa al nuovo **Identity Provider**).

Nel **Capitolo 4**, vengono descritti i tre percorsi per l'adesione dell'utente al Framework SCP, uno per ognuna delle tre tipologia di esigenza (nuova SCP, nuova Solution, connessione a inter-SCP).

Nel **Capitolo 5**, vengono descritte alcune sotto-attività che sono tutt'ora in corso e che, si ipotizza, continueranno nei prossimi mesi: tra queste sotto-attività abbiamo il completamento del **secondo caso studio**, "urbano", l'individuazione di **nuovi di KPI energetici** da utilizzare nella inter-SCP, e la definizione di diverse **dashboard per la visualizzazione dei dati** tramite tabelle e grafici, **su diverse scale** (tra cui quella **regionale**, che replicherebbe l'approccio adottato in questa sperimentazione).

Nel **Capitolo 6**, riportiamo le conclusioni sul lavoro svolto in questa linea di attività e alcune riflessioni sull'intero triennio (LA 1.19, 1.20, 1.21) e quelli che potrebbero essere gli sviluppi futuri per l'utilizzo di SCP e inter-SCP.

In **Appendice A**, vengono riportare le definizioni dettagliate dei nuovi UrbanDataset definiti per la sperimentazione e per l'individuazione dei KPI energetici.

In **Appendice B**, viene riportata l'analisi dei requisiti che ha portato allo sviluppo di una web application per la modifica agevolata dell'Ontologia degli UrbanDataset.

1.3 Gruppi di lavoro

Nella LA 1.21, hanno partecipato e operato in sinergia i seguenti gruppi di lavoro:

- ENEA TERIN-SEN-SCC
 - o coordinamento generale di tutte le attività;
 - o manutenzione specifiche SCPS Communication 2.0;
 - o sviluppo software UrbanDatasetGateway;
 - o progettazione e supervisione allo sviluppo software di SCP-GUI 2.0 e SCP-Scheduler;
 - o progettazione JOB che possono essere eseguiti dallo SCP-Scheduler;
 - o definizione Dashboard della inter-SCP;
 - o implementazione del caso studio pilota;
 - o sperimentazione finale tramite il prototipo inter-SCP 1.0;
 - o valutazione sperimentazione tramite indicatori per l'interoperabilità;
 - o definizione dei tre percorsi di adesione per l'utente;
- ENEA TERIN-SEN-CROSS
 - o manutenzione specifiche SCPS Information 2.0;
 - o analisi KPI;
 - o raccolta dei requisiti per sviluppo web application per modificare l'Ontologia;
- ENEA TERIN-ICT-HPC
 - o consolidamento prototipo SCP/inter-SCP e integrazione validatore Schematron;
- Heartwood Labs (società esterna)
 - o Sviluppo software relativo a SCP-GUI 2.0 e SCP-Scheduler.

Collegamento con altre linee di attività:

- Politecnico di Milano (LA 1.25 [16])
 - o definizione di un insieme di riferimento di dati urbani energetici su scala nazionale e municipale;
- Università di Bologna (LA 1.23 [17])
 - o sviluppo strumento software per modificare l'Ontologia (dove vengono definiti gli UrbanDataset) più agevolmente, direttamente da applicazione web;
- Università di Insubria (LA 1.26 [20])
 - o definizione dell'Urban Check Up Model per effettuare indagini preliminari verso i referenti di municipalità e aziende in vista di una transizione tecnologica in ambito Smart City;
- PELL ([21])
 - o Piattaforma PELL: avvio popolamento dati dinamici PELL IP, test dati statici PELL Edifici.

2 Sperimentazione piattaforma inter-SCP

In questo capitolo viene descritta la **sperimentazione** sulla “Piattaforma per la Governance dei Dati Urbani Energetici” avvenuta con l’**implementazione del caso studio pilota**.

La sperimentazione intende mostrare (e dimostrare) una metodologia, basata su specifiche per l’interoperabilità condivise e pubbliche, che permette il monitoraggio dei dati urbani (e quindi la governance) sia su scala urbana con la **SmartCityPlatform (SCP)**, sia su scala nazionale con la piattaforma **inter-SmartCityPlatform (inter-SCP o iSCP)**.

Al fine di presentare i diversi passaggi dell’implementazione e della relativa sperimentazione, riprendiamo i punti salienti della descrizione del caso studio pilota (fornita dettagliatamente nel report della LA1.20 [2]):

“il caso studio pilota è una descrizione specifica dell’utilizzo reale delle metodologie e delle tecnologie in oggetto che, per le sue stesse caratteristiche, viene considerato come caso fondamentale: è stato definito in modo tale da coprire una casistica più ampia possibile e non solo il caso concreto che descrive direttamente; implementando il caso studio pilota potremo configurare un numero consistente di casi ad esso riconducibili, che necessitano degli stessi componenti e utilizzano il medesimo approccio”.

La descrizione della sperimentazione è organizzata con un percorso analogo alla descrizione del caso studio pilota “Da SCP-Casaccia a inter-SCP” e, per questa ragione, riportiamo anche la figura schematica del caso studio pilota in cui vengono riportate graficamente diverse fasi, dal basso verso l’alto (Figura 2).

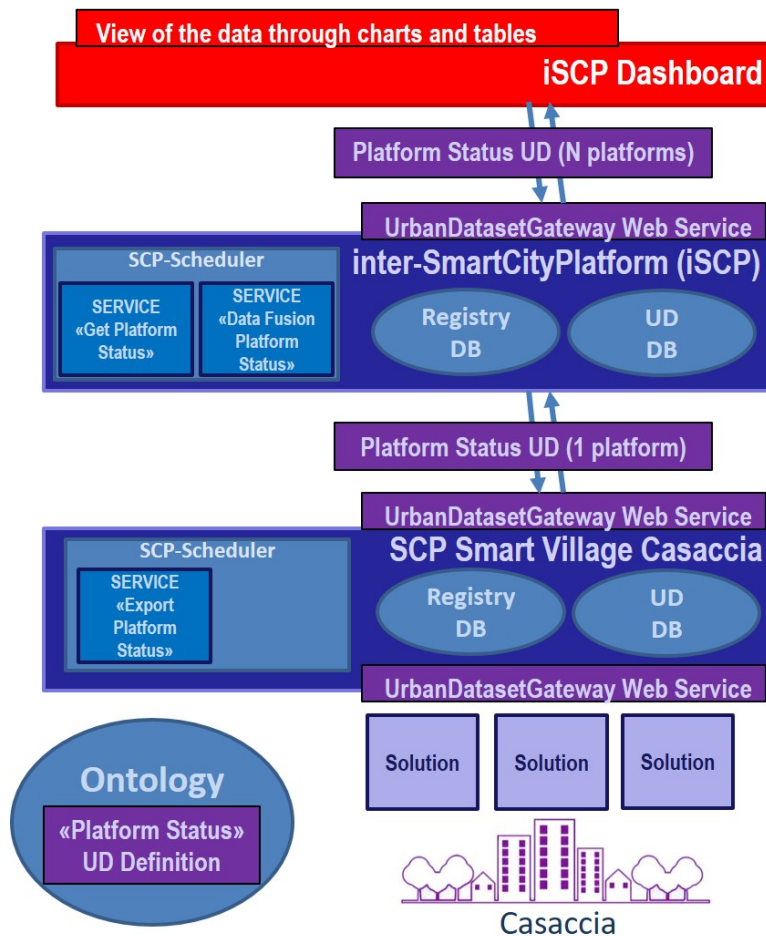


Figura 2. Caso Studio Pilota “Da SCP-Casaccia a inter-SCP”

Le fasi delineate nel caso studio pilota sono le seguenti:

- 1 definizione UrbanDataset (nell'Ontologia) per rappresentare i dati da trasmettere;
- 2 esportazione e pubblicazione dell'UD in una SCP reale e funzionante: verrà utilizzato allo scopo il prototipo SCP-Casaccia attualmente utilizzato per lo Smart Village Casaccia;
- 3 recupero dell'UD da parte della inter-SCP (iSCP) tramite SCP-Scheduler;
- 4 preparazione dei dati di input per la SCP-Dashboard, elaborando gli UD raccolti dalla iSCP: anche in questo caso si utilizzerà lo SCP-Scheduler per effettuare azione di DataFusion;
- 5 recupero degli UD nella SCP-Dashboard per la visualizzazione dati finale.

Come si evince dallo schema in Figura 2 del caso studio pilota, un ruolo centrale sarà ricoperto dallo SCP-Scheduler e dai moduli software che possono essere configurati ed eseguiti al suo interno.

Utilizzeremo nel testo i termini "SCP-Scheduler", "Job", "Service" e "Bridge"; ne forniamo qui una definizione sintetica:

- lo **SCP-Scheduler** è un componente della SCP che è in grado di registrare diversi Job, avviarli con una periodicità predefinita, fermarli, monitorarli, gestirli;
- con "**Job**" intendiamo un modulo software che può essere eseguito per completare uno o più task ben definiti;
- con "**Bridge**" intendiamo un particolare Job che, una volta eseguito, si preoccupa di essere ponte tra due sistemi, recuperando i dati dal primo sistema per poi inserirli nel secondo;
- con "**Service**" intendiamo identificare un elemento che può leggere e/o scrivere UrbanDataset nella SCP ma, mentre la Solution è visto come sistema "esterno" alla SCP, gestito da terze parti, il Service è un sistema "interno" alla SCP, gestito dall'amministratore della SCP.

Nell'implementazione del caso studio pilota, sono previsti 3 "Job", alcuni di loro funzionano come "Bridge", e tutti sono configurati nella SCP come "Service", per questo si farà spesso riferimento a loro come "Job/Service", per indicare che sono moduli software configurati per essere eseguiti nella SCP.

Tramite lo SCP-Scheduler e alcuni Job/Service sarà dunque possibile abilitare la comunicazione tra diverse SCP e quindi, nel caso specifico, **sarà possibile abilitare la comunicazione tra diverse SCP cittadine e la inter-SCP agente su scala nazionale.**

L'implementazione non ha previsto modifiche all'ipotesi teorica che era stata definita (Figura 2); **il caso studio pilota è stato implementato in tutte le sue parti.** Non solo, dopo aver implementato e sperimentato il caso studio pilota, con la connessione tra SCP-Casaccia e inter-SCP, **sono state connesse altre quattro SCP urbane alla piattaforma inter-SCP** agente su scala nazionale, dimostrando la tesi e la bontà dell'approccio, ottenendo una prima infrastruttura agente su scala nazionale (si veda in particolare il paragrafo 2.4 relativo alla replicabilità del caso studio pilota).

Descriviamo nei prossimi paragrafi ogni fase della sperimentazione seguendo quelle del caso studio pilota.

2.1 Consolidamento PlatformStatus

Nel corso della precedente annualità (LA 1.20 [2]) è stato definito e inserito, nell'Ontologia, l'UrbanDataset "Platform Status", che consente ad una piattaforma software di inviare periodicamente informazioni ad un'altra piattaforma che ne debba tenere monitorato lo stato di attività.

Nel corso di questa annualità l'utilizzo del "Platform Status" è stato concretamente testato nell'ambito della sperimentazione relativa al caso studio pilota "Da SCP-Casaccia a inter-SCP" (che descriviamo nel dettaglio in questo capitolo); in tale sperimentazione è emersa l'esigenza di ampliare il set di informazioni che possono essere inviate con questo UrbanDataset e, conseguentemente, è stata aggiornata la sua definizione semantica nell'Ontologia SCPS [8].

Nello specifico, la nuova esigenza nasce dal fatto che nella fase di datafusion (si veda par. 2.5) viene utilizzato l'UD "Platform Status" anche per descrivere lo stato di N piattaforme, di cui bisogna registrare le coordinate spaziali e temporali, differenziandole; quindi, i requisiti che sono stati individuati sono:

- consentire la georeferenziazione delle informazioni a livello di singola riga dell'UrbanDataset (nella versione iniziale dell'UD le informazioni di tutte le sue righe vengono automaticamente associate alle coordinate geografiche al centro geometrico dell'area gestita dalla piattaforma, ovvero quelle indicate a livello di *context*);
- consentire l'indicazione dell'istante temporale a cui si riferiscono le informazioni a livello di singola riga dell'UrbanDataset (nella versione iniziale dell'UD le informazioni di tutte le sue righe vengono automaticamente associate al suo istante di generazione).

Questi requisiti sono stati implementati aggiungendo le seguenti proprietà opzionali a quelle già presenti nella definizione semantica dell'UD:

- *coordinate* a livello di linea;
- *timestamp* a livello di linea.

Si fa presente che l'Ontologia ha subito un upgrade di versione (da 1.0 a 2.0) e che la nuova versione supporta nuove funzionalità, ad esempio la possibilità di definire proprietà opzionali (come fatto per questo aggiornamento dell'UD "PlatformStatus") o di associare più unità di misura alternative ad una stessa proprietà. La nuova definizione dell'UrbanDataset "PlatformStatus" è stata quindi implementata nella versione 2.0 dell'Ontologia SCPS.

Si rimanda all'Appendice A1 per la descrizione dettagliata dell'UrbanDataset "Platform Status".

2.2 Esportazione PlatformStatus

Una volta definito l'UrbanDataset "Platform Status" definitivo, è possibile procedere con l'esportazione di questo dataset nella SCP-Casaccia.

Questa fase prevede 5 task da attuare nella SCP-Casaccia che elenchiamo di seguito:

- 1 Sviluppo software del Job di esportazione;
- 2 Configurazione della produzione nella SCP-Casaccia (del SERVICE, dell'UD e della loro associazione);
- 3 Configurazione del Job nello scheduler della SCP-Casaccia;
- 4 Esecuzione del Job;
- 5 Verifica del risultato.

Descriviamo nel dettaglio questi 5 task.

2.2.1 Sviluppo software Job di Export

L'esportazione dell'UD "Platform Status" è un'azione effettuata da un Job/SERVICE che prevede

- la verifica delle funzionalità generali della SCP e
- la lettura dal Registry di alcune informazioni

allo scopo di preparare un breve report sullo stato della piattaforma che può essere reso disponibile, in questo caso, alla inter-SCP.

Questa azione di verifica funzionalità e lettura informazioni è stata implementata in un modulo software denominato "ExportPlatformStatus", sviluppato

- nel linguaggio di programmazione di riferimento per poter utilizzare lo scheduler, ovvero Java;
- utilizzando la libreria "job-template" necessaria per ereditare le funzionalità dei job che possono così essere eseguiti dallo SCP-Scheduler;
- utilizzando come output il template JSON dell'UrbanDataset "Platform Status";
- per ottenere, come risultato, un pacchetto .jar che deve essere caricato all'interno della SCP.

La verifica delle funzionalità riguarda:

- il controllo sul numero di utenti e di solution e sul numero di UrbanDataset prodotti e acceduti: un numero pari a zero è una situazione anomala anche in fase di SCP appena istanziata;
- la chiamata al metodo di test dell'UrbanDatasetGateway che ritorna un codice "00" se funzionante.

Dipendentemente dalla verifica delle funzionalità e dalla lettura errata o meno delle informazioni dalla bancadati Registry, verrà ritornato nell'UD "PlatformStatus" uno dei seguenti tre messaggi:

1. "INFO: Everything OK, the platform is working normally."
2. "WARNING: the platform is working but it requires not urgent checking action."
3. "ERROR: the platform is not working normally, it requires urgent checking action."

La lettura delle informazioni riguarda esattamente le proprietà dell'UD presenti nel Registry (proprietà introdotte con la prima versione, nel report LA 1.20, par. 3.2.2 "UrbanDataset PlatformStatus").

Si noti che questo sviluppo è stato fatto per esportare le informazioni dalla SCP-Casaccia, così come descritto nel caso studio pilota ma, essendo le SCP del tutto identiche, potrà essere riutilizzato in tutte le SCP (si veda a tal proposito il par. 2.4 sulla replicabilità del caso studio pilota).

2.2.2 Configurazione Produzione sulla SCP-Casaccia

L'azione di esportazione del Job/SERVICE "ExportPlatformStatus" produce l'UD "Platform Status" che deve essere salvato nella SCP per poi essere reso disponibile per la inter-SCP; ciò viene fatto riutilizzando il sistema principale di comunicazione interoperabile (basato su specifiche SCPS Communication 2.0 [11]) che permette di scrivere/leggere UrbanDataset dal DB locale della SCP chiamando i metodi dell'UrbanDatasetGateway.

L'azione di esportazione e salvataggio dell'UD può essere vista come una produzione a tutti gli effetti e come tale va configurata nella SCP.

È stato quindi necessario configurare tramite la SCP-GUI:

- un utente di sistema dedicato ai SERVICE interni: *scp.services@enea.it* e associare ad esso il Service "ExportPlatformStatus" (si veda la Figura 3);
- l'UD "Platform Status" versione 2.0 come supportato in questa SCP (si veda Figura 4);
- la produzione come associazione Service-UD (si veda Figura 5).

Smart City Platform SMART VILLAGE CASACCIA
SCP-GUI 2.0 (Interoperability Layer)

HOME HISTORY DEVELOPER ADMIN SOLUTION DATASET PRODUCTION ACCESS SCHEDULER HELP Cristiano Novelli DEVELOPER

ExportPlatformStatus-2

This is your solution-id, you will specify it in the Urban Datasets sending ("producer" element).

Solution

STATE: ENABLED (dropdown) ID: 2 SOLUTION NAME: Export Platform Status

SOLUTION URL (OPTIONAL):

User

USERNAME: scp.services@enea.it PASSWORD: ***** ROLE: SERVICE

NAME: SCP SURNAME: Services

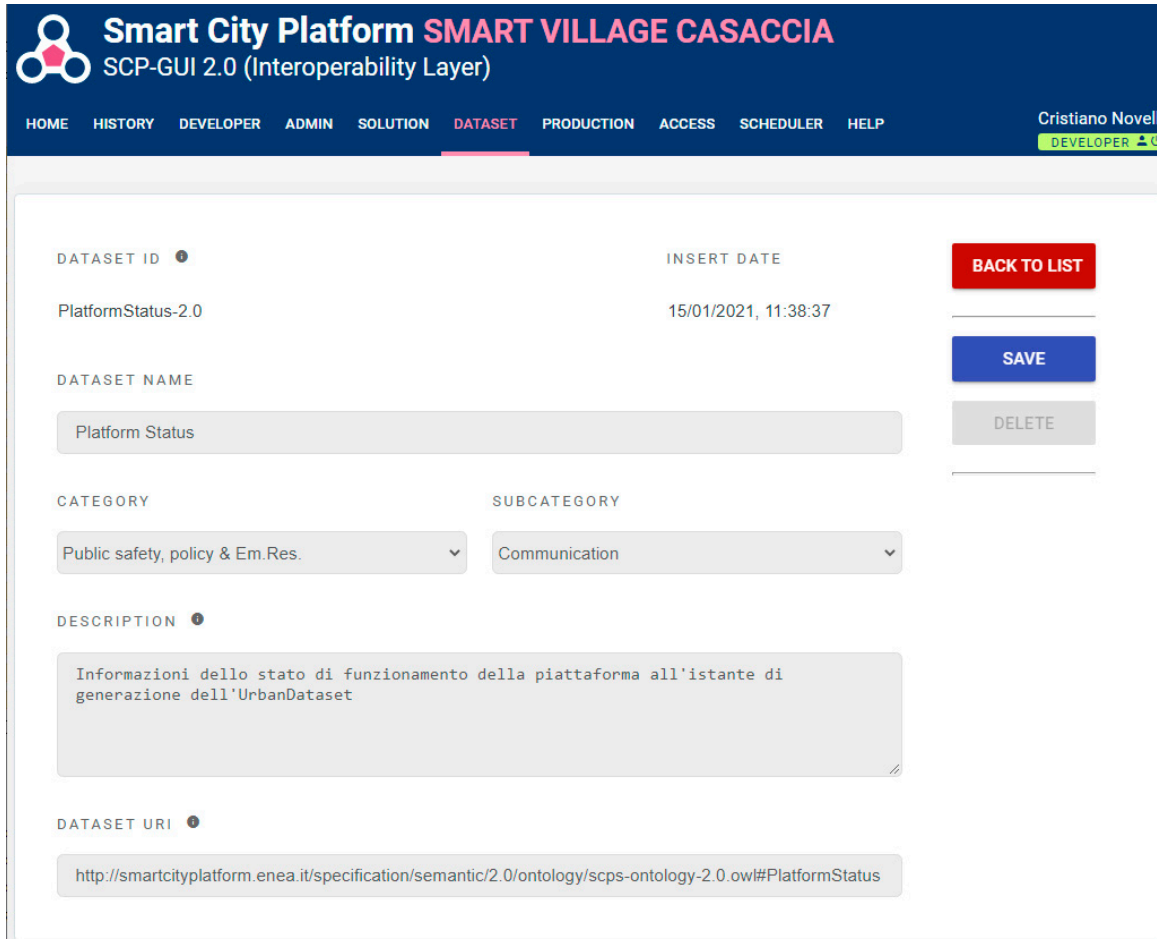
ADDRESS: ORGANIZATION: ENEA PHONE:

BACK TO LIST SAVE DELETE

Figura 3. Creazione SERVICE "Export Platform Status"

Si noti che, in rispetto della convenzione descritta nelle specifiche SCPS Collaboration 2.0 [10], alla creazione del SERVICE consegue la generazione del *solution_id* "ExportPlatformStatus-2"; tale ID verrà utilizzato nell'UD risultante in output (si veda il par. 2.2.4).

Parallelamente alla configurazione della Solution, è stato specificato l'UD "Platform Status 2.0" come supportato da questa SCP. Tale azione viene eseguita in maniera agevole tramite la funzionalità di navigazione dell'ontologia remota dalla quale vengono prese, in automatico, tutte le caratteristiche dell'UrbanDataset, richieste per completare la form (si veda par. 3.1.3).



The screenshot shows the 'Smart City Platform SMART VILLAGE CASACCIA' interface. The top navigation bar includes links for HOME, HISTORY, DEVELOPER, ADMIN, SOLUTION, DATASET (highlighted), PRODUCTION, ACCESS, SCHEDULER, and HELP. The user 'Cristiano Novelli' is logged in as a DEVELOPER. The main form is titled 'DATASET' and contains the following fields:

- DATASET ID:** PlatformStatus-2.0
- INSERT DATE:** 15/01/2021, 11:38:37
- DATASET NAME:** Platform Status
- CATEGORY:** Public safety, policy & Em.Res.
- SUBCATEGORY:** Communication
- DESCRIPTION:** Informazioni dello stato di funzionamento della piattaforma all'istante di generazione dell'UrbanDataset
- DATASET URI:** http://smartcityplatform.enea.it/specification/semantic/2.0/ontology/scps-ontology-2.0.owl#PlatformStatus

Buttons on the right side of the form include 'BACK TO LIST' (red), 'SAVE' (blue), and 'DELETE' (grey).

Figura 4. Inserimento Dataset "Platform Status 2.0"

Si noti che, in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration 2.0* [10], all'inserimento del DATASET consegue la generazione del *dataset_id* "PlatformStatus-2.0"; tale ID verrà utilizzato nell'UD risultante in output (si veda il par.2.2.4).

Una volta definito il modulo Service e specificato l'UrbanDataset come supportato, è stata definita la produzione relativa (Figura 5), ovvero il permesso per il Job/Service "Export Platform Status" di scrivere l'UD "Platform Status" nella SCP corrente (SCP-Casaccia).

La definizione della produzione richiede di specificare diversi aspetti, fondamentali per permettere alla SCP di abilitare servizi di monitoraggio, diagnostica e gestione degli UrbanDataset; tra le caratteristiche più importanti ricordiamo:

- la proprietà del dato, che in questo caso rimane di ENEA;
- il protocollo con cui verrà scritto l'UD nella SCP, in questo caso si tratta del classico metodo *push* dell'*UrbanDatasetGateway*, in rispetto delle specifiche *SCPS Communication* [11];
- la frequenza con cui l'UD verrà prodotto, qui impostata a "1 Day", una volta al giorno;

- la modalità di storage, ovvero la persistenza dell'UD, impostata in modalità "Overwrite", per fare in modo che ogni giorno l'UD generato sovrascrive il precedente (in quanto vogliamo che la SCP metta a disposizione ogni giorno il proprio stato corrente, non lo stato pregresso);
- le coordinate a cui afferisce questa produzione, che possono coincidere con quelle dell'ENEA Casaccia, ricercabile tramite il servizio google maps integrato tramite il tasto "Edit map".

The screenshot shows a configuration form for a production. Key fields include:

- ENABLED:** A dropdown menu set to 'ENABLED'.
- SOLUTION/SERVICE:** 'Export Platform Status' and **DATASET:** 'Platform Status 2.0'.
- RESOURCE ID:** 'SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000'.
- START:** '27/10/2021 00:00' and **END:** (empty).
- GRANT:** 'ENEA' and **PROPERTY:** 'ENEA'.
- Communication:** **PROTOCOL:** 'SOLUTION or SERVICE pushes UD' and **ENDPOINT:** 'https://scp-ws.portici.enea.it:8443/webservices/rest/UrbanDatasetGateway/'.
- Frequency:** 'ON DEMAND' (unchecked).
- Period:** '1' (count), 'Day' (unit), '01' (start time), '01' (end time).
- Storage:** **Way:** 'Overwrite' (dropdown), **Start:** '27/10/2021 00:00', **End:** (empty).
- Space:** **LATITUDE:** '42,0412384', **LONGITUDE:** '12,3020691', **HEIGHT:** (empty), **RADIUS (METERS):** '1081,34370894'.

Buttons for 'SAVE' (blue) and 'DELETE' (red) are located on the right side of the form.

Figura 5. Definizione Produzione "Export Platform Status" – "Platform Status 2.0"

Si noti che, in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration* [10], all'associazione SERVICE-DATASET consegue la generazione del *resource_id*

"SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000";

tale ID verrà utilizzato per salvare l'UD risultante in output (si veda il par. 2.2.4) ma non solo; tramite questo stesso *resource_id*, l'UD "Platform Status" sarà riferito univocamente dalla inter-SCP nel momento in cui dovrà esserne attuato il recupero (si veda par. 2.3.1).

2.2.3 Configurazione Job “Export” sulla SCP-Casaccia

Nei precedenti due paragrafi sono stati descritti:

- lo sviluppo del modulo software (Job) di esportazione dell’UD “Platform Status” (par. 2.2.1);
- la configurazione della produzione per permettere di salvare l’UD nella SCP (par. 2.2.2).

Descriviamo in questo paragrafo la configurazione del Job nello SCP-Scheduler.

Da notare questa differenza:

- la configurazione della produzione definisce il permesso per il Job/Service di scrivere l’UD in questa SCP con un *resource_id* univoco;
- la configurazione del Job definisce come il modulo software verrà eseguito dallo SCP-Scheduler, scrivendo in output il risultato nella SCP corrente (avendo il permesso con il *resource_id* suddetto).

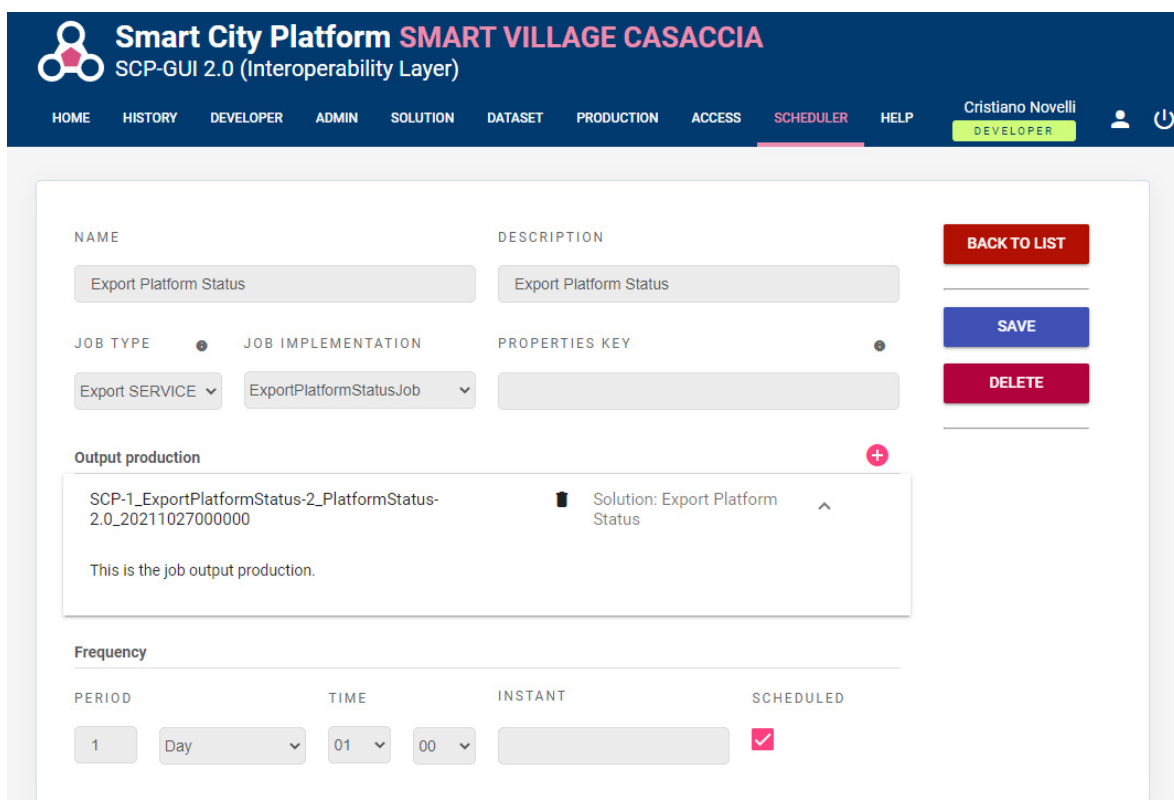


Figura 6. Configurazione Job “Export Platform Status”

Nella sezione “SCHEDULER” è possibile vedere la lista dei Job configurati, ed è possibile editarli o crearne di nuovi, nella classica modalità CRUD (Create, Read, Update, Delete) tipica di molte interfacce utente, tra cui la SCP-GUI. Nella Figura 6 viene mostrato uno screenshot relativo l’editazione del Job “Export Platform Status”, dove si evince, a parte nome e descrizione, una serie di informazioni fondamentali per l’esecuzione del Job da parte dello SCP-Scheduler:

- Job Type: una delle quattro tipologie di Job supportati:
 - **Export, per l’esportazione di dati generici;**
 - Get Remote, per recuperare UrbanDataset da endpoint remoti;
 - Push Remote, per inviare UrbanDataset verso endpoint remoti;
 - Data Fusion, per attuare attività di data fusion utilizzando UD come input e output;
nel nostro caso in oggetto è stata scelto il type “Export SERVICE”;
- Job Implementation: implementazione del Job da eseguire (in altre parole il modulo software .jar);
- Properties Key: proprietà da fornire in input al Job, in questo caso nessuna;

- Output production: premesso che il Job di tipo “Export” non ha elementi in input perché effettua la sola esportazione, viene quindi richiesto di indicare il *resource_id* di output, ovvero il permesso in produzione per questo Job/Service di salvare l’UD nella SCP corrente; tale *resource_id* è stato definito nel precedente par. 2.2.2 ed è il seguente
“SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000”;
- Frequenza con cui il Job viene eseguito, nel nostro caso una volta al giorno (“1 Day”) alle ore 01:00; ciò vuol dire che ogni notte all’una il Job/Service “Export Platform Status” esporta l’UrbanDataset “Platform Status” e lo salva nella SCP-Casaccia (per poi renderlo disponibile alla inter-SCP, così come verrà descritto nel par. 2.3);
- Checkbox “Scheduled”: opzione che indica che il Job corrente sarà eseguito dallo Scheduler alla data, ora e frequenza indicate.

N.B. Lo SCP-Scheduler è frutto di un’attività di consulenza esterna che trova completa descrizione nel report “Sviluppo SCP-Scheduler per l’esecuzione di Service nella SCP” [15].

2.2.4 Esecuzione del Job “Export” sulla SCP-Casaccia

Una volta che il Job è stato configurato può essere “scheduled” (opzione che abbiamo visto nella parte bassa della Figura 6 del paragrafo precedente) e quindi essere eseguito secondo la frequenza espressa. Per effettuare una corretta esportazione e successivo invio dell’UD all’*UrbanDatasetGateway* è necessario seguire le specifiche SCPS e quindi, tra i diversi aspetti, anche utilizzare gli ID (descritti nel par. 2.2.2) che sono stati generati definendo questa produzione:

- il *solution_id* “ExportPlatformStatus-2” deve essere indicato nell’UD nell’elemento *UrbanDataset/context/producer/id*;
- il *dataset_id* “PlatformStatus-2.0” deve essere coerente con quello indicato nell’UD nell’elemento *UrbanDataset/specification/id/value*;
- il *resource_id* “SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000” deve essere utilizzato come parametro nella chiamata al metodo push, secondo l’interfaccia definita da specifiche *SCPS Communication 2.0* [11].

Vediamo nel seguente screenshot una view della sezione “History” della SCP-GUI per monitorare l’esportazione del Job “Export Platform Status” negli ultimi 10 giorni.

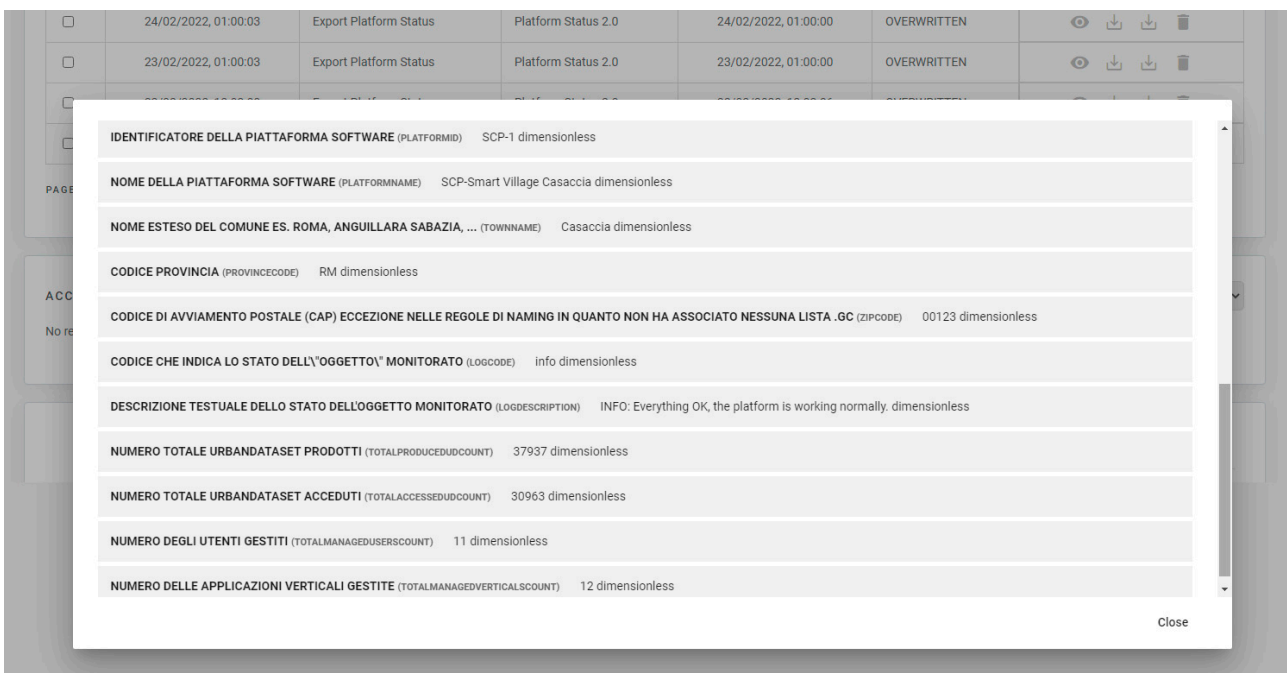
	TIMESTAMP	SOLUTION / SERVICE	URBAN DATASET	UD TIMESTAMP	STATE	
<input type="checkbox"/>	09/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	09/03/2022, 01:00:00	STORED	
<input type="checkbox"/>	08/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	08/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	07/03/2022, 01:00:15	Export Platform Status	Platform Status 2.0	07/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	06/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	06/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	05/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	05/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	04/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	04/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	03/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	03/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	02/03/2022, 01:00:03	Export Platform Status	Platform Status 2.0	02/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	01/03/2022, 01:00:06	Export Platform Status	Platform Status 2.0	01/03/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	28/02/2022, 01:00:03	Export Platform Status	Platform Status 2.0	28/02/2022, 01:00:00	OVERWRITTEN	

Figura 7. Esportazione quotidiana dell’UD “Platform Status”

Come si può notare, ogni notte alle ore 01:00:00 in punto (colonna “UD Timestamp”) l’UrbanDataset viene esportato e salvato nella SCP-Casaccia richiamando il metodo *push* dell’*UrbanDatasetGateway* (si veda a tal proposito il par. 3.3 relativo alle nuove funzionalità di validazione UrbanDataset del componente).

L’UD è scritto nella SCP in modalità “Overwrite” e quindi tutti i precedenti UD sono stati sovrascritti; sarà disponibile per la inter-SCP solo l’ultimo “Platform Status” prodotto, ovvero lo stato della piattaforma più aggiornato, che è esattamente ciò che si voleva ottenere.

Il “Platform Status” esportato presenta al suo interno le informazioni esportate, che mostriamo tramite la view della SCP-GUI.



IDENTIFICATORE DELLA PIATTAFORMA SOFTWARE (PLATFORMID)	SCP-1 dimensionless
NOME DELLA PIATTAFORMA SOFTWARE (PLATFORMNAME)	SCP-Smart Village Casaccia dimensionless
NOME ESTESO DEL COMUNE ES. ROMA, ANGUILLARA SABAZIA, ... (TOWNNAME)	Casaccia dimensionless
CODICE PROVINCIA (PROVINCECODE)	RM dimensionless
CODICE DI AVVIAMENTO POSTALE (CAP) ECCEZIONE NELLE REGOLE DI NAMING IN QUANTO NON HA ASSOCIATO NESSUNA LISTA .GC (ZIPCODE)	00123 dimensionless
CODICE CHE INDICA LO STATO DELL'OGGETTO\ MONITORATO (LOGCODE)	info dimensionless
DESCRIZIONE TESTUALE DELLO STATO DELL'OGGETTO MONITORATO (LOGDESCRIPTION)	INFO: Everything OK, the platform is working normally. dimensionless
NUMERO TOTALE URBANDATASET PRODOTTI (TOTALPRODUCEDUDCOUNT)	37937 dimensionless
NUMERO TOTALE URBANDATASET ACCEDUTI (TOTALACCESSEDUDCOUNT)	30963 dimensionless
NUMERO DEGLI UTENTI GESTITI (TOTALMANAGEDUSERSCOUNT)	11 dimensionless
NUMERO DELLE APPLICAZIONI VERTICALI GESTITE (TOTALMANAGEDVERTICALSCOUNT)	12 dimensionless

Figura 8. View dell’UD “Platform Status”

Oltre le informazioni di anagrafica e del traffico UD gestito tra la SCP e le proprie Solution verticali, l’informazione più rilevante sullo stato della piattaforma è la seguente:

LOGDESCRIPTION INFO: Everything OK, the platform is working normally.

Questo messaggio in particolare ci permetterà di monitorare tutte le SCP urbane dalla Dashboard della inter-SCP agente su scala nazionale (si veda par. 2.5).

A questo punto la SCP-Casaccia è pronta per essere connessa alla inter-SCP.

2.3 Trasporto UD da SCP-Casaccia a inter-SCP

Questo paragrafo, che riprende parzialmente il nome del caso studio pilota, descrive la parte fondamentale della comunicazione tra le due SCP, ovvero il trasporto dell’UrbanDataset “Platform Status” dalla SCP-Casaccia agente su scala urbana, alla inter-SCP agente su scala nazionale.

Questa azione, orchestrata ed effettuata dalla inter-SCP, prevede il recupero remoto dell’UD utilizzando la comunicazione interoperabile implementata con il web service *UrbanDatasetGateway*, basata su specifiche *SCPS Communication 2.0* [11], e il successivo salvataggio dell’UD in locale, utilizzando lo stesso sistema di comunicazione.

Questa fase, dunque, prevede 3 task da attuare nella inter-SCP che elenchiamo di seguito:

- 1 Configurazione Produzione (per salvare l'UD nella inter-SCP);
- 2 Configurazione Job nello SCP-Scheduler (per recuperare l'UD dalla SCP-Casaccia);
- 3 Esecuzione del Job.

Descriviamo nel dettaglio questi 3 task.

2.3.1 Configurazione Produzione "Platform Status" (singolo) sulla inter-SCP

L'azione di recupero e salvataggio dell'UD "Platform Status" dalla SCP-Casaccia alla inter-SCP, per concretizzarsi, deve utilizzare un permesso di produzione dell'UD nella inter-SCP, poiché qui verrà scritto; è necessario quindi configurare la relativa produzione.

È stato quindi necessario configurare tramite la SCP-GUI:

- un utente di sistema dedicato alle SOLUTION generiche monitorate: *scp.solutions@enea.it* e associare ad esso la **Solution "SCP Casaccia"** (si veda la Figura 9);
- l'UD "Platform Status" versione 2.0 come supportato in questa SCP;
- la produzione come associazione Solution-UD (si veda Figura 10).

The screenshot shows the 'Smart City Platform INTER-SCP' interface. The header includes the logo and title 'SCP-GUI 2.0 (Interoperability Layer)'. A navigation menu contains 'HOME', 'HISTORY', 'DEVELOPER', 'ADMIN', 'SOLUTION' (highlighted), 'DATASET', 'PRODUCTION', 'ACCESS', 'SCHEDULER', and 'HELP'. The user 'Cristiano Novelli' is logged in as a 'DEVELOPER'. The main content area is titled 'SCPCasaccia-1' and contains the following form fields:

- Solution ID:** 1
- Solution Name:** SCP Casaccia
- State:** ENABLED
- Solution URL (Optional):** https://smartcityplatform.enea.it/casaccia/
- User:**
 - Username:** scp.solutions@enea.it
 - Password:** [masked]
 - Role:** SOLUTION
 - Name:** SCP
 - Surname:** Solutions
 - Organization:** ENEA

Buttons for 'BACK TO LIST', 'SAVE', and 'DELETE' are visible on the right side of the form.

Figura 9. Creazione SOLUTION "SCP Casaccia"

Dal punto di vista della inter-SCP, le SCP urbane monitorate sono Solution verticali.

Si noti che avremmo potuto specificare, come utente associato alla SCP/Solution, l'indirizzo email di un reale amministratore della SCP-Casaccia ma, essendo esso coincidente con l'amministratore della inter-SCP, si è preferito creare un'utenza apposita a cui associare le diverse SCP monitorate nella sperimentazione.

Si noti, inoltre, che in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration 2.0* [10], alla creazione del SERVICE consegue la generazione del *solution_id* "SCP-Casaccia-1"; tale ID verrà utilizzato nell'UD salvato sulla inter-SCP (si veda il par. 2.3.3).

Parallelamente alla configurazione della SCP/Solution nella inter-SCP, è stato **specificato l'UD "Platform Status 2.0" come supportato nella inter-SCP** (nello stesso identico modo in cui, abbiamo visto, è stato configurato nella SCP-Casaccia (si veda par. 2.2.2 e Figura 3). Anche in questo caso,

- l'azione viene eseguita in maniera agevole tramite la funzionalità di navigazione dell'ontologia remota dalla quale vengono prese, in automatico, tutte le caratteristiche dell'UrbanDataset, richieste per completare la form (si veda par. 3.1.3);
- in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration 2.0* [10], all'inserimento del DATASET consegue la generazione del *dataset_id* "PlatformStatus-2.0";
- tale ID verrà utilizzato nell'UD salvato sulla inter-SCP (si veda il par. 2.3.3).

Una volta configurata la SCP/Solution nella inter-SCP e specificato l'UrbanDataset come supportato, è stata definita la produzione relativa ovvero **il permesso per la Solution "SCP-Casaccia" di scrivere l'UD "Platform Status" nella inter-SCP** (Figura 10).

SOLUTION/SERVICE		DATASET	
SCP Casaccia		Platform Status 2.0	
RESOURCE ID			
SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000			
START	END		•
05/11/2021 00:00			
GRANT	PROPERTY		•
<input type="checkbox"/> IS OPENDATA?	ENEA	ENEA	
Communication			
PROTOCOL		ENDPOINT	
SERVICE pushes UD from SOLUTION		https://scp-ws.portici.enea.it:8443/webservices/rest/UrbanDatasetGateway/	
USERNAME	PASSWORD	RESOURCE EXTERNAL REFERENCE	
scp.admin@enea.it	*****	SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_202110270	
Frequency			<input type="checkbox"/> ON DEMAND •
PERIOD	TIME	INSTANT	
1 Day	01 01	05/11/2021	
Storage			
WAY	START	END	
Append	05/11/2021 00:00		
Space			
LATITUDE	LONGITUDE	HEIGHT	RADIUS (METERS)
42,0414535	12,3020906		1500
Edit map			

Figura 10. Definizione Produzione "SCP-Casaccia" – "Platform Status 2.0"

La **definizione della produzione** richiede di specificare diversi aspetti, fondamentali per permettere alla inter-SCP di abilitare servizi di monitoraggio, diagnostica e gestione degli UrbanDataset; tra le caratteristiche più importanti ricordiamo:

- la proprietà del dato, che anche in questo caso rimane di ENEA;
- il protocollo con cui verrà scritto l'UD nella inter-SCP, in questo caso è stato configurato il protocollo "SERVICE pushes UD from SOLUTION" che prevede sia un **SERVICE (BRIDGE) ad effettuare il recupero dell'UD da un endpoint remoto** per poi scriverlo localmente nella inter-SCP utilizzando il metodo *push* dell'*UrbanDatasetGateway*, in rispetto delle specifiche SCPS 2.0 [4];
- poiché è stato scelto il suddetto protocollo con BRIDGE, la form richiede di inserire anche i dati necessari per chiamare il metodo di *request* dell'*UrbanDatasetGateway*, in rispetto delle specifiche *SCPS Communication 2.0* [11], ovvero:
 - o *endpoint* della Solution remota (SCP-Casaccia);
 - o *username* e *password* che la inter-SCP dovrà utilizzare per essere autenticata presso la SCP-Casaccia (si noti che in ogni SCP è stato configurato un account *scp.admin@enea.it* per finalità di questo tipo);
 - o *resource external reference*: ovvero il *resource_id* con cui la inter-SCP può identificare univocamente l'UD "Platform Status" esportato dalla SCP-Casaccia;
- la frequenza con cui l'UD verrà recuperato dal bridge, qui impostata a "1 Day", una volta al giorno alle ore 01:01 (si ricorda che l'esportazione è stata configurata alle ore 01:00);
- la modalità di storage, ovvero la persistenza dell'UD, impostata in modalità "Append", per fare in modo che ogni giorno l'UD generato venga salvato in coda al precedente (in quanto vogliamo mantenere in memoria uno storico della comunicazione con la SCP e del suo stato nel tempo); si noti che l'esportazione nella SCP-Casaccia è in modalità "Overwrite" perché in quel caso non si è ritenuto importante mantenere uno storico degli stati della piattaforma;
- le coordinate a cui afferisce questa produzione, che possono coincidere con quelle dell'ENEA Casaccia, ricercabile tramite il servizio google maps integrato tramite il tasto "Edit map".

Si noti che, in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration 2.0* [10], all'associazione SERVICE-DATASET consegue la generazione del *resource_id*

"SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000";

tale ID verrà utilizzato per salvare l'UD risultante in output (si veda il par. 2.3.3) ma non solo; tramite questo stesso *resource_id*, l'UD "Platform Status" sarà riferito univocamente dal modulo Job/Service di DataFusion nel momento in cui dovrà esserne attuato il recupero (si veda par. 2.4).

Si noti, inoltre, che anche se i due UrbanDataset (quello esportato nella SCP-Casaccia e quello recuperato e salvato nella inter-SCP) sono sostanzialmente lo stesso UD, i *resource_id* sono diversi in quanto si riferiscono a due SCP diverse nell'ambito di due produzioni diverse, che possono avere diverse caratteristiche di gestione (da quelle più tecniche relative al tipo di storage a quelle più legali relativamente la validità temporale e la proprietà del dato).

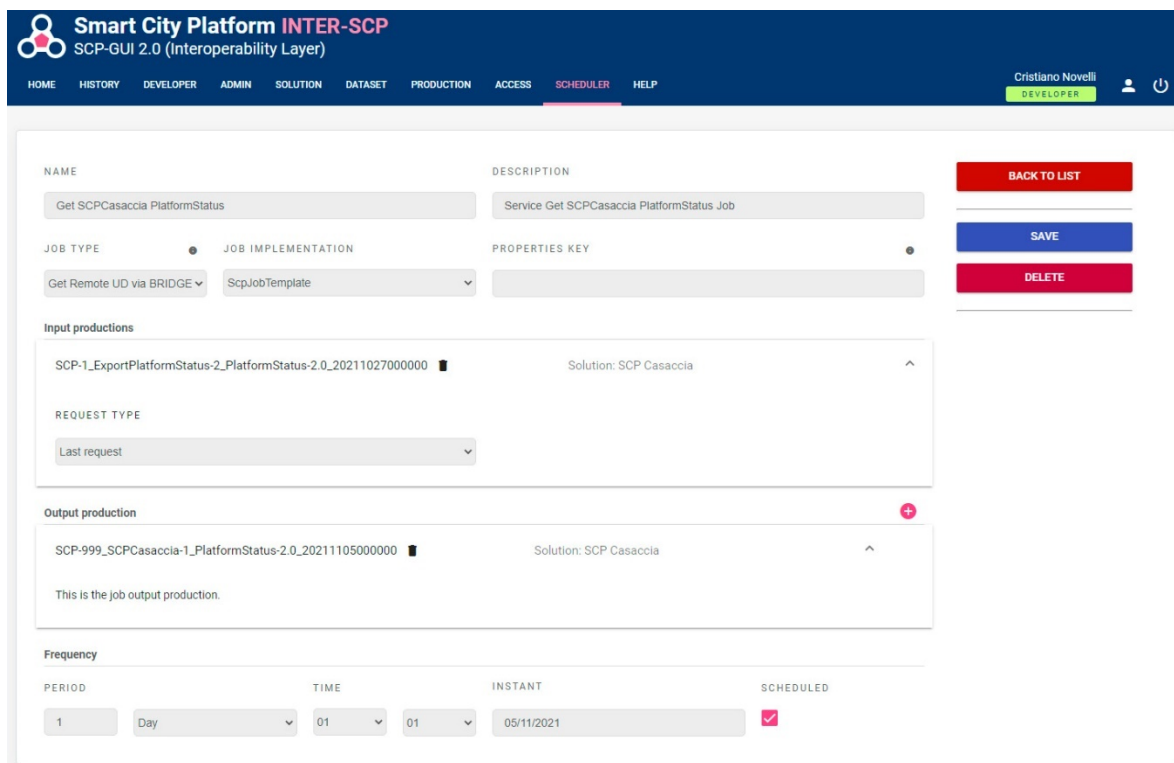
2.3.2 Configurazione Job “Get Remote” sulla inter-SCP

Nel precedente paragrafo è stata descritta la configurazione della produzione per permettere di salvare l’UD recuperato dalla SCP-Casaccia nella inter-SCP (par. 2.3.1).

Descriviamo in questo paragrafo la configurazione del Job “Get Remote” nello SCP-Scheduler.

Ricordiamo questa differenza:

- la configurazione della produzione definisce il permesso per il Job/Service di scrivere l’UD in questa inter-SCP con un *resource_id* univoco;
- la configurazione del Job definisce come il modulo software verrà eseguito dallo SCP-Scheduler, scrivendo in output il risultato nella inter-SCP (avendo il permesso con il *resource_id* suddetto).



The screenshot shows the configuration page for a job in the SCP-Scheduler. The job name is 'Get SCPCasaccia PlatformStatus' and its description is 'Service Get SCPCasaccia PlatformStatus Job'. The job type is 'Get Remote UD via BRIDGE' and the job implementation is 'ScpJobTemplate'. The input production is 'SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000' with the solution 'SCP Casaccia'. The output production is 'SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000' with the solution 'SCP Casaccia'. The frequency is set to '1' per 'Day' at '01:01' on '05/11/2021'.

Figura 11. Configurazione Job “Get SCPCasaccia PlatformStatus”

Nella Figura 11 viene mostrato uno screenshot relativo l’editazione del Job “Get SCPCasaccia PlatformStatus”, dove si evince, a parte nome e descrizione, una serie di informazioni fondamentali per l’esecuzione del Job da parte dello SCP-Scheduler:

- Job Type: una delle quattro tipologie di Job supportati:
 - Export, per l’esportazione di dati generici;
 - **Get Remote, per recuperare UrbanDataset da endpoint remoti;**
 - Push Remote, per inviare UrbanDataset verso endpoint remoti;
 - Data Fusion, per attuare attività di data fusion utilizzando UD come input e output;
nel nostro caso in oggetto è stata scelto il type “Get Remote UD via BRIDGE”;
- Job Implementation: implementazione del Job da eseguire, poiché il “Get Remote Job” è integrato nello SCP-Scheduler, viene selezionato lo “SCPJobTemplate” che contiene le funzionalità per il recupero remoto dell’UD;
- Properties Key: proprietà da fornire in input al Job, in questo caso nessuna;
- Input production: il Job di tipo “Get Remote” ha come elemento in input il *resource_id* dell’UD da recuperare ovvero il “Platform Status” che si trova sulla SCP-Casaccia e che è il seguente: “SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000”;

si noti che, trattandosi di un metodo di recupero di UD, viene anche richiesto il metodo dell'UrbanDatasetGateway con cui effettuare la chiamata di request secondo le specifiche SCP: in questo caso si è scelto il metodo *lastRequest* per recuperare l'ultimo UD prodotto con quel *resource_id* e sappiamo che, essendo l'UD sulla SCP-Casaccia prodotto in modalità "Overwrite", ce ne sarà sempre disponibile uno solo;

- Output production: il *resource_id* di output, invece, corrisponde al permesso in produzione per questo Job/Service di salvare l'UD nella inter-SCP corrente; tale *resource_id* è stato definito nel precedente par. 2.3.1 ed è il seguente
"SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000";
- Frequenza con cui il Job viene eseguito, nel nostro caso una volta al giorno ("1 Day") alle ore 01:01; ciò vuol dire che ogni notte,
 - o all'una in punto, il Job/Service "Export Platform Status" esporta l'UrbanDataset "Platform Status" e lo salva nella SCP-Casaccia e, subito dopo,
 - o all'una e un minuto, il Job/Service "Get SCPCasaccia PlatformStatus" recupera l'UrbanDataset "Platform Status" e lo salva nella inter-SCP;
- Checkbox "Scheduled": opzione che indica che il Job corrente sarà eseguito dallo Scheduler alla data, ora e frequenza indicate.

N.B. Lo SCP-Scheduler è frutto di un'attività di consulenza esterna che trova completa descrizione nel report "Sviluppo SCP-Scheduler per l'esecuzione di Service nella SCP" [15].

2.3.3 Esecuzione del Job "Get Remote" sulla inter-SCP

Una volta che il Job è stato configurato può essere "scheduled" (opzione che abbiamo visto nella parte bassa della Figura 11 del paragrafo precedente) e quindi essere eseguito secondo la frequenza espressa.

Per effettuare un corretto recupero e successiva importazione dell'UD tramite l'UrbanDatasetGateway, è necessario seguire le specifiche SCPS e quindi, tra i diversi aspetti, anche utilizzare gli ID (descritti nel par. 2.3.1) che sono stati generati definendo questa produzione:

- il *solution_id* "SCPCasaccia-1" deve essere indicato nell'UD nell'elemento *UrbanDataset/context/producer/id* in fase di salvataggio dell'UD;
- il *dataset_id* "PlatformStatus-2.0" deve essere coerente con quello indicato nell'UD nell'elemento *UrbanDataset/specification/id/value*;
- il *resource_id* "SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000" deve essere utilizzato come parametro nella chiamata al metodo push, secondo l'interfaccia definita da specifiche SCPS *Communication 2.0* [11].









































Vediamo nel seguente screenshot una view della sezione "History" della SCP-GUI per monitorare il recupero effettuato dal Job "Get SCPCasaccia PlatformStatus" nei 10 giorni precedenti (al momento in cui è stato effettuato lo screenshot).

FILTERS

STORED ONLY SHOW SOLUTIONS SOLUTION SERVICE SCP Casaccia DATASET Platform Status 2.0 START [] END []

SHOW SERVICES

PRODUCTION # PER PAGE 10 Actions on Selected UD's

	TIMESTAMP	SOLUTION / SERVICE	URBAN DATASET	UD TIMESTAMP	STATE	
<input type="checkbox"/>	25/02/2022, 01:01:02	SCP Casaccia	Platform Status 2.0	25/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	24/02/2022, 01:01:02	SCP Casaccia	Platform Status 2.0	24/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	23/02/2022, 01:01:02	SCP Casaccia	Platform Status 2.0	23/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	22/02/2022, 12:02:33	SCP Casaccia	Platform Status 2.0	22/02/2022, 12:02:06	STORED	   
<input type="checkbox"/>	21/02/2022, 01:01:03	SCP Casaccia	Platform Status 2.0	21/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	20/02/2022, 01:01:02	SCP Casaccia	Platform Status 2.0	20/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	19/02/2022, 01:01:02	SCP Casaccia	Platform Status 2.0	19/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	18/02/2022, 01:01:05	SCP Casaccia	Platform Status 2.0	18/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	17/02/2022, 01:01:04	SCP Casaccia	Platform Status 2.0	17/02/2022, 01:00:00	STORED	   
<input type="checkbox"/>	16/02/2022, 01:01:03	SCP Casaccia	Platform Status 2.0	16/02/2022, 01:00:00	STORED	   

PAGE 1 OF 11 (107 ELEMENTS) |< < 1 2 3 4 5 > >|

Figura 12. Produzione quotidiana UD da SCP-Casaccia a inter-SCP

Come si può notare, ogni notte alle ore 01:01 (colonna "Timestamp") l'UrbanDataset generato alle ore 01:00 (colonna "UD Timestamp") viene recuperato dalla SCP-Casaccia e salvato nella inter-SCP richiamando il metodo *push* dell'*UrbanDatasetGateway* (si veda a tal proposito il par. 3.3 relativo alle nuove funzionalità di validazione UrbanDataset del componente).

L'UD è scritto nella SCP in modalità "Append" e quindi tutti gli UD sono disponibili per poter verificare se, nel tempo, ci sono stati malfunzionamenti o per definire servizi di monitoraggio e confronto delle SCP urbane che siano esprimibili con una funzione temporale.

Fino ad ora è stato utilizzato un UrbanDataset con informazioni di stato della piattaforma urbana connessa, si pensi a un caso equivalente dove siano disponibili dati di tipo energetico che abiliterebbero un confronto tra città sul tema energetico.

2.4 Replicabilità del caso studio pilota

Abbiamo raggiunto, a questo punto, un primo importante obiettivo, ovvero permettere il trasferimento di un UrbanDataset da una SCP urbana quale è la SCP-Casaccia alla inter-SCP agente su scala nazionale; non solo, questo recupero è stato configurato ed automatizzato per avvenire regolarmente una volta al giorno, e abbiamo testato attraverso la SCP-GUI come questo avvenga con successo.

Il prossimo passo, seguendo le fasi del caso studio pilota, potrebbe essere il recupero dell'UD "Platform Status" dalla Dashboard della inter-SCP per visualizzarne le informazioni ma non è il nostro obiettivo; l'obiettivo del caso studio pilota è definire un metodo generalizzabile e replicabile a tutte le SCP urbane, in modo tale che la inter-SCP divenga piattaforma centrale agente su scala nazionale per monitorare un numero illimitato di SCP urbane.

Prima di procedere con le altre fasi del caso studio pilota, si è voluto dunque **replicare l'approccio fin qui descritto ad altre SCP urbane, per ottenere da ognuna di esse il "Platform Status" descrittivo dello stato corrente della piattaforma.** In questo modo abbiamo potuto:

- **verificare** che la prima parte della sperimentazione fin qui descritta fosse efficace anche su altre piattaforme urbane e quindi **che l'approccio fosse replicabile;**
- **preparare un set di dati più interessanti** in vista del completamento del caso studio pilota, **avendo più di una SCP urbana connessa alla inter-SCP.**

La prima SCP su cui è stato implementato l'approccio è la **SCP-Template**, ovvero il prototipo di SCP che viene utilizzato come matrice per istanziare altre SCP: si tratta di una SCP a tutti gli effetti, funzionante, su cui vengono trasferite tutte le innovazioni del prototipo dopo averle accuratamente testate.

The screenshot shows the 'Smart City Platform TEMPLATE' interface. The top navigation bar includes 'HOME', 'HISTORY', 'DEVELOPER', 'ADMIN', 'SOLUTION', 'DATASET', 'PRODUCTION', 'ACCESS', 'SCHEDULER', and 'HELP'. The user 'Cristiano Novelli' is logged in as a 'DEVELOPER'. The main content area is titled 'The HISTORY section allows the users to surf the UrbanDatasets (UD) managed by the Smart City Platform (SCP). Production and Access tables report, respectively, the UD's sent by the prod Solutions to the SCP and the UD's retrieved by the consumer Solutions from the SCP. Filters help the user to focus on the different aspects. At the bottom, a graphical schema provides a view UD's managed long the time by the SCP.'

Below the text, there are filter controls: 'STORED ONLY' (checkbox), 'SHOW SOLUTIONS' (checkbox), 'SHOW SERVICES' (checkbox), 'SOLUTION SERVICE' (checkbox), 'Export Platform Status' (dropdown), 'DATASET' (dropdown), 'START' (input), and 'END' (input).

The 'PRODUCTION' table is displayed with the following columns: 'TIMESTAMP', 'SOLUTION / SERVICE', 'URBAN DATASET', 'UD TIMESTAMP', 'STATE', and 'Actions on Selected'. The table contains 10 rows of data, all for 'Export Platform Status' on the 'Platform Status 2.0' dataset, with states ranging from 'STORED' to 'OVERWRITTEN'.

	TIMESTAMP	SOLUTION / SERVICE	URBAN DATASET	UD TIMESTAMP	STATE	Actions on Selected
<input type="checkbox"/>	01/03/2022, 01:00:06	Export Platform Status	Platform Status 2.0	01/03/2022, 01:00:00	STORED	
<input type="checkbox"/>	28/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	28/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	27/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	27/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	26/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	26/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	25/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	25/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	24/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	24/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	23/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	23/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	22/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	22/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	21/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	21/02/2022, 01:00:00	OVERWRITTEN	
<input type="checkbox"/>	20/02/2022, 01:00:02	Export Platform Status	Platform Status 2.0	20/02/2022, 01:00:00	OVERWRITTEN	

Figura 13. Produzione quotidiana UD sulla SCP-Template

Nella precedente Figura 13 viene riportata la lista degli ultimi 10 UrbanDataset esportati nella SCP-Template dal Job "Export Platform Status" e, come si può notare, la creazione dell'UD "Platform Status" risulta puntuale e affidabile così come lo era stato nella SCP-Casaccia.

Allo stesso modo, anche il recupero della inter-SCP dell'UD "Platform Status" è stato configurato, avviato e testato, ottenendo dei risultati ottimali.

Si effettuerà una valutazione della sperimentazione nel paragrafo 2.7, dove si analizzeranno una serie di indicatori tra cui anche quelli relativi all'esportazione e al recupero dell'UD sulla inter-SCP.

Si noti che per effettuare la **replica dell'approccio**, con l'obiettivo di connettersi alla inter-SCP:

- la SCP-Template è stata aggiornata all'ultima versione della SCP-GUI 2.0 contenente, oltre le ultime innovazioni, anche lo SCP-Scheduler;
- è stato copiato il Job "Export Platform Status", senza modificarlo, dalla SCP-Casaccia;
- sono state attuate le configurazioni, così come descritte nei precedenti paragrafi, per permettere alla SCP-Template di produrre il "Platform Status" e alla inter-SCP di recuperarlo;
- l'utilizzo di una definizione condivisa di formato dati (l'UD "Platform Status") e interfaccia di comunicazione (il web service "UrbanDatasetGateway"), ha reso l'operazione agevole e azzerato gli errori che comunemente avvengono nell'implementare uno scambio dati tra due sistemi;
- essendo la SCP-Template, come detto, la matrice da cui vengono istanziate tutte le nuove SCP, **automaticamente ogni nuova SCP eredita la potenzialità di connettersi alla inter-SCP.**

Dopo la SCP-Template, il metodo è stato replicato ulteriormente, in questo modo **sono state connesse alla inter-SCP un totale di 5 SmartCityPlatform** esistenti (si noti in Figura 14 il caso studio pilota esteso).

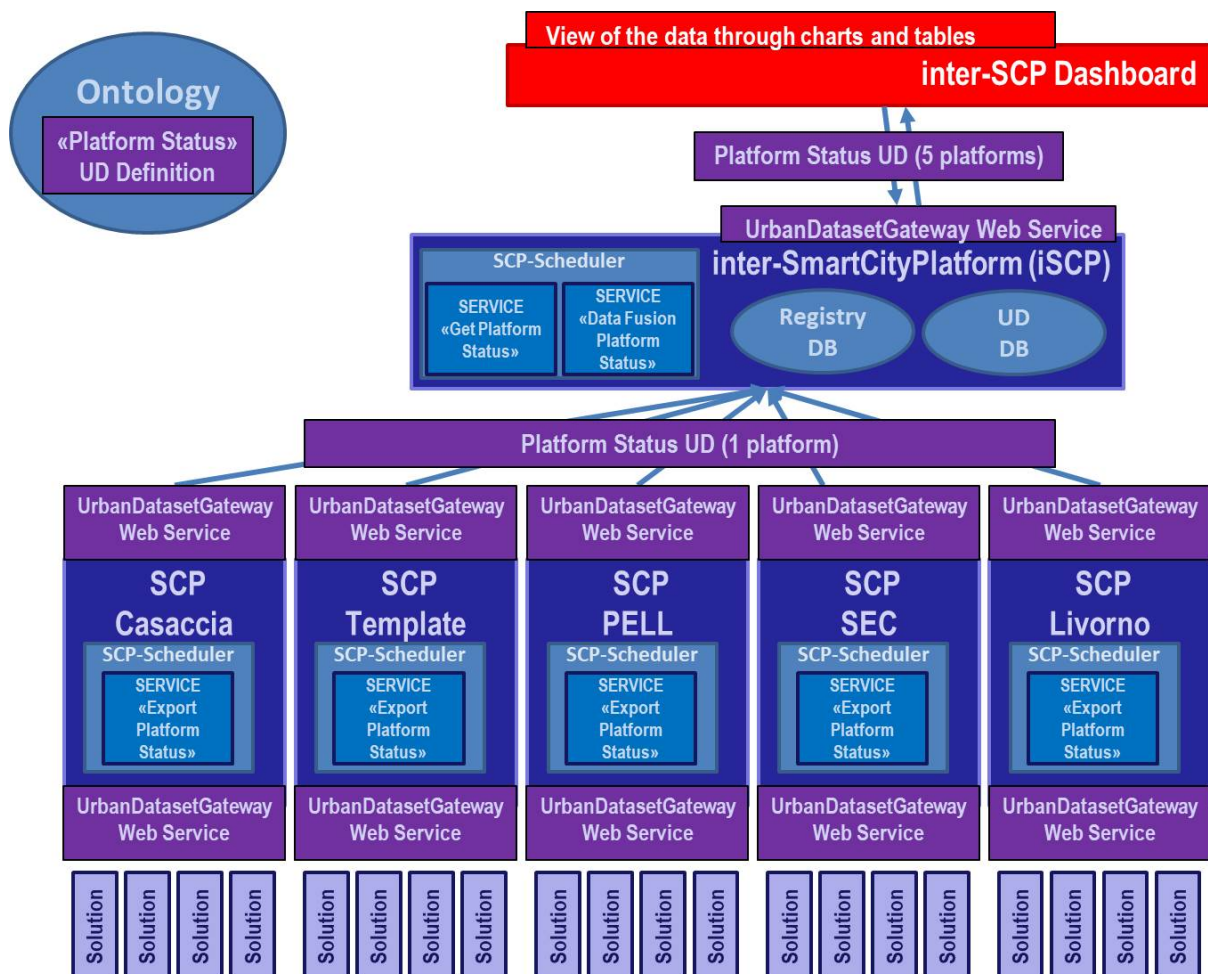


Figura 14. Caso studio pilota replicato ad altre 4 SCP

Oltre a SCP-Casaccia e SCP-Template, le altre SCP coinvolte sono le seguenti:

- SCP-PELL: integrata nella piattaforma PELL (si veda report [2]) che in questo modo permette un primo e reale scambio di dati tra le piattaforme agenti su scala nazionale PELL e inter-SCP;
- SCP-Livorno: sperimentazione nell’ambito del progetto ES-PA, a Livorno;
- SCP-SEC: sperimentazione per le Smart/Local Energy Communities a Roma (report [19]).

Ognuna di queste SCP è stata configurata come la SCP-Casaccia per esportare l’UrbanDataset “Platform Status” ogni notte alle ore 01:00:00; la inter-SCP, invece, è stata configurata per recuperare questi UD qualche minuto dopo, in modo tale da ottenere ogni notte lo stato di ogni piattaforma connessa. Queste configurazioni si traducono in altrettante produzioni e altrettanti job configurati nello scheduler, così come mostrato rispettivamente in Figura 15 e Figura 16.











STATE	SOLUTION / SERVICE	DATASET	OWNER	RESOURCE ID	START / END	
ENABLED	SCP SEC	Platform Status 2.0	ENEA	SCP-999_SCPSEC-8_PlatformStatus-2.0_20220217000000	17/02/2022, 00:00:00 - ENDLESS	 
ENABLED	SCP PELL	Platform Status 2.0	ENEA	SCP-999_SCPPELL-2_PlatformStatus-2.0_20211216000000	16/12/2021, 00:00:00 - ENDLESS	 
ENABLED	SCP Template	Platform Status 2.0	ENEA	SCP-999_SCPTemplate-99_PlatformStatus-2.0_20211105000000	05/11/2021, 00:00:00 - ENDLESS	 
ENABLED	SCP Casaccia	Platform Status 2.0	ENEA	SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000	05/11/2021, 00:00:00 - ENDLESS	 
ENABLED	SCP Livorno	Platform Status 2.0	Livorno	SCP-999_SCPLivorno-9_PlatformStatus-2.0_20211103000000	03/11/2021, 00:00:00 - 13/10/2050, 12:00:00	 

Figura 15. Produzioni definite per le 5 SCP connesse alla inter-SCP

























NAME	DESCRIPTION	JOB TYPE	
DataFusion PlatformStatus	Service DataFusion PlatformStatus Job	Data Fusion SERVICE	   
Get SCPCasaccia PlatformStatus	Service Get SCPCasaccia PlatformStatus Job	Get Remote UD via BRIDGE	   
Get SCPLivorno PlatformStatus	Service Get SCPLivorno PlatformStatus Job	Get Remote UD via BRIDGE	   
Get SCPELL PlatformStatus	Service Get SCPELL PlatformStatus Job	Get Remote UD via BRIDGE	   
Get SCPSEC PlatformStatus	Service Get SCPSEC PlatformStatus Job	Get Remote UD via BRIDGE	   
Get SCPTemplate PlatformStatus	Service Get SCPTemplate PlatformStatus Job	Get Remote UD via BRIDGE	   

Figura 16. Job/Service configurati per le 5 SCP connesse alla inter-SCP

Ottenuto questo risultato, possiamo descrivere la seconda parte della sperimentazione, ovvero:

- l’azione di Data Fusion;
- la visualizzazione finale nella Dashboard della inter-SCP.

2.5 Data Fusion PlatformStatus

A questo punto il dataset “Platform Status” viene recuperato quotidianamente da diverse SCP e salvato nella inter-SCP nel proprio database degli UrbanDataset.

Questo UrbanDataset potrebbe essere recuperato direttamente dalla Dashboard della inter-SCP ma l’obiettivo che ci si è posti è il monitoraggio di tutte le SCP urbane connesse alla inter-SCP agente su scala nazionale, quindi è necessaria una ulteriore azione, per “fondere” assieme tutti gli N “Platform Status” recuperati dalle N SCP connesse alla inter-SCP in un unico set di dati (che sarà l’input della Dashboard).

Tale azione di “fusione” è stata implementata con il Job “Data Fusion Platform Status” che, come gli altri Job, è stato configurato ed eseguito dallo SCP-Scheduler.

Questa fase prevede 5 task da attuare nella inter-SCP che elenchiamo di seguito:

- 1 Sviluppo software del Job di datafusion;
- 2 Configurazione della produzione nella iSCP (e quindi del SERVICE, dell’UD e della loro associazione);
- 3 Configurazione del Job nello scheduler della iSCP;
- 4 Esecuzione del Job;
- 5 Verifica del risultato.

Descriviamo nel dettaglio questi 5 task.

2.5.1 Sviluppo software Job di Data Fusion

L’azione di Data Fusion nella inter-SCP è un’azione effettuata da un Job/SERVICE che prevede di prendere in input gli N UD “Platform Status” (che nei precedenti paragrafi abbiamo visto sono recuperati da N SCP urbane) e raggruppare le informazioni in output in un singolo UD “Platform Status” che sarà utilizzato infine dalla dashboard della inter-SCP (si veda par. 2.6).

Il modulo software è stato denominato “Data Fusion Platform Status” e prevede questo algoritmo:

- recupero e lettura degli N UD “Platform Status”;
- per ogni UD prendo la singola linea presente che riporta le informazioni sullo stato della piattaforma che lo ha generato;
- aggiorno le coordinate della linea in oggetto con quelle che sono state associate alla produzione dell’UD da cui è stata presa; questa informazione sarà indispensabile per la georeferenziazione della piattaforma su un servizio di tipo mappa (in particolare “google maps”);
- aggiorno il timestamp e timezone della linea in oggetto con il timestamp e timezone di generazione dell’UD da cui è stata presa; questa informazione mi permette di sapere il timestamp di generazione di ogni linea e quindi di ogni stato di piattaforma una volta che saranno assemblate in un unico UD;
- inserisco la linea nell’UD di output che è un “Platform Status” predisposto a questo scopo;
- aggiorno timestamp e timezone di generazione dell’UD in output con il timestamp corrente;
- consegno allo scheduler l’UD “Platform Status” in output.

Vedremo successivamente come lo scheduler salverà il file di output (par. 2.5.4).

Si noti che questo sviluppo è stato fatto per “fondere” i dati provenienti da N SCP anche se, inizialmente, era disponibile solo la SCP-Casaccia, così come descritto nel caso studio pilota. Essendo le SCP identiche, è stato possibile aggiungere le nuove SCP connesse alla inter-SCP nel processo di DataFusion senza intervenire sul codice del modulo software ma solamente in fase di configurazione (si veda a tal proposito i par. 2.5.2 e par. 2.5.3) garantendo anche in questa fase la replicabilità del caso studio pilota (precedentemente discussa nel par. 2.4).

2.5.2 Configurazione Produzione “Platform Status” (multiplo) sulla inter-SCP

L’azione di datafusion dell’UD “Platform Status”, contenente gli stati delle N piattaforme connesse alla inter-SCP, deve utilizzare un permesso di produzione dell’UD nella inter-SCP, poiché qui verrà scritto dal Job/SERVICE della cui implementazione abbiamo parlato nel precedente par. 2.5.1); è necessario quindi configurare la relativa produzione.

È stato quindi necessario configurare tramite la SCP-GUI della inter-SCP:

- un utente di sistema dedicato ai SERVICE generici: *scp.services@enea.it* e associare ad esso il **Service “Data Fusion Platform Status”** (si veda la Figura 3);
- l’UD “Platform Status” versione 2.0 come supportato nella inter-SCP, operazione che è già stata effettuata quando è stata configurato il primo Job “Get Remote”, nel par. 2.3.2, e che non deve essere più eseguita per qualsiasi produzione utilizzi questo UD;
- la produzione come associazione Service-UD (si veda Figura 5).

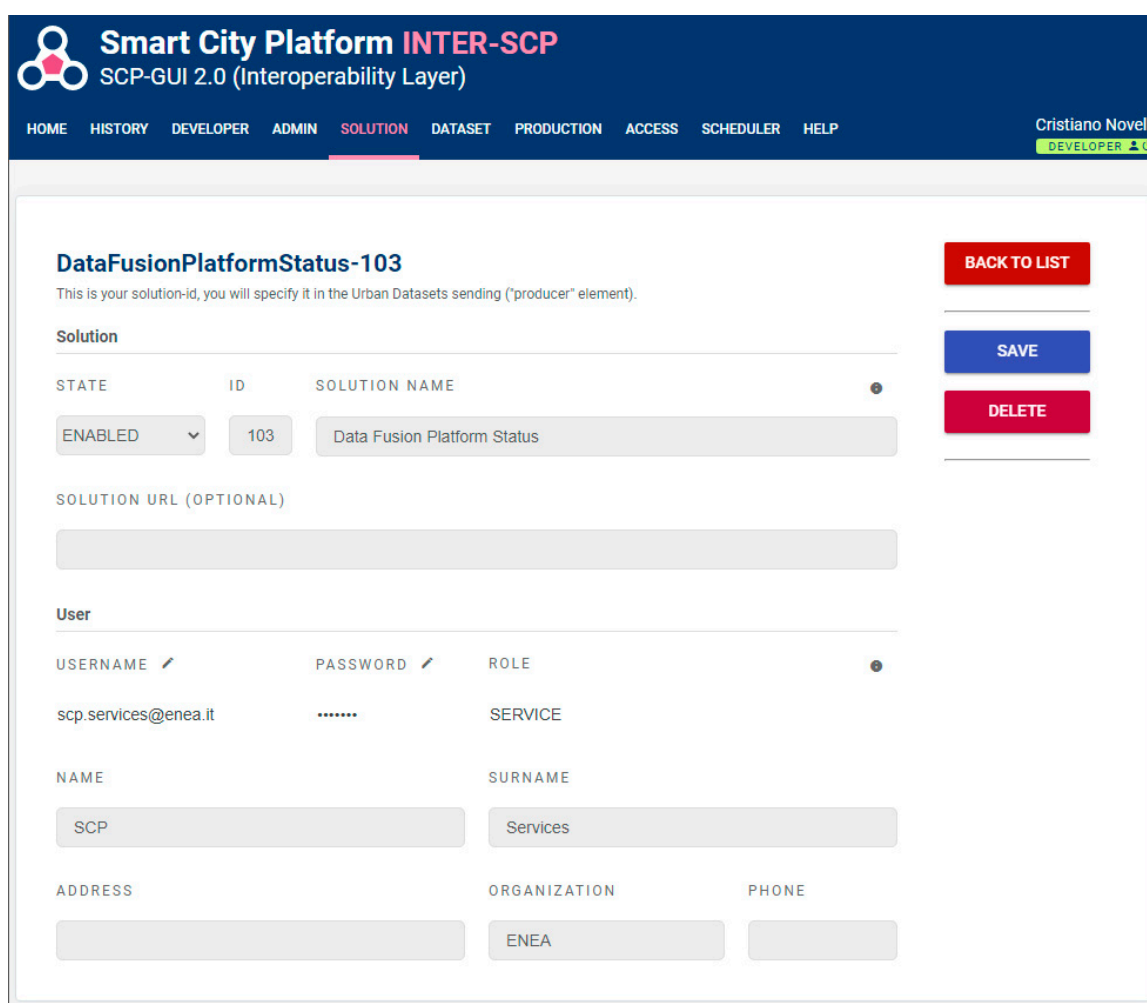


Figura 17. Creazione SERVICE "Data Fusion Platform Status"

Dal punto di vista della inter-SCP, i Job/Service sono monitorati in modo identico alle Solution verticali ma con apposita vista (si veda par. 3.1).

Si noti che avremmo potuto specificare, come utente associato al Job/Service, un indirizzo email di un reale amministratore della inter-SCP ma si è preferito creare un’utenza apposita a cui associare i diversi job/service monitorati nella sperimentazione.

Si noti, inoltre, che in rispetto della convenzione descritta nelle specifiche SCPS Collaboration 2.0 [10], alla creazione del SERVICE consegue la generazione del *service_id* "DataFusionPlatformStatus-103"; tale ID verrà utilizzato nell'UD salvato sulla inter-SCP (si veda il par. 2.5.4).

L'UD "Platform Status 2.0" è già stato dichiarato come supportato nella inter-SCP (si veda par. 2.3.1) e, in rispetto della convenzione descritta nelle specifiche SCPS Collaboration 2.0 [10], all'inserimento del DATASET consegue la generazione del *dataset_id* "PlatformStatus-2.0"; tale ID verrà utilizzato nell'UD salvato sulla inter-SCP (si veda il par. 2.5.4).

Una volta configurato il Job/Service nella inter-SCP e verificato l'UrbanDataset come supportato, è stata definita la produzione relativa ovvero il **permesso per il Service "Data Fusion Platform Status" di scrivere l'UD "Platform Status" nella inter-SCP** (Figura 18).

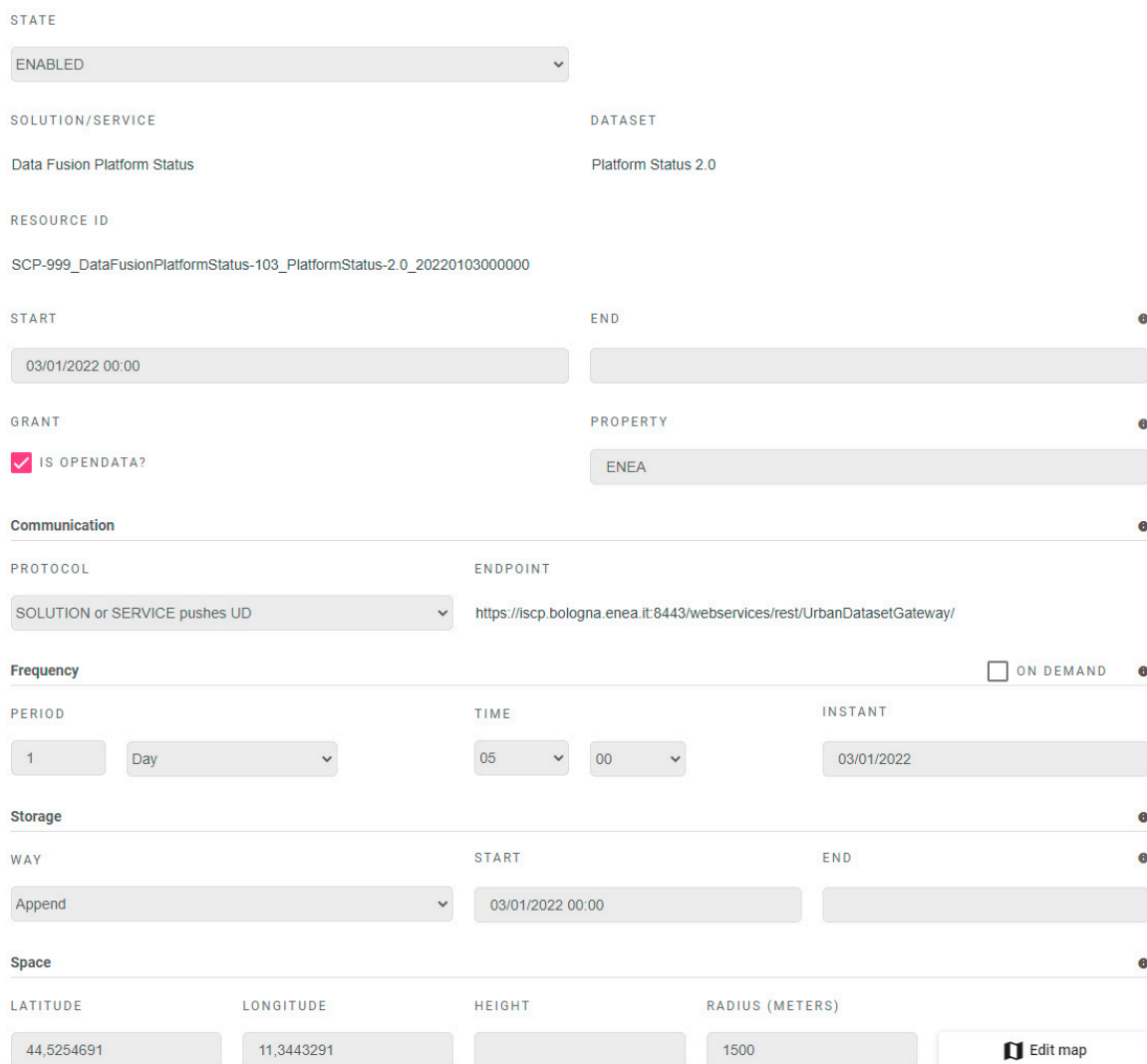


Figura 18. Definizione Produzione "Data Fusion Platform Status" – "Platform Status 2.0"

La **definizione della produzione** per il Job/Service, così come era stato per la SCP/Solution, richiede di specificare diversi aspetti, fondamentali per permettere alla inter-SCP di abilitare servizi di monitoraggio, diagnostica e gestione degli UrbanDataset; tra le caratteristiche più importanti ricordiamo:

- la proprietà del dato, che anche in questo caso rimane di ENEA ma è stato dichiarato come OPENDATA perché vogliamo che la dashboard finale non richieda login per poter visualizzare i dati nella fase finale del caso studio pilota;
- il protocollo con cui verrà scritto l'UD nella inter-SCP, in questo caso è stato configurato il protocollo "SOLUTION or SERVICE pushes UD" che scrive direttamente, localmente nella inter-SCP, utilizzando il metodo *push* dell'*UrbanDatasetGateway*, in rispetto delle specifiche *SCPS Communication 2.0* [11];
- la frequenza con cui l'UD verrà prodotto, qui impostata a "1 Day", una volta al giorno alle ore 05:00 (si ricorda che ogni recupero di UD tramite Job "Get Remote" è stata configurato tra le ore 01:00 e le ore 02:00 e quindi l'azione di DataFusion si effettuerà su UD sempre nuovi);
- la modalità di storage, ovvero la persistenza dell'UD, impostata in modalità "Append", per fare in modo che ogni giorno l'UD generato e salvato in coda al precedente (in quanto vogliamo mantenere in memoria uno storico degli stati delle SCP in modo da creare eventualmente dei grafici di monitoraggio di stato delle piattaforme nel tempo).

Si noti che, in rispetto della convenzione descritta nelle specifiche *SCPS Collaboration 2.0* [10], all'associazione SERVICE-DATASET consegue la generazione del *resource_id*

"SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000";

tale ID verrà utilizzato per salvare l'UD risultante dall'azione della datafusion in output (si veda il par. 2.5.4) ma non solo; tramite questo stesso *resource_id*, l'UD "Platform Status" sarà riferito univocamente dalla Dashboard della inter-SCP nel momento in cui dovrà esserne attuato il recupero (si veda par. 2.6).

2.5.3 Configurazione Job "Data Fusion" sulla inter-SCP

Nei precedenti due paragrafi sono stati descritti:

- lo sviluppo del modulo software (Job) di datafusion degli N UD "Platform Status" in un "Platform Status" che contiene tutti gli stati delle diverse piattaforme (par. 2.5.1);
- la configurazione della produzione per permettere di salvare l'UD in output nella SCP (par. 2.5.2).

Descriviamo in questo paragrafo la configurazione del Job nello SCP-Scheduler.

Ricordiamo questa importante differenza:

- la configurazione della produzione definisce il permesso per il Job/Service di scrivere l'UD in questa SCP con un *resource_id* univoco;
- la configurazione del Job definisce come il modulo software verrà eseguito dallo SCP-Scheduler, scrivendo in output il risultato nella SCP corrente (avendo il permesso con il *resource_id* suddetto).

Nella configurazione del Job "Data Fusion Platform Status" verrà scelto:

- l'output (che corrisponde alla produzione configurata nel precedente par. 2.5.2) e poi
- gli input compatibili con questo tipo di job "datafusion" (che corrispondono alle produzioni, una per ogni SCP connessa alla inter-SCP, configurate con i Job "Get Remote" per recuperare e salvare localmente tali UD, così come descritto nel par. 2.3.2 e par. 2.3.3).

Smart City Platform INTER-SCP
SCP-GUI 2.0 (Interoperability Layer)

HOME HISTORY DEVELOPER ADMIN SOLUTION DATASET PRODUCTION ACCESS SCHEDULER HELP
Cristiano Novelli
DEVELOPER

NAME DataFusion PlatformStatus	DESCRIPTION Service DataFusion PlatformStatus Job	<div style="background-color: #ff0000; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">BACK TO LIST</div> <div style="background-color: #003366; color: white; padding: 2px; text-align: center; margin-bottom: 5px;">SAVE</div> <div style="background-color: #ff0000; color: white; padding: 2px; text-align: center;">DELETE</div>
JOB TYPE Data Fusion SERVICE	JOB IMPLEMENTATION DataFusionPlatformStatusJob	PROPERTIES KEY
<div style="display: flex; justify-content: space-between; align-items: center;"> Input productions + </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000 Solution: SCP Casaccia </div> <div style="margin-top: 5px;"> <p>REQUEST TYPE</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Last request ▼ </div> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> SCP-999_SCPPELL-2_PlatformStatus-2.0_20211216000000 Solution: SCP PELL </div> <div style="margin-top: 5px;"> <p>REQUEST TYPE</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Last request ▼ </div> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> SCP-999_SCPLivorno-9_PlatformStatus-2.0_20211103000000 Solution: SCP Livorno </div> <div style="margin-top: 5px;"> <p>REQUEST TYPE</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Last request ▼ </div> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> SCP-999_SCPTemplate-99_PlatformStatus-2.0_20211105000000 Solution: SCP Template </div> <div style="margin-top: 5px;"> <p>REQUEST TYPE</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Last request ▼ </div> </div> </div> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; justify-content: space-between;"> SCP-999_SCPSEC-8_PlatformStatus-2.0_20220217000000 Solution: SCP SEC </div> <div style="margin-top: 5px;"> <p>REQUEST TYPE</p> <div style="border: 1px solid #ccc; padding: 2px; display: flex; justify-content: space-between;"> Last request ▼ </div> </div> </div>		

Output production
+

SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000
Solution: Data Fusion Platform Status

This is the job output production.

Frequency

PERIOD	TIME	INSTANT	SCHEDULED
1	Day	05:00	03/01/2022 <input checked="" type="checkbox"/>

Figura 19. Configurazione del Job “Data Fusion Platform Status”

Nella Figura 19 viene mostrato uno screenshot relativo l’editazione del Job “Export Platform Status”, dove si evince, a parte nome e descrizione, una serie di informazioni fondamentali per l’esecuzione del Job da parte dello SCP-Scheduler:

- Job Type: una delle quattro tipologie di Job supportati:
 - Export, per l’esportazione di dati generici;
 - Get Remote, per recuperare UrbanDataset da endpoint remoti;
 - Push Remote, per inviare UrbanDataset verso endpoint remoti;
 - **Data Fusion, per attuare attività di data fusion utilizzando UD come input e output;**
- nel nostro caso in oggetto è stata scelto il type “Data Fusion SERVICE”;

- Job Implementation: implementazione del Job da eseguire (in altre parole il modulo software .jar che implementa l'algoritmo descritto nel par. 2.5.1);
- Properties Key: proprietà da fornire in input al Job, in questo caso nessuna;
- Input productions: set di UrbanDataset, individuati tramite *resource_id*, che saranno passati in input al modulo software della datafusion; al momento sono stati configurati questi 5 *resource_id* corrispondenti agli UD prodotti dalle 5 SCP connesse alla inter-SCP:
 - "SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000",
 - "SCP-999_SCPPELL-2_PlatformStatus-2.0_20211216000000",
 - "SCP-999_SCPSEC-8_PlatformStatus-2.0_20220217000000",
 - "SCP-999_SCPLivorno-9_PlatformStatus-2.0_20211103000000",
 - "SCP-999_SCPTemplate-99_PlatformStatus-2.0_20211105000000";
- Output production: singolo UrbanDataset, individuato tramite *resource_id*, che sarà output del modulo software di datafusion; tale *resource_id* è stato definito nel precedente par. 2.5.2 ed è "SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000";
- Frequenza con cui il Job viene eseguito, nel nostro caso una volta al giorno ("1 Day") alle ore 05:00; ciò vuol dire che ogni giorno alle ore 5 del mattino il Job/Service "Data Fusion Platform Status" leggerà i "Platform Status" in input (recuperati poche ore prima) e genererà l'UrbanDataset "Platform Status", salvandolo nella inter-SCP (per poi renderlo disponibile alla Dashboard della inter-SCP, che verrà descritta nel par. 2.6);
- Checkbox "Scheduled": opzione che indica che il Job corrente sarà eseguito dallo Scheduler alla data, ora e frequenza indicate.

N.B. Lo SCP-Scheduler è frutto di un'attività di consulenza esterna che trova completa descrizione nel report "Sviluppo SCP-Scheduler per l'esecuzione di Service nella SCP" [15].

2.5.4 Esecuzione del Job "Data Fusion" sulla inter-SCP

Una volta che il Job è stato configurato può essere "scheduled" (opzione che abbiamo visto nella parte bassa della Figura 19 del paragrafo precedente) e quindi essere eseguito secondo la frequenza espressa.

Per effettuare una corretta datafusion, generando un UD "Platform Status" valido e il successivo invio dell'UD all'UrbanDatasetGateway è necessario seguire le specifiche SCPS e quindi è necessario utilizzare gli ID (descritti nel par. 2.5.2) che sono stati generati definendo questa produzione:































- il *service_id* "DataFusionPlatformStatus-103" deve essere indicato nell'UD risultante in output, nell'elemento *UrbanDataset/context/producer/id*;
 - il *dataset_id* "PlatformStatus-2.0" deve essere coerente con quello indicato nell'UD risultante in output, nell'elemento *UrbanDataset/specification/id/value*;
- il *resource_id* "SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000" deve essere utilizzato come parametro nella chiamata al metodo push, secondo l'interfaccia definita da specifiche SCPS Communication 2.0 [11].

Vediamo nel seguente screenshot una view della sezione "History" della SCP-GUI per monitorare l'azione di data fusion del Job "Data Fusion Platform Status" negli ultimi 10 giorni.

FILTERS

STORED ONLY SHOW SOLUTIONS SOLUTION SERVICE SHOW SERVICES Data Fusion Platform Status DATASET Platform Status 2.0 START END

PRODUCTION # PER PAGE 10 Actions on Selected UD's

	TIMESTAMP	SOLUTION / SERVICE	URBAN DATASET	UD TIMESTAMP	STATE	
<input type="checkbox"/>	08/03/2022, 05:00:05	Data Fusion Platform Status	Platform Status 2.0	08/03/2022, 05:00:03	STORED	  
<input type="checkbox"/>	07/03/2022, 05:00:03	Data Fusion Platform Status	Platform Status 2.0	07/03/2022, 05:00:02	STORED	  
<input type="checkbox"/>	06/03/2022, 05:00:04	Data Fusion Platform Status	Platform Status 2.0	06/03/2022, 05:00:02	STORED	  
<input type="checkbox"/>	05/03/2022, 05:00:04	Data Fusion Platform Status	Platform Status 2.0	05/03/2022, 05:00:02	STORED	  
<input type="checkbox"/>	04/03/2022, 05:00:03	Data Fusion Platform Status	Platform Status 2.0	04/03/2022, 05:00:02	STORED	  
<input type="checkbox"/>	03/03/2022, 05:00:03	Data Fusion Platform Status	Platform Status 2.0	03/03/2022, 05:00:02	STORED	  
<input type="checkbox"/>	02/03/2022, 05:00:02	Data Fusion Platform Status	Platform Status 2.0	02/03/2022, 05:00:01	STORED	  
<input type="checkbox"/>	01/03/2022, 05:00:03	Data Fusion Platform Status	Platform Status 2.0	01/03/2022, 05:00:01	STORED	  
<input type="checkbox"/>	28/02/2022, 05:00:03	Data Fusion Platform Status	Platform Status 2.0	28/02/2022, 05:00:02	STORED	  
<input type="checkbox"/>	27/02/2022, 05:00:02	Data Fusion Platform Status	Platform Status 2.0	27/02/2022, 05:00:01	STORED	  

PAGE 1 OF 7 (67 ELEMENTS) |< < 1 2 3 4 5 > >|

Figura 20. Data Fusion quotidiana dell'UD "Platform Status"

Come si può notare, ogni notte alle ore 05:00 (colonna "UD Timestamp") l'UrbanDataset viene esportato e salvato nella inter-SCP richiamando il metodo *push* dell'*UrbanDatasetGateway* (si veda a tal proposito il par. 3.3 relativo alle nuove funzionalità di validazione UrbanDataset del componente).

L'UD "Platform Status" risultante dalla Data Fusion è scritto nella inter-SCP in modalità "Append" e quindi tutti i precedenti UD sono stati salvati e sono disponibili per essere recuperati (per esempio, per creare nella dashboard un chart grafico di monitoraggio dello stato delle piattaforme che non sia solo istantaneo ma dia anche un andamento temporale).

Il set di dati in output è stato formalizzato utilizzando lo stesso formato dati "Platform Status", quindi:

- dalle N SCP connesse alla inter-SCP otteniamo N UD "Platform Status" descrittivi, ognuno, lo stato della singola SCP in cui tale UD è stato esportato;
- internamente alla inter-SCP avviene un'azione di data fusion degli N "Platform Status" recuperati e il cui output è un "Platform Status" contenente tutti gli N stati delle relative N piattaforme.

In altre parole, il "Platform Status" risultato della Data Fusion presenta al suo interno le stesse informazioni che avevamo visto nel "Platform Status" esportato da una SCP urbana come la SCP-Casaccia ma, a differenza di questo, non presenta una singola linea di dati relativi allo stato di una piattaforma ma **N linee di dati relativi agli stati di N piattaforme connesse alla inter-SCP**.

Si noti come i "Platform Status" gestiti nelle diverse fasi siano univocamente riconoscibili dal *resource_id*:

- 1) "SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000", indica il "Platform Status" esportato dal job "Export Platform Status" nella SCP-Casaccia (id "SCP-1");
- 2) "SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000", indica il "Platform Status" recuperato dal job "Get Remote Platform Status" dalla SCP-Casaccia alla inter-SCP (id "SCP-999");
- 3) "SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000", indica il "Platform Status" generato dal job "Data Fusion Platform Status" nella inter-SCP (id "SCP-999");

a dimostrare come le *SCPS Collaboration 2.0* [10] permettano di **identificare una risorsa in maniera non ambigua**.

2.6 Dashboard della inter-SCP

Siamo arrivati alla fase finale del percorso che descrive l'implementazione del caso studio pilota, che ha previsto i seguenti passi:

1. Consolidamento della definizione dell'UD "Platform Status" nell'Ontologia;
2. Esportazione dell'UD "Platform Status" nella SCP-Casaccia;
3. Trasporto dell'UD "Platform Status" dalla SCP-Casaccia alla inter-SCP;
4. Replica dell'approccio per ottenere un totale di 5 SCP connesse alla inter-SCP;
5. Azione di Data Fusion per fondere i 5 "Platform Status" (singoli) recuperati dalle 5 SCP connesse alla inter-SCP in un unico "Platform Status" (multiplo) che contiene tutti gli stati delle 5 piattaforme;
6. Recupero del "Platform Status" (multiplo) generato dalla Data Fusion per la **visualizzazione finale nella Dashboard della inter-SCP** (denominata anche "iSCP-Dashboard").

La Dashboard da associare a una generica SCP è implementata tramite il **componente SCP-DASH**, ampiamente descritto nel report LA 1.20 [2]. Mostriamo la Dashboard della inter-SCP nella Figura 21.

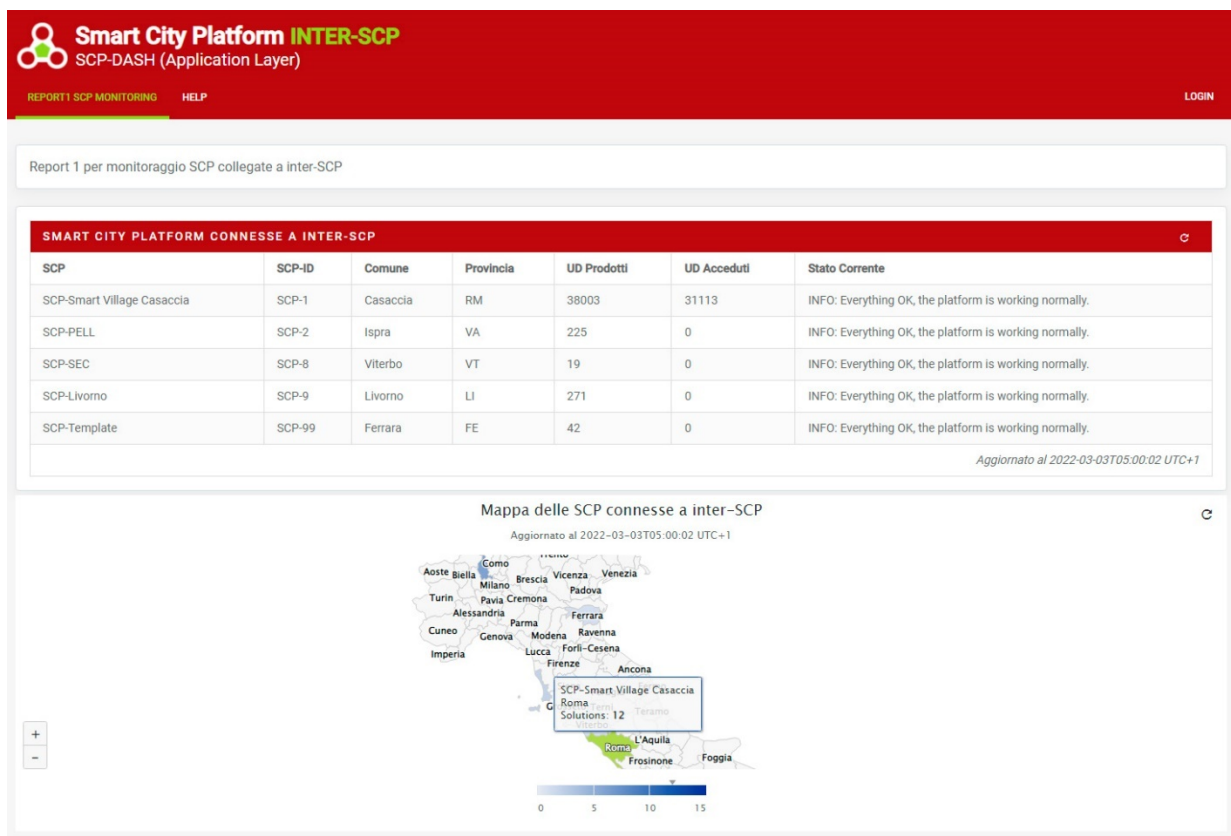


Figura 21. Dashboard della inter-SCP

La iSCP-Dashboard, al momento, presenta un solo report (ovvero una vista composta da tabelle e grafici) diviso in due parti:

- 1) una tabella di riepilogo delle SCP connesse alla inter-SCP con alcune informazioni e, cosa più importante, lo stato corrente per ognuna di esse;
- 2) una mappa in cui è possibile vedere ogni SCP localizzata sul territorio nazionale.

Questo report della dashboard utilizza il “Platform Status” che è generato dall’azione della Data Fusion (descritta nel precedente par. 2.5) e che è quindi sempre aggiornato ogni giorno alle 5 di mattina (si noti nella parte bassa della tabella in Figura 21, il timestamp dell’ultimo aggiornamento corrispondente al timestamp dell’ultimo UD generato nella data fusion).

La iSCP-Dashboard, definita tramite il componente SCP-DASH, è configurata tramite un file di configurazione json (uno per ogni report) che permette di specificare:

- nome, descrizione e posizione del report;
- tabelle e chart che saranno utilizzati e loro posizione nel layout;
- per ogni tabella/chart:
 - o id e descrizione;
 - o tipologia (tra le 5 fino ad oggi supportate);
 - o *resource_id* che verrà utilizzato per richiamare il metodo *lastRequest* dell’UrbanDatasetGateway sulla inter-SCP e che corrisponde esattamente a quello definito per la produzione da parte del Job/Service di datafusion, ovvero: “SCP-999_DataFusionPlatformStatus-103_PlatformStatus-2.0_20220103000000”,
 - o titolo, sottotitolo e informazioni di contorno;
 - o mapping tra i valori prelevati dall’UD e i valori di input utilizzati.

Si noti che, aggiungendo una nuova connessione tra la inter-SCP e una nuova piattaforma urbana, otterremo un ulteriore “Platform Status” (singolo) che andrebbe ad alimentare l’azione di datafusion con un UD di input aggiuntivo e otterrebbe un “Platform Status” (multiplo) con una linea in più al suo interno;

di conseguenza, la Dashboard potrebbe configurare agevolmente lo stesso report ma arricchito dalle informazioni recuperate dalla nuova SCP urbana.

Possiamo dunque affermare che **la connessione di una nuova SCP agente su scala urbana alla inter-SCP agente su scala nazionale non comporta alcuno sviluppo aggiuntivo**, ma solo le configurazioni del caso.

Un altro aspetto da sottolineare è che, a differenza di molte dashboard che si basano su dati raccolti in una fase precedente (addirittura, in alcuni casi, negli anni precedenti) **la Dashboard per SCP si basa su dati raccolti in tempo reale, la cui frequenza di aggiornamento è configurabile.**

Con questa ultima fase, la descrizione dell’implementazione del caso studio pilota è giunta a conclusione;

la sperimentazione di questa implementazione, oltre ad aver dimostrato la replicabilità dell’approccio con la connessione di altre SCP alla inter-SCP, **prevede anche una valutazione basata sull’analisi di alcuni indicatori** che vedremo nel prossimo par. 2.7.

Nel capitolo 6 delle conclusioni finali, si riporteranno alcune riflessioni relative alla iSCP-Dashboard finale e all’implementazione del caso studio pilota; se ne consiglia la lettura per completare il quadro sull’argomento.

2.7 Indicatori della Sperimentazione

La sperimentazione della piattaforma di governance agente su scala nazionale, oltre all'implementazione del caso studio pilota, esteso a un totale di 5 SCP connesse, termina con l'analisi di una serie di indicatori che misurano l'interoperabilità, la consistenza dei dati e, quindi, l'affidabilità dell'approccio.

Gli indicatori presi in esame sono calcolati su un intervallo temporale relativo all'ultima settimana, per considerare una finestra temporale più grande sarebbe necessario attendere più tempo dal termine dell'implementazione del caso studio pilota; tali indicatori sono i seguenti:

1. percentuale di UrbanDataset "Platform Status" esportati e salvati correttamente nella SCP agente su scala urbana negli ultimi 7 giorni;
2. percentuale di UrbanDataset "Platform Status" recuperati correttamente nella inter-SCP agente su scala nazionale negli ultimi 7 giorni;
3. percentuale di UrbanDataset "Platform Status" generati correttamente dal Job/Service di datafusion nella inter-SCP agente su scala nazionale negli ultimi 7 giorni.

Si noti che questi indicatori sono calcolati in real-time tramite la SCP-GUI 2.0 che, nella sua ultima versione, permette di monitorare oltre le SOLUTION anche i SERVICE, con un indice di interoperabilità che esprime la percentuale di quanti UrbanDataset sono stati inviati correttamente rispetto alla frequenza che è stata definita nella configurazione della produzione. L'utilizzo di questi indicatori hanno permesso la messa a punto dell'implementazione del caso studio pilota, permettendo di individuare prontamente eventuali anomalie e di migliorare sia lo SCP-Scheduler e i Job/Service dove occorreva.

Si evidenzia, inoltre, che sebbene l'insieme di KPI energetici siano stati individuati dal Politecnico di Milano nella LA 1.25 [16], è ancora in corso la loro valutazione per essere implementanti nella inter-SCP.

2.7.1 Indicatori su Esportazione

Analizzando il flusso dati così come è descritto nel caso studio pilota, il primo indicatore preso in esame esprime la percentuale di UrbanDataset "Platform Status" esportati dal Job/Service "Export Platform Status" (descritto nel par. 2.2.4) e salvati correttamente nella SCP-Casaccia negli ultimi 7 giorni, rispetto alla frequenza che è stata configurata nella relativa produzione (descritta nel par. 2.2.2) che è "1 Day" ovvero una produzione dell'UD quotidiana.

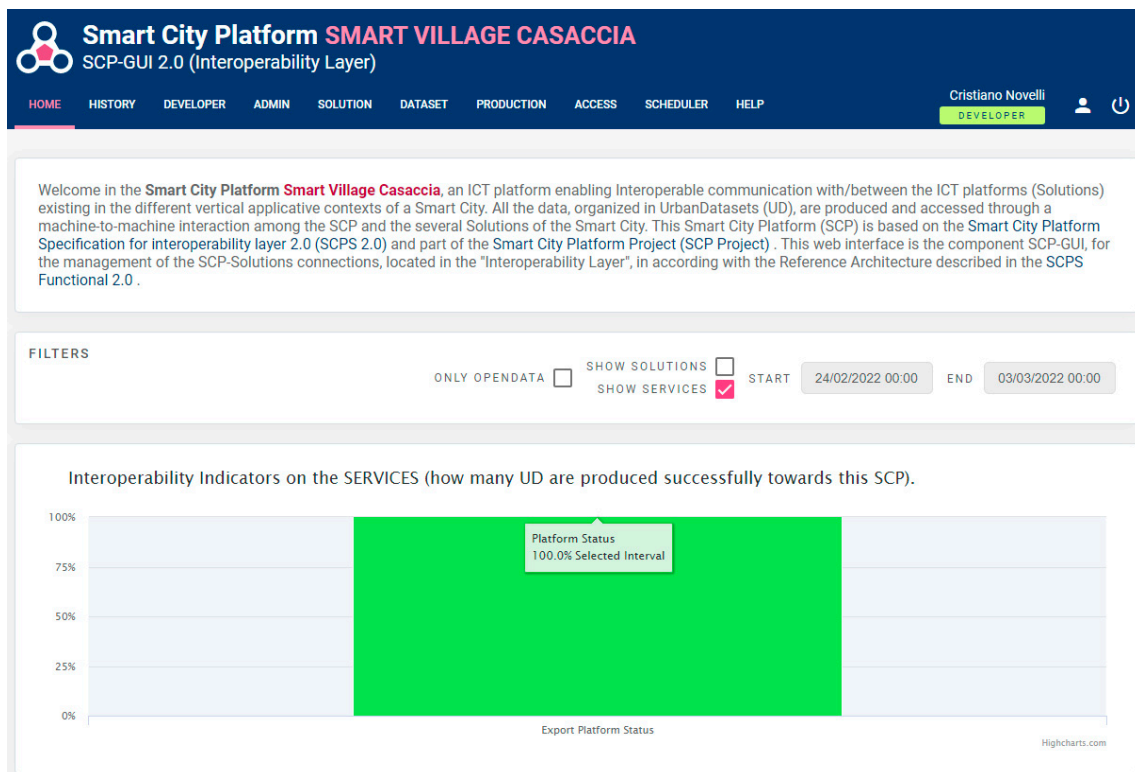


Figura 22. Indicatore su esportazione “Platform Status” nella SCP-Casaccia

Come si evince dal grafico precedente, la percentuale di UD correttamente esportati è del 100%, ovvero 1 UrbanDataset esportato e salvato ogni giorno, negli ultimi 7 giorni, in rispetto della produzione configurata.

Allo scopo di rendere maggiormente esaustiva l’analisi su questa fase e, allo stesso tempo, verificare la replicabilità dell’approccio su un’altra SCP, si considera un secondo indicatore, sempre relativo al Job/Service “Export Platform Status” ma, in questo caso, eseguito sulla SCP-Template, sempre negli ultimi 7 giorni, rispetto alla frequenza che è stata configurata nella relativa produzione ancora a “1 Day”.

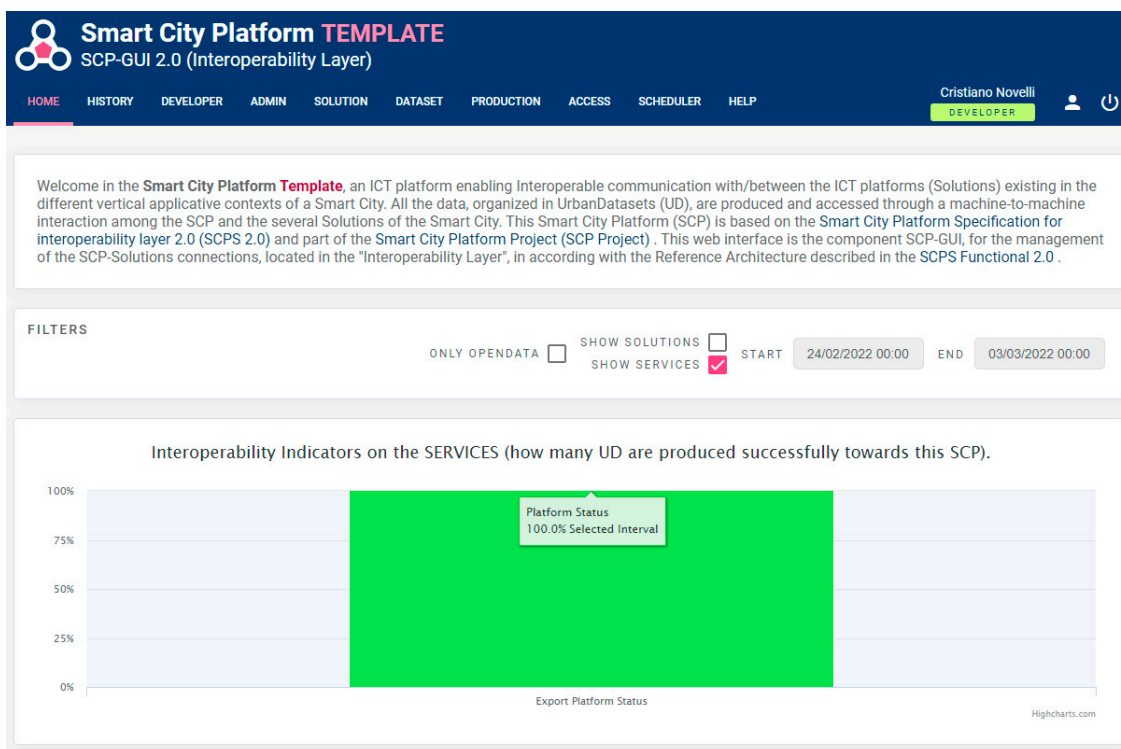


Figura 23. Indicatore su esportazione “Platform Status” nella SCP- Template

Anche in questo secondo caso, la percentuale di UD correttamente esportati è del 100%, ovvero 1 UrbanDataset esportato e salvato ogni giorno, negli ultimi 7 giorni, in rispetto della produzione configurata.

Monitorando tutte le SCP su cui è stato effettuato l’ultimo aggiornamento a SCP-GUI 2.0 e dove è stato configurato avviato il processo di esportazione con il modulo “Export Platform Status”, il risultato è identico: il 100% degli UD sono prodotti e salvati come da attese, andando a confermare che la replicabilità dell’approccio in questa prima fase è solido.

2.7.2 Indicatori su Trasporto

L'indicatore che esaminiamo in questo paragrafo è forse il più importante tra quelli presi in esame: esprime la percentuale di UrbanDataset "Platform Status" recuperati dalle SCP urbane e salvati nella inter-SCP agente su scala nazionale, tramite il Job/Service "Get Remote Platform Status" (descritto nel par. 2.3.3), negli ultimi 7 giorni, rispetto alla frequenza che è stata configurata nelle relative produzioni (descritta nel par. 2.3.2 per la SCP-Casaccia) che è sempre "1 Day" ovvero un recupero remoto quotidiano dell'UD da ogni SCP alla inter-SCP.

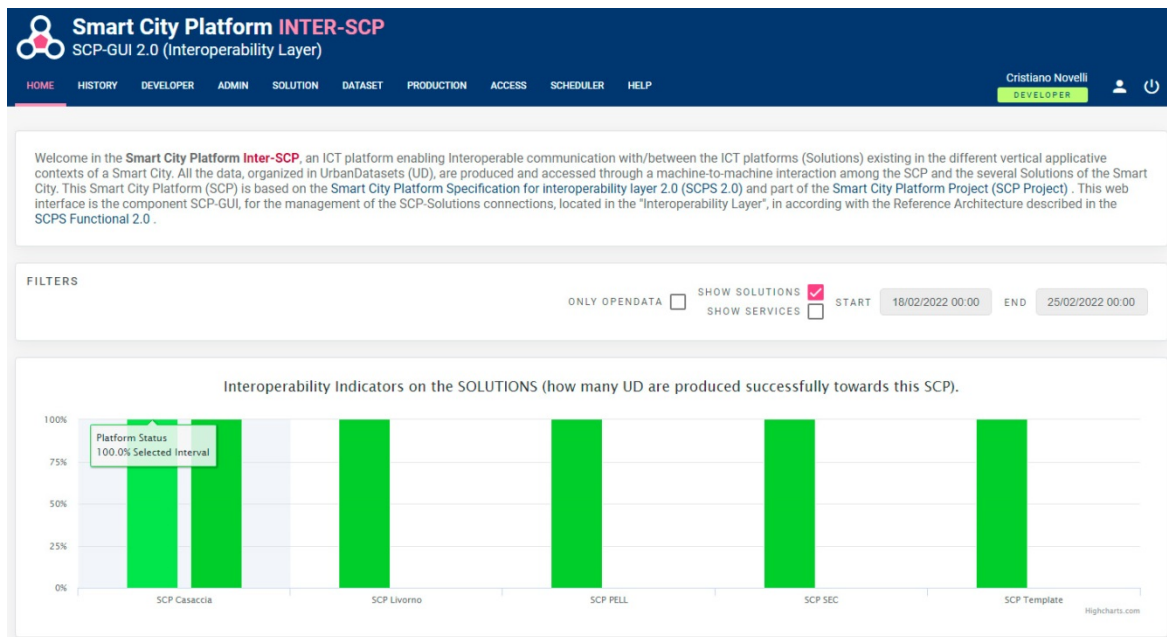


Figura 24. Indicatore su recupero remoto "Platform Status" nella inter-SCP

Come si può capire dal grafico, la percentuale di UD correttamente trasportati, per ognuna delle 5 SCP connesse alla inter-SCP, è del 100%, ovvero 1 UrbanDataset recuperato e salvato 1 volta al giorno, negli ultimi 7 giorni, in pieno rispetto delle produzioni configurate.

Questo indicatore ci conferma che **la comunicazione interoperabile è affidabile.**

Utilizzando questa vista è inoltre agevole effettuare un monitoraggio costante sull'invio degli UrbanDataset da tutte le SCP connesse alla inter-SCP e quindi correggere eventuali lacune.

Nel corso della sperimentazione, infatti, si è intervenuti diverse volte per migliorare la comunicazione che, tramite questi indicatori, mostrava un comportamento al di sotto delle attese:

- è stato individuato un mancato ripristino del token di sessione, utilizzato nelle chiamate al web service UrbanDatasetGateway; tale token non veniva rinnovato da una nuova login quando scadeva ma dopo l'intervento ciò avviene in modo sistematico;
- in alcuni casi, seppur rari, la push interna per salvare l'UrbanDataset falliva, ciò era dovuto a un simultaneo recupero di tutti e cinque gli UD dalle diverse cinque SCP nello stesso identico momento, poiché configurati alla stessa ora; per risolvere questa criticità sono stati configurati orari diversi per il recupero degli UD.

Altre azioni di consolidamento, minori ma non meno importanti, sono state effettuate sempre grazie a un monitoraggio costante degli indicatori e all'immediata individuazione e correzione degli errori o al miglioramento degli algoritmi implementati, fino a raggiungere uno stato della comunicazione che fosse il 100% affidabile.

2.7.3 Indicatori su DataFusion

L'ultimo indicatore che prendiamo in considerazione è relativo all'azione di DataFusion: esprime la percentuale di UrbanDataset "Platform Status" generati dal Job/Service "Data Fusion Platform Status" (descritto nel par. 2.5.4) e salvati correttamente nella inter-SCP negli ultimi 7 giorni, rispetto alla frequenza che è stata configurata nella relativa produzione (descritta nel par. 2.5.2) che è "1 Day" ovvero una produzione dell'UD quotidiana.

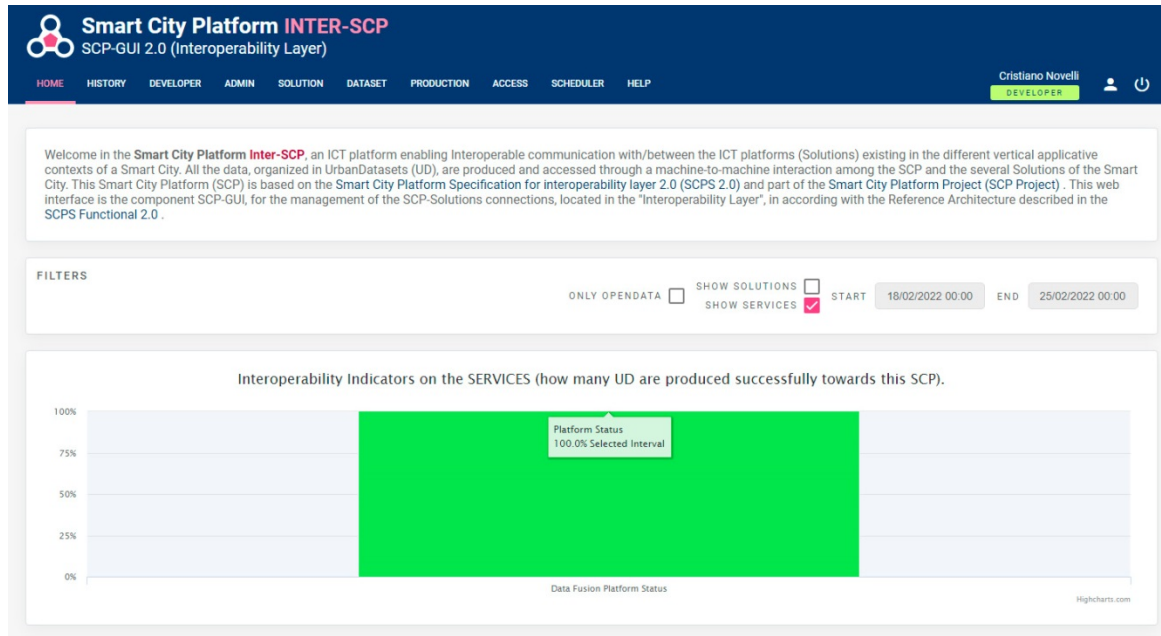


Figura 25. Indicatore su datafusion "Platform Status" nella inter-SCP

Come si nota dal grafico in Figura 25, la percentuale di UD correttamente generati dall'azione di datafusion è del 100%, ovvero 1 UrbanDataset esportato e salvato ogni giorno, negli ultimi 7 giorni, in rispetto della produzione configurata.

Questo era l'ultimo degli indicatori analizzati sia in fase di implementazione del caso studio, per individuare e risolvere le anomalie del caso, sia in fase di sperimentazione finale per valutare la bontà dell'approccio, per questa ragione non ci si deve sorprendere se tutti gli indicatori hanno dato un risultato ottimale, **sono stati proprio questi indicatori a guidare la sperimentazione fino ad avere un flusso di dati privo di anomalie in ognuna delle sue fasi.**

Oltre alle conclusioni che abbiamo tratto descrivendo questi indicatori, sono state descritte alcune riflessioni strettamente inerenti alla sperimentazione anche nel capitolo 6 delle conclusioni finali; ne suggeriamo la lettura per completare il quadro sull'argomento.

3 Sviluppo software inter-SCP 1.0

Poiché la LA 1.21 è l'ultima del triennio, vogliamo "tirare le somme" e sottolineare:

- quale è stato il risultato alla fine del triennio 2015-2017;
- quale è stato il risultato alla fine del triennio 2019-2021;

al fine di capire meglio l'obiettivo a cui si è giunti oggi, rispetto alla fase precedente.

Per chi ha letto la definizione del caso studio (report LA 1.20 [2]) e la sperimentazione che si basa sull'implementazione dello stesso (capitolo 2 di questo report) risulterà evidente che il risultato più grande raggiunto in questo triennio è la capacità di scambiare dati urbani tra SCP; ciò ha reso possibile l'implementazione della inter-SCP su scala nazionale, comunicante con altre SCP (si veda Figura 26).

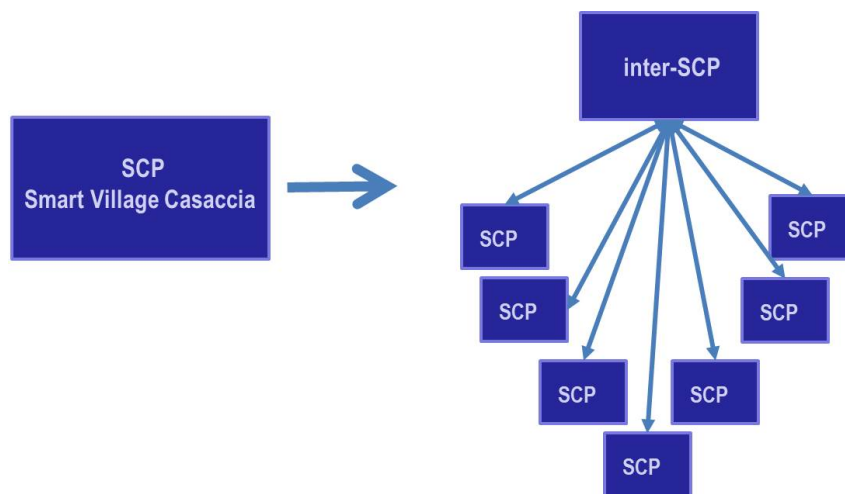


Figura 26. Passaggio da SCP urbana a inter-SCP nel corso del triennio 2019-2021

Nella seguente tabella riportiamo una comparazione tra la SCP versione 1.0 (triennio precedente) e la SCP-GUI 2.0 / inter-SCP 1.0 (triennio corrente), cercando di esprimere le macro-caratteristiche.

SCP 1.0	SCP 2.0 / inter-SCP
scala urbana/distrettuale	scala da urbana/distrettuale a nazionale
comunicazione «passiva» (le solution inviano UD alla SCP)	comunicazione «passiva» e «attiva» (le SCP possono recuperare UD o spedire UD a piattaforme terze)
no comunicazione tra SCP	comunicazione tra SCP possibile
UrbanDataset 1.0	UrbanDataset 2.0
6 macchine virtuali -> replica e configurazione complesse	6 macchine virtuali oppure 1 -> replica e configurazione agevolate
identity server privato	identity server pubblico condiviso con SSO
pubblicazione (URL) singola necessaria	pubblicazione (URL) agevolata tramite Proxy
SCP-GUI-IL 1.0 non accessibile	SCP-GUI-IL 2.0 accessibile
nessuna guida utente	percorsi di adesione utente
nessuna integrazione a PELL	integrazione con SCP-PELL -> comunicazione iSCP
no data fusion	data fusion tramite job/service
nessuna dashboard	SCP-DASH-AL (GUI per Dashboard) configurabile per i casi particolari

Figura 27. Comparazione tra SCP 1.0 e SCP-GUI 2.0 / inter-SCP

In questo capitolo verranno descritti gli sviluppi software più rilevanti che sono stati compiuti nella LA 1.21:

- evoluzione SCP-GUI 2.0;
- sviluppo SCP-Scheduler per l'esecuzione di JOB/Service;
- evoluzione UrbanDatasetGateway;

verrà inoltre data

- una panoramica generale dell'organizzazione dei progetti;
- l'intersezione con altre linee di attività.

3.1 Evoluzione SCP-GUI 2.0

SCP-GUI è il componente del prototipo SCP che permette l'interazione utente tramite interfaccia web. Nel corso della LA 1.21 ha raggiunto l'evoluzione alla versione 2.0, utilizzata da SCP e inter-SCP.

In questo paragrafo descriviamo le principali azioni di sviluppo e consolidamento:

- nuova sezione DEVELOPER;
- doppia vista Solution/Service;
- configurazione agevolata UrbanDataset da Ontologia;
- Map integrata nella produzione;
- logiche di alert tramite comunicazione via email.

Si sottolinea che il componente SCP-GUI 2.0, diversamente dalla versione precedente, presenta delle interfacce utente che rispettano i requisiti di accessibilità livello AA pubblicati da AGID¹.

3.1.1 Sezione DEVELOPER

Il componente SCP-GUI 2.0 presenta, per l'utente di tipo "Developer" (ovvero il personale ENEA che si occupa dello sviluppo e manutenzione software della SCP) una nuova sezione DEVELOPER a lui dedicata, contenente diversi strumenti per agevolare l'amministrazione della piattaforma e dei suoi componenti.

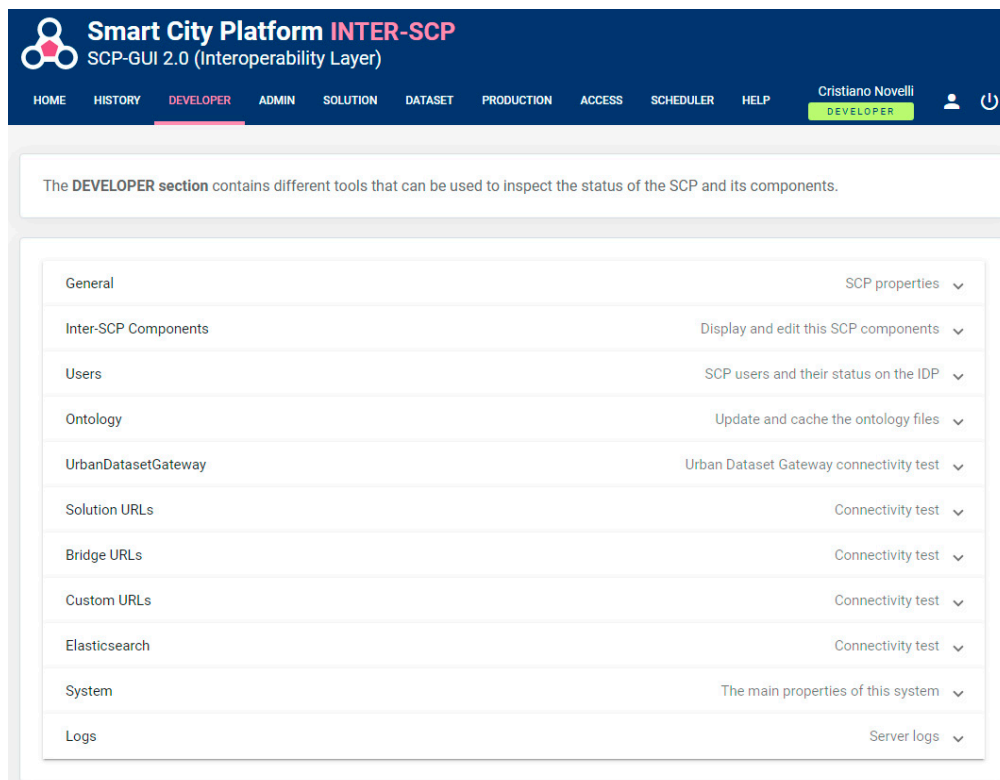


Figura 28. Nuova Sezione DEVELOPER in SCP-GUI 2.0

¹ Linee Guida per l'Accessibilità: le novità per le PA <https://www.agid.gov.it/it/agenzia/stampa-e-comunicazione/notizie/2020/01/09/linee-guida-laccessibilita-novita-pa>

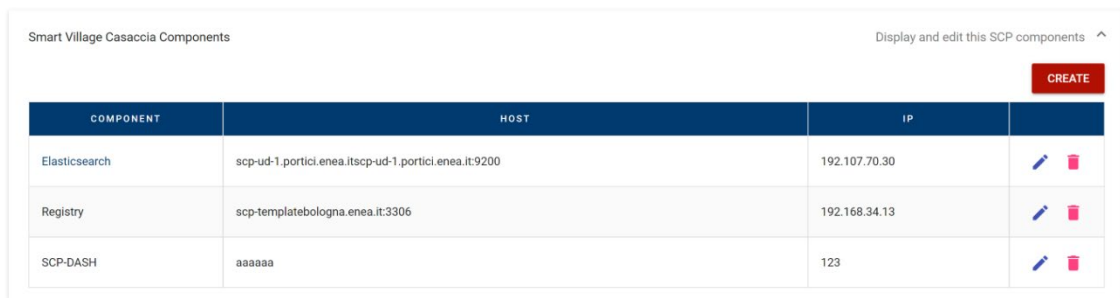
Vediamo brevemente ogni sotto-sezione.

General: permette l'editazione del record contenuto nella tabella *admin_smartcityplatform* del Registry. Nel caso in cui venga aggiornato l'ID, viene aggiornata anche la collaboration di test impostata di default.



Figura 29. Sezione Developer – Sotto-sezione General

Components: permette la visualizzazione e creazione dei componenti definiti nella tabella *admin_dev_component* del Registry. Se viene specificato un file di configurazione, questo può essere visualizzato in un pannello dedicato.








COMPONENT	HOST	IP	
Elasticsearch	scp-ud-1.portici.enea.it:9200	192.107.70.30	 
Registry	scp-templatebologna.enea.it:3306	192.168.34.13	 
SCP-DASH	aaaaaa	123	 

Figura 30. Sezione Developer – Sotto-sezione Components

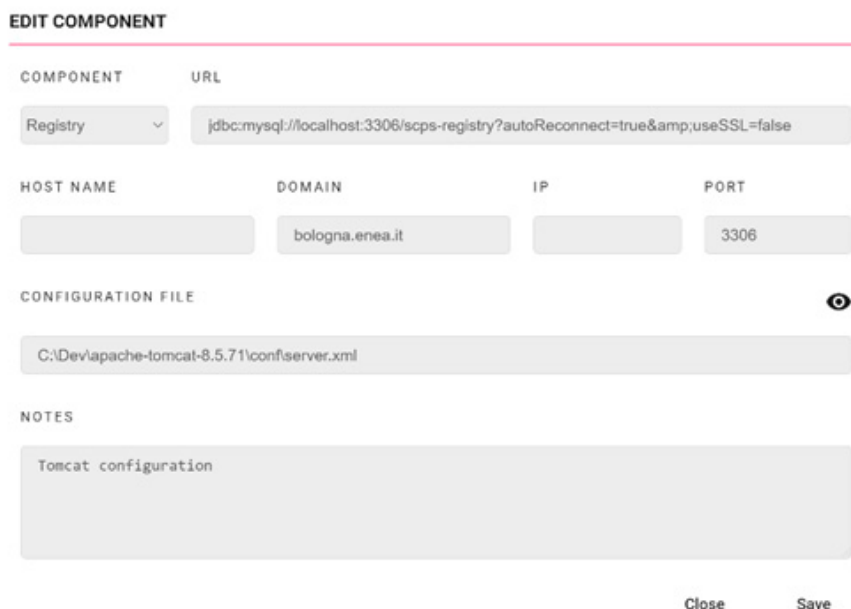


Figura 31. Sezione Developer – Edit Component

Users: permette di visualizzare lo stato di sincronizzazione degli utenti presenti nel Registry locale con quelli dell'Identity Provider remoto. Le email degli utenti sono state omesse per proteggerne la privacy.

USER	ROLE	PRESENT IN IDP	PRESENT IN REGISTRY
	ADMINISTRATOR	✓	✓
	CITIZEN	✓	✓
	DEVELOPER	✓	✓
	DEVELOPER	✓	✓
	READER	✓	✓
	M2MAGENT	✓	⊘
	DEVELOPER	✓	✓
	ADMINISTRATOR	✓	✓
	SERVICE	✓	✓
	SOLUTION	✓	✓
	DEVELOPER	✓	✓
	DEVELOPER	✓	✓

Figura 32. Sezione Developer – Users

Ontology: permette di effettuare il download e aggiornare localmente le ontologie in versione 1.0 e 2.0.

VERSION	URL	UPDATE STATUS
1.0	https://smartcityplatform.enea.it/specification/semantic/1.0/ontology/scps-ontology-1.0.owl	✓
2.0	https://smartcityplatform.enea.it/specification/semantic/2.0/ontology/scps-ontology-2.0.owl	✓

Figura 33. Sezione Developer – Ontology

UrbanDatasetGateway: permette di invocare gli URL dei metodi *test*, *login*, *push*, *specificRequest*, *delete* dell'UDGateway configurato, in modo da verificarne il corretto funzionamento.

TEST TYPE	TEST RESULT
Test	✓
Login	✓
UD Push	✓
Specific Request	⊘
UD Delete	✓

Figura 34. Sezione Developer – UrbanDatasetGateway

Solution URLs: permette di verificare la consistenza degli URL delle Solution configurate nella SCP.

Solution URLs			Connectivity test ^
			PING
LABEL	URL	STATUS	
WebGIS Casaccia	http://apic.casaccia.enea.it:8080/ciprnet/	?	
PELL	https://www.pell.enea.it/enea/	?	
Test1	http://test22.com	?	
Test2	http://test23.com	?	

Figura 35. Sezione Developer – Solution URLs

Bridge URLs: permette di verificare la consistenza degli URL dei Bridge configurati nella SCP.

Bridge URLs			Connectivity test ^
			PING
LABEL	URL	STATUS	
Test	http://localhost:8080/test-gw/	?	
	http://localhost:8080/test-gw2/	?	
	http://localhost:8080/test-gw3/	?	

Figura 36. Sezione Developer – Bridge URLs

Custom URLs: permette di verificare la consistenza degli URL configurati in un file *.properties*. Questa opzione è stata definita per poter inserire e verificare anche altri tool del framework, come per esempio la web application “UrbanDataset Web Library.

Custom URLs			Connectivity test ^
			PING
LABEL	URL	STATUS	
Test 1	https://www.google.com	?	
Test 2	https://www.enea.it/it	?	

Figura 37. Sezione Developer – Custom URLs

Log: mostra le ultime N righe del file di log. Il numero di righe può essere specificato tramite interfaccia.



Figura 40. Sezione Developer – Log

Come si può intuire, questa sezione il cui accesso è strettamente riservato a pochi utenti per la manutenzione ordinaria del sistema e l'individuazione di anomalie da correggere, permette un intervento tempestivo da qualunque device in sola lettura, per poi agire in modifica da un terminale interno alla rete ENEA su uno dei diversi componenti della SCP oggetto del bug fixing.

3.1.2 Duplice View Solution/Service

Nel corso della sperimentazione descritta nel Capitolo 2 abbiamo descritto le diverse fasi dell'intero flusso dati che dalla SCP-Casaccia arriva fino alla inter-SCP.

In questo percorso abbiamo visto come i dati possano essere inviati/recuperati:

- da una Solution verticale esterna alla SCP;
- da un JOB/Service interno alla SCP.

Monitorare il traffico da/verso gli attori esterni e quello relativo ai propri processi interni, come si può intuire, sono due cose molto diverse, anche se possono afferire allo stesso flusso dati generale.

La nuova SCP-GUI 2.0 permette di vedere le Solution e/o i Service in due distinte view tramite i filtri:

- **Show Solutions;**
- **Show Services;**

presenti nelle diverse sezioni dove si renda utile differenziare le due tipologie di entità.



Figura 41. Filtri per abilitare le view Solutions o Services

È quindi possibile personalizzare l'interfaccia in modo tale da vedere e quindi monitorare

- solo le Solution (opzione che potrebbe essere utilizzata da chi deve monitorare il traffico dati con i partner esterni);
- solo i Service (opzione che potrebbe essere utilizzata dal team di sviluppo dei processi interni);
- entrambe le tipologie di dati.

Non si ritiene necessario includere altre figure con screenshot della GUI, essendovene presenti in quantità nel capitolo 2, relativo alla sperimentazione; si suggerisce piuttosto di rivedere tale capitolo, notando l'utilizzo di un filtro piuttosto che un altro, per abilitare la visualizzazione desiderata.

3.1.3 Configurazione UrbanDataset da Ontologia

Nella SCP-GUI 2.0 è stata introdotta la funzionalità di configurazione “assistita” nella SCP per l’inserimento di un UrbanDataset direttamente dall’Ontologia.

Come sappiamo,

- l’Ontologia contiene la definizione centralizzata e univoca per ogni UrbanDataset definito;
- in ogni SCP, bisogna specificare quali UrbanDataset, tra quelli definiti nell’Ontologia, sarà utilizzato nella SCP in oggetto.

La configurazione “assistita” consiste nel permettere all’utente di **scegliere la versione dell’Ontologia** da utilizzare (l’ultima versione è la 2.0) e poi **ricercare il dataset** al suo interno tramite l’inserimento di una stringa che sia, anche parzialmente, il nome.

Per esempio,

- inserendo “plat” viene trovato già il solo “Platform Status”;
- selezionando questo UD, **tutti gli altri campi saranno completati prendendo le informazioni direttamente dall’Ontologia scelta** (come mostrato in Figura 42).

The screenshot shows the 'DATASET' configuration page in the SCP-GUI 2.0. The header includes the logo and navigation menu. The main form contains the following fields and values:

- ONTOLOGY VERSION:** 2.0
- SEARCH DATASET:** Platform Status
- DATASET ID:** PlatformStatus-2.0
- DATASET NAME:** Platform Status
- CATEGORY:** Public safety, policy & Em.Res.
- SUBCATEGORY:** Communication
- DESCRIPTION:** Informazioni dello stato di funzionamento della piattaforma all'istante di generazione dell'UrbanDataset
- DATASET URI:** https://smarcityplatform.enea.it/specification/semantic/2.0/ontology/scps-ontology-2.0.owl#PlatformStatus

Buttons for 'BACK TO LIST' and 'SAVE' are located on the right side of the form.

Figura 42. Configurazione UD da Ontologia

Questa funzionalità aggiuntiva è molto importante perché, rispetto a prima, impedisce di specificare nella SCP un UrbanDataset in modo errato, cosa che potrebbe renderne l’utilizzo impossibile. Prendendo i dati direttamente dall’Ontologia, stiamo specificando nella SCP l’esatta definizione che è stata registrata nell’Ontologia centrale.

3.1.4 Map integrata nella produzione

Nella sezione PRODUCTION della SCP-GUI è presente la sotto-sezione “Space” dove poter specificare le coordinate (*latitude, longitude, height*) e il raggio (*radius*, espresso in metri) che sono associati alla

produzione in oggetto in questa SCP; in questo modo è possibile delineare un'area circolare rappresentativa per i dati relativi a questa produzione.

Per facilitare la specificazione delle coordinate, con la SCP-GUI 2.0 è stata introdotta una nuova funzionalità, basata su google maps, che permette di ricercare una località sulla mappa e delineare l'area circolare, per poi ricavarne automaticamente le coordinate e il raggio, come valori numerici nella form.

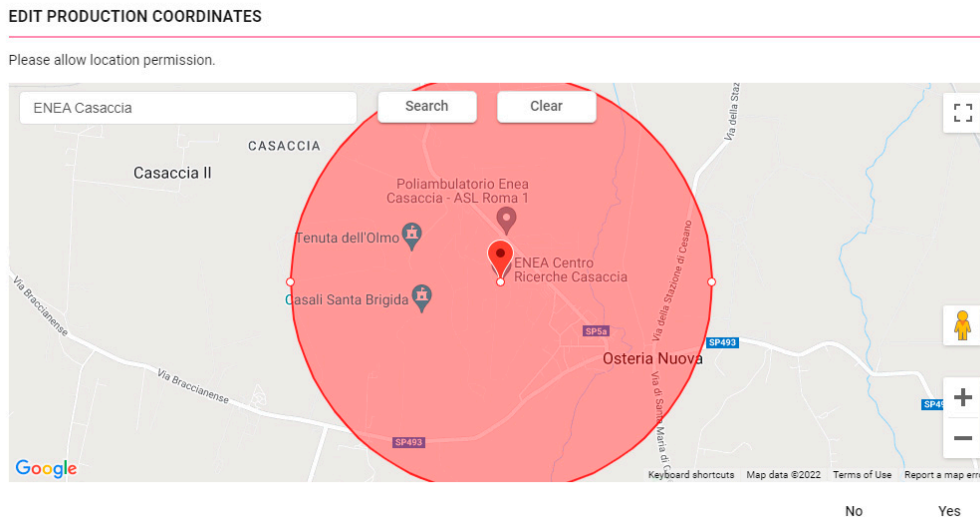


Figura 43. Funzionalità Map integrata

Anche in questo caso la funzionalità introdotta permette di velocizzare il lavoro di configurazione della SCP e minimizzare il rischio di errore in fase di specificazione delle coordinate.

3.1.5 Alert via email

L'ultima funzionalità che riteniamo importante presentare per la SCP-GUI 2.0 è il sistema di alert via email che è stato abilitato. Spieghiamo brevemente di cosa si tratta, utilizzando il simbolo

- ➔ Per sottolineare l'invio di una email.

Verifica Utente da parte dell'Identity Provider

Una volta registrato nell'Identity Provider (IDP) tramite il processo di registrazione,

- ➔ viene inviata una email all'utente per effettuare la **verifica utente**.

L'utente clicca sul link inviatogli e così diviene verificato.

Nella SCP-GUI deve essere impedito il login agli utenti che non hanno verificato la propria e-mail.

In caso di errore, viene generato un messaggio di errore specifico.

Gestione abilitazione da parte dell'Administrator

In alcune fasi di utilizzo della SCP, si rende necessario l'invio di email per allertare gli utenti riguardo i "permessi" richiesti/forniti. I casi in cui inviare email di alert sono i seguenti:

1) Registrazione e quindi richiesta di **abilitazione per una Solution**:

- ➔ Viene inviata una email all'utente Administrator (responsabile);

Una volta che l'abilitazione è effettuata:

- ➔ Viene inviata una email all'utente Solution per avvertirlo che la Solution è abilitata.

2) Richiesta di **abilitazione produzione/accesso UD** da parte di una Solution:

- ➔ Viene inviata una email all'utente Administrator (responsabile);

Una volta che l'abilitazione è effettuata:

- ➔ Viene inviata una email all'utente Solution per avvertirlo che la produzione/accesso è abilitata.

È stato utilizzato un server SMTP (ENEA) per abilitare le suddette funzionalità via email.

3.2 Sviluppo SCP-Scheduler e Job

Uno dei componenti chiave della sperimentazione è lo SCP-Scheduler per l'esecuzione di JOB/Service. Ciò si evince immediatamente dalla rappresentazione schematica del caso studio pilota (che abbiamo riportato all'inizio del Capitolo 2): **sono i JOB/Service che vengono eseguiti periodicamente dallo SCP-Scheduler a permettere la comunicazione tra SCP**, nonché le azioni di **esportazione e datafusion**.

Progettazione e sviluppo dello SCP-Scheduler sono stati parte di un'attività di consulenza descritta in un apposito report [15]. Riportiamo in questo paragrafo riprendiamo solo gli aspetti più importanti, lasciando l'approfondimento relativo alla realizzazione dello SCP-Scheduler nel suddetto report e l'utilizzo pratico dei diversi tipi di JOB nel capitolo 2 relativo alla sperimentazione del caso studio pilota.

La progettazione ha previsto l'analisi dei requisiti relativi a 3 Job/Service descritti nel caso studio pilota, per ognuno di essi è stata definita la tipologia astratta:

1. "Exporter",
2. "Get Remote UD",
3. "DataFusion UD",

A questi Job/Service astratti ne è stato aggiunto un quarto:

4. "Push Remote UD", che non è stato ancora utilizzato.

Dall'analisi di questi 4 JOB/Service astratti si sono definiti 6 step di configurazione validi per tutti:

1. Step1: Nome e Tipologia;
2. Step2: Algoritmo interno relativo all'implementazione del JOB;
3. Step3: UrbanDataset in input;
4. Step4: UrbanDataset in output;
5. Step5: configurazione della frequenza che, inizialmente, viene impostata con quella espressa nella produzione che è stata selezionata, poi può essere modificata;
6. Step6: salvataggio dell'output con push verso l'UrbanDatasetGateway (locale o remoto).

È stata così definita un'unica interfaccia utente che, in base alla tipologia di JOB scelto, ne permettesse la configurazione richiedendo gli input necessari. In Figura 44 sono evidenziati i suddetti step comuni.

The screenshot displays a configuration form for a Job/Service. It is divided into several sections, each representing a step in the configuration process:

- Step1:** Contains fields for 'NAME' (Get SCPCasaccia Platform Status) and 'JOB TYPE' (Get Remote UD via BRIC).
- Step2:** Contains fields for 'JOB IMPLEMENTATION' (ExportPlatformStatusJob) and 'DESCRIPTION' (Service Get SCPCasaccia PlatformStatus Job).
- Step3:** Titled 'Input productions', it shows a selected production (SCP-1_ExportPlatformStatus-2_PlatformStatus-2.0_20211027000000) and a 'REQUEST TYPE' dropdown set to 'Last request'.
- Step4 e 6:** Titled 'Output production', it shows a selected production (SCP-999_SCPCasaccia-1_PlatformStatus-2.0_20211105000000) and a note 'This is the job output production.'
- Step5:** Titled 'Frequency', it includes a table for scheduling:

PERIOD	TIME	INSTANT	SCHEDULED
1	Day	01 02	05/11/2021 <input checked="" type="checkbox"/>

On the right side of the form, there are three buttons: 'BACK TO LIST' (red), 'SAVE' (blue), and 'DELETE' (red).

Figura 44. Step di configurazione del JOB/Service

Il componente SCP-Scheduler è quindi in grado di:

- individuare i JOB implementati (sotto forma di librerie .jar opportunamente collocati nella webapp) e proporli nell'interfaccia utente;
- permettere la configurazione di 4 tipologie di JOB/Service;
- riutilizzare le produzioni definite nella SCP per configurare parzialmente i propri input;
- eseguire i JOB periodicamente, secondo la frequenza configurata.

Per implementare i JOB/Service è stato definito un apposito progetto "SCP-Scheduler-JOB-Template" che implementa un'interfaccia comune a tutti i JOB e necessaria per dialogare con lo SCP-Scheduler.

Si rimanda al report [15] appositamente dedicato a progettazione e sviluppo dello SCP-Scheduler, per una descrizione completa e dettagliata di questa attività di consulenza.

3.3 Evoluzione *UrbanDatasetGateway*

In questo paragrafo vengono descritti le diverse sotto-attività che hanno portato al consolidamento e all'evoluzione del componente *UrbanDatasetGateway (UDG)* web service, per l'invio e il recupero di *UrbanDataset* in una SCP.

Le sotto-attività che verranno descritte sono le seguenti:

1. sviluppo di 2 nuovi metodi secondo specifiche "*SCPS Communication*" [11];
2. consolidamento dell'UDG e risoluzione di anomalie, in particolare un'anomalia che gestisce il controllo di più solution in capo ad un singolo utente in funzione delle collaborazioni (*resourceId*);
3. validazione *UrbanDataset* e validazione della buona forma di alcuni parametri (p.es. timestamp);
4. integrazione dei nuovi validatori di *UrbanDataset 2.0* e loro riorganizzazione nel processo di ricezione (metodo *push* dell'UDG).

Di seguito verranno illustrate in dettaglio tutte le attività svolte.

3.3.1 Implementazione 2 nuovi metodi

Sono stati implementati due nuovi metodi per la comunicazione via web service RESTful con l'*UrbanDatasetGateway*, tali metodi sono stati progettati a livello di specifiche SCPS e discussi nella precedente LA 1.20 [2]; riprendiamo e integriamo le due descrizioni date:

- *lastRequest*: questo metodo serve a recuperare l'ultimo *UrbanDataset* prodotto in una certa collaborazione (ciò è particolarmente utile nella connessione tra SCP e inter-SCP e nelle dashboard realizzate con il componente SCP-DASH;
- *searchingRequestByProperty*: evoluzione del metodo *searchingRequest*, che permette di raffinare la ricerca ulteriormente, ricercando tra le proprietà dell'UD quelle che fanno match con la proprietà data (questo metodo può risultare molto utile in diversi ambiti, p.es. nel caso PELL per il recupero degli *UrbanDataset* secondo un dato codice ISTAT (si veda par. 3.5.3).

Le descrizioni di tali metodi sono state integrate nella specifica *SCPS Communication 2.0* [11].

L'implementazione dei due nuovi metodi è stata realizzata.

Si noti che il metodo *lastRequest* risulti indispensabile per la sperimentazione, per permettere:

- 1) alla inter-SCP di ottenere l'ultimo *UrbanDataset* "Platform Status" prodotto da una SCP con il job "Get Remote" (si veda in particolare il par. 2.3.2);
- 2) alla Dashboard di ottenere il risultato della datafusion più aggiornato (si veda par. 2.6).

3.3.2 Consolidamento e correzione anomalie

Il lavoro di consolidamento della SCP prevede continue azioni di test, debug e aggiornamenti del software.

Uno degli interventi più delicati effettuati nella LA 1.21 ha riguardato il *ResourceIdManager*, ovvero il gestore dei permessi per poter scrivere o leggere una collaborazione (identificata tramite il *resourceId*).

Nel corso dei precedenti sviluppi della SCP, in particolare nell'attività di integrazione della SCP nel PELL effettuata nella LA 1.20, è stata modificata la relazione tra il singolo utente e le sue Solution, passando da una associazione 1-1 a un'associazione 1-N.

Questo aggiornamento, implementato sia nel *Registry* (database relazionale che gestisce i permessi sulle produzioni e sugli accessi dei dati) e sia nella SCP-GUI (interfaccia utente di gestione della SCP) ha introdotto un'anomalia nel componente *UrbanDatasetGateway* che non gestiva correttamente il sistema di verifica dei permessi per la produzione/accesso di una collaborazione relativa a una particolare Solution.

L'anomalia, è stata individuata, analizzata e risolta con la revisione della relativa classe java *ResourceIdManager*, alla quale è appunto delegato il compito di effettuare le opportune letture sul database *Registry* e verificare i corretti permessi per produrre o accedere ai dati, aspetto cruciale della comunicazione.

3.3.3 Validatore Timestamp

Il parametro *timestamp* definito a livello di *context* in ogni singolo *UrbanDataset*, assieme al *resourceId*, risulta di notevole importanza per identificare in maniera univoca un nuovo dato inserito all'interno di una piattaforma SCP. Esso riporta la data e l'orario (eventualmente fino ai millisecondi) del momento in cui un *UrbanDataset* è stato prodotto da parte di un verticale. Nel formato JSON di un *UrbanDataset*, il valore della chiave *timestamp* è una stringa che secondo le specifiche SCPS è definita nel seguente modo:

“yyyy-MM-dd' T' HH:mm:ss [.SSS]”

dove, rispettivamente:

- yyyy rappresenta l'anno nel formato di 4 cifre;
- MM rappresenta il mese nel formato di 2 cifre, da 01 a 12;
- dd rappresenta il giorno nel formato di 2 cifre, da 01 a 31;
- 'T' rappresenta il separatore tra data e orario;
- HH rappresenta l'ora nel formato di 2 cifre, da 00 a 23;
- mm rappresenta i minuti nel formato di 2 cifre, da 00 a 59;
- ss rappresenta i secondi nel formato di 2 cifre, da 00 a 59;
- [.SSS] rappresenta opzionalmente i decimi, i centesimi e i millisecondi, rispettivamente, nel formato di 1-2-3 cifre, e quindi da 0 a 9, da 00 a 99, da 000 a 999.

È importante sottolineare che il *timeZone* è gestito nell'elemento omonimo dell'*UrbanDataset*.

I dati urbani di una smart city sono molto eterogenei ma vengono indicizzati, secondo l'approccio delle specifiche SCPS, tramite parametri temporali e spaziali. Si può dunque capire quanto importante sia la gestione corretta dell'elemento temporale tramite *timestamp* e come quasi tutti i servizi dell'*UrbanDatasetGateway* usino direttamente o indirettamente dei dati di questo tipo.

Ad esempio, il servizio (*push*) per inviare e memorizzare un *UrbanDataset* sulla piattaforma SCP, oltre al *resource_id*, legge ed utilizza il valore del *timestamp* definito nel *context* dell'*UD* per identificarlo univocamente; analogamente tutti i servizi (*specificRequest*, *lastRequest*, *basicRequest*, *searchingRequest*, *deepSearchingRequest*) che restituiscono uno o più *UrbanDataset* dopo aver effettuato una ricerca all'interno di una finestra temporale, filtrano i risultati basandosi appunto su query che prendono input uno o più parametri di tipo *timestamp*.

Nel corso delle sperimentazioni è stato osservato che aggiungere un controllo sul formato del timestamp in fase di acquisizione di un *UrbanDataset* (servizio *push*) o di accesso (servizi **Request*) avrebbe reso più robusta la componente *UrbanDatasetGateway* nella fase di verifica e individuazione di timestamp malformati dal punto sintattico e semantico.

Per tale motivo è stato integrato nel codice sorgente dell'*UrbanDatasetGateway* un validatore di timestamp. Con questa nuova funzionalità, sono considerate valide dal punto di vista della composizione timestamp espressi come quelli riportati in Tabella 1 mentre verranno gestiti e segnalati con opportuni messaggi di errore o di eccezione casi come quelli riportati in Tabella 2.

Tabella 1 - Esempi di timestamp validi

Timestamp	Nota
"2021-06-21T12:34:56"	forma standard
"2021-06-21T12:34:56.7"	con decimi di secondi
"2021-06-21T12:34:56.78"	con centesimi di secondi
"2021-06-21T12:34:56.789"	con millesimi di secondi
"2000-02-29T12:34:56"	2000 è un anno bisestile
"1900-02-28T12:34:56"	1900 non è un anno bisestile

Tabella 2 - Esempi di timestamp non validi

Timestamp	Nota
"2021-06-21T12:34"	senza secondi
"2021-06-21T12"	senza minuti e secondi
"2021-06-21"	senza ore, minuti e secondi
"2021-06"	senza giorni, ore, minuti e secondi
"2021-06-21 12:34:56"	senza 'T'
"2021-06-21T12:34:56.7898"	con decina di mii
"2021-06-21T12:34:56.abc"	millisecondi non validi
"2021-06-21T12:34:60"	secondi non validi, massimo 59
"2021-06-21T12:60:56"	minuti non validi, massimo 59
"2021-06-21T24:34:56"	ore non valide, massimo 23
"1998-09-31T12:34:56"	giorno non valido, massimo 30 a settembre
"1998-11-31T12:34:56"	giorno non valido, massimo 30 a novembre
"2008-02-2xT12:34:56"	giorno 2x non valido
"2008-0x-28T12:34:56"	mese 0x non valido
"20xx-02-28T12:34:56"	anno 20xx non valido
"20-11-02T12:34:56"	anno non valido, deve essere formato da 4 cifre yyyy
"2020/11/02T12:34:56"	formato data non valido, deve essere yyyy-MM-dd
"2020-11-32T12:34:56"	giorno invalido, 32 non fuori intervallo
"2020-13-30T12:34:56"	mese non valido, 13 fuori intervallo
"2020-A-20T12:34:56"	mese non valido
"2020-02-30T12:34:56"	anno bisestile, febbraio ha 29 giorni

"2019-02-29T12:34:56"	anno non bisestile, febbraio ha 28 giorni
"1900-02-29T12:34:56"	1900 non bisestile, febbraio ha 28 giorni
"12012-04-05T12:34:56"	anno non valido, massimo 4 cifre
Null	valore nullo
" "	spazio bianco
""	stringa vuota

3.3.4 Consolidamento validatore JSON-Schema

Sin dalla sua prima versione, l'*UrbanDatasetGateway* esegue una validazione del formato dei dati garantendo che essi siano conformi nella struttura al JSON previsto dalle specifiche. In altre parole, ogni *UrbanDataset* che viene inserito sulla piattaforma, o che viene rilasciato, è soggetto ad un controllo per verificare che esso sia un JSON dotato dei tre livelli obbligatori "specification", "context" e "values".

Il controllo si dirama anche nei successivi sottolivelli per assicurarsi che l'intera struttura nidificata del JSON sia quella prevista dalle specifiche. In sostanza, si verifica che un *UrbanDataset* venga convalidato secondo il mapping del JSON-Schema reperibile al seguente link:

<https://smartcityplatform.enea.it/specification/information/2.0/json/schemas/scps-urbandataset-schema-2.0.json>

Nelle ultime versioni dell'*UrbanDatasetGateway* questa operazione di convalida è demandata ad un nuovo modulo software appositamente preparato: "JSON Validator". In questo modo, lo sviluppo del validatore risulta indipendente dal progetto dell'*UrbanDatasetGateway* e può essere in qualsiasi momento aggiornato e caricato come libreria indipendente.

3.3.5 Integrazione validatore Schematron

Il modulo software "JSON Validator" (introdotto nella precedente sezione) è stato sviluppato in modo tale da estendere le funzionalità di validazione di un *UrbanDataset* non solo rispetto ad uno JSON-Schema ma anche rispetto ad uno Schematron. Questo secondo tipo di validazione interviene ad un livello più profondo andando a controllare il rispetto della definizione semantica (e quindi delle proprietà) dell'*UrbanDataset* così come è stata dichiarata nell'Ontologia centrale.

In altre parole, tramite lo Schematron che viene prodotto automaticamente dall'Ontologia, sulla singola istanza di *UrbanDataset* vengono effettuate le seguenti verifiche:

- la coerenza dei riferimenti alla specifica utilizzata, riportati nella sezione "specification"
- la correttezza delle dichiarazioni della proprietà previste per l'*UrbanDataset* riportate nella sezione "specification/properties/propertyDefinition" (ad es.: presenza di tutte le proprietà obbligatorie, correttezza di eventuale lista di codici indicata per una proprietà, correttezza dell'unità di misura e del tipo di dato indicato per una proprietà,...)
- la presenza, nella sezione dei valori ("values"), di tutte le proprietà previste per l'*UrbanDataset*, coerentemente con quanto previsto dalla specifica semantica e dichiarato nella sezione "specification/properties/propertyDefinition".

La descrizione di ogni tipologia di *UrbanDataset* attualmente definita può essere visionata nella *UrbanDataset Web Library* al seguente link:

<https://smartcityplatform.enea.it/UDWebLibrary/it/urbandataset>

Il validatore JSON integrato nell'*UrbanDatasetGateway*, effettua la validazione Schematron in fase di acquisizione di un nuovo *UrbanDataset* (metodo *push dell'UDG*) sulla piattaforma SCP e non in fase di recupero (servizi **Request*) poiché solo *UrbanDataset* validi possono essere immagazzinati nella SCP e quindi risulterebbe azione ridondante e inutile.

Si noti che, la validazione Schematron si attiva solo per UrbanDataset di seconda generazione (versione 2.0) e solo se un parametro booleano definito nelle proprietà della piattaforma SCP è settato a “true”; ciò permette di abilitare/disabilitare la validazione Schematron in caso sia necessario effettuare manutenzione o per individuare particolari anomalie.

Dal punto di vista operativo, il validatore JSON:

- legge all'interno dell'UrbanDataset il tipo di dataset (per es. WeatherCondition) che esso vuole rappresentare;
- recupera lo Schematron in formato XML interrogando tramite un'opportuna chiamata RESTful (GET) da un link del tipo:
<https://smartcityplatform.enea.it/UDWebLibrary/it/template/WeatherCondition?templ=schem;>
- esegue una validazione dell'UrbanDataset secondo lo Schematron generato a run-time dall'ultima versione dell'Ontologia.

3.3.6 Riorganizzazione ordine dei validatori

Come sappiamo, l'UrbanDatasetGateway permette di inviare UrbanDataset alla SCP; in questa fase vengono compiuti una serie di controlli molto importanti per verificare (oltre ai permessi di scrittura) se una serie di caratteristiche siano rispettate in modo rigoroso e siano valide.

Ogni controllo è demandato a un particolare “Validatore” (abbiamo visto nelle precedenti sezioni come, ad esempio, possa esserci un *JSON Validator* per gli UrbanDataset e un validatore *ResourceIdManager* relativo ai permessi di scrittura e lettura UD).

Al fine di migliorare le prestazioni complessive della piattaforma SCP è stato rivisto l'ordine di esecuzione dei diversi validatori. In sostanza, si è cercato di anticipare i validatori le cui azioni sono meno onerose dal punto di vista computazionale, mantenendo sempre un ordine logico di precedenza, al fine di massimizzare l'efficienza del metodo, ridurre i tempi di risposta e restituire opportuni messaggi di errori con un dettaglio più esauriente. D'altro canto, se invece tutte le validazioni sono soddisfatte, i tempi di esecuzione rimangono invariati rispetto al passato.

È per tale motivo che nel codice sorgente del servizio di *push* dell'*UrbanDatasetGateway*, l'ordine con il quale sono eseguite le operazioni di controllo, è stato variato come segue:

1. Validazione del token di autenticazione;
2. Validazione del *resourceId* (*permessi di scrittura*);
3. Validazione della dimensione dell'UrbanDataset;
4. Validazione del timestamp a livello di *context*;
5. Validazione della frequenza di inserimento del dato sulla piattaforma;
6. Validazione dell'UrbanDataset secondo il JSON-Schema;
7. Validazione dell'UrbanDataset secondo lo Schematron (se abilitata);
8. Validazione di verifica che l'ID del produttore (*elemento context/producer/id*) e l'ID del dataset (*elemento specification/id/value*) indicati nell'UrbanDataset siano quelli previsti dal *resourceId*.

Si noti che

- se la validazione 2 (sul *resourceId*) non fosse subito verificata, sarebbe inutile svolgere tutte le altre valutazioni, questo è il motivo per cui è meglio che venga effettuata subito dopo la verifica del token;
- la validazione 8 (verifica sul *producer* e sul *dataset ID*) ha senso farla solo se l'UD inviato è corretto dal punto di vista dello schema JSON, per cui tutti i campi relativi a produttore e tipologia di dataset sono sicuramente presenti; questo è il motivo per cui la validazione 8 segue la validazione 6 (JSON-schema), al più potrebbe inserirsi prima della validazione 7 (Schematron).

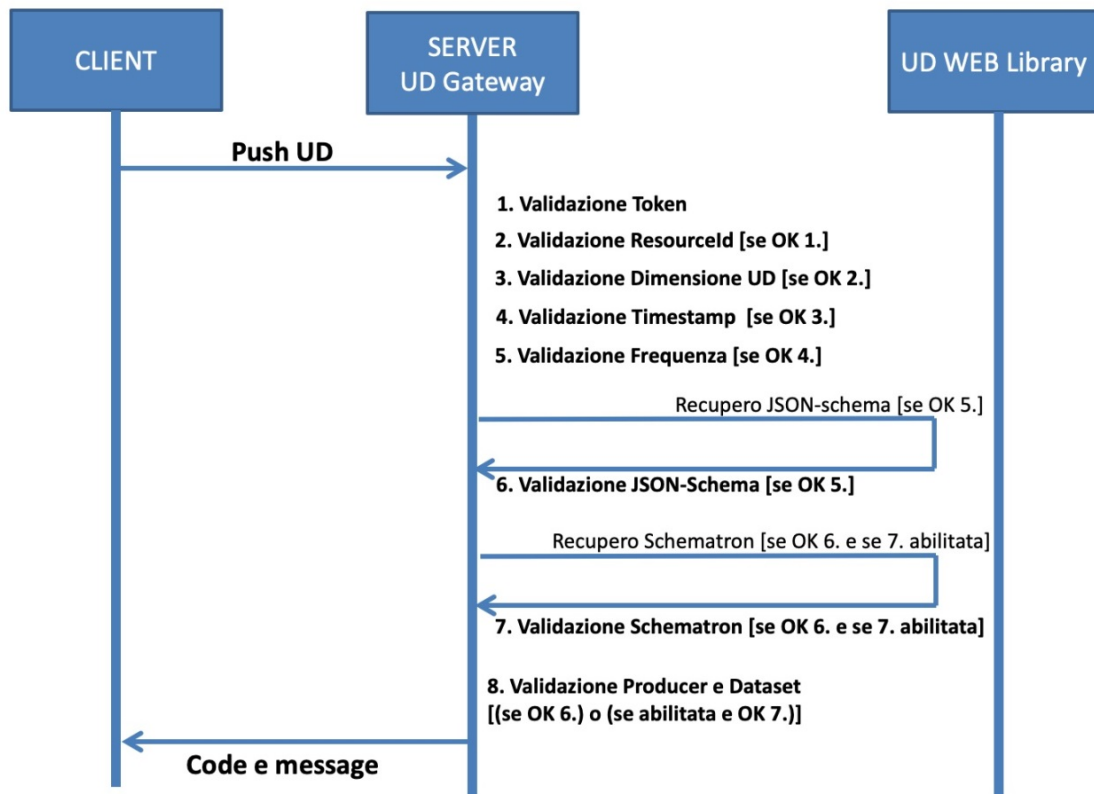


Figura 45 - Schema flusso di validazione del metodo *push*

In Figura 45 è riportato l'attuale schema di flusso di validazioni eseguite durante l'inserimento di un nuovo UrbanDataset sulla Piattaforma SCP (servizio *push*). Lo schema tiene conto del nuovo ordinamento e delle regole per attivare un processo di validazione in funzione del successo o del fallimento di quello precedente. Sono inoltre riportate in maniera schematica le due interrogazioni al server web UrbanDataset WEB Library per richiedere il JSON-Schema e lo Schematron per eseguire, rispettivamente, le validazioni indicate nei punti 6. e 7. (quando abilitata) del precedente elenco.

3.4 Organizzazione Progetti

In questo paragrafo si vuole offrire una panoramica dei progetti software che permettono di implementare i diversi componenti del framework SCP.

Tali progetti software vengono condivisi tra i diversi team di lavoro tramite tecnologia GIT, che consente inoltre di effettuare continui backup del codice sorgente.

Tramite la tecnologia MAVEN è possibile invece esportare e importare in maniera automatica librerie, come dipendenze, e in questo modo massimizzare la modularità e il riuso.

Quello che segue è uno schema dell'organizzazione dei diversi progetti, di cui si offrirà subito dopo, una breve descrizione.

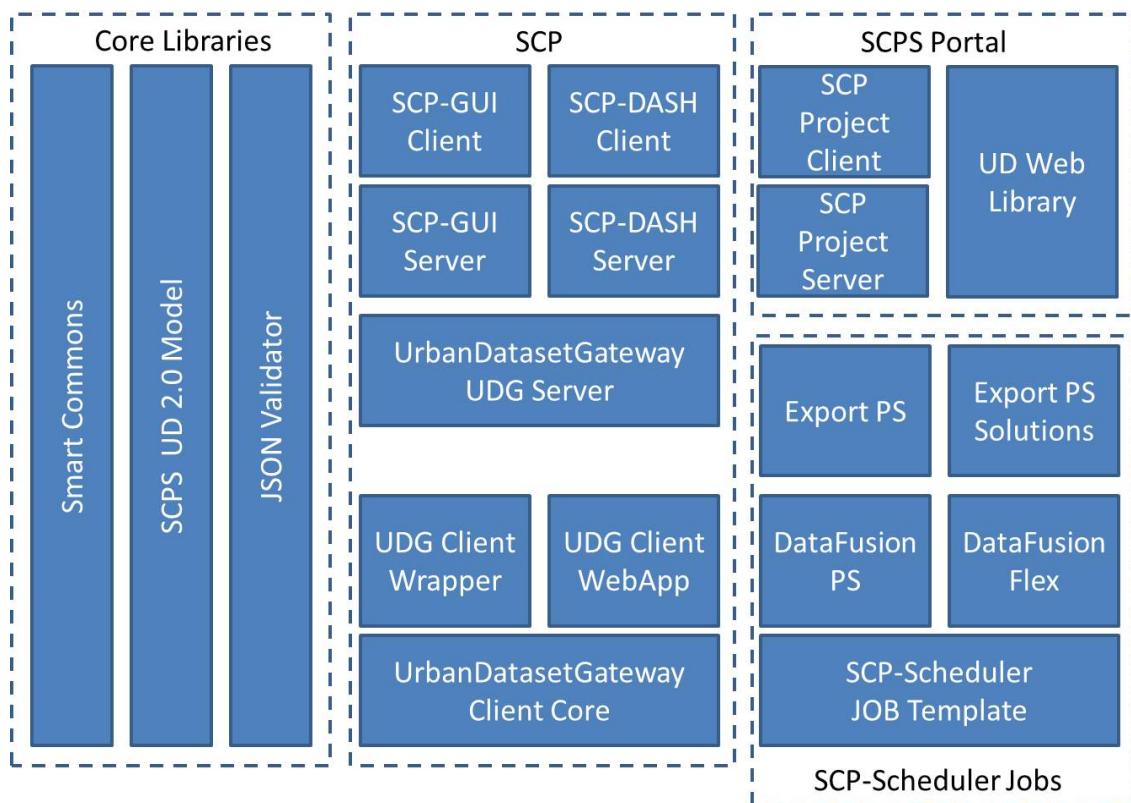


Figura 46. Schema Progetti SCP

Nella parte più a sinistra abbiamo le “**Core Libraries**” ovvero i progetti da cui vengono esportate librerie che vengono utilizzate da molti altri progetti. In questo gruppo troviamo:

- **Smart Commons:** libreria che contiene diverse utilities, p.es. per l’accesso ai file .properties, per l’accesso ai database, per trattare i timestamp nei diversi formati, ecc.;
- **SCPS UD 2.0 Model:** è il nuovo modello delle classi java per effettuare il marshalling/unmarshalling degli UrbanDataset dalla loro versione JSON;
- **JSON Validator:** è il validatore degli UrbanDataset che permette di effettuare due tipi di validazione degli UrbanDataset: la validazione della macro-struttura con JSON-Schema e la validazione delle proprietà con lo Schematron XML.

Il secondo gruppo, centrale, contiene i componenti principali del prototipo SCP 2.0 / inter-SCP:

- **SCP-GUI:** componente SCP-GUI 2.0, client (il front-end) e server (il back-end); è importante ricordare che questo componente contiene anche lo SCP-Scheduler;
- **SCP-DASH:** componente per definire Dashboard, client (il front-end) e server (il back-end);
- **UrbanDatasetGateway:** web service che implementa le specifiche “SCPS Communication” [11] per inviare o recuperare UrbanDataset dalla SCP;

- **UrbanDatasetGateway Client Core:** è il client del web service UrbanDatasetGateway, è la parte “core”, ovvero l’implementazione centrale delle chiamate ai diversi metodi, che verranno utilizzate dal Client Wrapper e dal Client WebApp;
- **UrbanDatasetGateway Client Wrapper:** è il client del web service UrbanDatasetGateway, utilizza l’UDG Client Core e vi aggiunge tutti gli input necessari pre-configurati per completare la chiamata;
- **UrbanDatasetGateway Client WebApp:** è il client del web service UrbanDatasetGateway in versione Web Application, utilizza l’UDG Client Core e vi aggiunge tutti gli input necessari prendendoli da interfaccia utente GUI web.

Sulla destra, in basso, abbiamo voluto riportare alcuni **JOB** che sono eseguiti dallo **SCP-Scheduler**:

- **SCP-Scheduler JOB-Template:** è la libreria “core” da importare in un progetto per creare dei nuovi JOB, eseguibili dallo SCP-Scheduler;

in questo gruppo vengono poi elencati **diversi JOB** che sono stati implementati, alcuni di questi sono quelli utilizzati nella sperimentazione descritta nel capitolo 2. Si sottolinea che ognuno di questi progetti “JOB” importa la suddetta libreria “SCP-Scheduler JOB-Template” ed esporta a sua volta una libreria .jar che rappresenta l’implementazione del JOB e viene quindi installata all’interno della SCP-GUI per essere individuata dallo SCP-Scheduler.

Sulla destra, in alto, abbiamo voluto riportare anche **altri progetti** importanti, tra cui:

- **SCP Project Website:** portale web ufficiale dello “SCP Project”, client (front-end) e server (back-end);
- **UrbanDataset Web Library:** è la web application che permette di navigare l’Ontologia, scaricare i template in formato JSON/XML degli UrbanDataset, nonché i relativi Schematron e guide.

3.5 Intersezioni con altre LA

Oltre agli sviluppi software di cui abbiamo parlato in questo capitolo, ve ne sono altri che sono afferenti ad altre linee di attività ma hanno una forte intersezione con il framework SCP.

3.5.1 Nuovo Identity Provider

Nel corso dell'ultimo anno è stato sviluppato un nuovo Identity Provider (IDP) che fornisce:

1. un sistema centralizzato per censire le piattaforme (ENEA) al fine di autenticare gli utenti;
2. il meccanismo di Single-Sign-On (SSO) per riutilizzare lo stesso token di sessione utente passando da una piattaforma all'altra.

Questa attività di progettazione e sviluppo è stata attuata nelle Linea di Attività 1.47 [18] e 1.48 [19], a cui si rimanda per una descrizione dettagliata del componente.

I componenti SCP-GUI 2.0 e UrbanDatasetGateway (e quindi anche la piattaforma inter-SCP 1.0 che li utilizza) sono stati aggiornati con successo al fine di utilizzare il nuovo Identity Provider.

Gli utenti hanno quindi una definizione centralizzata nel nuovo Identity Provider e, dunque, il passaggio da una SCP a un'altra, per gli utenti registrati, avviene senza necessità di effettuare un nuovo login apposito, tramite il meccanismo di SSO.

3.5.2 Web Application per modifica Ontologia

Per registrare i nuovi UrbanDataset o modificare quelli esistenti, è necessario agire direttamente sull'Ontologia [8], azione che richiede una certa conoscenza dell'ontologia OWL e che quindi rimane riservata agli esperti ENEA che ne curano la manutenzione e l'aggiornamento.

Si è reso quindi necessario realizzare uno strumento che:

- 1) semplificasse la gestione e l'implementazione degli UrbanDataset, evitando errori di inserimento e permettendolo anche a membri del team che non conoscono a fondo l'ontologia di modificarla;
- 2) permetta agli utenti esterni di proporre nuovi UrbanDataset.

A tal fine è stata sviluppata una web application per la modifica dell'Ontologia degli Urban Dataset.

La raccolta dei requisiti, avvenuta nella LA corrente, è riportata in Appendice B.

Lo sviluppo della web application è descritto nel Report Tecnico "Web Application per l'editing dell'Ontologia del Framework" inserito nella LA 1.23 "Strumenti basati su ontologia per il Framework per la Governance dei Dati Urbani Energetici" [17].

3.5.3 Connessione SCP-PELL a inter-SCP

La SCP-PELL (che è integrata con la piattaforma PELL, si veda report LA 1.20 [2]) è stata messa in connessione con la inter-SCP (si veda il capitolo 2 relativo alla sperimentazione).

A questo punto si aprono una serie di scenari molto interessanti in cui la piattaforma PELL (che recupera i consumi elettrici agendo su scala nazionale e calcola una serie di KPI energetici) può comunicare con la inter-SCP (che potenzialmente può recuperare dati da tutte le SCP distribuite su scala nazionale).

Sono stati classificati, nell'ambito dell'attività PELL, un certo numero di KPI energetici statici e dinamici (con "statici" si intendono i KPI il cui valore non cambia nel tempo, con "dinamici" si intendono i KPI che vengono calcolati con input il cui valore varia continuamente, p.es. gli ultimi consumi elettrici rilevati).

Nel corso dell'annualità è stato progettato il flusso dell'invio dei dati statici del PELL alla SCP sfruttando le apposite API REST. I KPI statici del PELL (Public Energy Living Lab) Illuminazione Pubblica sono calcolati sulla

base delle informazioni strutturali dell'impianto di illuminazione, e danno informazioni comparative sulle caratteristiche di tale impianto rispetto ai CAM (Criteri Ambientali Minimi) di riferimento.

Sono stati definiti cinque KPI, di cui quattro afferenti al POD (Point Of Delivery), punto in cui viene contabilizzato il consumo relativo alla porzione di area illuminata su cui tale POD è attestato, ed uno afferente alla zona omogenea, ovvero una porzione di strada avente alcune caratteristiche illuminotecniche uniformi.

I KPI definiti sono:

1. KPI tecnologico: indica la qualità dell'efficienza luminosa della sorgente adottata (cioè la quantità di luce prodotta in relazione alla potenza elettrica fornita), consente il confronto della prestazione luminosa della sorgente adottata rispetto al valore minimo previsto per la medesima tecnologia dai CAM. Questo KPI è calcolato per ciascuna zona.
2. KPI geometrico: indica se la potenza elettrica installata della sorgente luminosa per m² di superficie è all'interno dell'intervallo ammesso per la categoria illuminotecnica, definita dallo standard vigente (UNI 11248), riferita alla zona omogenea considerata (cioè l'area che necessita di uguali condizioni luminose per garantire la sicurezza della circolazione veicolare, pedonale e la fruizione degli spazi). Il calcolo viene effettuato a livello del reticolo di calcolo. Questo KPI è calcolato per ciascuna zona.
3. KPI confronto (vs BAU - Business As Usual): indica il vantaggio, in termini energetici, derivanti dall'utilizzo della sorgente installata rispetto all'utilizzo della BAU (la tecnologia usualmente adottata) affiancata anche dall'utilizzo di sistemi di riduzione del flusso in modalità adattiva). Il calcolo viene effettuato a livello del reticolo di calcolo. Questo KPI è calcolato per ciascuna zona.
4. KPI confronto (vs BAT - Best Available Technology): indica il vantaggio, in termini energetici, derivanti dall'utilizzo di una BAT in sostituzione della sorgente installata. La BAT costituisce la migliore tecnologia sul mercato, considerando l'impianto a piena potenza. Il calcolo viene effettuato a livello del reticolo di calcolo. Questo KPI è calcolato per ciascuna zona.
5. KPI dimming: indica il risparmio energetico derivante dall'utilizzo di strategie di dimming, statiche o adattive, rispetto al caso in cui l'impianto preso in considerazione funzioni sempre alla massima potenza, secondo i dati inseriti nella scheda censimento. Questo KPI è calcolato per ciascun POD.

Per ulteriori approfondimenti in merito ai KPI si rimanda al report Report RdS/PTR(2019)/008.

Al fine di rendere il formato dei KPI compatibile con la SCP, è stato definito un apposito UrbanDataset: **PodKPIpell**.

Poiché i valori dei KPI sono associati ai POD ed alle zone omogenee sono stati definiti i seguenti campi al fine di identificare univocamente i dati inviati alla SCP:

- **PODID**. Codice POD che identifica univocamente il punto di prelievo
- **TownCode**. Codice ISTAT del comune
- **KPIValue**. Valore calcolato per il KPI specifico.
- **PellKPICode**. KPI tecnici specifici del PELL IP
- **PLHomogeneousAreaID**. Identificatore della zona omogenea secondo criteri illuminotecnici di un'area urbana con illuminazione pubblica
- **EvaluationIndicator**. Valutazione del valore del KPI/indicatore es. 'ok', 'fuori soglia', 'non disponibile'

- **Period.** Periodo durante il quale sono stati rilevati i dati riportati nella riga

Il valore di questi KPI viene calcolato in maniera asincrona quando l'utente esegue la sottomissione di una scheda censimento ovvero il documento contenente le caratteristiche dell'impianto di illuminazione. Per cui l'invio degli UrbanDataset alla SCP avverrà contestualmente all'esito corretto del calcolo dei KPI sulla piattaforma PELL.

Tra i prossimi sviluppi, di cui parleremo nel par. 5.3 sono previste diverse dashboard per la visualizzazione di KPI, tra cui

- un'apposita dashboard per la SCP-PELL, per monitorare i consumi elettrici (a tal proposito è stato già sviluppato un nuovo metodo per l'UrbanDatasetGateway, *searchingRequestByProperty*, di cui si è parlato nel par. 3.3.1 per il recupero dei consumi elettrici data una specifica proprietà, che nel caso dei consumi elettrici del PELL sarebbe il codice ISTAT);
- un'estensione della dashboard della inter-SCP che permetta di recuperare e visualizzare i KPI prodotti dal PELL, sia quelli statici che dinamici.

4 Guida Utente per l'Adesione

In questo capitolo si descrive la guida utente per l'adesione al Framework SCP.

L'intero capitolo riporta il testo integrale della sezione "ADESIONE" presente sul sito web ufficiale dello "SCP Project" [3], dove sono pubblicate anche le specifiche *Smart City Platform Specification (SCPS) for interoperability layer* [4] e tutte le informazioni inerenti il prototipo Smart City Platform.

L'adesione allo Smart City Platform Project può avvenire tramite uno dei tre percorsi di adesione graduali per divenire nodo del framework. Si noti che ognuno di questi percorsi si riferisce a una particolare situazione architeturale descritta nella sezione "ARCHITETTURA" (questa sezione, come la precedente, è pubblicata sul sito web dello "SCP Project" [3] ed è stata riportata nel report della LA 1.19 del 2019 [1]).

In tale sezione sono descritti tutti i concetti chiave utilizzati (p.es. SCP, Solution, inter-SCP, ecc.)

I percorsi di adesione sono:

1. Adesione con una nuova SCP;
2. Adesione con una Solution a una SCP;
3. Adesione con una SCP alla inter-SCP.

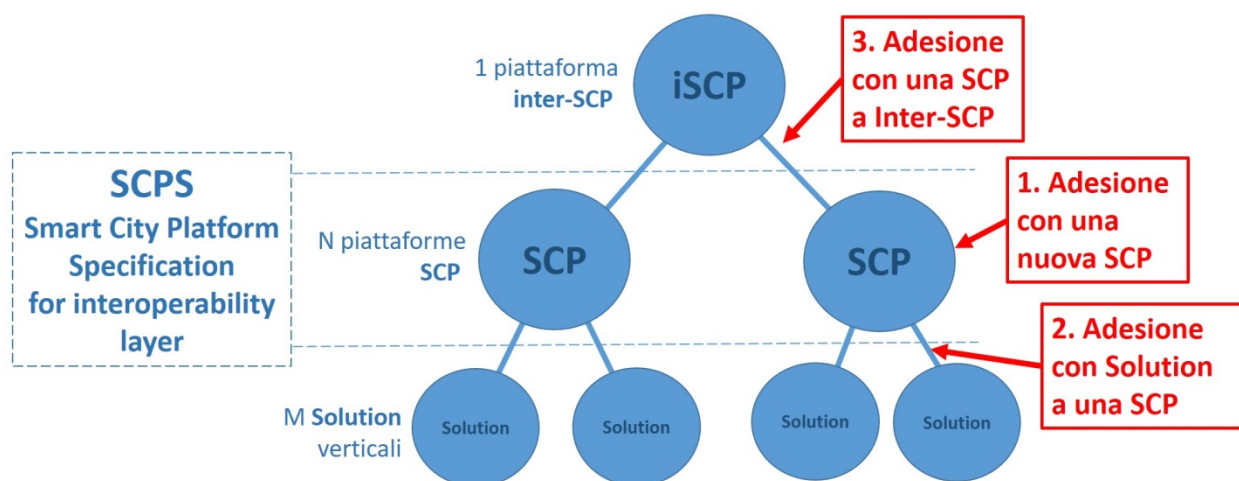


Figura 47. Adesione allo SCP Project

In particolare, si può intuire che:

- l' Adesione con una SCP (percorso 1) o con una Solution (percorso 2) si riferiscono ad un contesto su scala urbana (si veda architettura intra-city nella suddetta sezione "ARCHITETTURA" al par. 2.1);
- l' Adesione con una SCP alla inter-SCP (percorso 3) si riferisce ad una scala extra-urbana (si veda architettura inter-city nella suddetta sezione "ARCHITETTURA" al par. 2.2).

In ognuno di questi percorsi, per abilitare la comunicazione interoperabile, si farà ricorso alle già citate specifiche *Smart City Interoperability Specification (SCPS) for interoperability layer*.

4.1 Percorso 1 - Adesione con una nuova SCP

Questo percorso di adesione è ideato per chi intende dotarsi di una Smart City Platform (SCP), piattaforma ICT di raccolta dati e monitoraggio di una Smart City, seguendo l'approccio ENEA che si basa sull'utilizzo di specifiche pubbliche per l'interoperabilità, denominate *Smart City Platform Specification for interoperability layer (SCPS)*.

Si noti che le cinque specifiche SCPS non descrivono come implementare una piattaforma SCP ma descrivono tutti gli aspetti che devono essere implementati per ottenere la comunicazione interoperabile tramite l'interfaccia di comunicazione comune.

Questo approccio è stato pensato inizialmente per le municipalità che vogliono dotarsi di una piattaforma per la gestione della Smart City, connettendo le diverse Solution/piattaforme verticali esistenti, mediante un formato e protocollo condivisi. Ciò non preclude un utilizzo della SCP che sia diverso, su diversa scala o per una tematica verticale (p.es. si pensi a una ipotetica SCP per l'analisi della qualità dell'aria che riceve dati da diverse città).

In questa pagina descriveremo un percorso di adesione specifico che ha come obiettivo finale la definizione di una SCP funzionante. Si veda la sezione relativa all'ARCHITETTURA per una prima introduzione concettuale.

Requisito 0: Tipologia di Smart City Platform (SCP)

Le specifiche SCPS sono pubbliche e liberamente utilizzabili.
Per dotarsi di una SCP ci sono dunque due possibili vie:

1. utilizzare il prototipo SCP sviluppato da ENEA
 - a. su infrastruttura ICT ENEA;
 - b. su infrastruttura proprietaria;
2. sviluppare un proprio prototipo SCP basato su specifiche SCPS
 - a. da zero;
 - b. rendendo una propria piattaforma esistente, aderente alle specifiche.

N.B. In questo secondo caso, il soggetto che ha utilizzato gratuitamente le specifiche nel proprio prodotto software dovrà semplicemente indicare nel footer della propria GUI, l'utilizzo delle SCPS 2.0 definite da ENEA, con link alla pagina di riferimento.

Viene qui di seguito suggerito un percorso di adesione graduale alle cinque specifiche SCPS. Ogni specifica potrebbe essere adottata indipendentemente dalle altre, o gradualmente, ma soltanto aderendo a TUTTE le cinque specifiche è possibile ottenere la piena interoperabilità tra i sistemi.

Il prototipo SCP di ENEA supporta tutte le cinque specifiche SCPS.

Il requisito si ritiene soddisfatto nel momento in cui si è individuato:

1. il referente e responsabile della nuova SCP;
2. la tipologia di SCP che si intende utilizzare.

4.1.1 Passo 1 - Adesione alla specifica SCPS Functional 2.0

L'introduzione all'architettura di riferimento che descrive schematicamente la comunicazione tra Solution e SCP è parte della specifica SCPS Functional 2.0. Si richiede, come primo passo, di aderire a tale specifica, allo scopo di condividere l'architettura di riferimento e i concetti chiave (come SCP, Solution, UrbanDataset, Ontologia, Registry, ecc.)

Si veda anche il Glossario che contiene le definizioni dei concetti chiave ed è parte integrante di questa specifica.

Lo sviluppo di una SCP, basata su SCPS Functional 2.0, richiede di implementare tutte le macro-funzionalità descritte negli use-case descritti, che a loro volta possono richiamare e implementare le altre specifiche SCPS richiamate nei prossimi passi.

Risultato dell'adesione a questa specifica è una piena condivisione dell'architettura e dei concetti chiave. Se la SCP utilizzata NON è il prototipo ENEA, deve essere compiuto uno studio di fattibilità sull'adozione delle cinque specifiche SCPS, alla cui conclusione il responsabile della nuova SCP conferma o meno la fattibilità di sviluppare le macro-funzionalità descritte nella specifica (funzionalità che trovano compimento con i passi successivi).

Nel momento in cui la potenziale nuova SCP entra a far parte della rete del framework "SCP Project", utilizzando l'ontologia centrale e aderendo alle cinque specifiche, ENEA assegna un identificatore univoco alla SCP per censirla centralmente e permettere che i propri UD siano identificabili univocamente (come verrà spiegato nel PASSO 4).

È possibile contattare via email il personale ENEA (dal menù principale, [AIUTO] > [FEEDBACK]), per valutare l'avvio di un percorso di adesione o ricevere assistenza sull'adozione delle specifiche SCPS.

4.1.2 Passo 2 - Adesione alla specifica SCPS Semantic 2.0

Questo secondo passo richiede di utilizzare l'Ontologia di ENEA per avere i riferimenti univoci delle definizioni UrbanDataset e, se queste definizioni non fossero sufficienti o fossero assenti, partecipare al processo di estensione e definizione degli UD, contattando il personale ENEA (dal menù principale, [AIUTO] > [FEEDBACK]).

Si veda, come introduzione, la descrizione concettuale e schematica relativa al ruolo dell'Ontologia per poi approfondire con la specifica vera e propria SCPS Semantic 2.0.

Un valido strumento per effettuare una prima definizione di nuovi UrbanDataset è la Scheda Proposta UrbanDataset. Tale definizione sarà valutata dal team ENEA per poi valutarne la registrazione nell'Ontologia centrale.

Risultato dell'adesione a questa specifica sono le definizioni di UrbanDataset correttamente registrati (da ENEA) nell'Ontologia centrale, nonché la loro configurazione nella SCP, con cui li si dichiara come supportati.

4.1.3 Passo 3 - Adesione alla specifica SCPS Information 2.0

L'adesione alla specifica SCPS Information 2.0 comporta la gestione degli UrbanDataset secondo il formato condiviso, ovvero l'implementazione della relativa validazione.

N.B. Il prototipo SCP di ENEA supporta il formato SCPS Information 2.0 per rappresentare gli UrbanDataset nel solo formato JSON (non XML).

La validazione viene eseguita su due livelli

1. validazione della macro-struttura comune a tutti gli UrbanDataset, grazie al json-schema;
2. validazione delle specifiche proprietà dello specifico UrbanDataset, grazie al relativo schematron (questa validazione è in corso di integrazione nel prototipo SCP di ENEA).

Nella sezione Strumenti è possibile accedere alle risorse per effettuare queste validazioni:

1. URL dello json-schema 2.0;
2. Navigatore dell'Ontologia UrbanDataset Web Library, con il quale è possibile navigare tra gli UrbanDataset definiti ed effettuare il download del relativo schematron.

Risultato dell'adesione a questa specifica è la capacità, da parte della SCP in oggetto, di validare gli UrbanDataset sui due livelli di validazione forniti.

4.1.4 Passo 4 - Adesione alla specifica SCPS Collaboration 2.0

L'adesione alla specifica SCPS Collaboration 2.0 comporta l'implementazione del meccanismo che permette di registrare nella SCP:

- gli UrbanDataset (tra quelli definiti nell'Ontologia) che saranno supportati;
- le Solution verticali che si conetteranno alla SCP;
- le produzioni, intese come collaborazioni dove le Solution invieranno UD alla SCP;
- gli accessi, intesi come permessi per le solution di recuperare particolari UD dalla SCP.

Si suggerisce di implementare tale meccanismo tramite una interfaccia utente GUI e un database relazionale Registry così come suggerito nell'architettura di riferimento delle SCPS Functional 2.0.

Il prototipo SCP di ENEA implementa questi aspetti:

- con il componente SCP-GUI, sviluppato con tecnologia Angular/J2EE;
- con la banca dati Registry, implementata con DBMS MySQL.

Risultato dell'adesione a questa specifica è la capacità, da parte della SCP in oggetto, di registrare correttamente UrbanDataset, Solution, Produzioni e Accessi, utilizzando le convenzioni di identificazione, quali: *scp-id*, *UrbanDataset-ID*, *solution_id*, *resource_id*.

N.B. l'identificatore *scp-id* viene assegnato da ENEA al PASSO 1.

4.1.5 Passo 5 - Adesione alla specifica SCPS Communication 2.0

L'adesione alla specifica SCPS Communication 2.0 comporta l'implementazione del web service RESTful *UrbanDatasetGetaway*. Tale web service permette l'invio e il recupero di UrbanDataset presso la SCP. Gli UrbanDataset vengono memorizzati in una banca dati apposita.

Il prototipo SCP di ENEA implementa questi aspetti:

- con un web service RESTful sviluppato con tecnologia J2EE;
- con la banca dati UD DB, implementata con Elastic Search.

Si noti che, in questo passo, si sta implementando la comunicazione interoperabile vera e propria, che farà diretto riferimento alle configurazioni delle collaborazioni, espresse mediante identificatori di risorse (si veda il PASSO 4), in particolare:

- l'identificatore *solution_id* deve essere usato per valorizzare il campo dell'UrbanDataset *UrbanDataset/context/producer*;
- l'identificatore *resource_id* deve essere usato nelle diverse chiamate al web service.

Risultato dell'adesione a questa specifica è la capacità, da parte della SCP in oggetto, di ricevere correttamente UrbanDataset o consentire l'accesso a UrbanDataset, via web service, validando la sintassi JSON di tali file (si veda PASSO 3) e validando i permessi in produzione e accesso configurati in precedenza (si veda PASSO 4).

4.1.6 Conclusioni Percorso 1

L'implementazione di SCP, realizzata aderendo alle cinque specifiche, deve permettere a un utente di tipo "Solution" di poter eseguire il percorso completo di Adesione con una Solution a una SCP (descritto nel par. 4.2); in caso ciò non sia possibile, l'implementazione non può dirsi totalmente SCPS-compliant.

Una volta che l'adesione alle cinque specifiche SCPS è stata ultimata, nel caso in cui NON si stia utilizzando il prototipo SCP di ENEA, si dovrà richiedere ad ENEA la "verifica SCP di adesione SCPS".

Per avviare il percorso di adesione o per ricevere assistenza, è possibile contattare via email il personale ENEA (dal menù principale, AIUTO > FEEDBACK).

4.2 Percorso 2 - Adesione con una Solution a una SCP

Questo percorso di adesione prevede che una Solution/Piattaforma di gestione di un contesto applicativo verticale (p.es. mobilità, illuminazione pubblica, ecc.) si debba connettere a una SmartCityPlatform (SCP), piattaforma di gestione della città/distretto (si veda l'architettura relativa alla comunicazione intra-city tra Solution e SCP) allo scopo di produrre o accedere a UrbanDataset (formato dati comune per rappresentare dati urbani relativi alla gestione verticale).

Requisito 0 - Accordo formale tra le parti

Per avviare questo percorso è necessario stabilire un accordo formale preso tra le parti coinvolte:

- Referente della Solution/Piattaforma verticale;
- Referente della SmartCityPlatform (SCP).

Tale accordo tra le parti coinvolte può essere definito, per esempio:

- all'interno di un progetto in cui ENEA mette a disposizione il prototipo SCP basato su specifiche SCPS;
- in un bando di gara indetto da una municipalità, che richiede alla società vincitrice l'adesione alle specifiche SCPS per l'invio di alcuni KPI relativi alla gestione che verrà fatta. In questo modo, oltre la gestione del contesto applicativo verticale (p.es. mobilità, illuminazione pubblica, rifiuti, ecc.) con una particolare Solution Verticale (p.es. piattaforma mobilità, piattaforma di gestione distribuzione energia elettrica, ecc.) la società dovrà inviare anche alcuni dati relativi alla gestione, su base tipicamente quotidiana o settimanale.

L'accordo formale stabilisce una serie di azioni che le parti saranno tenute ad eseguire, entro un periodo di tempo concordato, in modo tale da instaurare la comunicazione interoperabile tra la SCP centrale orizzontale e la Solution/Piattaforma verticale. Queste azioni sono schematizzate in una serie di passi, riportati in questa tabella, utilizzando le specifiche SCPS.

L'accordo formale può, per semplicità, riprendere il seguente schema di passi.

4.2.1 Passo 1 - Adesione all'Architettura di Riferimento

La comunicazione interoperabile SCPS-based tra Solution e SCP è descritta attraverso l'architettura di riferimento nel capitolo 2 della specifica SCPS Functional 2.0. Si richiede dunque, come primo passo informale, di leggere tale specifica, allo scopo di condividere l'architettura di riferimento e i concetti chiave (come SCP, Solution, UrbanDataset, Ontologia, Registry, ecc.)

Sebbene informale, si tratta di un passo da non sottovalutare, in quanto getta le fondamenta a un dialogo privo di ambiguità.

Si veda anche il Glossario che contiene le definizioni dei concetti chiave ed è parte integrante della specifica SCPS Functional 2.0.

4.2.2 Passo 2 - Analisi e Progettazione dei flussi dati

Nel caso in cui l'accordo formale tra le parti (si veda REQUISITO) non abbia previsto e non specifichi già i dati da utilizzare nella comunicazione, si dovrà procedere con l'analisi dei flussi di dati che si intendono implementare.

In altre parole, dovranno essere individuati i dataset (denominati UrbanDataset) per rappresentare i dati di scambio.

Ciò può avvenire tramite:

- l'individuazione degli UrbanDataset necessari, tra quelli già definiti nell'Ontologia centrale, ed eventualmente proporre un'estensione;
- la progettazione di nuovi UrbanDataset, non ancora presenti nell'Ontologia centrale.

Il risultato dell'analisi dei flussi dati è tipicamente un caso studio che qualifica la Solution (o Piattaforma Verticale) e specifica

- quali dati sono recuperati dalla rete applicativa di sorgenti dati (p.es. sensori hardware) e
- quali dati verranno aggregati e rappresentati in UrbanDataset da inviare alla SCP.

Nel caso di modifica o progettazione di nuovi UrbanDataset, dovranno essere coinvolti gli specialisti di settore (ENEA), individuando, per ciascun dataset:

- un nome,
- una categorizzazione (tra le categorie e sotto-categorie fornite),
- la finestra temporale su cui è aggregato il dato,
- le proprietà che lo compongono

Questa progettazione viene solitamente fatta utilizzando due strumenti presenti nella sezione Strumenti:

- il Navigatore dell'Ontologia UrbanDataset Web Library, con il quale si possono vedere gli UrbanDataset definiti, organizzati in categorie e sottocategorie;
- la Scheda Proposta UrbanDataset.

Al termine di questo passo, gli UrbanDataset necessari alla comunicazione saranno correttamente certificati e registrati da ENEA nell'Ontologia centrale.

N.B. Ogni SCP deve specificare nel proprio Registry quali UrbanDataset, tra quelli definiti nell'Ontologia, devono essere supportati. Questa configurazione deve essere svolta dall'utente ADMINISTRATOR, tramite la SCP-GUI, nella sezione DATASET.

4.2.3 Passo 3 - Esportazione degli UrbanDataset

Una volta che gli UrbanDataset necessari alla comunicazione in oggetto, tra Solution e SCP, sono correttamente definiti nell'Ontologia centrale (si veda PASSO 2), sarà possibile effettuare il download dei relativi template in forma JSON o XML.

N.B. Il prototipo SCP di ENEA supporta, al momento, solo il formato JSON.

Per effettuare il download è possibile utilizzare il Navigatore dell'Ontologia (UrbanDataset Web Library), dopo aver individuato l'UrbanDataset desiderato, tramite i link "JSON Message Template" o "XML Message Template".

La struttura sintattica degli UrbanDataset, comune a entrambi i formati JSON e XML, è descritta nel dettaglio tramite il "Modello Astratto dei Dati" che fa parte della specifica SCPS Information 2.0.

I template JSON o XML degli UrbanDataset saranno utilizzati per rappresentare i dati nello scambio dati. La Solution/Piattaforma verticale, quindi, dovrà implementare l'esportazione dei dati dal proprio sistema locale nel formato UrbanDataset, di cui possiede il template, e verificare che tale file sia conforme alle specifiche superando i test di validazione (Schema e Schematron).

N.B. il campo dell'UrbanDataset *UrbanDataset/context/producer*, che deve essere aggiornato nel template, verrà spiegato nel PASSO 4.

4.2.4 Passo 4 - Configurazione della Collaborazione sulla SCP

A questo punto, è necessario configurare la collaborazione tra la Solution Verticale in oggetto e la SmartCityPlatform (SCP) secondo la specifica SCPS Collaboration 2.0.

Questa azione è fatta dall'utente ADMINISTRATOR della SCP.

Si noti che nel prototipo SCP di ENEA questa configurazione può essere fatta come proposta dall'utente SOLUTION per poi essere abilitata dall'utente ADMINISTRATOR in un secondo tempo. La nuova collaborazione viene specificata tramite la SCP-GUI nelle sezioni PRODUCTION (per le produzioni di UD) o ACCESS (per l'accesso a UD); la configurazione verrà salvata opportunamente nel Registry della SCP.

La configurazione della collaborazione implica:

- a. la registrazione della Solution/Piattaforma verticale presso la SCP,

a ciò consegue la generazione automatica di un *solution_id* da inserire negli UrbanDataset nel campo *UrbanDataset/context/producer*, quando la Solution verticale comincia a produrre e inviare UD verso la SCP;
- b. la specificazione presso la SCP delle associazioni "Solution – UrbanDataset" (in produzione), ovvero quali UrbanDataset saranno generati e inviati dalla Solution/Piattaforma verticale alla SCP e in quale modo (protocollo, frequenza di invio, modalità di storage, ecc.).

a ciò consegue la generazione automatica di un *resource_id*, identificatore dell'UD prodotto da una specifica Solution in uno specifico periodo temporale, in una specifica SCP; tale *resource_id* dovrà essere utilizzato nella comunicazione (si veda PASSO 5);
- c. la specificazione presso la SCP delle associazioni "Solution – UrbanDataset" (in accesso), ovvero quali UrbanDataset potranno essere acceduti da una Solution/Piattaforma verticale registrata nella SCP e in quale modo (protocollo, frequenza di accesso, ecc.).

in questo caso la Solution/Piattaforma verticale potrà accedere a uno specifico UrbanDataset tramite riferimento al *resource_id*.

4.2.5 Passo 5 - Implementazione della Comunicazione

Il passo finale è l'implementazione della comunicazione tra la Solution Verticale in oggetto e la SCP.

La Solution/Piattaforma verticale, dopo aver esportato i dati dal proprio sistema locale e averli rappresentati con il formato UrbanDataset (si veda PASSO 3) dovrà inviare tali dati richiamando il web service *UrbanDatasetGateway* pubblicato sulla SCP di riferimento.

L'*UrbanDatasetGateway* è un web service RESTful la cui interfaccia è descritta dettagliatamente nella specifica SCPS Communication 2.0. Per richiamare tale web service la solution dovrà implementare un client con i metodi che dovrà utilizzare (tipicamente *push* per l'invio di UD e una *request* per il recupero di UD).

N.B. uno dei parametri richiesti in ogni chiamata è il *resource_id* configurato al PASSO 4.

Una volta implementato il client web service, nel caso l'invio o il recupero di UD sia configurato come periodico, la Solution dovrà automatizzare il processo secondo la frequenza stabilita e concordata.

4.2.6 Conclusioni Percorso 2

Una volta ultimati i 5 PASSI, la Solution Verticale si impegna a:

- monitorare periodicamente, tramite l'interfaccia grafica della SCP, il corretto invio/recupero di UD;
- provvedere a rimettere in funziona il meccanismo di invio secondo la periodicità stabilita, qualora dal monitoraggio della SCP dovessero risultare delle irregolarità di invio;
- mantenere attivo il meccanismo per il tempo concordato (stabilito nell'accordo formale tra le parti);
- ad effettuare eventuali aggiornamenti necessari al buon funzionamento del sistema durante il periodo di collaborazione concordato.

Per ricevere assistenza sui percorsi di adesione, è possibile contattare via email il personale ENEA (dal menù principale, [AIUTO] > [FEEDBACK]).

4.3 Percorso 3 - Adesione con una SCP alla iSCP

Questo percorso di adesione prevede che una SmartCityPlatform (SCP) di una città/distretto si debba connettere alla inter-SmartCityPlatform (abbreviato in "inter-SCP" o "iSCP"), piattaforma ICT sviluppata da ENEA, agente su scala nazionale (si veda l'architettura relativa alla comunicazione inter-city tra SCP e iSCP) allo scopo di produrre o accedere a UrbanDataset (formato dati comune per rappresentare KPI o dati relativi alla gestione Smart City).

Si noti che i **passi descritti in questo terzo percorso, riprendono in sostanza ciò che è stato fatto per configurare il caso studio pilota** nella sperimentazione del capitolo 2 di questo report.

Requisito 0 - Accordo formale tra le parti

Per avviare questo percorso è necessario stabilire un accordo formale tra le parti coinvolte:

- Referente della SmartCityPlatform (SCP);
- Referente della inter-SmartCityPlatform (inter-SCP), ovvero ENEA.

Al momento, ogni SCP definita con il prototipo di ENEA, su server di ENEA, è automaticamente idonea per avviare la connessione alla inter-SCP. Eventuali SCP sviluppate da terze parti (ma che comunque abbiano adottato le specifiche SCPS) o SCP installate su server esterni alla rete ENEA, saranno valutate prima di intraprendere il percorso di adesione per connettere una SCP urbana alla inter-SCP agente su scala nazionale.

Per richiedere tutte le informazioni del caso, utilizzare il contatto che si trova nel menù principale in [Aiuto] > [Feedback] (presente anche nel footer di ogni pagina).

Prima di intraprendere questo percorso di adesione, è necessario che la SCP in oggetto abbia effettuato con successo i precedenti due percorsi di adesione:

1. Adesione con una nuova SCP;
2. Adesione con una Solution a una SCP.

L'accordo formale esprime, tra le altre cose, la lista degli UrbanDataset che la SCP invierà alla inter-SCP (p.es. l'UD "Platform Status").

Una volta che l'accordo formale è raggiunto e i requisiti sono stati verificati, la municipalità (o società) che gestisce la SCP dovrà compiere una serie di passi per instaurare la comunicazione con la inter-SCP utilizzando le specifiche SCPS (ovvero i passi seguenti).

Si noti che, nel caso in cui la SCP sia il prototipo SCP di ENEA, ed esso sia installato e configurato su server ENEA, per abilitare la comunicazione tra SCP e inter-SCP, **TUTTI I PASSI SEGUENTI** sono già configurati e implementati di default nei prototipi SCP e inter-SCP di ENEA per tutti gli UrbanDataset concordati nell'accordo formale tra le parti.

4.3.1 Passo 1 - Adesione all'Architettura di Riferimento

La comunicazione interoperabile SCPS-based tra SCP e inter-SCP (iSCP) è equivalente alla comunicazione tra Solution e SCP così come descritta attraverso l'architettura di riferimento nel capitolo 2 della specifica SCPS Functional 2.0. Si richiede dunque, come primo passo informale, di leggere tale specifica, allo scopo di condividere l'architettura di riferimento e i concetti chiave (come SCP, Solution, UrbanDataset, Ontologia, Registry, ecc.)

Si noti che la comunicazione inter-city tra SCP e iSCP (N SCP connesse a 1 iSCP) è analoga alla comunicazione intra-city tra Solution e SCP (N Solution connesse a 1 SCP): in entrambi i casi si utilizzano le specifiche SCPS per implementare la comunicazione interoperabile.

Vi è però una differenza architetturale:

1. nella comunicazione intra-city, tra Solution e SCP, è la Solution ad inviare gli UD alla SCP;
2. nella comunicazione inter-city, tra SCP e inter-SCP, è la inter-SCP a recuperare gli UD dalla SCP.

Vedremo come ciò sarà configurato nel PASSO 4, relativo alla configurazione della collaborazione.

4.3.2 Passo 2 - Progettazione dei flussi dati

L'accordo formale preso con ENEA (si veda REQUISITO 0) prevede che la SCP coinvolta invii uno o più UrbanDataset alla inter-SCP.

Gli UrbanDataset che le SCP inviano alla inter-SCP sono definiti da ENEA.

Diversamente, il referente della SCP:

1. dovrà valutare se tutti gli UrbanDataset richiesti sono realizzabili, ovvero se tutti i dati e KPI che dovranno essere inviati alla inter-SCP sono dati recuperabili o KPI calcolabili;
2. nel caso gli UD richiesti non siano realizzabili o siano insufficienti, potrà contribuire al processo di modifica e/o definizione di nuovi UrbanDataset.

Nel caso di modifica o creazione di UrbanDataset, dovranno essere coinvolti anche gli specialisti di settore (ENEA), individuando, per ciascun dataset:

1. un nome,
2. una categorizzazione (tra le categorie e sotto-categorie fornite),
3. la finestra temporale su cui è aggregato il dato,
4. le proprietà che lo compongono.

Questa progettazione viene solitamente fatta utilizzando due strumenti presenti nella sezione Strumenti:

- il Navigatore dell'Ontologia UrbanDataset Web Library, con il quale si possono vedere gli UrbanDataset definiti, organizzati in categorie e sottocategorie;
- la Scheda Proposta UrbanDataset.

Al termine di questo passo, gli UrbanDataset necessari alla comunicazione saranno correttamente certificati e registrati da ENEA nell'Ontologia centrale.

N.B. la inter-SCP dovrà inoltre specificare nel proprio Registry quali UrbanDataset, tra quelli definiti nell'Ontologia, devono essere supportati. Questa configurazione viene svolta da ENEA come utente ADMINISTRATOR della inter-SCP, tramite la SCP-GUI, nella sezione DATASET.

4.3.3 Passo 3 - Esportazione degli UrbanDataset

Una volta che gli UrbanDataset necessari alla comunicazione in oggetto, tra SCP e iSCP, sono correttamente definiti nell'Ontologia centrale (si veda PASSO 2), sarà possibile effettuare il download dei relativi template in formato JSON o XML.

N.B. Il prototipo SCP di ENEA supporta, al momento, solo il formato JSON.

Per effettuare il download è possibile utilizzare il Navigatore dell'Ontologia (UrbanDataset Web Library), dopo aver individuato l'UrbanDataset desiderato, tramite i link "JSON Message Template" o "XML Message Template".

La struttura sintattica degli UrbanDataset, comune a entrambi i formati JSON e XML, è descritta nel dettaglio tramite il "Modello Astratto dei Dati" che fa parte della specifica SCPS Information 2.0.

I template JSON o XML degli UrbanDataset saranno utilizzati per rappresentare i dati nello scambio dati. La SCP, quindi, nell'ottica di abilitare la comunicazione con la inter-SCP, dovrà implementare l'esportazione dei dati dal proprio sistema locale nel formato UrbanDataset, di cui possiede il template, e verificare che tale file sia conforme alle specifiche superando i test di validazione (Schema e Schematron).

N.B. il campo dell'UrbanDataset *UrbanDataset/context/producer*, che deve essere aggiornato nel template, verrà spiegato nel PASSO 4.

4.3.4 Passo 4 - Configurazione della Collaborazione

A questo punto, è necessario configurare la collaborazione tra la SCP e la inter-SCP (iSCP) secondo la specifica SCPS Collaboration 2.0.

Vi sono due configurazioni, relative a due collaborazioni, da effettuare:

1. lato SCP (PASSO 4.1) e
2. lato inter-SCP (PASSO 4.2).

Passo 4.1 - Configurazione della Collaborazione lato SCP

Questa configurazione viene eseguita per permettere al modulo di esportazione dell'UrbanDataset (descritto al PASSO 3), di essere identificato dalla SCP in oggetto come SERVICE (equiparabile a un utente di tipo SOLUTION) e così di poter pubblicare l'UD esportato sulla SCP, rendendolo disponibile al recupero da parte della inter-SCP.

Questa azione è fatta dall'utente ADMINISTRATOR della SCP tramite la SCP-GUI.

La configurazione della collaborazione implica la pubblicazione della risorsa con un *resource_id* univoco a cui la inter-SCP deve avere accesso.

Descriviamo sinteticamente in cosa consiste la configurazione di questa collaborazione nel prototipo SCP (in modo tale possa essere da esempio per piattaforme sviluppate da terze parti):

- a. la registrazione del SERVICE presso la SCP,
N.B. ciò genera il *solution_id* da inserire negli UD nel campo *UrbanDataset/context/producer*, quando il SERVICE produce UD (nello stesso modo in cui accade per le Solution verticali della SCP);
- b. la specificazione dei permessi in produzione presso la SCP, esprimendo l'associazione "SERVICE – UrbanDataset", ovvero quali UrbanDataset saranno generati e pubblicati sulla SCP dal modulo SERVICE di esportazione e in quale modo (protocollo, frequenza di invio, modalità di storage, ecc.);
- c. si noti che nella configurazione dell'associazione "SERVICE – UrbanDataset" (in produzione) viene automaticamente associato un *resource_id*, identificatore dell'UD prodotto dallo specifico SERVICE (o SOLUTION), in uno specifico periodo temporale, per la specifica SCP (seguendo la convenzione sintattica descritta nella specifica SCPS Collaboration 2.0; tale *resource_id* viene utilizzato sia per effettuare la push dell'UD dal modulo SERVICE sul sistema locale (si veda PASSO 5.1), sia per fornire il conseguente accesso alla inter-SCP sulla specifica risorsa (si veda il punto seguente) e sia per configurare la comunicazione sulla inter-SCP (si veda PASSI 4.2 e 5.2);
- d. per permettere alla inter-SCP di accedere all'UD pubblicato, bisognerà definire l'accesso relativo (salvo l'UD non sia dichiarato OPENDATA); si noti che, in questo caso, la iSCP viene vista dalla SCP come una normale Solution/Piattaforma verticale, basterà quindi definire un'associazione "SOLUTION – UrbanDataset" (in accesso), ovvero quale UrbanDataset potrà essere acceduto dalla iSCP (che deve risultare registrata presso la SCP), e con quale protocollo.

A questo punto il SERVICE di esportazione è abilitato a pubblicare l'UrbanDataset che la iSCP potrà recuperare.

Passo 4.2 - Configurazione della Collaborazione lato inter-SCP

Questa configurazione viene eseguita per permettere alla inter-SCP (iSCP) di recuperare l'UrbanDataset esportato e pubblicato dalla SCP (si vedano PASSI 3 e 4.1).

Questa azione è fatta dall'utente ADMINISTRATOR della inter-SCP (ovvero il responsabile di ENEA) tramite la SCP-GUI della inter-SCP.

La configurazione della collaborazione lato inter-SCP implica:

- a. registrazione della SCP (vista come Solution/Piattaforma verticale) presso la iSCP,
N.B. ciò permette di generare il *solution_id* da inserire negli UrbanDataset nel campo *UrbanDataset/context/producer*, quando la SCP produce UD per la iSCP;
- b. specificazione presso la iSCP delle associazioni "SCP – UrbanDataset" (in produzione), ovvero quali UrbanDataset saranno generati dalla SCP (vista come Solution/Piattaforma verticale) alla iSCP e con quali parametri (protocollo, endpoint del web service, *resource_id* remoto (configurato al PASSO 4.1), frequenza di produzione, modalità di storage, ecc.);
- c. si noti che anche alla configurazione dell'associazione "SCP – UrbanDataset" (in produzione, verso la iSCP) viene automaticamente associato un *resource_id*, identificatore dell'UD prodotto da una specifica SCP in uno specifico periodo temporale, verso la inter-SCP; tale *resource_id* sarà quello utilizzato nella comunicazione (descritta al PASSO 5);

N.B. la presenza di un diverso *resource_id* sembrerebbe una complessità aggiuntiva non necessaria. In realtà i due *resource_id*, anche se afferiscono idealmente agli stessi UrbanDataset, sono in realtà ben distinti:

1. il *resource_id* descritto nel PASSO 4.1 è l'identificatore dell'UrbanDataset esportato e immagazzinato nella SCP in oggetto;

2. il *resource_id* descritto nel PASSO 4.2 è l'identificatore dell'UrbanDataset recuperato e immagazzinato nella inter-SCP.

4.3.5 Passo 5 - Implementazione della Comunicazione

Il passo finale è l'implementazione della comunicazione tra SCP in oggetto e la inter-SCP.

Vi sono due comunicazioni da implementare:

1. una relativa al flusso dati "SERVICE - SCP" (PASSO 5.1) e
2. una relativa al flusso dati "SCP - iSCP" (PASSO 5.2).

Passo 5.1 - Comunicazione SERVICE - SCP

La SCP, dopo aver esportato i dati dal proprio sistema locale e averli rappresentati con il formato UrbanDataset (si veda PASSO 3), e dopo aver preparato la configurazione della produzione da parte del modulo SERVICE (si veda PASSO 4.1), dovrà pubblicare tale dati tramite il web service UrbanDatasetGateway sulla SCP di riferimento per renderli disponibili alla inter-SCP.

Il modulo SERVICE effettuerà dunque una chiamata al metodo *UrbanDatasetGateway.push* della SCP fornendo in input l'UrbanDataset esportato e il *resource_id* configurato appositamente (PASSO 4.1).

L'UrbanDatasetGateway è un web service RESTful la cui interfaccia è descritta dettagliatamente nella specifica SCPS Communication 2.0.

Per richiamare tale web service la solution dovrà implementare un client implementando i metodi che dovrà utilizzare. Una volta implementato il client web service, nel caso l'invio o il recupero di UD sia configurato come periodico, la Solution dovrà automatizzare il processo secondo la frequenza stabilita e concordata.

A questo punto l'UD è pubblicato e pronto per essere recuperato dalla iSCP.

Passo 5.2 - Comunicazione SCP - iSCP

Il passo finale è l'implementazione della comunicazione tra SCP e inter-SCP.

Come abbiamo descritto, a questo punto la SCP esporta e pubblica l'UD concordato periodicamente (si veda PASSO 5.1). A questo punto la iSCP, seguendo la configurazione della collaborazione preparata in 4.2, può prelevare l'UD periodicamente, tramite una chiamata presso la SCP al metodo *L'UrbanDatasetGateway.lastRequest* che restituisce l'ultimo UD prodotto per un particolare *resource_id*. La inter-SCP, a questo punto, automatizza il processo secondo la frequenza stabilita e concordata.

N.B. uno dei parametri richiesti è il *resource_id* configurato al PASSO 4.2, che identifica univocamente l'UrbanDataset da recuperare.

L'UrbanDatasetGateway è un web service RESTful la cui interfaccia è descritta dettagliatamente nella specifica SCPS Communication 2.0.

4.3.6 Conclusioni Percorso 3

In conclusione, una volta ultimati i 5 PASSI, il referente della SCP si impegna a:

1. monitorare periodicamente, tramite l'interfaccia grafica della SCP-GUI (nel caso del prototipo ENEA), la corretta produzione dell'UD da parte dei SERVICE interni e il corretto accesso agli UD da parte della inter-SCP;
2. qualora dal monitoraggio dovessero risultare delle anomalie, provvedere a ripristinare il meccanismo di produzione dell'UD da parte del SERVICE, secondo la periodicità stabilita;
3. mantenere funzionante il meccanismo di comunicazione SCP-iSCP per il tempo concordato (stabilito nell'accordo formale tra le parti);
4. effettuare eventuali aggiornamenti necessari al funzionamento del sistema, durante il periodo di collaborazione concordato.

Per ricevere assistenza sui percorsi di adesione, è possibile contattare via email il personale ENEA (dal menù principale, [AIUTO] > [FEEDBACK]).

5 Sviluppi in Corso e Applicazioni Future

Parallelamente all'implementazione del caso studio pilota (che, lo ricordiamo è stato l'obiettivo principale, raggiunto, per la corrente LA 1.21) altre attività sono state avviate e, sebbene non abbiano trovato ancora conclusione, le vogliamo segnalare in quanto inerenti al lavoro effettuato.

5.1 Caso studio urbano

Oltre al caso studio pilota "Da SCP-Casaccia a inter-SCP", oggetto della sperimentazione, è stato definito nel Report della LA 1.20 [2] un secondo caso studio denominato "Da SCP-Casaccia a SCP-Dashboard".

È stato avviato lo sviluppo per implementare anche questo secondo caso studio, strettamente urbano, il cui obiettivo è di definire un primo cruscotto (dashboard) per gli utenti che devono gestire una municipalità attraverso la piattaforma SCP. Riprendiamo lo schema del caso studio urbano.

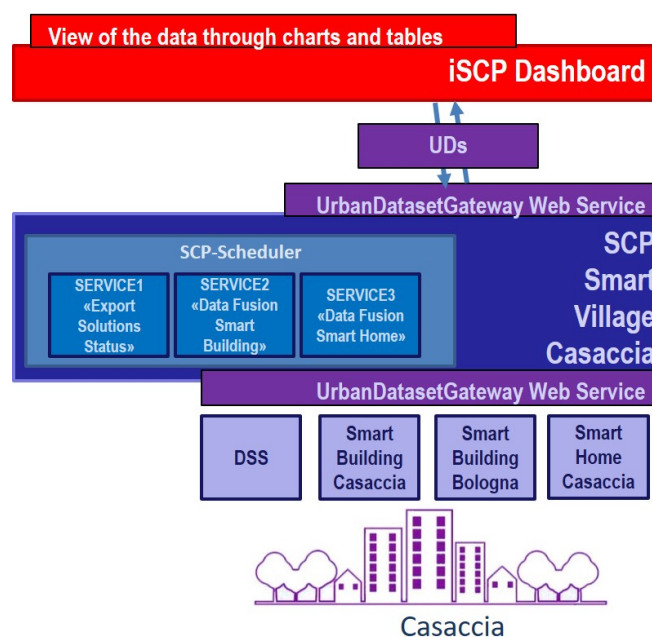


Figura 48 - Caso Studio "Urbano"

Lo sviluppo relativo il caso studio urbano "Da SCP-Casaccia a SCP-Dashboard" prevede lo sviluppo di 4 moduli Job/Service funzionanti nello SCP-Scheduler e i cui risultati in output (UrbanDataset secondo specifiche SCPS di ENEA) si collegano direttamente alla SCP-Dashboard.

I 4 moduli SERVICE che sono stati sviluppati sono:

1. "Export Solutions Platform Status": esportazione UD «Platform Status»;
2. "Data Fusion Smart Building": fusione di due UrbanDataset "Smart Building Consumption", inviati da Solution diverse, ed esportazione in un UD «Whatever» (dedicato);
3. "Data Fusion Smart Home": fusione di due UrbanDataset "Home Aggregated Electric Production" e "Home Aggregated Electric Consumption", ed esportazione in un UD «Whatever» (dedicato);
4. "Export Map View": inizialmente non contemplato, viene aggiunto per utilizzare il servizio di visualizzazione georeferenziata su mappa interattiva.

I suddetti moduli sono stati sviluppati utilizzando il Job-Template (progetto Java per creare dei Job/Service che possono essere configurati ed eseguiti dallo SCP-Scheduler, si veda par. 3.2), nella direzione di essere utilizzati per implementare il caso studio urbano.

5.1.1 “Export Solutions Platform Status”

Il Job/SERVICE “*Export Solutions Platform Status*” è stato sviluppato per esportare, per ogni Solution configurata nel Registry, un set di informazioni relative a quella Solution.

Le informazioni da esportare e il template dell’UrbanDataset, “*PlatformStatus-2.0-OUTPUT.json*”, sono stati forniti dal personale ENEA. Il risultato in output del SERVICE è, quindi, un UrbanDataset “PlatformStatus” che descrive, per ogni Solution, una linea che contiene il set di proprietà relative a quella Solution. Il modulo software è stato progettato, sviluppato, configurato e testato per effettuare l’export della libreria che sarà utilizzata direttamente dentro lo SCP-Scheduler.

5.1.2 “Data Fusion Smart Building”

Il Job/SERVICE “*Data Fusion Smart Building*” è stato sviluppato per effettuare un’operazione di DataFusion, prendendo in input N UrbanDataset “*Smart Building Electric Consumption*” di N Solution diverse, e scrivendo in output un singolo UrbanDataset “*Whatever*” predisposto in modo tale che i dati saranno in un secondo tempo visualizzati nella Dashboard di una SCP urbana (p.es. SCP-Casaccia). Le proprietà di riferimento (consumi elettrici dell’edificio), gli esempi di UrbanDataset da usare in input e il template dell’UrbanDataset per generare l’output, sono stati tutti forniti dal personale ENEA. Il risultato in output del SERVICE è, quindi, un UrbanDataset “*Whatever*” che descrive, per ogni edificio delle N Solution in input, i relativi consumi elettrici. Il modulo software è stato progettato, sviluppato, configurato e testato per effettuare l’export della libreria che sarà utilizzata direttamente dentro lo SCP-Scheduler.

5.1.3 “Data Fusion Smart Home”

Il Job/SERVICE “*Data Fusion Smart Home*” ha il compito di effettuare una data fusion:

- prendendo in input, 2 UrbanDataset “*Home Aggregated Electric Consumption*”, “*Home Aggregated Electric Production*” (stessa Solution), aggregati sulla singola giornata, per N giorni;
- scrivendo in output un singolo UrbanDataset “*Whatever*”.

Questo output sarà poi utilizzato dalla SCP-Dashboard di una SCP urbana (p.es. SCP-Casaccia).

Le proprietà di riferimento (consumo e produzione elettrici dell’edificio), gli esempi di UrbanDataset da usare in input e il template dell’UrbanDataset per generare l’output, sono stati forniti dal personale ENEA. Il risultato in output del SERVICE è, quindi, un UrbanDataset “*Whatever*” che descrive per N giorni, i relativi consumi e produzione elettrici della smart home in oggetto.

Il modulo software è stato progettato, sviluppato, configurato e testato per effettuare l’export della libreria che sarà utilizzata direttamente dentro lo SCP-Scheduler.

5.1.4 “Export Map View”

Il Job/SERVICE “*Export Map View*”, eseguibile nello SCP-SCHEDULER, è stato definito per utilizzare il servizio di Google Maps per la visualizzazione georeferenziata su mappa interattiva. Per fare in modo di utilizzare questo SERVICE assieme ai precedenti già definiti, il personale ENEA ha declinato l’utilizzo di questo modulo su scala urbana del caso studio (Appendice B dell’allegato tecnico) ma cercando di riutilizzare lo stesso modulo anche su scala nazionale (utilizzando la inter-SCP). Il SERVICE è stato sviluppato per esportare, per ogni Solution configurata nel Registry, le coordinate per la georeferenziazione più alcune informazioni a corredo. Il risultato in output del SERVICE è, quindi, un UrbanDataset “*Whatever*” che descrive, per ogni Solution, una linea che contiene le coordinate e il set di proprietà relative a quella Solution.

Il modulo software è stato progettato, sviluppato, configurato e testato per effettuare l’export della libreria che sarà utilizzata direttamente dentro lo SCP-Scheduler.

La Dashboard finale per il caso studio “urbano” è in corso.

5.2 KPI energetici

Abbiamo visto nel corso della **sperimentazione** descritta nel capitolo 2, come l'UrbanDataset "Platform Status" sia stato raccolto dalle diverse SCP fino a divenire input per la Dashboard finale.

Se qui abbiamo "solamente" un UD con informazioni di stato della piattaforma urbana connessa, **si pensi a un caso equivalente dove siano disponibili dati di tipo energetico che abiliterebbero un confronto tra città sul tema energetico.**

Nel corso dell'ultima annualità di RdS, il team del politecnico di Milano impegnato nella LA 1.25 [16] ha prodotto un insieme di KPI per monitorare e gestire gli ambiti energetici del contesto urbano.

ENEA intende utilizzare in un futuro prossimo alcuni dei KPI individuati dal politecnico nelle sue attività che coinvolgono la SCP. I KPI verranno calcolati da alcune SCP cittadine grazie ai dati raccolti tramite UrbanDataset dalle solution verticali connesse, per essere poi inviati alla iSCP.

La selezione dei KPI al momento più interessanti è stata effettuata in base ai seguenti parametri:

- Valore informativo dell'indicatore
- Disponibilità dei dati presso le SCP attualmente implementate da ENEA
- Dinamicità dell'indicatore.

La lista dei KPI attualmente in valutazione è la seguente:

1) Consumo di energia elettrica annuo del Comune

- **Descrizione:** *L'indicatore misura i consumi finali di energia elettrica che avvengono all'interno della città considerando i seguenti settori: residenziale, edifici pubblici, comparto industriale, terziario, agricoltura e trasporto pubblico e privato.*
- **Metodo di calcolo:** *Sommatoria per tutti gli n settori (dei consumi finali legati all'energia elettrica del settore "i" esimo)*
- **Valore informativo:** dà una valutazione complessiva dei consumi della città e, nel tempo, rende conto di miglioramenti o peggioramenti di tale consumo globale
- **Disponibilità dei dati:** ad esempio, sono disponibili su Terna², ma la lettura pare difficile da automatizzare (da verificare)
- **Periodicità:** annuale
- **Valutazione iniziale:** pur avendo un valore informativo interessante, la periodicità annuale lo rende poco interessante per la SCP. Inoltre, si tratterebbe di dati non raccolti dalla città ma presi da fonte esterna. Se si trovasse il modo di prelevarlo automaticamente dalle applicazioni potrebbe comunque essere utile da affiancare alle informazioni prelevate direttamente dalle applicazioni cittadine

2) Domanda energetica primaria specifico edificio pubblico – illuminazione / raffrescamento

- **Descrizione:** *Gli indicatori misurano la domanda di energia primaria per i consumi legati all'illuminazione / raffrescamento dell'i-esimo edificio pubblico all'interno del Comune. Indicatore da calcolare per ogni singolo edificio pubblico del territorio comunale*
- **Metodo di calcolo:** *Consumo di energia primaria legato all'illuminazione /al raffrescamento dell'edificio i-esimo nell'anno j-esimo / superficie dell'edificio i-esimo*
- **Valore informativo:** dà una valutazione specifica su singolo edificio pubblico
- **Disponibilità dei dati:** se vengono installati dei sensori possono essere calcolati dinamicamente. ENEA li ha disponibili per es. su alcuni edifici dei centri di ricerca
- **Periodicità:** annuale, ma se vengono installati sensori può essere calcolato con frequenze maggiori

² Link a cui consultare i dati annuali del consumo elettrico della città:
<https://app.powerbi.com/view?r=eyJrIjoiaNzkyMTFjNWUeNmJhNy00YjAxLWJjOGYtMjVjZjRlODZmNjZlIiwidCI6ImVjY2QzMzRlTcwMjltNDcwOS1hYmE1LWE1ZGQ3NzkyOWUyNyIsImMiOj9&pageName=ReportSection>

- **Valutazione iniziale:** pur avendo un valore informativo più limitato, possono essere indicativi delle politiche di efficienza energetica pubblica perseguite in un comune. Inoltre, possono essere estesi inserendo anche il consumo elettrico alle prese, o, in mancanza delle singole componenti, può essere calcolato il consumo elettrico totale

3) Inquinamento dell'aria - NO₂ / O₃ / PM10 / PM2.5 / SO₂

- **Descrizione:** Gli indicatori misurano la concentrazione di diversi inquinanti nell'aria all'interno del Comune
- **Metodo di calcolo:** Massa dell'inquinante (µg) / Volume di aria campionato all'interno del Comune (m³)
- **Valore informativo:** dà una valutazione dell'effetto delle combustioni di carburanti fossili sulla qualità dell'aria
- **Disponibilità dei dati:** sempre più spesso i comuni installano le proprie centraline. Esistono inoltre le rilevazioni ufficiali dell'ARPA
- **Periodicità:** mensile, ma si può pensare anche di calcolarli giornalmente
- **Valutazione iniziale:** sono molto importanti poiché impattano fortemente sulla salute dei cittadini. Indirettamente questo indicatore è correlato anche alla rete elettrica, visto che l'aumento della diffusione dei veicoli elettrici ci si aspetta porti a una diminuzione di questi inquinanti

Al fine di predisporre le Smart City Platform a ricevere dati utili al calcolo di KPI energetici, sono stati progettati e implementati nell'Ontologia SCPS 2.0 **due nuovi tipi UrbanDataset:**

- *"Electric Meter Reading"* fornisce i dati relativi ai consumi elettrici rilevati a livello di meter o POD (Point Of Delivery) presente in una unità immobiliare;
- *"Gas Meter Reading"* fornisce i dati relativi ai consumi di gas rilevati a livello di meter o PDR (Punto Di Riconsegna) presente in una unità immobiliare.

Le descrizioni dettagliate di tali UD sono riportate rispettivamente in Appendice A2 e A3.

5.3 Dashboard nazionale e regionale

Tra gli sviluppi in corso e quelli futuri ricopre un ruolo particolarmente centrale quello relativo alle Dashboard per la visualizzazione finale dei dati tramite Report composti da tabelle e grafici/chart.

Abbiamo visto come **la Dashboard della inter-SCP**, agente su scala nazionale, oggetto della sperimentazione presentata nel capitolo 2, sia ad oggi molto spartana e riporti solamente pochi dati relativi alle informazioni di stato di 5 piattaforme che sono geograficamente disposte sul tessuto nazionale.

La sua evoluzione è sicuramente oggetto dei lavori di sviluppo nel prossimo futuro, riutilizzando l'infrastruttura consolidata di comunicazione, andando ad utilizzare gli altri grafici disponibili per visualizzare i dati finali, anche con curve di andamento temporale e anche e soprattutto relativamente i **KPI energetici** che si sta valutando di implementare nella inter-SCP per la governance su scala nazionale.

Abbiamo anche presentato lo sviluppo fino ad ora svolto riguardo il caso studio urbano, nel par. 5.1.

Il suo completamento, allo scopo di ottenere una **Dashboard urbana** per la SCP-Casaccia, sarà eseguito nei prossimi mesi. La riflessione che si sta facendo a riguardo, è quella relativa ad una dashboard urbana che possa essere vetrina con cui diffondere il prototipo SCP verso le municipalità: **potrebbero proprio essere i servizi finali visualizzati sul cruscotto finale a guidare la transizione "smart" di una municipalità.**

Una terza riflessione riguarda la **Dashboard declinata su scala regionale.**

Partendo dall'esperienza fatta in questo triennio, si potrebbe ripetere la stessa metodologia declinata su un insieme di comuni afferenti la stessa zona geografica; a quel punto la inter-SCP del caso funzionerebbe allo stesso modo ma sarebbe alimentata da dati provenienti da piattaforme urbane di municipalità tra loro geograficamente vicine, non sarebbe dunque agente su scala nazionale ma su scala regionale.

Abbiamo visto, nella LA 1.20 [2] come la SCP sia già stata integrata nella **piattaforma PELL** con una SCP-PELL e come quest'ultima piattaforma sia stata connessa alla inter-SCP.

Il passo successivo è duplice:

- da un lato, si vogliono recuperare i KPI esportati dalla piattaforma PELL (si veda par. 3.5.3) nella Dashboard della inter-SCP;
- dall'altro lato, si vuole definire una **Dashboard apposita per la SCP-PELL** che permetta il **monitoraggio dei consumi elettrici il più possibile in real-time.**

Un'altra **connessione** che potrebbe rivelarsi interessante è quella **tra SCP e sistema CIPCAST** per il supporto alle decisioni critiche (Decision Support System, DSS). Un'integrazione tra i due sistemi è già stata fatta nel primo triennio della RdS [6] con la connessione tra SCP-Casaccia e sistema WebGIS.

Si potrebbe valutare un caso studio per riutilizzare la connessione tra SCP e sistema CIPCAST e pensare come affiancare i **due cruscotti per una stessa municipalità** (o meglio ancora, per una stessa **regione**).

Infine, come già detto, nel prossimo futuro si ritiene molto utile **investire nelle Dashboard per valorizzare l'infrastruttura di comunicazione interoperabile sottostante e diffondere maggiormente i risultati.**

In questa direzione, si prevedono nei prossimi mesi diverse ore di **formazione** per trasferire il know-how relativo a SCP e Dashboard a diversi colleghi ENEA dei laboratori TERIN-SEN-SCC e TERIN-SCC-CROSS.

6 Conclusioni

Questo report ha descritto i risultati finali della LA 1.21 (anno 2021), terza di tre linee di attività che compongono un unico percorso.

Il risultato più importante e immediatamente più tangibile è l'implementazione avvenuta con successo del caso studio pilota e quindi l'aver implementato la **versione 1.0 della piattaforma inter-SmartCityPlatform (inter-SCP) agente su scala nazionale**, a cui sono state connesse 5 piattaforme (tra cui la SCP-PELL) con cui avviene uno scambio dati periodico, automatico, monitorabile. Da questo punto di vista il risultato è andato oltre quanto pianificato, poiché il caso studio pilota prevedeva la connessione con 1 sola SCP.

La sperimentazione è avvenuta con successo (capitolo 2), testando con successo l'implementazione del caso studio pilota; ciò è misurato anche grazie agli indicatori utilizzati per quantificare quanto affidabile è il flusso dati nelle sue diverse fasi (si veda par. 2.7 "Indicatori della Sperimentazione").

Al momento, i dati urbani scambiati sono relativi alla comunicazione e allo stato delle piattaforme, quindi non sono ancora utilizzati **KPI energetici** come inizialmente pianificato, sebbene in questa direzione l'analisi sia terminata (si veda report [16]) nella direzione della successiva implementazione (si veda par. 3.5.3 sui KPI statici del PELL e par. 5.2 sui KPI energetici) e siano già stati definiti dei nuovi UrbanDataset (Appendice A2 e A3) proprio per questo scopo. Di conseguenza anche la **Dashboard finale** che riporta con successo dati provenienti da tutte le SCP connesse alla inter-SCP, può risultare a un primo sguardo ancora "povera" di contenuti (si veda par. 2.6). Ma ciò che va considerato è che **il framework che permette la comunicazione interoperabile tra SCP è pronto per trasportare qualunque tipologia di dataset che si vorrà definire**.

Grazie alla **SCP-GUI 2.0**, il monitoraggio del traffico degli UrbanDataset diviene efficace e immediato,

- sia per i dati ricevuti dalle piattaforme esterne (Solution verticali)
- sia per i dati elaborati internamente (JOB/Service).

Il **sistema di adesione per l'utente** si è concretizzato in **3 percorsi** per le differenti esigenze:

1. Adesione con una nuova SCP che vuole entrare nel network di SCP;
2. Adesione con una Solution verticale a una SCP esistente;
3. Adesione con una SCP urbana alla inter-SCP agente su scala nazionale.

Queste guide sono state definite e pubblicate sul sito dello "SCP Project" [3] nella sezione ADESIONE.

In conclusione, sebbene sia stato effettuato uno studio apposito per individuare KPI energetici è ancora in corso la valutazione della loro implementazione; una **criticità** che è emersa è la difficoltà di individuare degli indicatori chiave che possano essere generati da tutte le SCP/municipalità connesse alla inter-SCP; sarebbe inutile, infatti, che ogni SCP/municipalità generasse dei propri KPI, non comparabili tra loro; lo sforzo che si sta facendo è quindi quello di **individuare indicatori comuni** che possano trovare riscontro ovunque (si pensi **all'illuminazione pubblica, alla mobilità e al monitoraggio della CO2**, in primis) ma anche considerare l'ipotesi che sia un'azione governativa "dall'alto" a richiedere particolari indicatori, per legge;

si ritiene di aver comunque perseguito la maggior parte degli obiettivi pianificati per il triennio e aver consolidato un **framework di comunicazione interoperabile, esteso alla scala nazionale**, che permette il **monitoraggio delle Smart City in tempo reale**; è questo uno degli aspetti più innovativi rispetto alle dashboard esistenti che, nella grande maggioranza dei casi esaminati, utilizzano dati raccolti in una fase temporale antecedente (a volte anche l'anno precedente) rendendo il monitoraggio poco aderente alla situazione reale. L'obiettivo a cui si vuole tendere, invece, è il **monitoraggio in tempo reale**, per poter avere in ogni momento la fotografia "energetica" corrente della propria Smart City, per agire tempestivamente con azioni di efficientamento e di riqualificazione e per una gestione più sostenibile.

In questa direzione, si ritiene che **gli sforzi per il prossimo futuro debbano essere spesi sull'evoluzione e definizione di Dashboard tramite il componente SCP-DASH (su diverse scale, come spiegato nel par. 5.3)** per valorizzare i risultati fin qui raggiunti e **avviare un'azione di diffusione delle tecnologie verso le municipalità**.

7 Riferimenti

1. C. Novelli, A. Brutti, A. Frascella, L. Gazzotti, N. Gessa, F. Moretti, F. Niccolai, C. Petrovich, S. Pizzuti, "Progettazione Framework per la Governance dei Dati Urbani Energetici", Report RdS/PTR(2019)/007
2. C. Novelli, G. Santomauro, A. Frascella, A. Brutti, C. Petrovich, F. Moretti, M. Chinnici, G. Ponti, S. Pizzuti, "Sviluppo Piattaforma per la Governance dei Dati Urbani Energetici", Report RdS/PTR2020/013
3. "SCP Project", sito web ufficiale
<https://smartcityplatform.enea.it/>
4. Smart City Platform Specification for interoperability layer, SCPS 2.0
<https://smartcityplatform.enea.it/#/it/specification/>
5. Smart City Platform (SCP) Smart Village Casaccia
<https://smartcityplatform.enea.it/casaccia/>
6. RdS 2015-2017, Report RdS/PAR2018/020, RdS/PAR2017/040 e SCPS in RdS/PAR2017/103 fino a 108
7. C. Novelli, A. Frascella, Settembre 2020, "SCPS Functional 2.0"
<https://smartcityplatform.enea.it/#/it/specification/functional/2.0/>
8. A. Brutti, N. Gessa, A. Frascella, "SCPS Semantic 2.0", SCPS Ontology Web Tool
<https://smartcityplatform.enea.it/UDWebLibrary/it/ontologyinfo>
9. A. Brutti, A. Frascella, N. Gessa, C. Novelli, Novembre 2020, "SCPS Information 2.0"
<https://smartcityplatform.enea.it/#/it/specification/information/2.0/>
10. C. Novelli, Settembre 2020, "SCPS Collaboration 2.0"
<https://smartcityplatform.enea.it/#/it/specification/collaboration/2.0/>
11. C. Novelli, A. Frascella, G. Santomauro, Settembre 2020, "SCPS Communication 2.0"
<https://smartcityplatform.enea.it/#/it/specification/communication/2.0/>
12. C. Novelli, A. Frascella, Settembre 2020, "SCPS Glossario 2.0"
<https://smartcityplatform.enea.it/#/it/specification/glossary.html>
13. inter- SmartCityPlatform (inter-SCP o iSCP), Piattaforma per la Governance dei Dati Urbani Energetici
<https://smartcityplatform.enea.it/inter-scp/>
14. Dashboard della inter-SCP (iSCP-Dashboard)
<https://smartcityplatform.enea.it/inter-scp/dash/>
15. M. Pirruccio, Heartwood Labs SRLS, nell'ambito della LA 1.21 è stata una sotto-attività di consulenza "Sviluppo SCP-Scheduler per l'esecuzione di Service nella SCP", Report RdS/PTR(2021)/014
16. S. Franzò, D. Chiaroni, V. Chiesa, F. Frattini, Report Tecnico "Definizione di un dataset di KPI per monitorare e gestire gli ambiti energetici del contesto urbano" inserito nella LA 1.25 "Definizione di un insieme di riferimento di dati urbani energetici su scala nazionale e municipale", PoliMI, Report RdS/PTR(2021)/016

17. C. Aguzzi, F. Chesani, M. Patella, M. Milano, P. Mello, Report Tecnico "Web Application per l'editing dell'Ontologia del Framework" inserito nella LA 1.23 "Strumenti basati su ontologia per il Framework per la Governance dei Dati Urbani Energetici", UniBO, Report RdS/PTR(2021)/015
18. G. D'Agosta, C. Meloni, A. Annunziato, S. Branchetti, G. D'Agostino, P. De Sabbata, N. Gessa, O. Gregori, G. Massa, F. Moretti, C. Naccarato, G. Nigliaccio, C. Petrovich, S. Romano, F. Paolucci, S. Pizzuti, S. Sylos Labini, A. Tofani, LA 1.47, "Progettazione e implementazione di una piattaforma a servizio di energy community", Report RdS/PTR(2021)/036
19. D'Agosta G., Meloni C., D'Agostino G., Tofani A., Massa G., Petrovich C., Nigliaccio G., De Sabbata P., Romano S., Pizzuti S., Gessa N., Caldera M., Chinnici M., Novelli C., LA 1.48, "Implementazione servizi a valore aggiunto per la LEC", Report RdS/PTR(2021)/039
20. R. Pezzetti, P. Gazzola, E. Pavione, LA 1.26, "Urban Check Up Model: Modello di mappatura, valutazione e governance dei dati urbani strategici", Report RdS/PTR(2021)/017
21. Piattaforma PELL: Avvio popolamento dati dinamici PELL IP, test dati statici PELL Edifici
Blaso L., Brutti A., Caldera M., Clemente P., Fumagalli S., Giovinazzi S., Giuliani G., Gozo N., Leonardi G., Moretti F., Pizzuti S., Pollino M., Rosato V., Sylos Labini S., Zinzi M. RdS/PTR(2021)/018

Appendice A – Definizioni UrbanDataset

In questa Appendice riportiamo le definizioni di UrbanDataset (aggiornate o create da zero) che sono state realizzate nel corso della Linea di Attività 1.21:

- 1) "Platform Status" 2.0;
- 2) "Electric Meter Reading" 2.0;
- 3) "Gas Meter Reading".

A1 UrbanDataset "Platform Status" 2.0

La nuova definizione semantica del Platform Status è la seguente:

Tabella 1 - Tabella di sintesi dell'UrbanDataset "Platform Status" versione 2.0

UrbanDataset	Platform Status			
Scopo	Informazioni dello stato di funzionamento della piattaforma all'istante di generazione dell'UrbanDataset.			
Categoria / Sottocategoria	Public Safety Policy Emergency Response/Communication			
SPECIFICAZIONE DELL'URBANDATASET				
UrbanDatasetId	PlatformStatus-2.0			
UrbanDatasetUri	http://smartcityplatform.bologna.enea.it/UDWebLibrary/it/urbandataset?uri=smartcityplatform%3Aenea%3Asemantic%3Aontology%3Aurbandataset%23PlatformStatus			
UrbanDatasetName	Platform Status			
CONTESTUALIZZAZIONE DELL'URBANDATASET				
	Unità di Misura	Formato	Descrizione/Esempio	
Timestamp	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Tempo di generazione dell'UrbanDataset (determina anche l'istante temporale di validità delle informazioni fornite)	
Timezone	Adimensionale	String	Es. UTC, UTC+01, UTC-05	
Coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate del centro geometrico dell'area gestita dalla piattaforma.	
Producer	Adimensionale	String	Identificatore del Sistema che ha prodotto i dati	
PROPRIETÀ DELL'URBANDATASET				
	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
PlatformID	Adimensionale	String	Identificatore della piattaforma software	sì
PlatformName	Adimensionale	String	Nome della piattaforma software	sì
LogCode	Adimensionale	String (vincolata a lista di codici)	Codice che indica lo stato dell'"oggetto" monitorato	sì
LogDescription	Adimensionale	String	Descrizione testuale dello stato dell'oggetto monitorato	sì
DataDescription	Adimensionale	String	Descrizione testuale dei dati inviati	sì
TownName	Adimensionale	String	Nome esteso del Comune gestito dalla piattaforma (es. Roma, Anagnina Sabazia, ...)	sì
ZIPCode	Adimensionale	String	Codice di Avviamento Postale(CAP) della zona gestita dalla piattaforma	sì

	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
ProvinceCode	Adimensionale	String	Codice Provincia a cui appartiene l'area gestita dalla piattaforma	sì
TotalManagedUsersCount	Adimensionale	Integer	Numero degli utenti gestiti	sì
TotalManagedVerticalsCount	Adimensionale	Integer	Numero delle applicazioni verticali gestite	sì
TotalAccessedUDCount	Adimensionale	Integer	Numero totale di UrbanDataset acceduti	sì
TotalProducedUDCount	Adimensionale	Integer	Numero totale di UrbanDataset prodotti	sì
coordinates	Adimensionale	WGS84 (World Geodetic System)	Riferimento geografico associato ai dati della riga	no
timestamp	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Istante temporale a cui si riferiscono i dati della riga	no

A2 UrbanDataset “Electric Meter Reading” 2.0

L'UD “Electric Meter Reading” fornisce i dati relativi ai consumi elettrici rilevati a livello di meter o POD (Point Of Delivery) presente in una unità immobiliare.

L'UD Electric Meter Reading si differenzia dagli UD Counter Reading e Single Phase Counter Reading, già presenti nella precedente versione dell'ontologia, in quanto è specifico per l'invio di informazioni relative ai consumi elettrici di unità immobiliari (appartamenti, edifici, ...), diversamente dagli altri due che sono specifici per i consumi elettrici degli impianti di illuminazione pubblica. Le due tipologie di flusso di dati hanno caratteristiche parzialmente diverse e si collocano in scenari diversi, per cui è stato ritenuto opportuno gestirle diversamente anche a livello di UrbanDataset.

È stato progettato in modo da supportare le seguenti esigenze:

- creazione di istanze contenenti dati di una ma anche più unità immobiliari (per supportare, ad esempio, i casi in cui si voglia inviare con un unico messaggio i consumi disaggregati di un insieme di unità immobiliari monitorate da una stessa Solution);
- consentire l'invio dei dati sui consumi con un dettaglio a livello di POD e/o a livello di meter (per supportare, ad esempio, la rilevazione dei dati in unità immobiliari che abbiano un POD ma più meter ciascuno dedicato al rilevamento dei consumi in base all'uso finale dell'energia elettrica);
- non imporre l'invio di informazioni che potrebbero essere in contrasto con aspetti di privacy.

La definizione semantica dell'UD “Electric Meter Reading”, realizzata in base all'analisi dei requisiti svolta, è mostrata in Tabella 2. Si noti che, tra le altre, le proprietà che consentono l'identificazione dell'unità immobiliare, del POD e del meter sono opzionali; questa scelta è stata fatta per supportare le esigenze esposte ai punti b) e c) del precedente elenco.

Tabella 2 - Tabella di sintesi dell'UrbanDataset “Electric Meter Reading”

UrbanDataset	Electric Meter Reading
Scopo	Fornire i dati relativi ai consumi elettrici rilevati a livello di meter o di POD. In una stessa istanza di questo UrbanDataset possono essere inseriti i dati di meter di una o più unità immobiliari; per una stessa unità immobiliare possono essere presenti dati di meter diversi (è il caso in cui per uno stesso POD sono presenti meter diversi dedicati al rilevamento dei diversi usi finali dell'energia elettrica). Se indicato il meter, i dati di una riga si riferiscono alla lettura effettuata su singolo meter.

UrbanDataset Electric Meter Reading

Categoria / Sottocategoria Energy Management/Energy demand

SPECIFICAZIONE DELL'URBANDATASET

UrbanDatasetId ElectricMeterReading-2.0

UrbanDatasetUri <http://smartcityplatform.bologna.enea.it/UDWebLibrary/it/urbandataset?uri=smartcityplatform%3Aenea%3Asemantic%3Aontology%3Aurbandataset%23ElectricMeterReading>

UrbanDatasetName Electric Meter Reading

CONTESTUALIZZAZIONE DELL'URBANDATASET

	Unità di Misura	Formato	Descrizione/Esempio
Timestamp	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Tempo di generazione dell'UrbanDataset (determina anche l'istante temporale di validità delle informazioni fornite)
Timezone	Adimensionale	String	Es. UTC, UTC+01, UTC-05
Coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate del centro geometrico dell'area gestita dalla piattaforma.
Producer	Adimensionale	String	Identificatore del Sistema che ha prodotto i dati

PROPRIETÀ DELL'URBANDATASET

	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
ActiveEnergy	kilowattHour	double	Energia attiva consumata	sì
period	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Periodo durante il quale sono stati rilevati i dati riportati nella riga	sì
coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate dell'unità immobiliare a cui fanno riferimento i dati della riga	no
PODID	Adimensionale	string	Codice POD che identifica univocamente il punto di prelievo	no
MeterID	Adimensionale	string	Identificativo univoco del dispositivo che rileva i consumi elettrici	no
MeterSchemeID	Adimensionale	string	Identificativo dello schema di codifica a cui appartiene l'identificativo fornito nel campo MeterID	no
RealEstateUnitID	Adimensionale	string	Identificativo dell'unità immobiliare (es. appartamento, edificio, abitazione indipendente...)	no
SchemeID	Adimensionale	string	Identificativo dello schema di codifica a cui appartiene l'identificativo fornito nel campo RealEstateUnitID	no
ReactiveEnergy	kilovoltamperereactivePerHour	double	Energia reattiva (inductiveEnergy + capacitiveEnergy)	no
ActivePower	kilowatt	double	Potenza attiva fase	no
ReactivePower	kilovoltamperereactive	double	Potenza reattiva	no
LineCurrent	ampere	double	Corrente della linea	no

	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
PhaseVoltage	volt	double	Tensione della fase	no
ElectricityEndUseCode	Adimensionale	String (vincolata a lista di codici)	Uso finale dell'energia elettrica consumata (ad es. illuminazione, forza motrice, generale, condizionamento,...)	no

La lista di codici a cui è stata vincolata la proprietà "*ElectricityEndUseCode*" è la seguente:

Tabella 3 - Lista di codici "ElectricityEndUseCode"

Nome sintetico	ElectricityEndUseCode
Versione	2.0
URI	https://smartcityplatform.enea.it/specification/semantic/2.0/gc/ElectricityEndUseCode.gc
Agency	ENEA
Codice	Valore
LGH	illuminazione
CND	condizionamento
PWR	forza motrice
ALL	generale

A3 UrbanDataset “Gas Meter Reading” 2.0

L'UD “Gas Meter Reading” fornisce i dati relativi ai consumi di gas rilevati a livello di meter o PDR (Punto Di Ricosegna) presente in una unità immobiliare.

È stato progettato in modo da supportare le seguenti esigenze:

- a. creazione di istanze contenenti dati di una ma anche più unità immobiliare (per supportare, ad esempio, i casi in cui si voglia inviare con un unico messaggio i consumi disaggregati di un insieme di unità immobiliari monitorate da una stessa Solution)
- b. non imporre l'invio di informazioni che potrebbero essere in contrasto con aspetti di privacy.

La definizione semantica dell'UD “Gas Meter Reading”, realizzata in base all'analisi dei requisiti svolta, è mostrata in Tabella 4. Si noti che, tra le altre, le proprietà che consentono l'identificazione dell'unità immobiliare, del PDR e del meter sono opzionali; questa scelta è stata fatta per supportare l'esigenza esposta al punto b) del precedente elenco.

Tabella 4 - Tabella di sintesi dell'UrbanDataset “Gas Meter Reading”

UrbanDataset		Gas Meter Reading		
Scopo	Fornire i dati relativi ai consumi di gas rilevati a livello di meter o di PDR. In una stessa istanza di questo UrbanDataset possono essere inseriti i dati di meter di una o più unità immobiliari.			
Categoria / Sottocategoria	Energy Management/Energy demand			
SPECIFICAZIONE DELL'URBANDATASET				
UrbanDatasetId	GasMeterReading-2.0			
UrbanDatasetUri	http://smartcityplatform.bologna.enea.it/UDWebLibrary/it/urbandataset?uri=smartcityplatform%3Aenea%3Asemantic%3Aontology%3Aurbandataset%23GasMeterReading			
UrbanDatasetName	Gas Meter Reading			
CONTESTUALIZZAZIONE DELL'URBANDATASET				
	Unità di Misura	Formato	Descrizione/Esempio	
Timestamp	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Tempo di generazione dell'UrbanDataset (determina anche l'istante temporale di validità delle informazioni fornite)	
Timezone	Adimensionale	String	Es. UTC, UTC+01, UTC-05	
Coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate geografiche del centro dell'area monitorata.	
Producer	Adimensionale	String	Identificatore del Sistema che ha prodotto i dati	
PROPRIETÀ DELL'URBANDATASET				
	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
GasConsumption	cubicMetre	double	Consumo gas	sì
period	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Periodo durante il quale sono stati rilevati i dati riportati nella riga	sì
coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate dell'unità immobiliare a cui fanno riferimento i dati della riga	no
PDRID	Adimensionale	string	Codice PDR cui è associata l'utenza	no

	Unità di Misura	Formato	Descrizione/Esempio	Obbligatoria
MeterID	Adimensionale	string	Identificativo univoco del dispositivo che rileva i consumi di gas	no
MeterSchemeID	Adimensionale	string	Identificativo dello schema di codifica a cui appartiene l'identificativo fornito nel campo MeterID	no
RealEstateUnitID	Adimensionale	string	Identificativo dell'unità immobiliare (es. appartamento, edificio, abitazione indipendente...)	no
SchemeID	Adimensionale	string	Identificativo dello schema di codifica a cui appartiene l'identificativo fornito nel campo RealEstateUnitID	no

APPENDICE B – Analisi Requisiti per WebApp Ontology-based

Questa appendice descrive l'analisi dei requisiti su cui si sono basati la progettazione e lo sviluppo avvenuti nella Linea di Attività 1.23 "Strumenti basati su ontologia per il Framework per la Governance dei Dati Urbani Energetici" [17]. Si riportano qui i requisiti richiesti.

Per vedere la progettazione e le scelte finali si veda il report RdS/PTR(2021)/015.

Prima di tale sviluppo, per inserire nuovi UrbanDataset, o modificare quelli esistenti, bisognava agire direttamente sull'Ontologia [8] OWL (per esempio usando il software open source Protege³). Inoltre, se qualcuno di esterno al gruppo di lavoro (per esempio un partner di progetto, un comune, ecc.) avesse voluto richiedere l'inserimento di un nuovo UrbanDataset, avrebbe dovuto proporli tramite un form Excel.

Si riporta, di seguito, l'analisi dei requisiti avvenuta.

Lo scopo che ci si pone è di progettare e sviluppare uno strumento che:

- 3) semplifichi la gestione e l'implementazione degli UrbanDataset, evitando errori di inserimento e permettendolo anche a membri del team che non conoscono a fondo l'ontologia di modificarla;
- 4) permetta agli utenti esterni di proporre nuovi UrbanDataset.

La soluzione più idonea sembra essere un'applicazione Web che permetta:

- di accedere tramite login, interfacciandosi con l'Identity Provider (IDP) centralizzato della SCP (se l'IDP non dovesse essere disponibile da subito, prevedere interfacciamento con Mock-Up che utilizzi la stessa interfaccia WS dell'IDP, per poter effettuare velocemente lo Switch verso di esso una volta che sarà pronto);
- di inserire e modificare in alcune parti l'ontologia delle SCPS;
- di salvare le modifiche come nuova versione dell'ontologia SCPS, lasciando comunque un backup della versione precedente dell'Ontologia, generando al contempo un report che documenti le modifiche effettuate.

Alcuni dei requisiti iniziali sono i seguenti:

- l'applicazione deve, per quanto possibile, riutilizzare tecnologie e funzioni dell'applicazione già esistente per la navigazione dell'ontologia;
- l'accesso alla modifica dell'ontologia è possibile solo tramite login;
- dovrebbe essere prevista la possibilità di supportare la doppia lingua, in italiano e inglese.

Funzionalità dell'applicazione

- Scelta della lingua
- Proprietà:
 - Ricerca di una proprietà (*sulla base della funzionalità già esistente nel navigatore*)
 - Inserimento di una nuova proprietà
 - Modifica di una proprietà esistente
 - Cancellazione di una proprietà
- UrbanDataset:
 - Ricerca di un UrbanDataset (*sulla base della funzionalità esistente*)
 - Modifica di un UrbanDataset esistente
 - Inserimento di un nuovo UrbanDataset

³ Software "Protege". Disponibile qui: <https://protege.stanford.edu/>

- Cancellazione di un UrbanDataset
- Ricerca, inserimento, modifica, cancellazione di una nuova categoria di proprietà
- Unità di misura
 - Ricerca di una unità di misura
 - Inserimento di una nuova unità di misura
- Esportazione del log come report del lavoro svolto (da decidere se log in file txt o xml)
- Esportazione della nuova versione dall'ontologia, backup e deploy

Vediamo nel seguito le funzionalità, facendo riferimento alle operazioni che di solito vengono effettuate con il software *Protege* e che dovrebbero essere sostituite da quelle dell'applicazione.

Scelte progettuali di base

1. Modifica accessibile solo per utenti abilitati e loggati: la modifica dell'ontologia deve essere possibile solo per utenti loggati e abilitati
2. Maschera di proposta semplificata: la proposta di nuovi UrbanDataset da parte di utenti esterni avverrà tramite una maschera di input semplificata che non andrà a modificare l'ontologia, ma permetterà di scaricare un file (possibilmente nel formato Excel usato finora) che contenga le proposte di modifica
3. Deploy e backup automatico delle modifiche: l'applicazione deve consentire di caricare automaticamente l'ontologia sul DB virtuoso e di pubblicarla online.

Funzionalità discusse ma che, al momento, **non** saranno implementate

1. Mantenimento dello stato (applicazione per utente generico): sarebbe interessante che l'applicazione per creare delle proposte potesse mantenere lo stato della proposta, per consentire l'editing in sessioni diverse
2. Funzionalità di contrattazione e lavoro condiviso (applicazione per utente generico): le opzioni alternative sono:
 - a. supportare la condivisione fra utenti diversi della proposta in fase di editing. In questo modo una proposta può essere vista da più utenti (e dall'amministratore) che lavorano sullo stesso "documento"
→ troppo complesso rispetto ai vantaggi che può dare in questo momento
 - b. la proposta è gestita sempre e solo da un utente, che alla fine la invia per una approvazione
3. Gestione diverse ontologie: sarebbe interessante la possibilità di caricare ontologie differenti. Questo potrebbe portare così a gestire anche la gestione di diverse versioni dell'ontologia. Però si pone il problema che si è utilizzato lo stesso namespace nelle diverse ontologie. Questo rende impossibile a Virtuoso di caricare più di una ontologia. Dunque per il momento non sarà implementata.
→ Il problema andrà prima risolto a livello di specifiche SCPS.