



Agenzia nazionale per le nuove tecnologie,
l'energia e lo sviluppo economico sostenibile



MINISTERO DELLO SVILUPPO ECONOMICO



Ricerca di Sistema elettrico

Progettazione Framework per la Governance dei Dati Urbani Energetici

C. Novelli, A. Brutti, A. Frascella, L. Gazzotti, N. Gessa, F. Moretti, F.
Niccolai, C. Petrovich, S. Pizzuti

Report RdS/PTR(2019)/007

Progettazione Framework per la Governance dei Dati Urbani Energetici

C. Novelli, A. Brutti, A. Frascella, L. Gazzotti, N. Gessa, F. Moretti, F. Niccolai, C. Petrovich, S. Pizzuti (ENEA)

Dicembre 2019

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - I annualità

Obiettivo: Tecnologie

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

Work package 1: Local Energy District

Linea di attività: 19 - Servizi Urbani Energivori: Progettazione Framework per la Governance dei Dati Urbani Energetici

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Claudia Meloni, ENEA

Indice

SOMMARIO	5
1 INTRODUZIONE	6
2 ARCHITETTURA FRAMEWORK	7
2.1 ARCHITETTURA CONCETTUALE	7
2.2 ARCHITETTURA DEL FRAMEWORK	8
3 COMUNICAZIONE INTEROPERABILE	10
3.1 DIFFERENTI SCENARI	10
3.2 SCP BRIDGE	11
4 STUDIO DI FATTIBILITÀ INTEGRAZIONE SCP - PELL	13
4.1 INTRODUZIONE ARCHITETTURALE	13
4.2 IPOTESI DI INTEGRAZIONE	15
4.3 FASE 1 CONFIGURAZIONE	16
4.3.1 <i>Task 1.1 "Creazione Account"</i>	16
4.3.2 <i>Task 1.2 "Creazione Solution"</i>	16
4.3.3 <i>Task 1.3 "Creazione Produzione"</i>	17
4.3.4 <i>Task 1.4 Generalizzazione Bridge</i>	18
4.4 FASE 2 VERIFICA	19
4.4.1 <i>Task 2.1 "Autenticazione Account"</i>	19
4.4.2 <i>Task 2.2 "Autenticazione Admin"</i>	20
4.4.3 <i>Task 2.3 "Verifica ACL"</i>	20
4.5 FASE 3 INVIO UD	20
4.6 CONCLUSIONI STUDIO DI FATTIBILITÀ	21
5 CONSOLIDAMENTO GUI	22
5.1 TASK "CONSOLIDAMENTO"	22
5.1.1 <i>Sotto-Task "Aggiustamenti"</i>	22
5.1.2 <i>Sotto-task "Riorganizzazione HOME"</i>	22
5.1.3 <i>Sotto-task "URL della Solution"</i>	23
5.1.4 <i>Sotto-Task "HELP parametrizzabile"</i>	23
5.2 TASK "MODIFICA ALGORITMO GRAFICO INDICATORI"	23
5.3 TASK "NEW USERS"	23
5.3.1 <i>Sotto-Task "New READER user"</i>	23
5.3.2 <i>Sotto-Task "New SERVICE user"</i>	23
5.3.3 <i>Sotto-Task "1 account – N solution"</i>	23
5.4 TASK "UPDATE USER EMAIL/PASSWORD"	23
5.4.1 <i>Sotto-Task "Update Username"</i>	23
5.4.2 <i>Sotto-Task "Update Password"</i>	23
5.5 TASK "SCP PER SMARTPHONE/MOBILE"	24
5.6 TASK "INTEGRAZIONE BRIDGE"	24
5.7 TASK "ENABLING SEZIONE REPORTS"	24
5.8 TASK "COACHING ON-THE-JOB"	24
6 SISTEMA DI ADESIONE ALLA INTER-SCP	26
7 STATO DELL'ARTE SERVIZI DI VISUALIZZAZIONE DATI	28
7.1 TIPOLOGIE DI GRAFICO	29
7.2 TIPOLOGIE DI SOFTWARE	35
7.2.1 <i>Tableau</i>	36
7.2.2 <i>D3.js</i>	37

7.2.3	<i>Highcharts</i>	38
7.2.4	<i>Google Charts</i>	39
7.2.5	<i>Chart.js</i>	40
7.2.6	<i>Matplotlib</i>	41
7.2.7	<i>Dash-Plotly</i>	42
7.2.8	<i>Grafana</i>	43
7.2.9	<i>JfreeChart</i>	44
7.2.10	<i>Snap4city</i>	45
7.2.11	<i>KIBANA</i>	46
7.3	ESEMPI DI SITI WEB	48
7.3.1	<i>OECD</i>	48
7.3.2	<i>EIGE</i>	49
7.3.3	<i>Flightradar24</i>	50
7.3.4	<i>Singapore</i>	51
7.3.5	<i>EUROSTAT</i>	52
7.3.6	<i>ISTAT</i>	53
7.3.7	<i>OPEN DATA Campania</i>	54
7.3.8	<i>Bristol</i>	55
7.3.9	<i>Comune di Bologna</i>	56
7.3.10	<i>Comune di Firenze</i>	57
8	CONCLUSIONI	58
9	RIFERIMENTI BIBLIOGRAFICI	59

Sommario

La linea di attività 19, “Progettazione Framework per la Governance dei Dati Urbani Energetici”, eseguita nell’annualità 2019, si è posta come obiettivo di pianificare un sistema che permetta il recupero centralizzato dei dati urbani energetici da diverse città, tramite un prototipo di piattaforma centrale agente su scala nazionale, denominata “inter-SmartCityPlatform” (iSCP).

Oltre alla piattaforma centrale, il framework prevede un insieme di risorse, di protocolli, di strumenti, di metodologie per il coinvolgimento degli utenti che verranno dettagliati in altre linee di attività; qui il focus sarà prevalentemente sulla piattaforma centrale. In particolare vengono presentati:

- nel capitolo 1: l’architettura di riferimento;
- nel capitolo 2: la comunicazione interoperabile e gli sviluppi software necessari;
- nel capitolo 3: lo studio di fattibilità per l’integrazione al framework di altre piattaforme nazionali;
- nel capitolo 4: sviluppo sull’interfaccia GUI;
- nel capitolo 5: metodologia di adesione alla piattaforma inter-city (iSCP);
- nel capitolo 6: uno stato dell’arte sulle tecnologie per la visualizzazione dati.

La linea di attività 19 ha raggiunto tutti i risultati prefissati ed è andata oltre, effettuando un ampio studio dello stato dell’arte delle interfacce di reportistica inizialmente non pianificato, in modo da preparare nel modo migliore le basi per la progettazione e sviluppo che avverranno nella annualità successiva.

1 Introduzione

ENEA già da diversi anni sta lavorando a un progetto su scala nazionale che preveda un “framework”, inteso come insieme di tecnologie e supporti agli utenti, per accompagnare le città in un percorso graduale per divenire Smart City.

La linea di attività 19, “Progettazione Framework per la Governance dei Dati Urbani Energetici”, si pone come obiettivo di pianificare un sistema che permetta il recupero centralizzato dei dati urbani energetici da diverse città, quindi agente su scala nazionale.

La progettazione ha previsto:

- un prototipo di piattaforma centrale, su scala nazionale, che possa raccogliere i dati dalle piattaforme locali alle città, tramite un approccio sistematico e condiviso, basato su modelli comuni di rappresentazione dei dati e protocolli per una comunicazione interoperabile (specifiche per l’interoperabilità);
- la definizione di un approccio per il coinvolgimento degli utenti che permetta un’adesione graduale alla piattaforma centrale tramite l’applicazione delle specifiche.

La “piattaforma centrale su scala nazionale” è denominata “inter-SmartCityPlatform” (iSCP).

L’ “approccio per il coinvolgimento degli utenti” sarà inizialmente definito come una guida per l’utente che, successivamente, sarà pubblicato assieme alle altre risorse del Framework.

Con “Framework per la Governance dei Dati Urbani Energetici” intendiamo, quindi, sia la piattaforma nazionale iSCP con le piattaforme locali ad essa integrate, sia le risorse, intese come specifiche, strumenti e guide per il coinvolgimento e formazione degli utenti.

In questa direzione sono stati adottati due importanti risultati del triennio precedente della Ricerca di Sistema Elettrico¹, nell’ambito dell’obiettivo “Piattaforma ICT per lo Smart District”:

- le specifiche per l’interoperabilità, Smart City Platform Specification (SCPS)²,
- il prototipo di piattaforma ICT per lo Smart District / Smart City, Smart City Platform (SCP),

effettuando azioni di consolidamento ed evoluzione verso il framework su scala nazionale.

In questo rapporto si descriveranno le sotto-attività relative alla progettazione del framework:

1. architettura di riferimento del framework su scala nazionale;
2. comunicazione interoperabile tra piattaforma inter-city e piattaforme delle città/distretti aderenti (che comprende la prima definizione del modulo software “bridge” che permette l’integrazione con la piattaforma iSCP);
3. studio di fattibilità sull’integrazione della piattaforma iSCP con la piattaforma PELL;
4. consolidamento della GUI della Smart City Platform, valida sia per SCP che per iSCP;
5. individuazione metodologia di adesione alla piattaforma inter-city;
6. analisi dello stato dell’arte delle tecnologie per la “visualizzazione dei dati”.

Si conclude il rapporto con alcune riflessioni sulle attività svolte.

¹ Triennio RdS 2015-2017, Report RdS/PAR2018/020, RdS/PAR2017/040 e SCPS in RdS/PAR2017/103 fino a 108

² Liberamente consultabili qui: <http://smartcityplatform.enea.it/specification/>

2 Architettura Framework

Il “Framework per la Governance dei Dati Urbani Energetici” intende risolvere il problema della rappresentazione dei dati urbani (prevalentemente energetici) in ambito Smart City e il loro recupero tramite una comunicazione interoperabile, agendo su scala nazionale.

In altre parole, il Framework permette di recuperare KPI sui consumi energetici da diverse città che abbiano aderito alla metodologia connettendo la propria piattaforma di gestione della smart city alla piattaforma di raccolta dati centrale, tramite la comunicazione interoperabile proposta.

Con “Framework” intendiamo, quindi,

- sia la piattaforma inter-SmartCityPlatform (iSCP);
- sia le piattaforme Smart City locali che comunicheranno con la iSCP;
- sia le risorse pubbliche, intese come specifiche, strumenti per la configurazione e partecipazione al framework, tra cui la metodologia di adesione alla piattaforma iSCP.

2.1 Architettura Concettuale

Presentiamo, di seguito, l’architettura concettuale schematica del “Framework per la Governance dei Dati Urbani Energetici” cominciando ad introdurre i concetti chiave che la compongono.

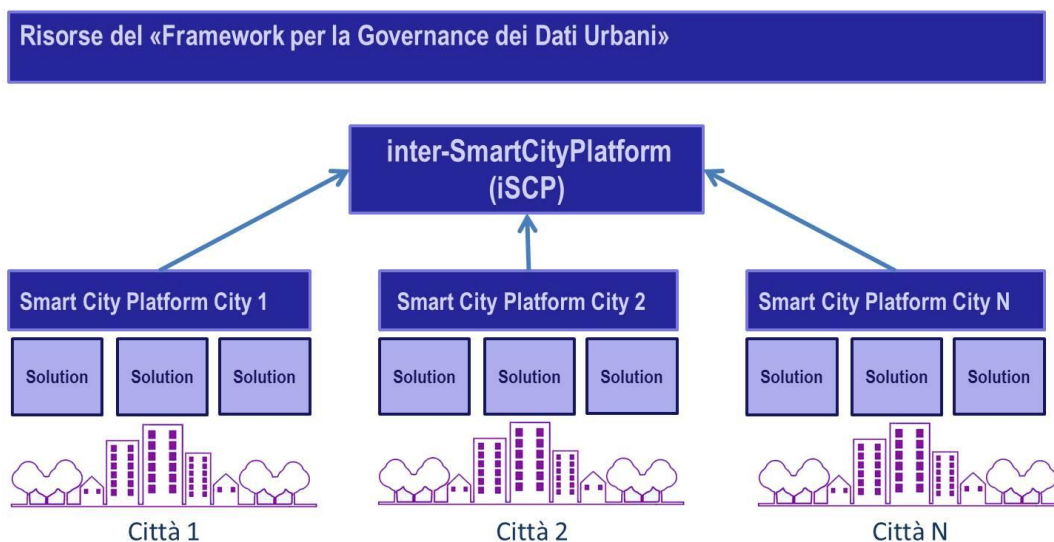


Figura 1. Architettura concettuale del Framework

Il Framework prevede uno scenario in cui sono coinvolte diverse città, ognuna delle quali dotata di una propria Piattaforma Smart City per la gestione dei dati urbani energetici, che sia capace di connettersi alla iSCP (verrà introdotta nel seguito la metodologia di adesione relativa).

Una “Piattaforma Smart City” che abbia implementato tutte le specifiche SCPS è denominata SmartCityPlatform (SCP); ad ogni modo, l’integrazione alla iSCP può avvenire sia con una SCP che con una Piattaforma Smart City generica che implementi l’interfaccia di comunicazione.

La gestione dei dati urbani energetici, nel contesto locale cittadino, è a sua volta gestita verticalmente da diverse Solution.

La inter-SmartCityPlatform (iSCP) raccoglie da ogni Piattaforma Smart City i dati concordati.

Le “Risorse del Framework”, intese come specifiche, strumenti, guide e il supporto per l’adesione alla piattaforma inter-city, saranno risorse pubblicate sul web, così come la gestione centralizzata dei modelli dei dati urbani attraverso l’Ontologia.

Nella Linea di Attività 26 (annualità 2020 e 2021), in collaborazione con l’Università degli Studi dell’Insubria, si definirà uno strumento “Urban Check Up Model” per il coinvolgimento degli utenti nella definizione dei Dati Urbani, attraverso un approccio di indagine che recupererà gli input necessari dai contesti reali.

2.2 Architettura del Framework

Aggiungiamo a questo punto alcuni elementi all'architettura concettuale di partenza per arrivare all'architettura di riferimento del Framework.

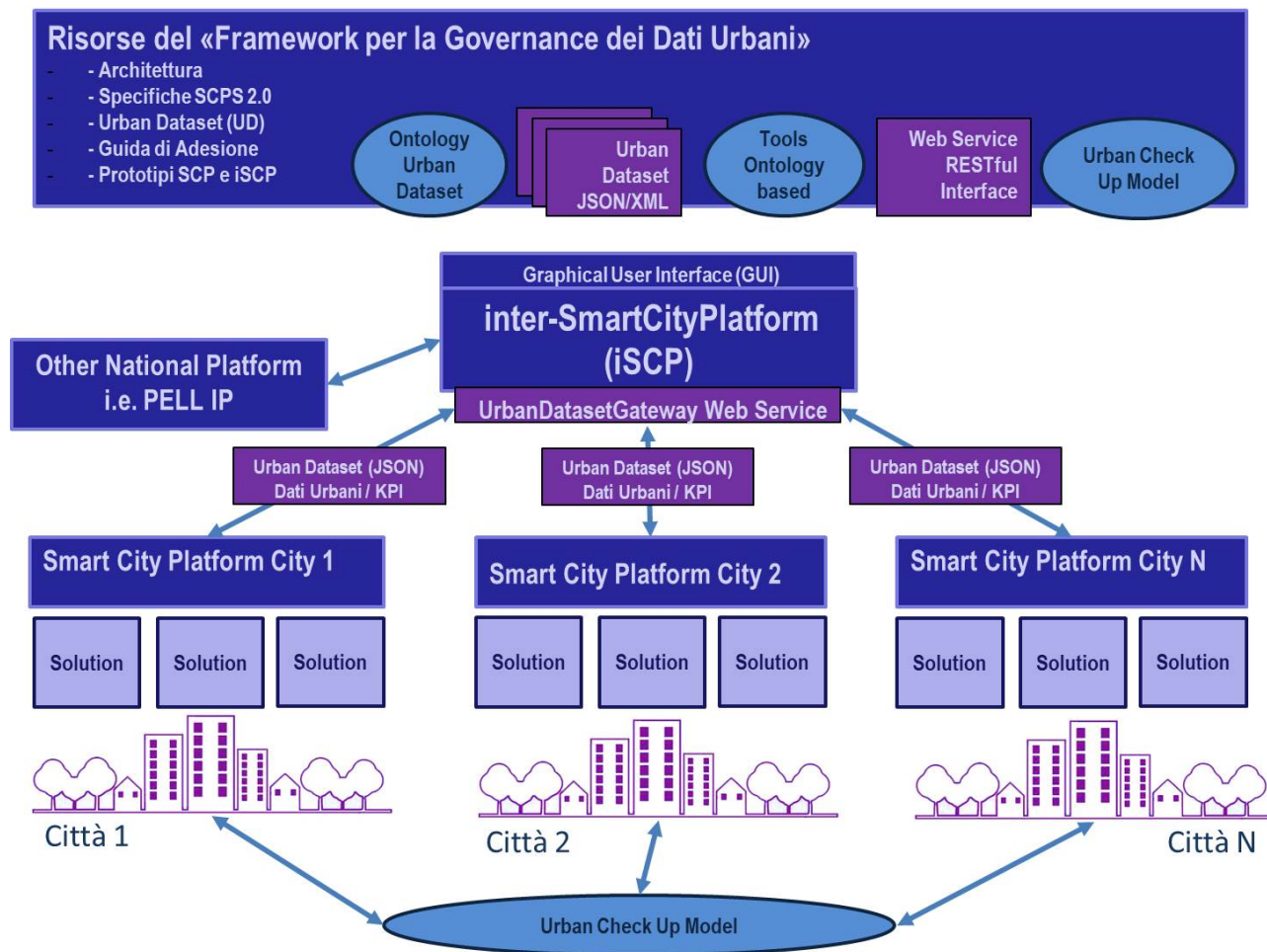


Figura 2. Architettura di riferimento del Framework

Nell'architettura di riferimento del Framework sono evidenziati alcuni concetti:

- la comunicazione tra SCP e iSCP, configurata nel Registry delle SCP registrate, basata su rappresentazione dei dati urbani nel formato comune UrbanDataset (sintassi JSON) e sull'utilizzo di web service RESTful (se ne parlerà nel capitolo 2);
- l'integrazione tra iSCP e altre piattaforme nazionali (come per esempio PELL IP, sulla cui fattibilità si effettuerà uno studio nel capitolo 3);
- il componente Graphical User Interface (GUI), che ricopre un ruolo fondamentale per ottenere uno strumento finale efficace (se ne parlerà nel capitolo 4);
- la Guida Utente per l'Adesione alla iSCP, di cui si parlerà nel cap.5;
- gli strumenti Ontology-based, oggetto delle linee di attività 22 "Ontologia per il Framework per la Governance dei Dati Urbani Energetici" e 23 "Strumenti basati su ontologia per il Framework per la Governance dei Dati Urbani Energetici";
- la definizione di nuovi Urban Dataset per il caso nazionale, oggetto delle linee di attività 24 "Inquadramento europeo dei dati urbani energetici su scala nazionale" e 25 "Definizione di un insieme di riferimento di dati urbani energetici su scala nazionale e municipale";
- l'Urban Check Up Model oggetto della linea di attività 26 "Mappatura quali-quantitativa delle infrastrutture urbane e strutturazione di un Urban Data Model per la transizione smart delle città".

L'architettura del Framework è definita partendo dal basso, dove risiedono i sensori nell'area campo, salendo verso l'alto dove i dati vengono via via aggregati e resi più astratti:

- area campo: sensori e sorgenti dati; questi potrebbero essere l'input fondamentale dell'Urban Check Up Model, strumento del framework per agevolare la raccolta di informazioni dalla città nella direzione di creare gli Urban Dataset necessari;
- area piattaforma locale (o Solution verticale): raccolta ed elaborazione dei dati all'interno del contesto applicativo verticale;
- area piattaforma della smart city (o SmartCityPlatform, SCP): raccolta, nella piattaforma orizzontale della città, dei dati provenienti dalle diverse Solution verticali;
- area piattaforma inter-city (o inter-SmartCityPlatform, iSCP): raccolta, nella piattaforma nazionale, dei dati provenienti dalle diverse città.

Facciamo un esempio concreto riprendendo questi 4 livelli di astrazione del dato:

- area campo: sensori presenti negli del CR ENEA Casaccia (Roma) che permettono di recuperare i consumi elettrici di alcuni edifici;
- area piattaforma locale: la Solution "Smart Building Casaccia" raccoglie questi dati sui consumi elettrici rappresentati con l'UrbanDataset "Building ElectricConsumption" e offre alcuni servizi di gestione e monitoraggio;
- area piattaforma della smart city: la SmartCityPlatform ENEA Casaccia può a questo punto raccogliere i consumi elettrici quotidiani degli edifici dalla "Smart Building Casaccia" che è stata connessa in maniera interoperabile con successo;
- area piattaforma inter-city: la inter-SmartCityPlatform (iSCP) può a questo punto raccogliere i consumi elettrici quotidiani dalla SmartCityPlatform ENEA Casaccia (generati come KPI locali alla città/distretto), nonché da altre piattaforme Smart City, per effettuare un monitoraggio su scala nazionale.

Se una SCP permette di abilitare il monitoraggio su una città o distretto e può abilitare anche la comunicazione tra Solution della stessa città; la iSCP, in maniera simile, permette di abilitare il monitoraggio su scala nazionale e può abilitare la comunicazione tra SCP di diverse città.

Nel capitolo 5 viene riportata una prima ipotesi di "Guida Utente per l'Adesione alla inter-SCP", in cui tramite una serie di passi graduali e consecutivi, viene fornito all'utente il know-how necessario ad implementare la connessione tra la propria SCP e la inter-SmartCityPlatform (iSCP).

Parallelamente alla definizione dell'architettura del framework a livello teorico, si è potenziata l'infrastruttura ICT di ENEA, a livello hardware: è bene ricordare che l'infrastruttura ICT di ENEA è un complesso sistema cloud che consente la creazione di macchine virtuali che permetteranno di definire e ospitare il framework oggetto di questa linea di attività (e che inoltre mantiene funzionante la Smart City Platform Smart Village Casaccia³, piattaforma ICT risultato del precedente triennio della ricerca di sistema elettrico e principale candidata alla prima integrazione con iSCP).

³ Smart City Platform Smart Village Casaccia: <https://scp-gui.portici.enea.it:8445/enea-gui/dist/#/dashboard>

3 Comunicazione Interoperabile

Nel precedente capitolo è stata presentata l’architettura di riferimento del “Framework per la Governance dei Dati Urbani Energetici” che mette in evidenza la presenza di un canale di comunicazione interoperabile tra la singola inter-SmartCityPlatform (iSCP) e le differenti Piattaforme Smart City. In questo capitolo viene descritta, con un dettaglio maggiore, la comunicazione tra una iSCP e N SCP.

3.1 Differenti Scenari

Prima di tutto dobbiamo essere consci di un importante aspetto: la iSCP è una SCP.

Con questa affermazione intendiamo dire che, a livello implementativo, si tenterà di mantenere un unico prototipo software valido sia su scala municipale (SCP) sia su scala nazionale (iSCP) che permetta di essere configurato per una o per l’altra esigenza, garantendo una comunicazione interoperabile sia tra Solution e SCP, sia tra SCP e iSCP, utilizzando le specifiche per l’interoperabilità Smart City Platform Specification (SCPS) consolidate ed estese a questo scopo.

Si presentano quindi due scenari generali possibili relativi alla comunicazione:

- comunicazione tra Piattaforma Smart City generica e iSCP;
- comunicazione tra SCP e iSCP.

Il primo caso si presenta *relativamente* semplice, la Piattaforma Smart City generica dovrà implementare:

- un modulo software di esportazione che mappi i propri dati interni, verso il formato UrbanDataset (UD) (a questo scopo si rimanda allo studio delle specifiche “SCPS Information” e “SCP Semantic” e ai tool semantici⁴ messi a disposizione per agevolare questa operazione;
- un client WS per inviare i dati alla SCP (a questo scopo, si rimanda lo studio delle specifiche “SCPS Communication”, con particolare riferimento all’interfaccia del web service *UrbanDatasetGateway*, relativa al metodo “push”). Il prototipo SCP, infatti, permette di ricevere in maniera passiva gli UrbanDataset (UD) e non permette al momento un recupero attivo di UrbanDataset da altre piattaforme.

Nel secondo caso, sorge naturale una domanda: come possono comunicare tra loro due SCP?

Considerando lo stato pregresso dell’implementazione, la iSCP rimarrebbe in attesa degli UD che le SCP dovrebbero inviarle e, in modo speculare, anche le SCP rimarrebbero in attesa che la iSCP andasse a recuperare gli UD. Entrambi i WS delle due SCP rimarrebbero in attesa.

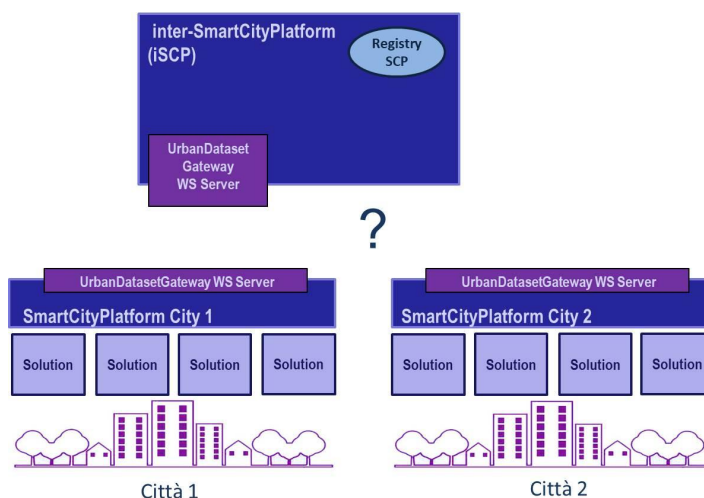


Figura 3. Assenza di comunicazione tra iSCP e SCP

⁴ SCPS Ontology Navigator: <http://smartcityplatform.enea.it:8080/SCPSWebLibrary/ontologyinfo>

3.2 SCP Bridge

La soluzione per risolvere l'assenza di comunicazione tra iSCP e SCP prevede un'evoluzione del prototipo per permettere il recupero attivo di UrbanDataset tramite un nuovo componente software: lo "SCP Bridge". Lo SCP Bridge è un modulo software che:

1. legge la configurazione delle collaborazioni nel Registry della iSCP;
2. per ogni SCP connessa, secondo il periodo configurato, utilizzando il pattern di comunicazione Request/Response (già implementato nell'apposito WS Client) chiama l'UrbanDatasetGateway WS Server presso la SCP, per recuperare gli UD configurati;
3. per ogni UD recuperato, effettua l'inserimento nell'UrbanDatasetGateway Server locale.

Questo funzionamento è mostrato nella seguente figura.

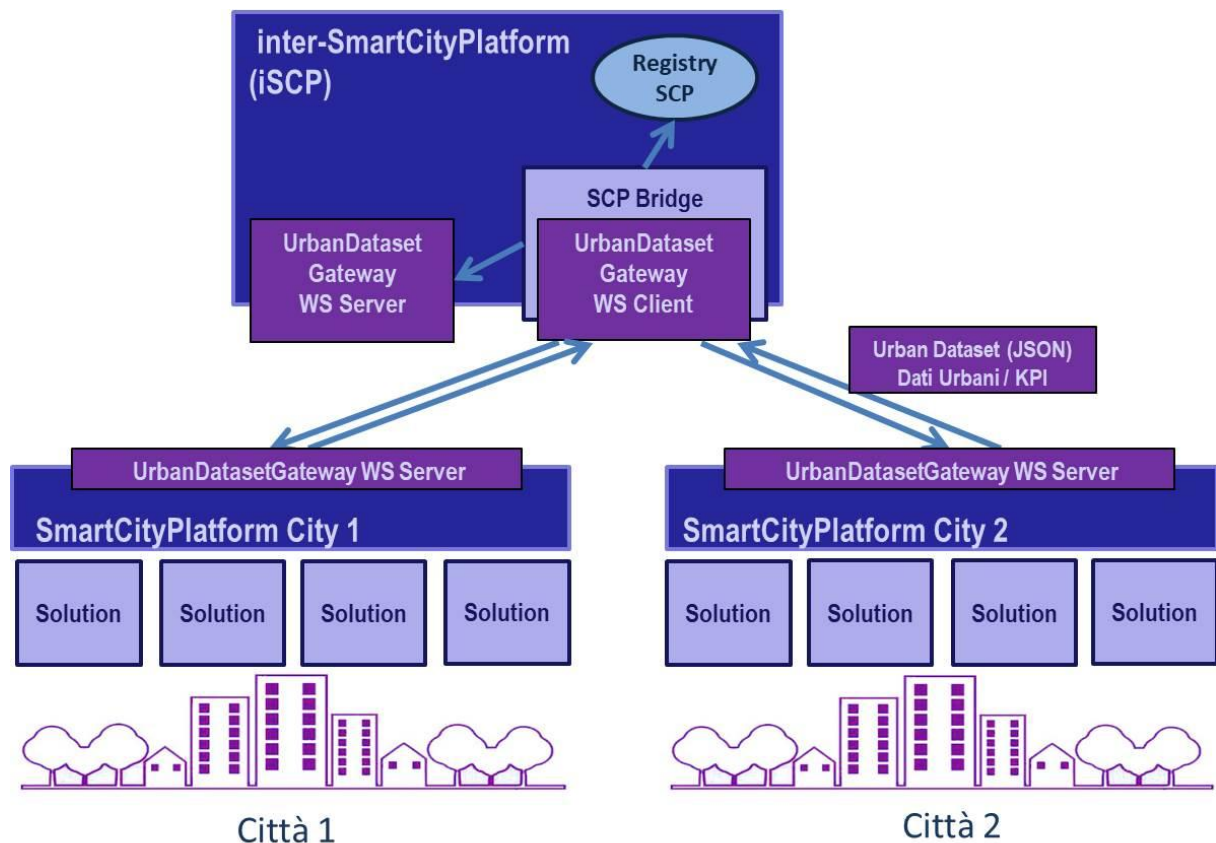


Figura 4. "SCP Bridge" per abilitare comunicazione tra iSCP e SCP

C'è un altro importante aspetto da considerare: la configurazione della comunicazione nei Registry della iSCP e di ogni SCP. Faremo qui di seguito un'ipotesi.

Supponiamo che sia stato definito un UrbanDataset "*City Aggregated Electric Consumption KPI*" che contenga tutte le informazioni relative ai consumi elettrici della città; tale UD dovrà essere trasmesso da ogni SCP alla iSCP centrale. La configurazione di questa trasmissione prevede che:

- l'UD sarà definito e registrato nell'Ontologia;
- sia la iSCP che ogni SCP deve configurare questo UD come supportato;
- ogni SCP deve configurare la produzione di questo UD da parte di una propria Solution/Service che sarà il generatore locale del dataset;
- la iSCP deve configurare la produzione di questo UD da parte di ogni SCP collegata.

Si noti che per implementare la comunicazione tra iSCP e SCP tramite SCP Bridge, saranno necessari due *resource_id*⁵ diversi; ciò è corretto, in quanto la produzione dell'UD della Solution X verso la SCP Y, non è la stessa produzione dell'UD della SCP Y verso la iSCP, anche se l'informazione può essere la stessa.

Lo SCP Bridge dovrà mappare un *resource_id* con un altro per consentire la trasmissione.

Per esempio, lo SCP Bridge utilizzerà:

- il *resource_id* =

SCP-1_DataFusionModule-99_CityAggregatedElectricConsumptionKPI-1.0_20191220120000

per recuperare i dati dalla SCP 1;

- il *resource_id* =

SCP-999_CityAggregatedElectricConsumptionKPI-1.0_20191222000000

per inserire i dati nella iSCP.

Sarà necessario associare un secondo *resource_id* ad ogni produzione configurata nella iSCP.

Ciò è molto interessante perché la stessa identica soluzione è stata individuata nello studio di fattibilità dell'integrazione SCP – PELL (capitolo 3) relativamente la necessità di associare un *topicMQTT* alla configurazione della produzione nella SCP-PELL. Essendo *resource_id* e *topic MQTT* delle normalissime stringhe, la soluzione può essere valida per risolvere entrambe le soluzioni, dando nomi coerenti e abbastanza generali alle nuove informazioni gestite nel Registry e nella GUI. La progettazione del componente SCP Bridge, dovrà dunque prevedere:

- configurazione nel Registry della iSCP della comunicazione con pattern Request/Response (conseguentemente anche la GUI dovrà essere modificata coerentemente);
- configurazione nel Registry di un secondo *resource_id* associato alla produzione (conseguentemente anche la GUI dovrà essere modificata coerentemente);
- clock manager per effettuare il recupero degli UD ad orari prestabiliti;
- integrazione con il WS Client per le chiamate di request (verso le SCP);
- gestione credenziali per lo SCP Bridge
 - o di tipo "reader" per accedere alle SCP (bridge.reader) configurato di default;
 - o di tipo "admin" per scrivere nel UDGateway locale (bridge.admin);

N.B. una Piattaforma Smart City no-SCP può solo fare push dell'UD alla iSCP;

- integrazione con il WS Client per le chiamate di push (verso proprio UDGateway).

Dal punto di vista architetturale è stato considerato ogni aspetto per implementare la comunicazione tra una CP e N iSCP. La progettazione vera e propria dello SCP Bridge, seguendo questa analisi, verrà eseguita il prossimo anno.

Si fa notare, in conclusione, che riutilizzando il prototipo SCP per implementare la iSCP, si potranno riutilizzare sul caso nazionale tutte le funzionalità già sviluppate per il caso cittadino, prima tra tutte la capacità di essere

⁵ "Il *resource_id* è l'identificatore di uno specifico UrbanDataset, prodotto da una specifica Solution verticale, all'interno di una specifica Smart City Platform, in un determinato periodo temporale."
(da "SCPS Core 1.0", Report RdS/PAR2017/103)

nodo di inter-scambio tra due o più Solution che, nel caso iSCP, sarà la capacità di essere nodo di inter-scambio tra due o più SmartCityPlatform e quindi tra due o più città (per questa ragione la nuova piattaforma è stata denominata inter-SCP).

4 Studio di Fattibilità Integrazione SCP - PELL

In questo capitolo viene presentato lo studio di fattibilità relativo all'integrazione di una Smart City Platform (SCP) con la piattaforma PELL IP (Linea di Attività 27 "Ricognizione ed avvio sperimentale all'uso della piattaforma PELL: dati statici") per la gestione dei consumi elettrici dell'illuminazione pubblica su scala nazionale. Questo studio di fattibilità è stato quindi affrontato in entrambe le linee di attività, da due punti di vista diversi: in questo report della LA19 viene presentato lo sforzo richiesto lato SCP, nel report della LA27 viene presentato lo sforzo richiesto lato piattaforma PELL.

4.1 Introduzione Architetture

Introduzione architetture a SCP e PELL per richiamare i principali concetti chiave. L'architettura del PELL prevede:

- diverse declinazioni di piattaforma ICT su scala nazionale: per esempio "PELL IP" è il PELL per il monitoraggio energetico dell'Illuminazione Pubblica (dati dinamici);
- una Big Data Platform per la raccolta e recupero dei dati inviati;
- un MQTT Broker per interfacciarsi con le sorgenti (o Solution di tipo "Smart Lighting") che inviano i dati tramite il protocollo MQTT;
- i dati sono rappresentati con il formato JSON UrbanDataset "Counter Reading", che permette di esprimere i consumi elettrici con un formato condiviso, la cui definizione è nell'Ontologia centrale.

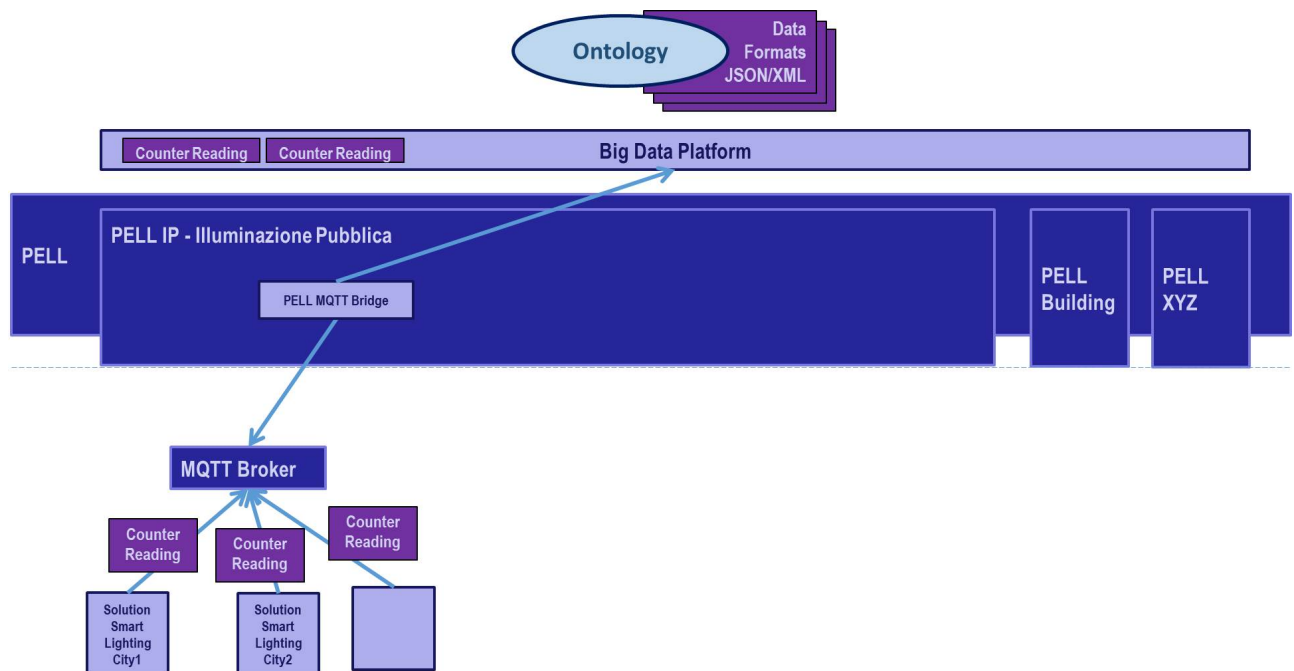


Figura 5. Architettura schematica PELL

L'architettura della piattaforma SCP prevede:

- un SCP-GUI utilizzato per permettere all'utente di interfacciarsi alla piattaforma;
- un SCP-Registry per memorizzare gli utenti, le Solution, e le collaboration configurate;
- un Identity Server per autenticare gli utenti / solution;
- due UrbanDatasetGateway per ricevere e pubblicare Urban Dataset dalle Solution verticali;
- un UrbanDataset Storage per memorizzare e recuperare gli UD.

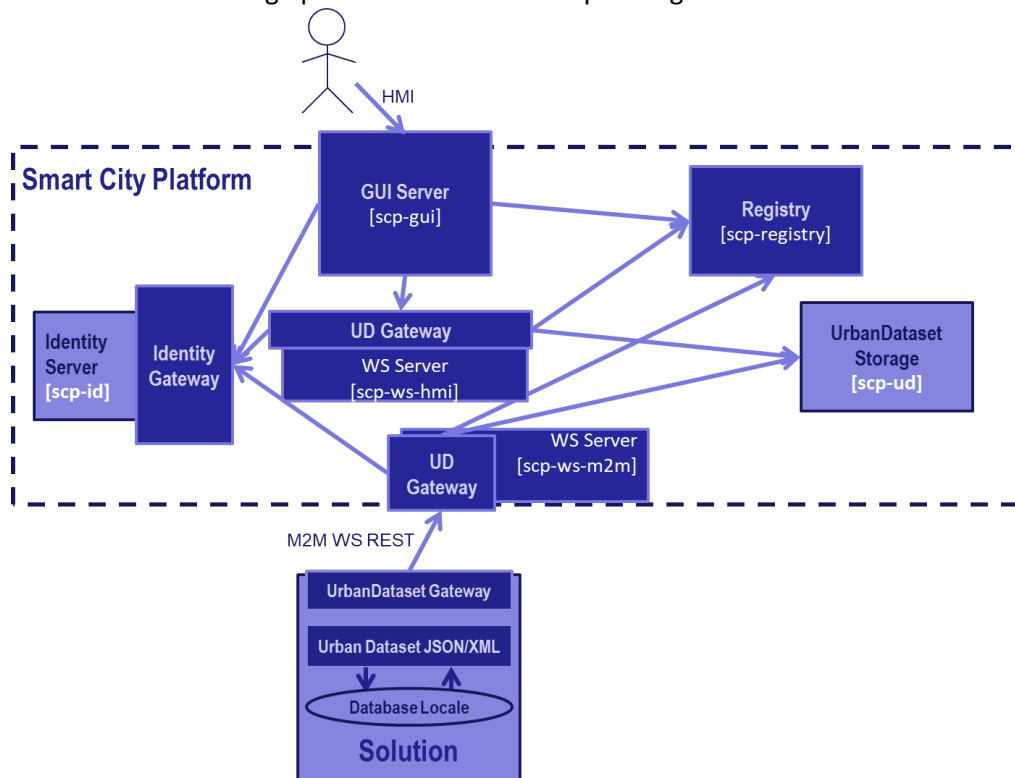


Figura 6. Architettura software della SCP

L'integrazione tra le piattaforme PELL IP e SCP potrebbe portare notevoli vantaggi in termini di riuso di funzionalità sviluppate basate sulle specifiche Smart City Platform Specification (SCPS).

Ci sono notevoli punti in comune tra PELL e SCP:

- 1) l'approccio architetturale (N solution verticali devono inviare i dati a una piattaforma orizzontale);
- 2) la rappresentazione dei dati (tramite l'UrbanDataset definito nelle SCPS);
- 3) il gateway di ricezione dati (sebbene in un caso abbiamo WS RESTful e nell'altro un broker MQTT);
- 4) il concetto di risorsa (nelle SCPS) che equivale al topic (nel broker MQTT del PELL).

Si ritiene possa portare un consistente vantaggio l'integrazione di una istanza SCP apposita per PELL (SCP-PELL) con il broker MQTT Mosquitto del PELL: si intende quindi valutarne la fattibilità.

4.2 Ipotesi di Integrazione

Il corrente studio di fattibilità descrive un'ipotesi di integrazione tra PELL IP e un'istanza di SCP apposita per PELL, denominata SCP-PELL.

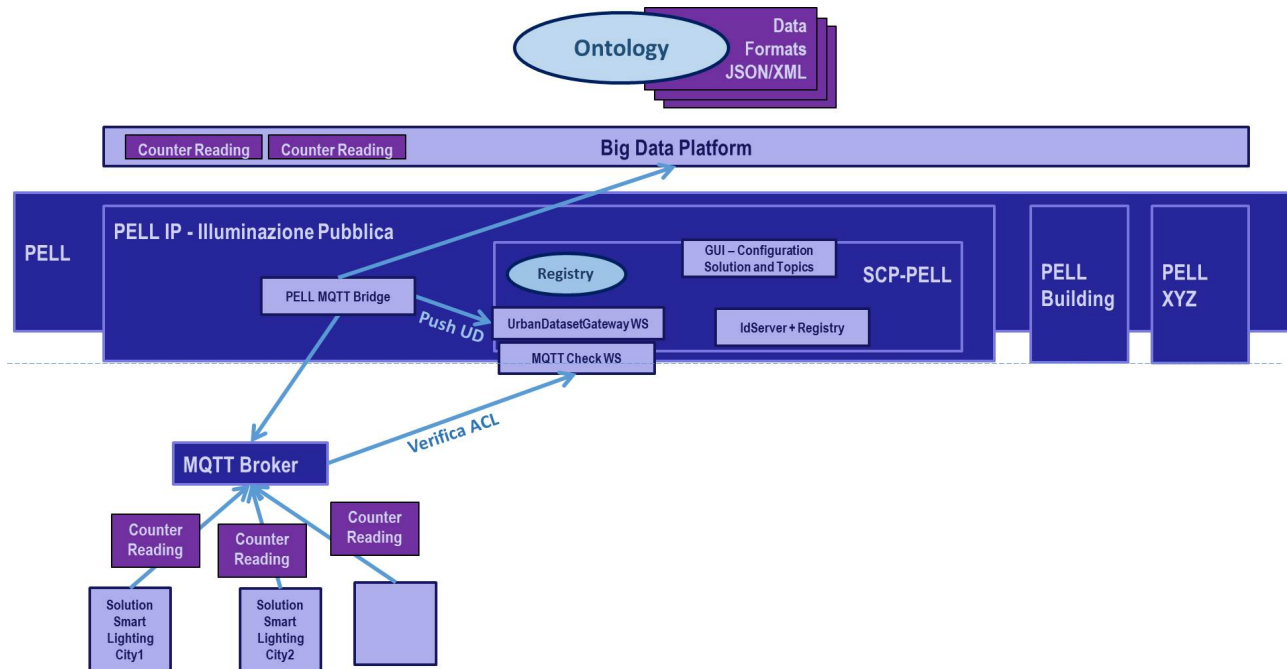


Figura 7. Architettura PELL IP integrato con SCP-PELL

L'ipotesi di integrazione prevede 3 fasi, ogni fase deve essere considerata e presenta le proprie criticità che, in questa sede, si è cercato di risolvere. Le tre fasi da considerare nell'integrazione, a loro volta suddivise in task, sono:

Fase 1 Configurazione

1. Creazione di un account
2. Creazione di 1 o N Solution
3. Creazione della produzione/accesso al topic

Fase 2 Verifica (plug-in Mosquitto del Broker MQTT)

1. Autenticazione User(*checkUser*)
2. Autenticazione Admin (*checkAdmin*)
3. Verifica degli accessi in scrittura (*checkACL*)

Fase 3 Invio Dati (PELL MQTT Bridge)

1. login (username, password)
2. Recupero del resource_id (*getResourceId*)
3. Invio "Counter Reading" tramite push.

PUNTO APERTO: questa prima integrazione comporta una ridondanza dei dati in due sistemi (piattaforma Big Data e SCP-UD-DB basato su Elasticsearch). Si può pensare in futuro, in caso di integrazione avvenuta con successo, a una convergenza anche su questo fronte.

4.3 Fase 1 Configurazione

1. Creazione di un account
2. Creazione di 1 o N Solution
3. Creazione della produzione/accesso al topic.

4.3.1 Task 1.1 “Creazione Account”

Overview

PELL con Broker MQTT necessita di un sistema per la creazione di un account utente (credenziali di accesso come coppia username, password) che la SCP offre.

La creazione di un account di tipo Solution può avvenire da parte dello stesso utente oppure da parte dell’Admin (nel primo caso l’account Solution sarà disabilitato e l’Admin lo dovrà abilitare).

La creazione di un account di tipo Administrator può avvenire solo da parte di un utente Administrator.

Creato un nuovo account, esso finisce nell’Identity Server, basato su tecnologia WSO2, a cui ci si interfaccia con un web service sviluppato da ENEA (IdentityGateway WS).

Intervento

Nessun intervento richiesto.

Si consiglia sia sempre l’ADMIN a creare gli account di tipo SOLUTION.

Manca un sistema di notifica all’admin quando una Solution si iscrive.

4.3.2 Task 1.2 “Creazione Solution”

Overview

Contestualmente alla creazione di un account utente di tipo Solution, nella SCP viene creata anche una Solution associata alle credenziali di accesso. In precedenza era possibile definire una sola Solution per un solo account utente, mancava quindi l’associazione “1 account – N solution” (descritta nel par.5.3.3).

Dopo questo intervento si possono creare più Solution associate a 1 account.

Utilizzando questa funzione nella SCP-PELL si definiranno le diverse sorgenti di lighting.

Intervento

Nella SCP occorre modificare la GUI per permettere l’associazione “1 account – N solution”.

Il Registry è invece già predisposto a questa modifica.

Creando una Solution, verrà creato un SolutionID che deve essere indicato nell’UrbanDataset “Counter Reading”, elemento UrbanDataset/context/producer.

N.B. Le specifiche PELL dovranno descrivere l’utilizzo del SolutionIDnelCounter Reading (nell’elemento context/producer/id; generato in fase di creazione Solution sulla SCP).

4.3.3 Task 1.3 “Creazione Produzione”

Overview

PELL con Broker MQTT necessita di un sistema di definizione delle ACL (Access Control List, ovvero lista di controllo degli accessi), per definire i topic da associare agli username.

Dalla documentazione del plugin per Mosquitto relativo all’autenticazione

<https://github.com/jpmens/mosquitto-auth-plugin#http-auth>

sarebbe necessaria una tabella *acls* di questo tipo

```
=> SELECT * FROM acls;
+-----+-----+-----+-----+
| id | username | topic          | rw |
+-----+-----+-----+-----+
| 1 | jjolie   | loc/jjolie     | 1  |
| 2 | nop      | mega/secret    | 1  |
+-----+-----+-----+-----+
```

La definizione di una ACL come topic associato a un utente corrisponde, nella SCP, alla definizione di una produzione di UrbanDataset associata a una Solution.

N.B. Il seguente Intervento è stato sovrascritto da quello specificato nel Task successivo.

Intervento

Nel SCP-Registry occorre creare una tabella *dataset_production_acl* che metta in relazione il *collaboration_id* (ovvero l’UrbanDataset prodotto da una specifica Solution) con il topic (ovvero l’equivalente MQTT del *resource_id*).

```
=> SELECT * FROM dataset_production_acl;
+-----+-----+
| collaboration_id | topic          |
+-----+-----+
| 2                | test/topic    |
+-----+-----+
```

Se esiste dunque il collegamento `user>solution> collaboration_id > topic` allora quello user può scrivere quella risorsa.

Nella SCP-GUI occorre:

- inserire nella form grafica, relativa l’inserimento della produzione, un campo testo “Topic” dove inserire in questo caso il topic relativo a questo utente; una checkbox “MQTT” deve permettere di abilitare o meno questo campo testo;
- l’inserimento della produzione comporterà l’inserimento del topic relativo in *dataset_production_acl*
- la cancellazione della collaboration cancellerà anche il record *acl* ad esso associato.

4.3.4 Task 1.4 Generalizzazione Bridge

Overview

Il “PELL MQTT Bridge” ha molte affinità con lo “SCP Bridge” (trattato precedentemente nell’architettura della comunicazione interoperabile della inter-SmartCityPlatform, iSCP).

In particolare, le necessità di modificare Registry e GUI per gestire un’informazione, di tipo stringa, associata alla produzione di un UD è identica, anche se:

- nel PELL MQTT Bridge questa stringa sarà il topic;
- nello SCP Bridge questa stringa sarà un secondo resource_id.

Vale la pena, dunque, generalizzare queste modifiche come segue, sovrascrivendo il Task 1.3.

Intervento

Nel SCP-Registry occorre creare una tabella dataset_production_bridge che metta in relazione:

- il collaboration_id (ovvero l’UrbanDataset prodotto da una specifica Solution)
- con un’informazione supplementare di tipo stringa (extra).

```
=> SELECT * FROM dataset_production_bridge;
+-----+-----+
| collaboration_id | extra          |
+-----+-----+
| 2                | società/bologna |
+-----+-----+
```

Se esiste dunque il collegamento user>solution> collaboration_id > extra allora quello user può scrivere quella risorsa con topic = extra.

Nella SCP-GUI occorre:

- inserire nella form grafica, relativa l’inserimento della produzione, un campo testo “Extra” dove inserire l’informazione aggiuntiva, in questo caso il topic relativo a questo utente; una checkbox “Bridge” deve permettere di abilitare o meno questo campo testo;
- l’inserimento della produzione comporterà l’inserimento del topic relativo indataset_production_bridge
- la cancellazione della collaboration cancellerà anche il record ad esso associato.

4.4 Fase 2 Verifica

Questa fase è strettamente relativa all'utilizzo del plug-in Mosquitto.

1. Autenticazione User (*http_checkUser_uri*) [post]
2. Autenticazione Admin (*http_checkAdmin_uri*) [post]
3. Verifica delle ACL (*http_aclcheck_uri*) [post]

È stato testato il plugin sulle tre chiamate.

L'algoritmo del plugin che utilizza queste tre chiamate è il seguente:

- call *checkUser* :
 1. se va a buon fine, restituisce codice http 200,
 - call *checkAdmin* :
 - se va a buon fine, restituisce codice http 200,
 - INVIO UD
 - altrimenti codice http 401,
 - call *checkACL*:
 - se va a buon fine, restituisce codice http 200,
 - INVIO UD
 - altrimenti codice http 401;
 - 2. altrimenti codice http 401.
- In caso di altre eccezioni non relative all'autorizzazione e ai permessi, viene ritornato un generico errore con codice http 500.

4.4.1 Task 2.1 "Autenticazione Account"

Overview

Il broker MQTT necessita di verificare l'account utente (create tramite la SCP-PELL nel Task 1.1).

L'utente MQTT, tramite il plugin di Mosquitto, si deve autenticare fornendo: *username, password*.

Intervento

Sulla SCP è necessario definire un nuovo web service sulla SCP: MQTTCheck che esponga il 1° metodo: *checkUser(username, password)*

questo metodo, internamente, richiamerà il metodo esistente

UrbanDatasetGateway/login(username, password)

che resta inalterato.

La *login* ritornerà

- un messaggio di success: in quel caso la *checkUser* ritornerà 200;
- un messaggio di failure: in quel caso la *checkUser* ritornerà 401.

Il JWS Token (che chiameremo *scp-token*) in questa fase viene ignorato.

4.4.2 Task 2.2 “Autenticazione Admin”

Overview

Il broker MQTT necessita di verificare se l'account utente è di tipo admin/checkAdmin.

L'utente MQTT, tramite il plugin di Mosquitto, invocherà il metodo

checkAdmin(username, password)

Intervento

Sulla SCP, il nuovo web service: MQTTCheck

Deve esporre il 2° metodo: *checkAdmin(username, password)*

questo metodo, internamente:

- 1) richiamerà il metodo esistente

UrbanDatasetGateway/login(username, password)

- 2) in caso di success, verificherà se l'utente è di tipo admin o meno nel Registry, e
 - se admin, ritornerà 200;
 - se non admin, ritornerà 401.

4.4.3 Task 2.3 “Verifica ACL”

Overview

Il plugin del broker MQTT, seguendo il proprio algoritmo, se la checkUser ha restituito 200 e la checkAdmin ha restituito 401, necessita di verificare le ACL (create tramite la SCP-PELL nel Task 1.3).

Il metodo che deve essere chiamato, secondo il plugin di Mosquitto, fornendo:

username, clientId, topic, accdove:

- *username*: è lo username che si è precedentemente autenticato;
- *clientId*: può essere ignorato;
- *topic*: è il topic configurato con la produzione della risorsa;
- *acc*: con valore “2” indica la produzione di questo topic (non è prevista la lettura).

Intervento

Sulla SCP, sul nuovo web service MQTTGateway

deve essere pubblicato un 3° metodo: *checkACL(username, clientId, topic, acc)*

dove:

- *clientId*: deve essere "" stringa vuota e comunque sarà ignorato;
- *acc* deve essere 1e comunque sarà ignorato;

che:

- verifica l'associazione user-topic e se consistente ritorna 200;
- in caso contrario ritorna 401.

PUNTO APERTO: per evitare che qualcuno, conoscendo lo username, invochi questo metodo, è necessario rendere il WS MQTTCheck privato, richiamabile solo dal Broker MQTT.

SOLUZIONE: Configurare il Firewall della SCP-PELL, in modo tale che riceva chiamate solo dal MQTT Broker.

4.5 Fase 3 Invio UD

Una volta effettuata la parte di configurazione e verifica, l'invio degli Urban Dataset è abilitato secondo il seguente flusso:

- Smart Lighting Solution -> Broker MQTT;
- Broker MQTT -> PELL MQTT Bridge;

- PELL MQTT Bridge -> Big Data Platform.

Allo scopo di riutilizzare tutte le funzionalità di monitoraggio traffico dati sviluppate nella SCP, si tenterà di aggiungere un ulteriore passaggio, in modo tale che gli UD vengano inviati anche alla SCP tramite il metodo di push, ovvero: PELL MQTT Bridge -> SCP.

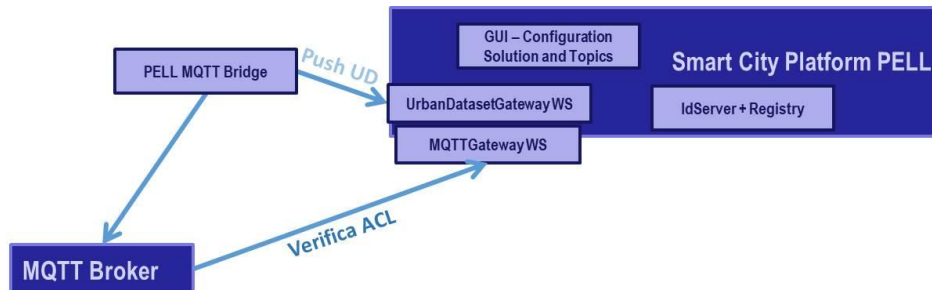


Figura 8. Dettaglio Architettura PELL IP integrato con SCP-PELL

Ciò che dovrebbe avvenire per abilitare l'invio di UrbanDataset da PELL MQTT Bridge a SCP:

1. Recupero di UrbanDataset e topic dal Messaggio MQTT;
2. Chiamata (1di3) a metodo WS *login*
3. Chiamata (2di3) a metodo WS *getResourceId* dato il topic;
4. Chiamata (3di3) a metodo WSpush.

Overview

Il PELL MQTT Bridge, per ogni Messaggio MQTT ricevuto dal MQTT Broker:

- recupera dal Messaggio MQTT i relativi "topic" e "UrbanDataset";
- recupera le credenziali (username, password) per l'accesso a SCP (utilizzerà credenziali di tipo Administrator create ad hoc);
- effettua la chiamata (1di3) al metodo *login* nella SCP tramite l'UrbanDatasetGateway WS;
- salva temporaneamente il "jwt-token", ritornato dal metodo *login* in caso di successo;
- effettua la chiamata (2di3) al metodo *getResourceId* fornendo "jwt-token" e "topic";
- salva temporaneamente il "resource_id", ritornato dal metodo *getResourceId*;
- effettua la chiamata (3di3) al metodo *push* fornendo "token", "resource_id" e l' "UrbanDataset".

Intervento

Lato PELL MQTT Bridge, è necessario implementare l'algoritmo suddetto.

Lato SCP, è necessario

- creare l'account di tipo Administrator per il PELL MQTT Bridge;
- implementare nel *WebServiceUrbanDatasetGateway* il metodo *getResourceId(jwt-token, topic)* che, dato il topic, ritorna il *resource_id* relativo.

N.B. È stato verificato con successo che, utilizzando l'account di tipo Administrator, la SCP registrerà l'UrbanDataset come se la chiamata al metodo push fosse stata fatta direttamente dalla Solution originaria, riuscendo così a usufruire del completo sistema di monitoraggio della SCP.

4.6 Conclusioni Studio di Fattibilità

Sebbene non banale e articolato in diversi interventi, lo studio di fattibilità ha dato esito positivo: si avvierà di conseguenza l'implementazione dell'integrazione tra SCP e PELL nella Linea di Attività 20 (annualità 2020) al fine di riutilizzare il sistema di autenticazione e monitoraggio dati offerto dalla SCP in ambito PELL e consentire, in questo modo, anche la comunicazione con la piattaforma inter-SmartCityPlatform (iSCP).

5 Consolidamento GUI

Abbiamo descritto nei precedenti capitoli che la nuova piattaforma che agisce su scala nazionale inter-SmartCityPlatform (iSCP) è basata sul prototipo Smart CityPlatform (SCP).

La SCP è un prototipo di Piattaforma ICT per Smart District / Smart City, risultato del precedente triennio della Ricerca di Sistema Elettrico; come tale ha subito un periodo di sperimentazione (tuttora in corso) che permettesse di individuare errori, anomalie e bug; conseguentemente sono stati individuati i miglioramenti da effettuare per ottenere un generale consolidamento della piattaforma, sia per l'uso cittadino, sia per l'uso che se ne farà a livello nazionale.

Descriviamo in questo capitolo l'aggiornamento del componente Graphical User Interface (GUI) della SCP. Verrà di seguito riportata l'esatta lista di funzionalità correttamente implementate nell'aggiornamento software, con una organizzazione dettagliata in Task e Sotto-Task.

5.1 Task "Consolidamento"

Il Task di "Consolidamento" è un insieme di modifiche non incluse nei successivi Task più specifici.

5.1.1 Sotto-Task "Aggiustamenti"

Si tratta di una serie di "aggiustamenti" della GUI considerati minori:

- il layout del form "Register" (alternativa a Login) è stato sistemato: le barre di scorrimento erano inutili e sono state eliminate;
- sezione HISTORY: i dataset nel menu a tendina sono stati ordinati per nome;
- sezione HISTORY: le colonne TIMESTAMP e UD TIMESTAMP ora permettono il riordinamento dei record sottostanti (con "ricordo" di quanto scelto);
- sezione HISTORY: i Filters ora hanno "memoria" di quanto preselezionato;
- nella definizione di una prod/access la form è stata migliorata con i valori di default "Choose" e il "ricordo" di quanto immesso, p.es. quando clicco 2 volte "on demand";
- nella home e nella history, l'utente Solution ora ha un filtro "ONLY MINE" settato di default per vedere solo i propri UD e nascondere gli OPENDATA degli altri (per questo utente, questo filtro sostituisce il precedente "HIDE OPENDATA");
- per l'utente Admin, il filtro "HIDE OPENDATA" è stato sostituito con "ONLY OPENDATA";
- le unità di misura, se presenti, ora sono riportate nella vista/view UD;
- nel grafico degli indicatori, ora non sono presi in considerazione gli UD con state=DELETED
- le chiamate del metodo isAlive venivano riportate nel log di tomcat, diverse chiamate ogni minuto, ma non c'era bisogno di loggarle tutte (al fine di avere un file di log più piccolo e più leggibile); il log è quindi stato alleggerito;
- sostituita l'icona web;
- modificata la posizione e rinominare il file di configurazione in: */etc/scp/scp-gui.properties*.

5.1.2 Sotto-task "Riorganizzazione HOME"

La sezione HOME ora presenta, nell'ordine, dall'alto verso il basso:

- Testo di benvenuto;
- la sezione FILTERS indicatori e il grafico degli indicatori di interoperabilità;
- la sezione FILTERS REPORTS e la sezione REPORTS (oggetto del task omonimo);
- la tabella di overview delle Solution integrate.

5.1.3 Sotto-task “URL della Solution”

Gestione della proprietà URL associata a una Solution.

- ora è visualizzato l’URL sia nella Home che nella sezione Solution nella tabella delle solution (rendendo il nome un link all’URL);
- ora permette l’editazione dell’URL.

5.1.4 Sotto-Task “HELP parametrizzabile”

Le voci del menù “HELP” della SCP ora sono parametrizzabili prendendo l’input da un file di testo esterno, in questo modo il personale ENEA può modificare questo menù in maniera indipendente.

5.2 Task “Modifica Algoritmo Grafico Indicatori”

E’ stato parametrizzato il periodo su cui si effettua la verifica nel grafico degli indicatori per l’interoperabilità presente nella Home.

Al momento ogni UD viene verificato in base la propria periodicità di invio, che viene usata per calcolare il rispettivo periodo su cui effettuare la verifica (p.es. 1 UD inviato ogni ora, viene verificato su 1 settimana).

E’ stato scelto di impostare, da GUI, un periodo di verifica valido per tutti (1 giorno, 1 settimana, 1 mese, 1 anno).

N.B. Nel caso il periodo di verifica superi il periodo di ricezione, la colonna relativa compare comunque.

5.3 Task “New Users”

5.3.1 Sotto-Task “New READER user”

E’ stato introdotto di un utente di tipo READER a fianco di quelli esistenti.

L’utente READER è un utente “osservatore” che deve poter accedere alle funzionalità della SCP senza il potere di compiere azioni di modifica nella SCP stessa; l’utente READER vede, dunque, tutte le sezioni e i dati così come li vede l’utente ADMINISTRATOR ma non può editare o cancellare.

5.3.2 Sotto-Task “New SERVICE user”

E’ stato introdotto di un utente di tipo SERVICE a fianco di quelli esistenti.

L’utente SERVICE si comporta come SOLUTION per accedere a uno o più UrbanDataset e implementare servizi strettamente relativi alla Smart City Platform.

5.3.3 Sotto-Task “1 account – N solution”

In precedenza la Smart City Platform permetteva di associare a 1 Account Utente 1 sola Solution.

E’ stata modificata la GUI per associare 1 Account Utente a 1 o N solution.

Per la distinzione tra Solution e Service si faccia riferimento al precedente sotto-task relativo, tenendo presente che un utente può essere di uno ed un solo tipo (ruolo) e quindi potrà istanziare solo quel tipo N volte (p.es. un utente di tipo solution potrà essere associato a 1 o più solution, un utente di tipo service, potrà essere associato a 1 o più servizi).

5.4 Task “Update User email/password”

In precedenza la SCP permetteva di editare gli utenti (Admin e Solution) solamente sui dati dell’anagrafica. Mancava il supporto alla modifica dell’account (username, password) che ora è presente.

5.4.1 Sotto-Task “Update Username”

L’utente DEVELOPER ora è in grado di modificare lo username di tutti, escluso se stesso.

5.4.2 Sotto-Task “Update Password”

L’utente Developer ora può modificare la password di tutti, compreso se stesso.

L’utente ADMINISTRATOR ora può modificare la password di tutti, escluso l’utente Developer.

L’utente SOLUTION (e SERVICE) ora può modificare solo la propria password.

5.5 Task “SCP per Smartphone/Mobile”

La visualizzazione della GUI SCP presentava diverse anomalie “orizzontali” relative al layout su device di dimensione “smartphone” (android, apple). E’ stato realizzato un importante aggiustamento del layout per permettere la corretta visualizzazione in modo da usufruire di tutte le funzionalità SCP anche su device portatili di piccole dimensioni.

5.6 Task “Integrazione Bridge”

Questo task è relativo l’integrazione tra un’istanza SCP/iSCP con un Bridge (sia che si tratti dello SCP Bridge, sia che si tratti del PELL MQTT Bridge).

E’ stata modificata la GUI per associare una stringa (*topic o resource_id*, astratti nel campo “external reference”) a una produzione di UrbanDataset andando a supportare, in questo modo, un pattern di comunicazione⁶ diverso dalla classica *push* per la ricezione passiva degli UD, ovvero il pattern request/response per il recupero attivo degli UD presso un endpoint esterno.

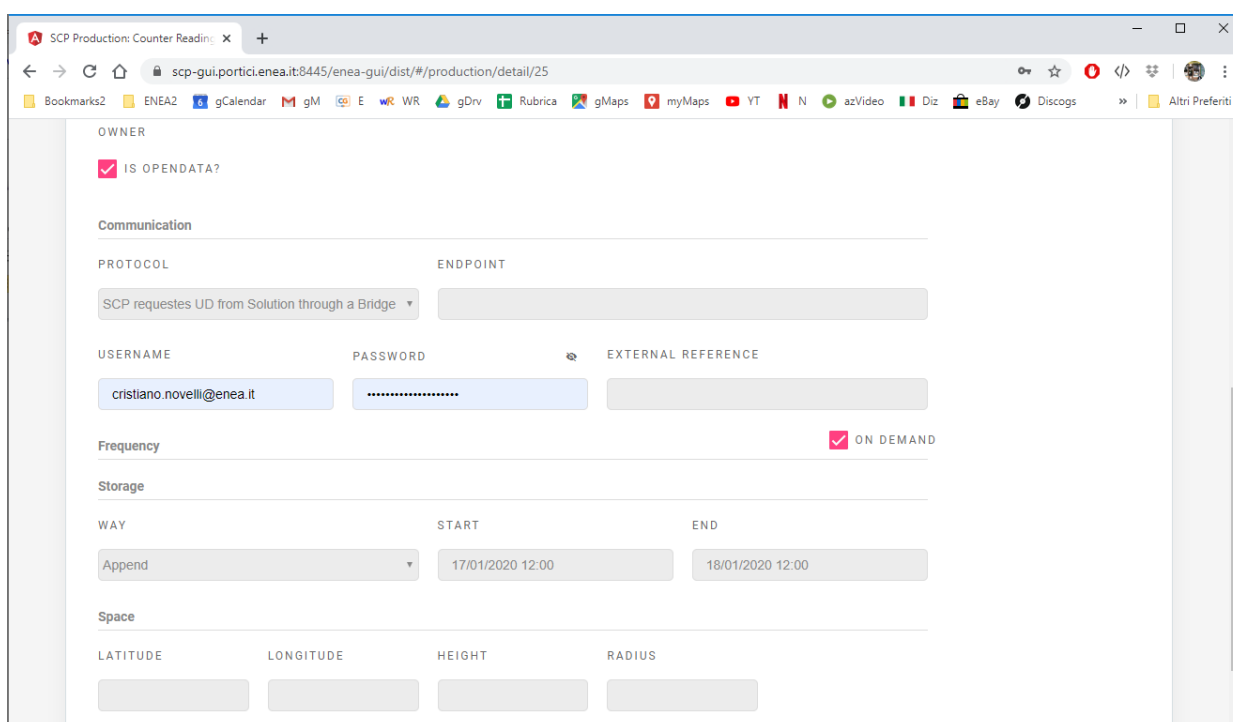


Figura 9. SCP-GUI – Definizione della comunicazione con Bridge

5.7 Task “Enabling Sezione REPORTS”

E’ stata circoscritta una sezione “REPORTS” (interna alla HOME) dedicata alla visualizzazione di “Report”. I Report consentano il monitoraggio dal punto di vista energetico della città/distretto attraverso la presentazione, in formato tabellare o di schema grafico, dei dati (UrbanDataset) che la SCP metterà a disposizione e che il componente GUI già recupera.

La sezione “REPORTS” ora viene attivata / disattivata tramite checkbox nella GUI che agisce su un campo presente nel file *enea-gui.properties*.

5.8 Task “Coaching on-the-job”

⁶ Si veda l’argomento dei pattern e dei metodi push e request/response nelle specifiche “SCPS Communication Level”, Report Rds/PAR2017/108

Al fine di assistere il personale ENEA, è stata programmata una giornata di formazione, presso la sede ENEA di Bologna, con il seguente percorso:

1. Download, Setup, Configurazione ambiente di sviluppo;
2. Apertura progetto "SCP-GUI" e modifica di un testo di una pagina;
3. Build in locale;
4. Deployment e Test in locale;
5. Deployment su VirtualMachine.

6 Sistema di Adesione alla inter-SCP

Tra i risultati attesi in questa linea di attività, è stata individuata una metodologia di supporto all'utente per l'adesione alla piattaforma inter-SmartCityPlatform (iSCP), ovvero un sistema che sia punto di riferimento per i diversi utenti per aderire alla piattaforma di raccolta dati su scala nazionale, tramite una serie di passi graduali, con riferimenti a specifiche pubbliche e strumenti, in modo tale che le informazioni siano fruibili dagli utenti in modo efficace.

Nell'intento di individuare la metodologia di adesione, ci si è posti le seguenti 4 domande:

- A quali tipi di utenti si rivolge questa metodologia di adesione alla piattaforma iSCP?
- Cosa comporta l'adesione alla piattaforma iSCP?
- Come agevolare l'utente con un percorso di adesione alla iSCP?
- Come formalizzare il percorso di adesione e renderlo disponibile agli utenti?

In questo capitolo si tenterà di dare una risposta a queste domande, andando così a delineare la metodologia di adesione alla piattaforma inter-city (iSCP).

A quali tipi di utenti si rivolge questa metodologia di adesione alla piattaforma iSCP?

I tipi di utente a cui si rivolgerà questa metodologia di adesione sono due:

- Il rappresentante amministrativo della Smart City
- Il tecnico della Piattaforma Smart City, che dovrà implementare l'integrazione.

Il responsabile amministrativo della Smart City potrà capire, attraverso la lettura dell'introduzione architeturale, i concetti chiave per comprendere cosa comporta aderire alla piattaforma iSCP e quale sforzo sia richiesto. Il suo benessere è indiscusso requisito.

Il tecnico della Piattaforma Smart City è colui che implementa l'integrazione software vera e propria con la piattaforma inter-SCP, seguendo il percorso completo della metodologia di adesione che provvederà a fornirgli tutte le indicazioni tecniche del caso.

Per esempio:

- Il "responsabile amministrativo" della Smart City può essere la municipalità stessa, nella veste del funzionario amministrativo o suo delegato; oppure il responsabile della multi-utility a cui è stato dato il compito di gestire la piattaforma centrale della città;
- Il "tecnico della Piattaforma Smart City" è in genere uno sviluppatore software (o un team) che si occupa della manutenzione della Piattaforma Smart City locale e sa come interfacciarsi per implementare il canale interoperabile che verrà descritto nel seguito.
-

Cosa comporta l'adesione alla piattaforma iSCP?

L'adesione alla piattaforma iSCP comporta l'invio di set di informazioni relativi alla gestione energetica della città, dalla Piattaforma Smart City locale di una città verso la inter-SmartCityPlatform (iSCP), piattaforma di monitoraggio energetico su scala nazionale.

Nel primo capitolo abbiamo presentato l'architettura concettuale e di riferimento del Framework, ponendo l'accento sulla comunicazione interoperabile tra

- N Piattaforme Smart City (piattaforme di gestione energetica delle città) e
- 1 Piattaforma su scala nazionale (denominata inter-SmartCityPlatform, iSCP).

L'adesione al framework richiede che la Piattaforma Smart City locale della città di riferimento sia capace di connettersi alla piattaforma iSCP.

Come agevolare l'utente con un percorso di adesione alla piattaforma iSCP?

Per agevolare l'utente nel processo di connessione della propria piattaforma Smart City alla iSCP, è stato delineato un percorso di adesione strutturato in passi graduali, che guidino gli utenti verso la realizzazione della comunicazione interoperabile e automatizzata tra i due sistemi, attraverso l'utilizzo di specifiche ed esempi.

Le specifiche che verranno utilizzate sono denominate Smart City Platform Specification (SCPS), risultato del lavoro del precedente triennio della Ricerca di Sistema Elettrico; in questo contesto tali specifiche saranno riviste e adattate al caso di framework su scala nazionale.

Il percorso di adesione alla iSCP può essere riassunto in questa serie di passi:

1. Architettura di Riferimento: descrizione e condivisione dell'architettura di riferimento del framework e dei concetti chiave;
2. Registrazione e Configurazione sulla inter-SCP della piattaforma Smart City e degli Urban Dataset che verranno inviati, secondo quale canale e periodicità;
3. Verifica dei Dati Urbani / KPI da inviare: descrizione e comprensione dei dati che dovranno essere generati da ogni Piattaforma Smart City verso la inter-SCP;
4. Recupero degli UrbanDataset "template": download e comprensione dei template in formato JSON degli UrbanDataset (definiti secondo le specifiche "SCPS Information" e "SCPS Semantic") che verranno utilizzati per la rappresentazione digitale dei dati urbani / KPI;
5. Implementazione del processo di esportazione dei dati urbani, presso la piattaforma Smart City, usando i template Urban Dataset JSON;
6. Implementare la comunicazione, dipendentemente dallo scenario:
 - La Piattaforma Smart City è una SCP: dovrà generare l'UD e lo pubblicherà tramite il proprio WS; la iSCP lo recupererà tramite lo SCP bridge;
 - La Piattaforma Smart City è generica, no-SCP: dovrà sviluppare un client web service (specifiche SCPS dell'UrbanDatasetGateway) per inviare i dati urbani alla iSCP secondo il periodo configurato tramite metodo "push";
7. Monitoraggio del traffico di UrbanDataset, individuazione di anomalie, ripristino della comunicazione interoperabile, agendo di concerto con la iSCP.

I diversi passi del percorso di adesione verranno maggiormente dettagliati nel corso del triennio, con precisi riferimenti alle specifiche e con esempi a corredo, per poi essere pubblicati sul web assieme alle altre risorse del framework, in modo da creare un punto di riferimento stabile, chiaro ed efficace.

7 Stato dell'Arte Servizi di Visualizzazione Dati

Negli ultimi anni si è registrata una mole incredibilmente crescente di dati, sotto la spinta di una forte innovazione tecnologica delle telecomunicazioni, della diffusione degli *smartphone*, dei *social network* e di numerosi servizi per gli utenti, di leggi e siti web sugli Open Data^{7,8,9,10}, della diffusione di Big Data, nonché della necessità di una gestione energetica più sostenibile, della disponibilità di una sensoristica più efficace a prezzi ridotti e di un numero crescente di progetti sulle Smart City [1]. Di conseguenza sono stati sviluppati, e continuano ad essere sviluppati, centinaia di strumenti software e servizi che consentono la visualizzazione di dati.

Oltre ai software di visualizzazione, rimane centrale la *scelta di KPI (Key Performance Indicators)* che possano descrivere e rappresentare un fenomeno. I KPI possono riferirsi a diverse unità di misura e grandezze (relative o assolute) e spesso risulta efficace per un quadro completo:

- a. **osservarne la *variazione temporale*** (anche per monitorare il raggiungimento di obiettivi);
- b. **osservarne la *variazione spaziale*** attraverso mappe (in particolare nelle Smart Cities);
- c. **visualizzare *indicatori*** contemporaneamente in appositi grafici (p.es. grafici a ragnatela);
- d. **confrontare *indicatori*** con altre realtà simili oppure con indici di riferimento, con limiti di legge, con medie (p.es. confronto del consumo energetico per abitante con altre città).

Si possono dividere i tipi di visualizzazioni di dati in:

- **singoli grafici** di poche variabili e/o KPI (p.es. visualizzazione di Open Data rilasciati dalla PA e non solo) oppure **insieme di grafici** e pannelli (istogrammi, grafici a torta, percentuali, infografiche, ecc.) per rappresentare un fenomeno, fino a cruscotti e dashboard;
- **mappe fisse o interattive**. Nel secondo caso l'utente può esplorare una zona di interesse, attivando e disattivando diverse variabili, su differenti scale;
- **grafici con variabili statiche o dinamiche**, ovvero con valori che possono aggiornarsi automaticamente ad intervalli di periodo prefissati o addirittura in **real-time**, sfruttando la connettività di alcuni sensori (p.es. traffico e posizione degli aerei in volo in Italia).

Gli obiettivi principali degli strumenti/servizi di visualizzazione dei dati sono di:

1. **sintetizzare** e mostrare un gran numero d'informazioni in una sola immagine e quindi in un solo sguardo ("un'immagine vale più di mille dati");
2. **monitorare** l'andamento temporale di una variabile e quindi il *trend* di un fenomeno;
3. **aumentare la conoscenza**, esplorare, sviluppare e valutare ipotesi, scoprire errori e anomalie nei dati, "estrarre valore" dai dati, trovare *patterns*, evidenziare il contributo relativo di diversi parti e la correlazione tra diversi variabili, estrarre nuove informazioni e permettere di creare nuove interpretazioni dei fenomeni (*insights*);
4. **facilitare l'accesso alle informazioni**, agevolando la diffusione dei dati, in quanto la rappresentazione grafica può essere più rivelatrice del calcolo statistico;
5. **offrire consapevolezza e base per decisioni**, comunicando l'informazione in modo chiaro, intuitivo ed efficace. Una comunicazione efficiente può aumentare consapevolezza e partecipazione dei cittadini, condivisione e collaborazione, oltre a stimolare coinvolgimento e attenzione (*engagement*) [2].

La visualizzazione di dati è insieme "un'arte e una scienza": fa parte dell'approccio scientifico, ma presenta anche una parte soggettiva e arbitraria in diverse scelte, come nella selezione del *tipo di grafico* e delle *variabili* da

⁷ http://www.datiopen.it/it/cosa_sono_gli_open_data

⁸ <https://www.dati.gov.it/>

⁹ Decreto Legge 18.10.2012, n. 179.

¹⁰ Direttiva 2013/37/UE del Parlamento europeo e del Consiglio.

rappresentare, nella *correlazione* da visualizzare, degli *estremi delle scale* degli assi, del *testo* utilizzato, dei *colori*, ecc. Tutte queste scelte possono essere più o meno consapevoli e comunque evidenziano il fatto che la visualizzazione dei dati **non è mai un'attività completamente automatizzabile**, anche quando queste scelte sono gestite dal software. anche alcune automatizzazioni, infatti, comportano delle scelte implicite.

Questo stato dell'arte ha lo scopo di presentare, senza pretesa di esaustività, gli strumenti software più diffusi di rappresentazione grafica, con un focus sui programmi *freeware*¹¹, *open-source*, sviluppati per applicazioni web, con possibilità di aggiornamento dinamico delle variabili, utilizzabili anche per *smartphone*, e le *best practice* in questo campo.

In questo modo, si potranno valutare le migliori soluzioni tecnologiche disponibili per un eventuale utilizzo nello sviluppo delle piattaforme ICT in ambito Smart City.

7.1 Tipologie di Grafico

Non esiste una lista definitiva di tipologie di grafico per la visualizzazione dati perché emergono continuamente nuovi tipi di visualizzazione (come ad esempio alcuni istogrammi animati estremamente originali ed efficaci, diventati addirittura virali nei social media¹²). Una galleria di esempi di grafici, anche innovativi e originali, è fornita dal software D3.js¹³.

Una classificazione dei tipi di grafico per la visualizzazione dati diventato comune è quello del Financial Times: *il Financial Times Visual Vocabulary* [3] (creato per i giornalisti ma menzionato anche a livello europeo nell'EU Open Data Portal¹⁴).

Questo vocabolario suddivide i grafici in 9 tipologie, come mostrato nelle seguenti figure estrapolate dalla suddetta classificazione.

¹¹La differenza tra prodotti software a pagamento o gratuiti non è netta, essendovi spesso una numerosa gamma di opzioni e pacchetti di vendita per diversi tipi di usi. Il focus sarà tenuto per software gratuiti oppure con una versione gratuita non troppo limitante (almeno per le organizzazioni no-profit).

¹²Il video "Top 10 Richest People In The World (1995-2019)" mostra con degli istogrammi come evolvono nel tempo i patrimoni delle persone più ricche al mondo e ha ottenuto più di 7 milioni di visualizzazioni su YouTube. Con la stessa tecnica di visualizzazione, il video "Best Selling Music Artists from 1969 to 2019" ha raggiunto 6 milioni di visualizzazioni (rilevazione 5 dic 2019). Questi istogrammi animati sono stati creati con il linguaggio R e con i pacchetti: *ggplot2* e *gganimate*.

¹³<https://github.com/d3/d3/wiki/Gallery>.

¹⁴European Open Data Portal: <http://data.europa.eu/euodp/en/node/8011>

1. **Deviazione:** variazioni rispetto a dei valori di riferimento (p.es. un target o una media);
2. **Correlazione:** relazione tra due o più variabili.

DEVIAZIONE

Example FT uses
Trade surplus/deficit, climate change

Diverging bar



A simple standard bar chart that can handle both negative and positive magnitude values.

Diverging stacked bar



Perfect for presenting survey results which involve sentiment (eg disagree/neutral/agree).

Spine



Splits a single value into two contrasting components (eg male/female).

Surplus/deficit filled line



The shaded area of these charts allows a balance to be shown – either against a baseline or between two series.

CORRELAZIONE

Example FT uses
Inflation and unemployment, income and life expectancy

Scatterplot



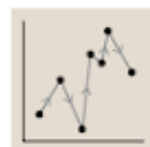
The standard way to show the relationship between two continuous variables, each of which has its own axis.

Column + line timeline



A good way of showing the relationship between an amount (columns) and a rate (line).

Connected scatterplot



Usually used to show how the relationship between 2 variables has changed over time.

Bubble



Like a scatterplot, but adds additional detail by sizing the circles according to a third variable.

XY heatmap



A good way of showing the patterns between 2 categories of data, less effective at showing fine differences in amounts.

Figura 10. Grafici di tipo “deviazione” e “correlazione”

3. **Ranking:** posizione e importanza di una voce rispetto alle altre.
4. **Distribuzione:** frequenza e valore di alcuni variabili.

RANKING

Example FT uses

Wealth, deprivation, league tables, constituency election results

Ordered bar



Standard bar charts display the ranks of values much more easily when sorted into order.

Ordered column



See above.

Ordered proportional symbol



Use when there are big variations between values and/or seeing fine differences between data is not so important.

Dot strip plot



Dots placed in order on a strip are a space-efficient method of laying out ranks across multiple categories.

Slope



Perfect for showing how ranks have changed over time or vary between categories.

Lollipop



Lollipops draw more attention to the data value than standard bar/column and can also show rank and value effectively.

Bump



Effective for showing changing rankings across multiple dates. For large datasets, consider grouping lines using colour.

DISTRIBUZIONE

Example FT uses

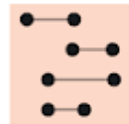
Income distribution, population (age/sex) distribution, revealing inequality

Histogram



The standard way to show a statistical distribution - keep the gaps between columns small to highlight the 'shape' of the data.

Dot plot



A simple way of showing the change or range (min/max) of data across multiple categories.

Dot strip plot



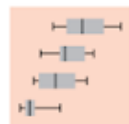
Good for showing individual values in a distribution, can be a problem when too many dots have the same value.

Barcode plot



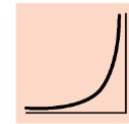
Like dot strip plots, good for displaying all the data in a table, they work best when highlighting individual values.

Boxplot



Summarise multiple distributions by showing the median (centre) and range of the data

Cumulative curve



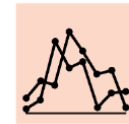
A good way of showing how unequal a distribution is: y axis is always cumulative frequency, x axis is always a measure.

Violin plot



Similar to a box plot but more effective with complex distributions (data that cannot be summarised with simple average).

Frequency polygons



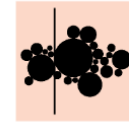
For displaying multiple distributions of data. Like a regular line chart, best limited to a maximum of 3 or 4 datasets.

Population pyramid



A standard way for showing the age and sex breakdown of a population distribution; effectively, back to back histograms.

Beeswarm



Use to emphasise individual points in a distribution. Points can be sized to an additional variable. Best with medium-sized datasets

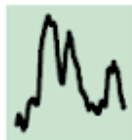
Figura 11. Grafici di tipo "ranking" e "distribuzione"

5. **Cambiamento nel tempo:** trend di cambiamento.

Example FT uses

Share price movements, economic time series, sectoral changes in a market

Line



The standard way to show a changing time series. If data are irregular, consider markers to represent data points.

Column



Columns work well for showing change over time - but usually best with only one series of data at a time.

Column + line timeline



A good way of showing the relationship over time between an amount (columns) and a rate (line).

Slope



Good for showing changing data as long as the data can be simplified into 2 or 3 points without missing a key part of story.

Area chart



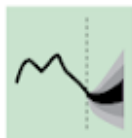
Use with care - these are good at showing changes to total, but seeing change in components can be very difficult.

Candlestick



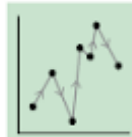
Usually focused on day-to-day activity, these charts show opening/closing and high/low points of each day.

Fan chart (projections)



Use to show the uncertainty in future projections - usually this grows the further forward to projection.

Connected scatterplot



A good way of showing changing data for two variables whenever there is a relatively clear pattern of progression.

Calendar heatmap



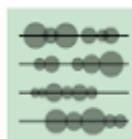
A great way of showing temporal patterns (daily, weekly, monthly) - at the expense of showing precision in quantity.

Priestley timeline



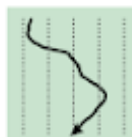
Great when date and duration are key elements of the story in the data.

Circle timeline



Good for showing discrete values of varying size across multiple categories (eg earthquakes by continent).

Vertical timeline



Presents time on the Y axis. Good for displaying detailed time series that work especially well when scrolling on mobile.

Seismogram



Another alternative to the circle timeline for showing series where there are big variations in the data.

Streamgraph



A type of area chart; use when seeing changes in proportions over time is more important than individual values

Figura 12. Grafici di tipo “cambiamento nel tempo”

6. **Grandezza:** confronti tra grandezze (relative o assolute).

Example FT uses
Commodity production, market capitalisation, volumes in general

Column



The standard way to compare the size of things. Must always start at 0 on the axis.

Bar



See above. Good when the data are not time series and labels have long category names.

Paired column



As per standard column but allows for multiple series. Can become tricky to read with more than 2 series.

Paired bar



See above.

Marimekko



A good way of showing the size and proportion of data at the same time – as long as the data are not too complicated.

Proportional symbol



Use when there are big variations between values and/or seeing fine differences between data is not so important.

Isotype (pictogram)



Excellent solution in some instances – use only with whole numbers (do not slice off an arm to represent a decimal).

Lollipop



Lollipop charts draw more attention to the data value than standard bar/column – does not have to start at zero (but preferable).

Radar



A space-efficient way of showing value of multiple variables – but make sure they are organised in a way that makes sense to reader.

Parallel coordinates



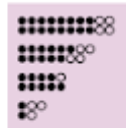
An alternative to radar charts – again, the arrangement of the variables is important. Usually benefits from highlighting values.

Bullet



Good for showing a measurement against the context of a target or performance range.

Grouped symbol



An alternative to bar/column charts when being able to count data or highlight individual elements is useful.

Figura 13. Grafici di “Grandezza”

tipo (Magnitude)

- 7. **Parte-del-tutto:** contributo degli elementi che costituiscono una determinata grandezza.
- 8. **Spaziale.**
- 9. **Flusso.** volumi e intensità di movimento tra due stati o condizioni (p.es. tra 2 sequenze logiche o posizioni geografiche)

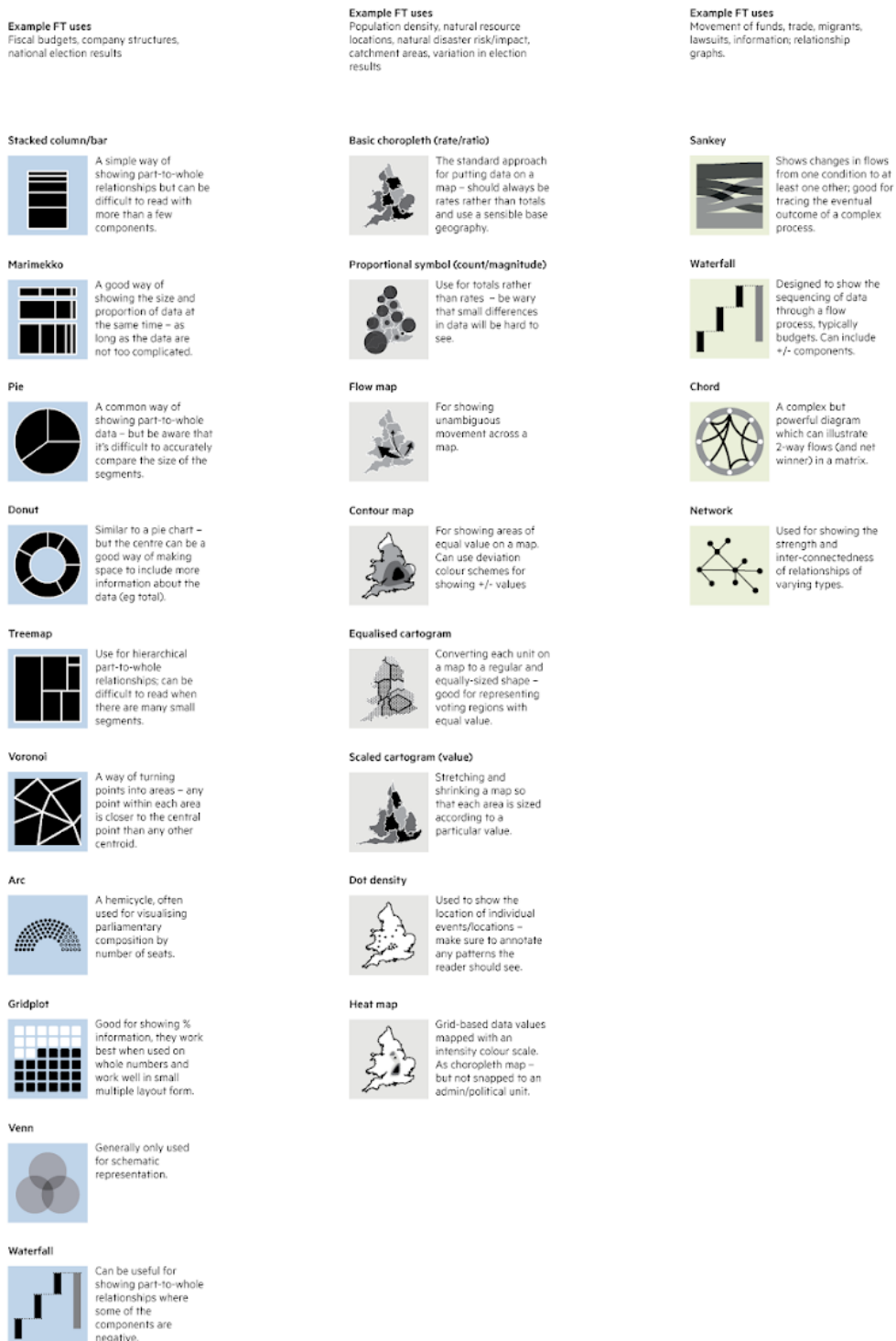


Figura 14. Grafici di tipo “parte-del-tutto” (a sinistra), “spaziali” (al centro), “flusso” (a destra).

Per ricercare quale software di visualizzazione grafica utilizzare, a partire dal tipo di grafico desiderato, è possibile consultare la mappa “ChartmakerDirectory”¹⁵, che mette in connessione circa 50 tipi di grafico e plot con circa 40 differenti software (la mappa mostra quindi quali grafici ogni software è in grado di supportare).

È possibile consultare delle linee guida per la formattazione dei grafici e per mostrare meglio “la storia che i dati vogliono raccontare”: la “Data Visualization Checklist”¹⁶ indica 24 linee guida per formattare in modo appropriato il testo, gli assi, le scale, i colori, i simboli per i dati, ecc.

Un aspetto interessante della rappresentazione grafica è quella dell’*engagement* e del *gaming*, ovvero di quelle tecniche che, rivolgendosi ai cittadini e quindi ad un pubblico più vasto, aumentano la capacità di attrazione, di “dialogo” e di coinvolgimento con chi fruisce del dato (p.es. grafici comunicati ai cittadini per incentivare l’uso della mobilità sostenibile).

7.2 Tipologie di Software

In questa sezione vengono descritti molto brevemente i principali software utilizzati come *visualisation tools*¹⁷, con alcune delle principali caratteristiche.

I software si differenziano, tra le altre cose, per:

- essere liberi o a pagamento¹⁸;
- il tipo di licenza;
- il fatto di essere o meno *opensource*;
- il numero e il tipo di grafici a disposizione;
- il numero e tipo di connettori verso file dati di input (p.es. *json*, *postgresql*, *csv*, ecc.);
- essere strutturati come *web-tool* dinamici aggiornabili in *real-time* oppure pensati come software di post-processamento;
- consentire un’installazione locale autonoma o per possedere dei servizi gestiti da remoto;
- il grado di autonomia del software verso aggiornamenti o variazioni esterne;
- la possibilità lasciata all’utente di cambiare e scegliere il tipo di grafico (oppure il tipo di grafico viene prefissato a priori);
- presenza o meno di *analytics*;
- essere orientati alla manipolazione di Big Data;
- il tipo di *device* a cui sono destinati (p.es. smartphone).

Vengono qui elencati alcuni software liberi (o che, anche se commerciali, hanno una versione con licenza libera). Come si è già detto, questo elenco non ha pretesa di esaustività, esistendo centinaia differenti di software di visualizzazione, alcuni dei quali in veloce evoluzione.

¹⁵ <https://chartmaker.visualisingdata.com/>

¹⁶ datavizchecklist.stephanieevergreen.com

¹⁷ In rete esistono centinaia di recensioni riguardo i visualization tools.

Si veda per esempio: [4], <http://selection.datavisualization.ch/> oppure <https://medium.com/sciforce/best-libraries-and-platforms-for-data-visualization-b986a43aee3f>

¹⁸ Vi sono poi tutte le opzioni intermedie, con versioni del software libere solo per uno o pochi utenti, libere solo per agenzie no-profit, libere solo per versioni del software più limitate.

7.2.1 Tableau

Tableau¹⁹ è un tool di visualizzazione dati tra i migliori esistenti [5], con una piattaforma professionale usata da molte imprese e industrie, molto sviluppato e specializzato in *business intelligence*, user-friendly nell'interfaccia interattiva, appropriato per *machine learning* e *Big Data* (usato tra gli altri anche dall'ISTAT²⁰). Permette di entrare nella descrizione dei dati in modo estremamente approfondito, consente l'uso di *analytics* e di soluzioni server senza che sia richiesta l'installazione in loco. In realtà il software Tableau è libero solo nella sua versione base *Tableau public*, ma viene qui descritto, essendo nella sua versione a pagamento uno dei più diffusi e potenti strumenti di visualizzazione, con un'ampia galleria²¹ di grafici. Si integra bene con MySQL, Hadoop, Amazon AWS, SAP e Teradata.

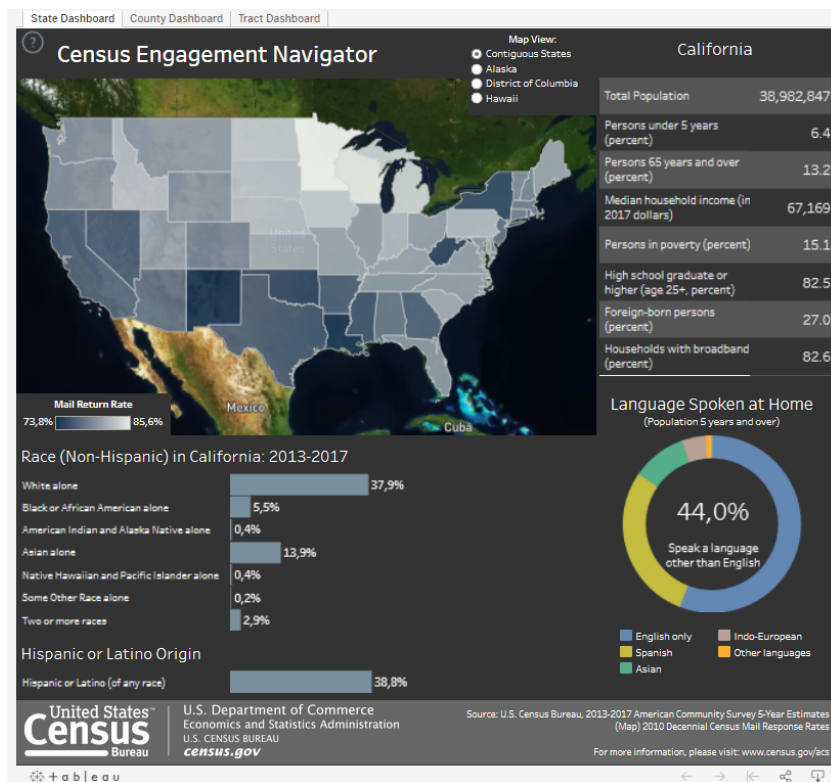


Figura 15. Esempio di dashboard con Tableau Public

¹⁹ <https://public.tableau.com/en-us/s/>

²⁰ https://public.tableau.com/views/noitalia2019Confrontoeuropeo/Confrontoeuropeo-sintesi?:embed=y&:display_count=yes?:showVizHome=no#2

²¹ <https://public.tableau.com/it-it/gallery/?tab=viz-of-the-day&type=viz-of-the-day>

7.2.2 D3.js

D3²² è l'abbreviazione di Data-Driven Document e combina componenti di visualizzazione molto potenti con i metodi di manipolazione dei *data-driven* DOM (Document Object Model).

D3 è una libreria JavaScript attenta agli standard del web e consente di utilizzare tutte le capacità dei browsers moderni senza doversi legare ad un framework di proprietà. Usa Javascript, HTML, SVG e CSS. D3 è uno strumento molto tecnico e mirato agli sviluppatori. Possiede una vastissima galleria con diverse centinaia di grafici, mappe, diagrammi interattivi e innovativi.

Questi sono forniti con esempi, già pronti per essere utilizzati^{23,24}.

Unemployment rate by county, August 2016. Data: Bureau of Labor Statistics

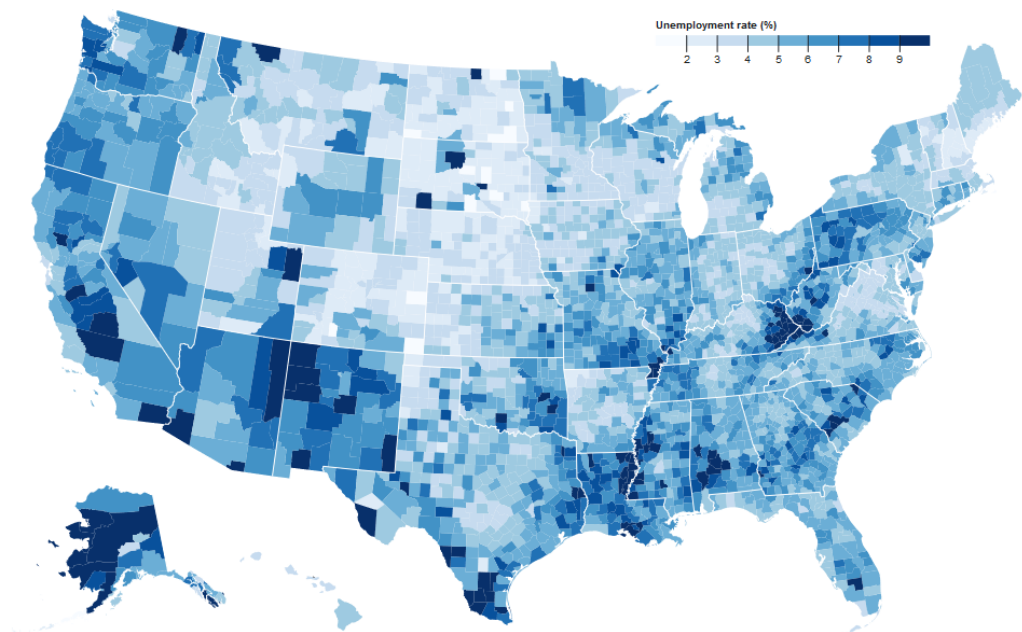


Figura 16. D3.js - Esempio

²²<https://d3js.org/>

²³ Alcune gallerie si possono trovare in: <https://github.com/d3/d3/wiki/Gallery>,

<http://techslides.com/over-1000-d3-js-examples-and-demos>

<https://www.d3-graph-gallery.com/index.html>

²⁴<https://observablehq.com/@d3/choropleth>

7.2.3 Highcharts

Highchart²⁵ è una libreria software molto potente, scritta in JavaScript puro e rilasciata dal 2009 in Norvegia, per siti web, anche per smartphone. È stata sviluppata per grafici, mappe e visualizzazioni applicate anche nel campo della finanza. È un prodotto commerciale ma che è gratuito per organizzazioni non commerciali, con licenza Attribution-NonCommercial 3.0 Unported (CC BY-NC 3.0). Una galleria di grafici²⁶, circa una quarantina, già in formato codice e pronti per l'uso, è fornita sul sito web ufficiale (un esempio²⁷ in figura).

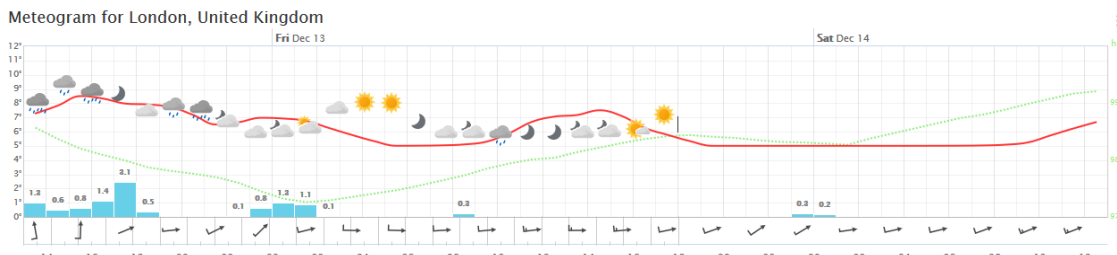


Figura 17. Highcharts – Esempio

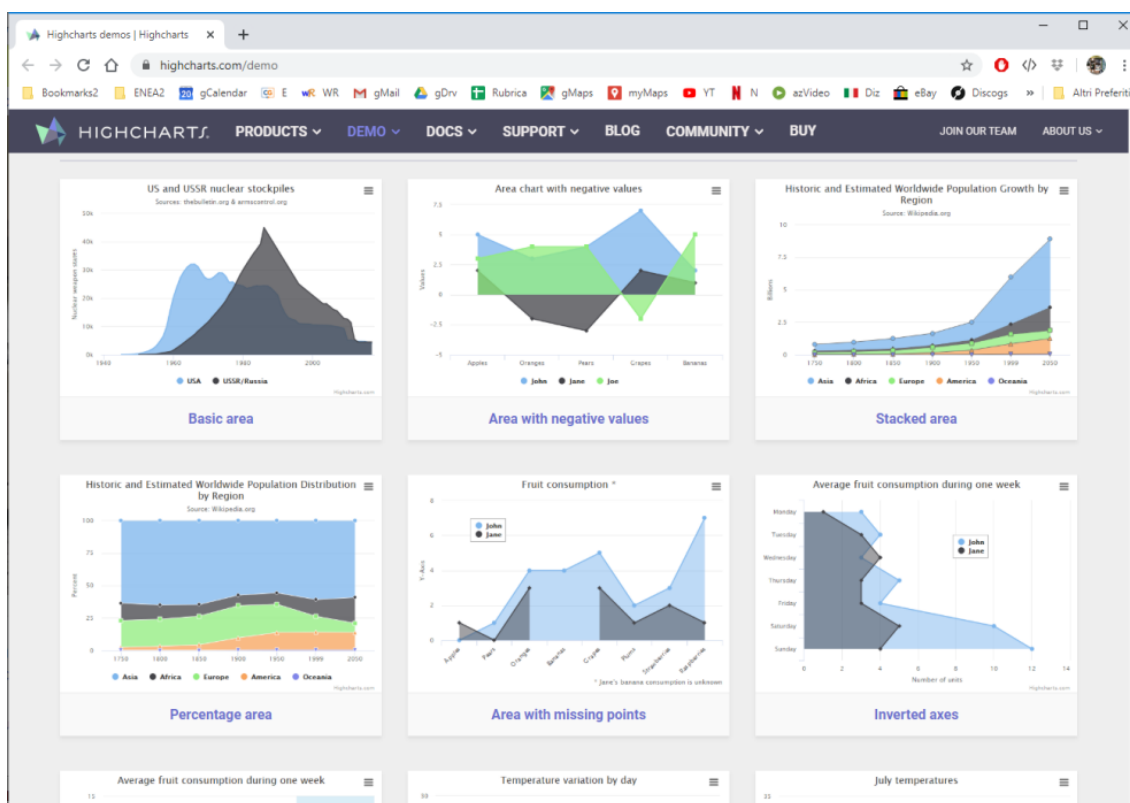


Figura 18. Highcharts web site

²⁵ www.highcharts.com

²⁶ <https://www.highcharts.com/demo>

²⁷ https://www.highcharts.com/demo/combo-meteorogram#https://www.yr.no/place/United_Kingdom/England/London/forecast_hour_by_hour.xml

7.2.4 Google Charts

Google Charts²⁸ è un software con licenza libera, basato sulla tecnologia pura HTML5/SVG (non sono richiesti plugin). Usa un semplice Javascript integrato nella pagina web e possiede una larga compatibilità con diversi browser. Funziona su Android, iOS o altre piattaforme.

È un tool molto user-friendly e comprende una galleria di circa 30 tipi di visualizzazioni e grafici²⁹. È appropriato per l'uso di grafici interattivi, per dati dinamici e per l'uso di dashboards. Inoltre esiste già una numerosa serie di connettori, tool e protocolli per utilizzare **Google Charts** con diversi formati di dati. Un esempio³⁰ di grafico viene mostrato nella seguente figura.

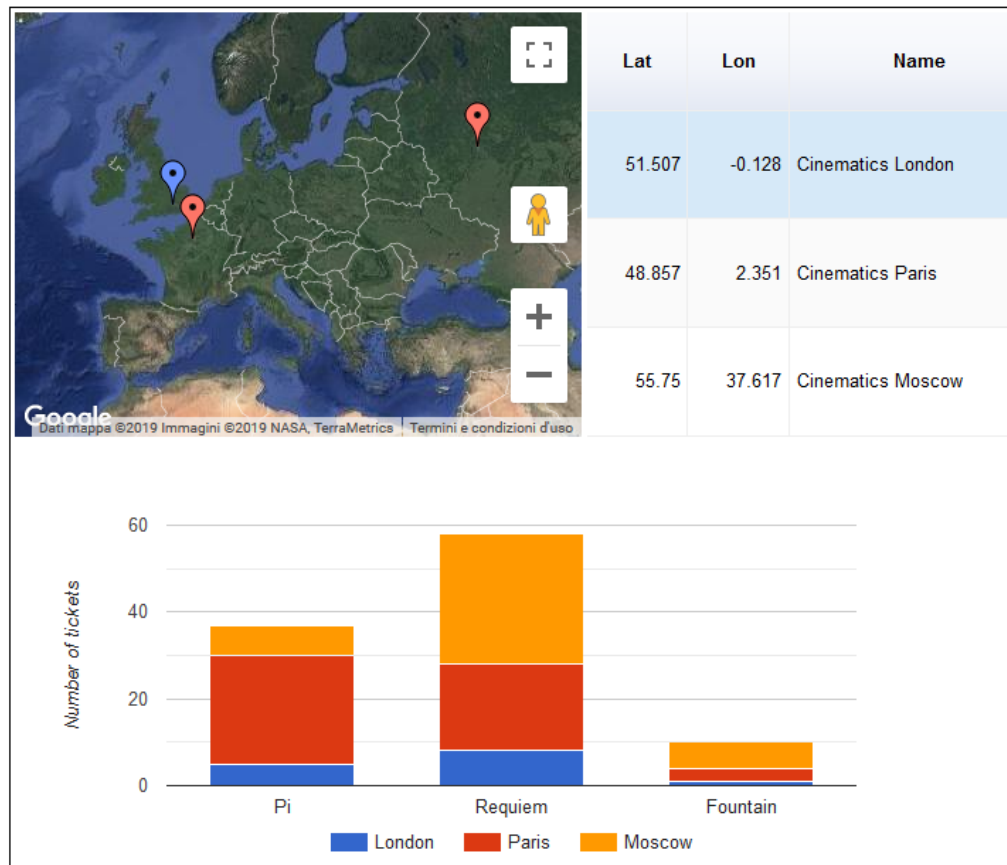


Figura 19. Google Charts - Statistica relativa a una compagnia di cinema

²⁸ <https://developers.google.com/chart>

²⁹ <https://developers.google.com/chart/interactive/docs/gallery>

³⁰ <https://developers.google.com/chart/interactive/docs/examples?hl=it>

7.2.5 Chart.js

Chart.js³¹ è una libreria Javascript open-source, libera e disponibile attraverso una licenza MIT. È mirata agli sviluppatori, ma è utilizzata anche da designers.

Creata nel 2013, è una libreria mantenuta dalla community stessa ed è considerata di semplice applicazione, molto più semplice da usare ad esempio rispetto a *D3.js*. D'altra parte è meno adatta alla customizzazione.

Chart.js usa la potenzialità di HTML5 Canvas³².

Una galleria³³ di tipologie di grafici si può trovare sul sito ufficiale.

Un esempio di dashboard³⁴ viene mostrato nella seguente figura.



Figura 20. Chart.js - Esempio di dashboard

³¹<https://www.chartjs.org/>

³²https://www.w3schools.com/html/html5_canvas.asp

³³<https://www.chartjs.org/samples/latest/>

³⁴<https://devpost.com/software/my-personal-dashboard>

7.2.6 Matplotlib

Matplotlib³⁵ è la libreria storica di Python, adatta anche per applicazioni web, e contiene centinaia di grafici interattivi e animazioni, già pronti. Una galleria³⁶ ricchissima di esempi si trova sul sito ufficiale. Alcuni grafici vengono mostrati nella seguente figura³⁷.

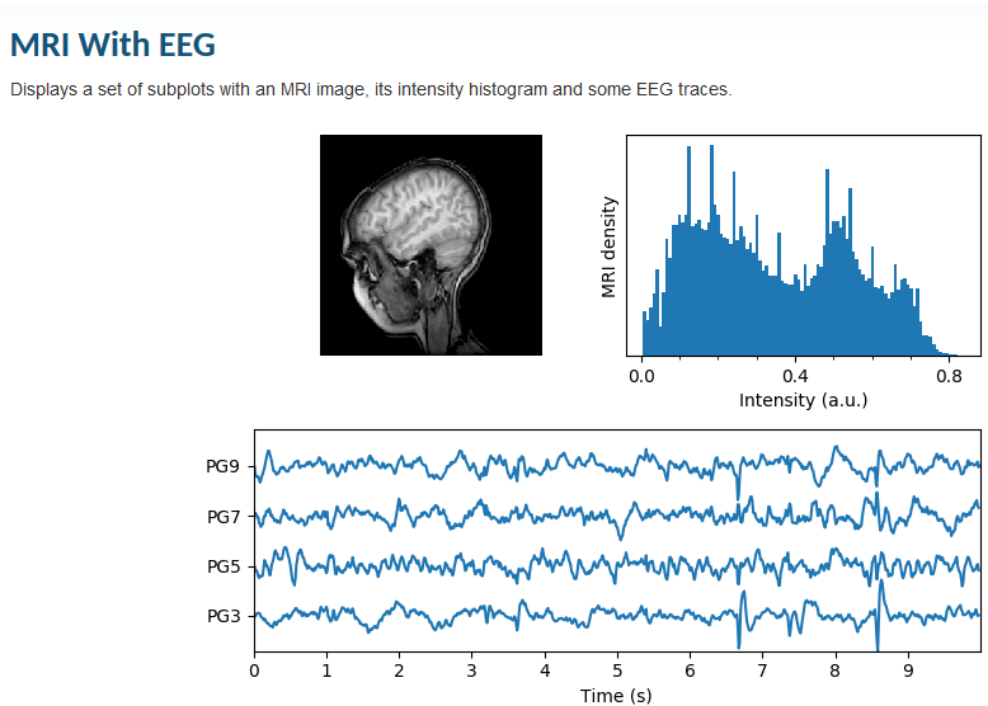


Figura 21. Matplotlib - Esempi

³⁵<https://matplotlib.org/>

³⁶<https://matplotlib.org/gallery/index.html>

³⁷https://matplotlib.org/gallery/specialty_plots/mri_with_eeg.html#sphx-glr-gallery-specialty-plots-mri-with-eeg-py

7.2.7 Dash-Plotly

Dash³⁸ è un framework Python per applicazioni web.

È costruito intorno a Flask, Plotly.js e React.js.

È mirato a costruire apps di visualizzazione dati, specialmente con grafici interattivi, interfacce flessibili e “customizzabili”. Risulta particolarmente adatto per chi elabora dati con Python.

È un’applicazione molto semplice da usare, incorporando già le tecnologie e i protocolli richiesti per costruire un’applicazione web interattiva. Dash è una libreria open source, rilasciata sotto licenza MIT. Include la progettazione di dashboard, di cui riportiamo un esempio³⁹ in figura.

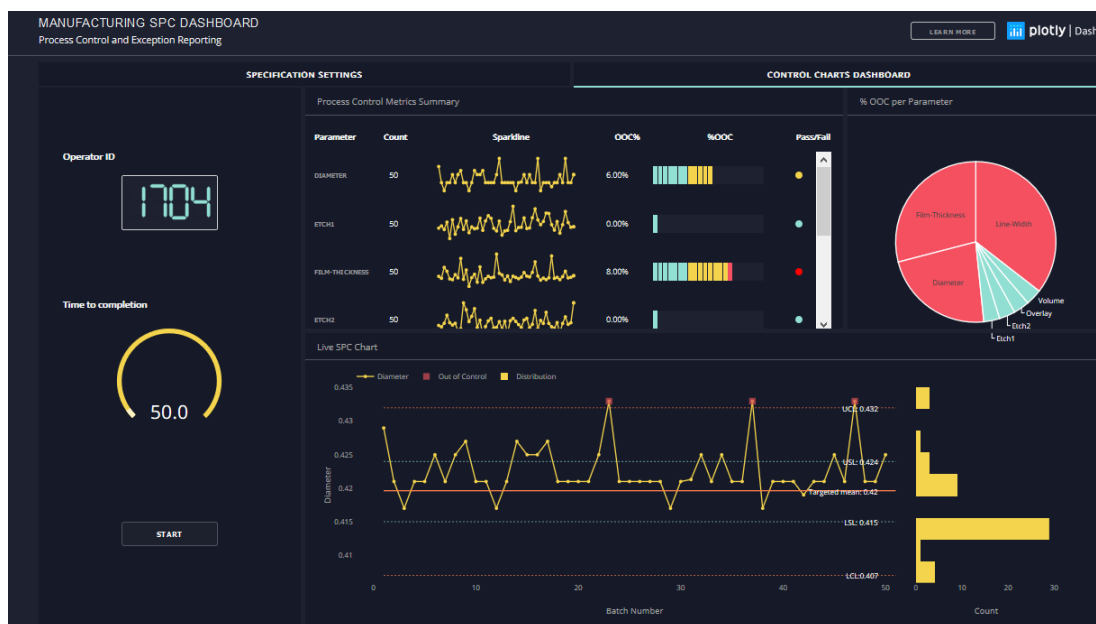


Figura 22. Dash - Esempio di Dashboard

³⁸ https://dash.plot.ly/?_ga=2.265487942.852944674.1573218018-320457490.1573218018

³⁹ <https://dash-gallery.plotly.host/dash-manufacture-spc-dashboard/>

7.2.8 Grafana

Grafana⁴⁰ è un software libero, open source, creato come “monitoring tool” per infrastrutture internet e analisi software, ma che si è diffuso anche in altri settori, come il monitoraggio di sensori industriali, la domotica, le applicazioni meteo ed il controllo di processi. Grafana si interfaccia con Graphite, Elasticsearch, postgresQL, Cloudwatch, Prometheus, InfluxDB e altro.

È un software di uso immediato e semplice, soprattutto per la costruzione di dashboard, e può sfruttare ricerche dati attraverso il linguaggio SQL. D'altra parte fornisce un numero limitato di grafici e di opzioni di customizzazione. Un esempio⁴¹ di applicazione è fornito in figura.



Figura 23. Grafana - Esempio di dashboard

⁴⁰<https://grafana.com/>

⁴¹<https://grafana.com/grafana/dashboards/11074>

7.2.9 JfreeChart

JfreeChart⁴² è una libreria Java, libera e open source, per la visualizzazione di dati e grafici.

I grafici sono interattivi e possono essere modificati dall'utente.

Ha il vantaggio di essere altamente flessibile, adattabile e modificabile, in quanto utilizza la potenza del codice Java. D'altra parte non presenta una gamma particolarmente ricca di esempi di grafici ed è un progetto che non sembra essere più sviluppato da una forte comunità.

Degli esempi di grafici⁴³ sono mostrati nella seguente figura.

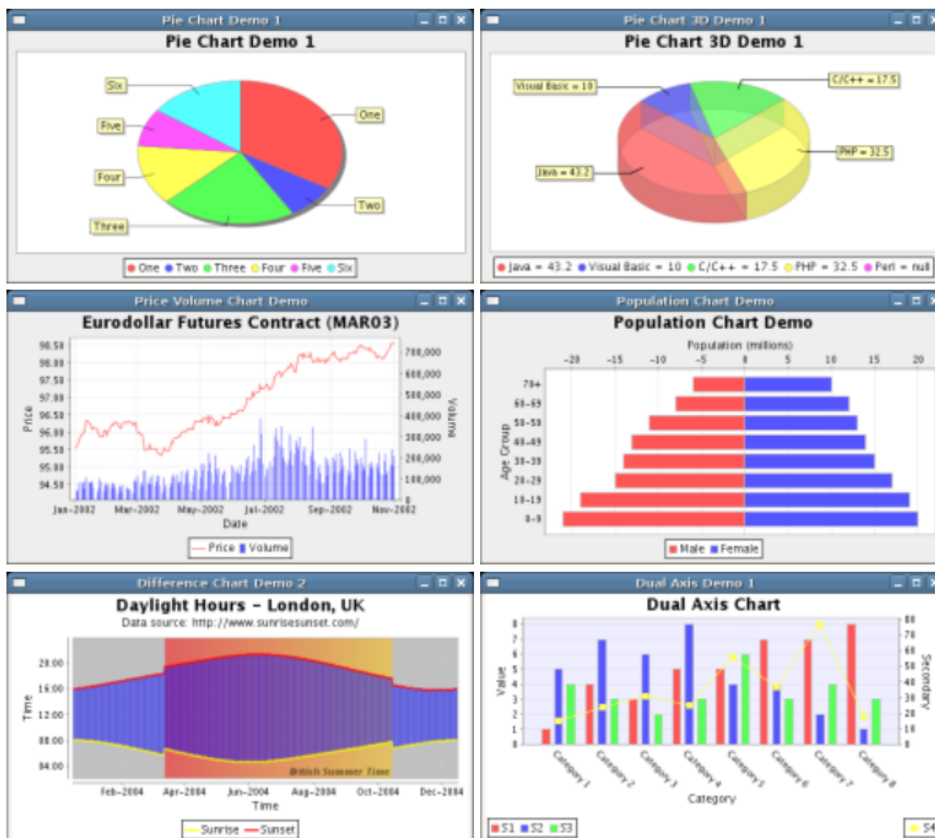


Figura 24. JfreeChart - Esempi

⁴² <http://www.jfree.org/jfreechart/>

⁴³ <http://www.jfree.org/jfreechart/samples.html>

7.2.10 Snap4city

Snap4city⁴⁴ (scalable Smart aNalytic APplication builder for sentient Cities) è una piattaforma già pensata per essere applicata alle Smart City [6].

È una piattaforma sviluppata da DISIT (Distributed Systems and Internet Technologies Lab) dell'Università di Firenze, open source, scalabile, su cloud, sicura e flessibile, realizzata con micro-servizi e strumenti sostituibili. Analizza e aggrega una grande quantità di dati e, attraverso l'uso dell'Intelligenza Artificiale, offre servizi, come la rivelazione di anomalie, warning a valutazioni di rischio. Snap4city è già stata applicata in diverse città (Firenze, Helsinki, Antwerp, Santiago di Compostela, Pisa, Prato, Cagliari) e regioni (Toscana, Emilia-Romagna). Permette di costruire dashboard personalizzate. Un esempio di dashboard⁴⁵ è mostrato nella seguente figura.

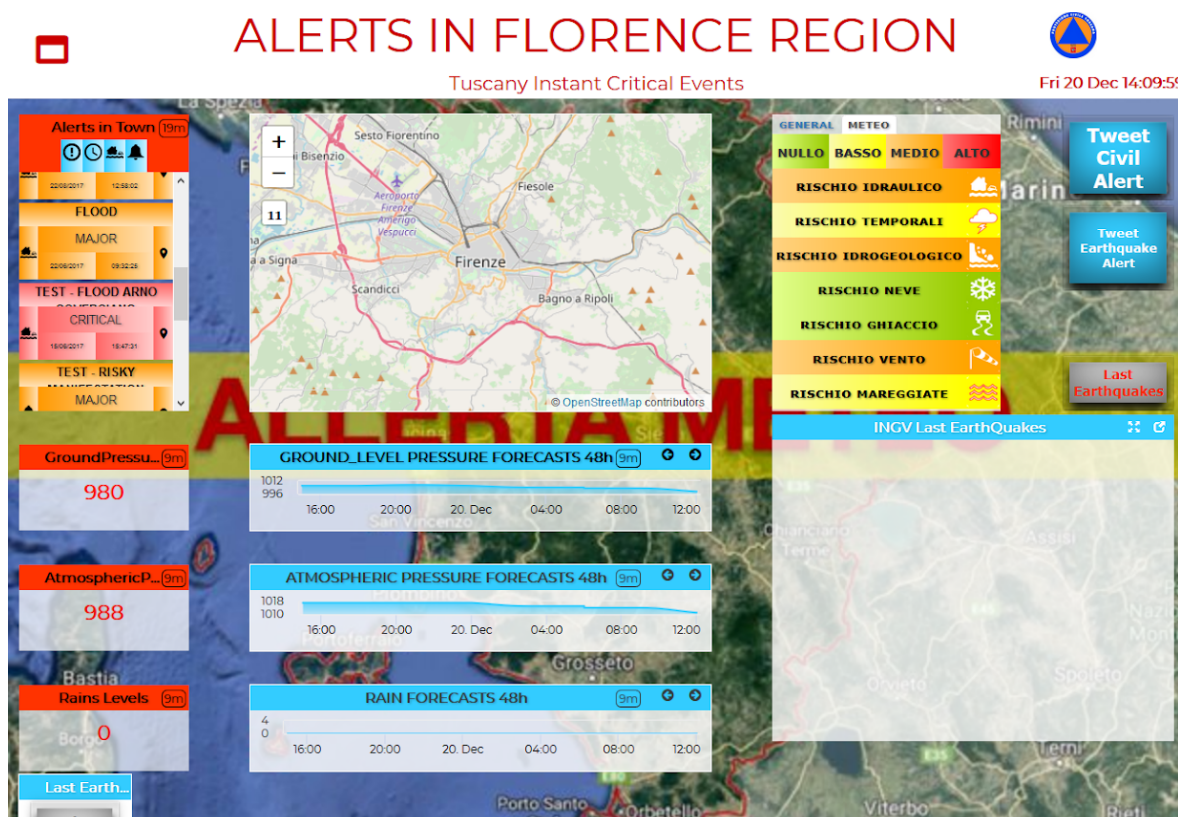


Figura 25. Snap4city – Dashboard con monitor di eventi per allerte a Firenze

⁴⁴<https://www.snap4city.org>

⁴⁵<https://www.snap4city.org/dashboardSmartCity/view/index.php?iddashboard=MTUzMW==>

7.2.11 KIBANA

Kibana è un software open source sviluppato per la visualizzazione, tramite dashboard dinamiche, di dati indicizzati in Elasticsearch (un motore di ricerca con capacità Full Text). Kibana è in grado di gestire e monitorare una gran mole di dati e di creare diversi tipi di tabelle e di grafici (grafici a dispersione, grafici a torta, istogrammi, mappe, serie temporali, ecc.). Permette anche analisi avanzate di dati, visualizzazioni interattive, rappresentazioni con grafi e con “nuvole di etichette”. Fa ora anche parte del prodotto commerciale Elastic Stack⁴⁶, combinazione di Elasticsearch, Logstash e Kibana. Un esempio di dashboard è mostrato nella seguente figura⁴⁷.

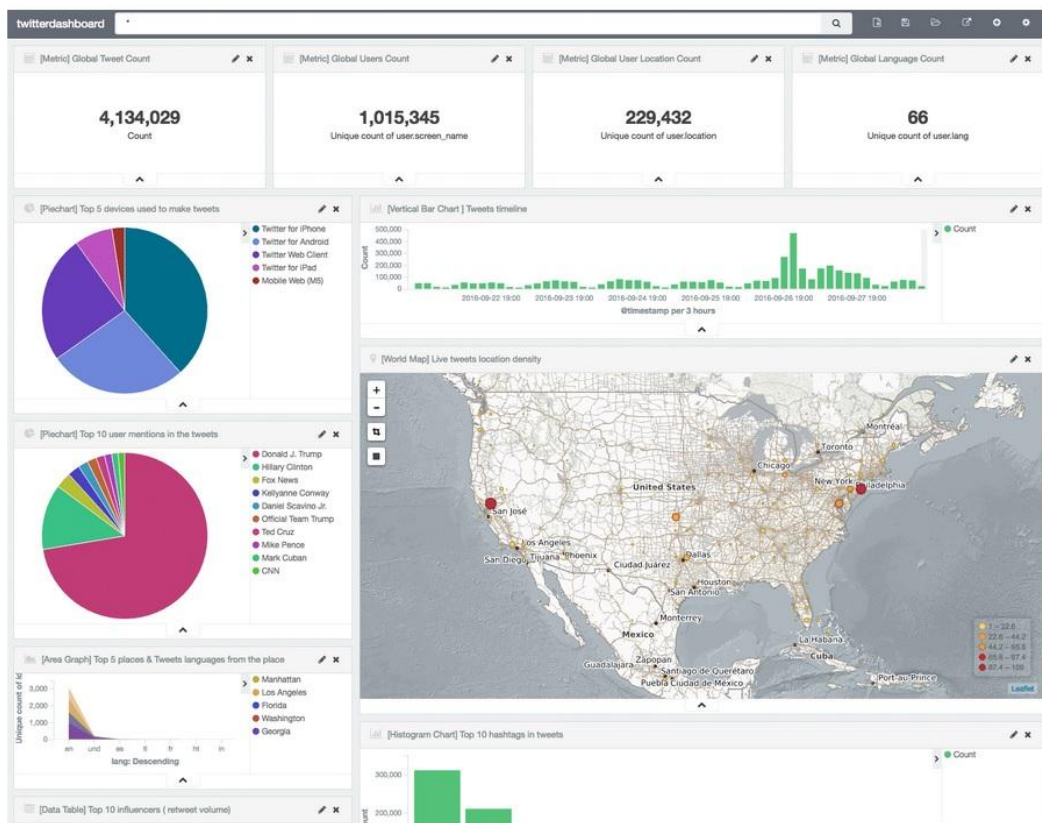


Figura 26. Esempio di dashboard creato con Kibana

⁴⁶ <https://www.elastic.co/kibana>

⁴⁷ <https://www.objectrocket.com/blog/data-connectors/streaming-twitter-to-elasticsearch-and-kibana/>

Esistono centinaia di altri software per la visualizzazione di dati, che per questione di spazio non sono stati qui descritti ma che vogliamo riportare:

- **Echarts** (strumento Javascript),
- **Datawrapper**,
- **Imply Pivot**,
- **Gapminder**,
- **Gephi** (per visualizzazione di reti),
- **ChartBlocks** (crea grafici on line),
- **Chartbuilder**,
- **NetData** (per monitoring attraverso dashboards),
- **NVD3** (sfrutta la potenza di D3.js),
- **ManyEyes** (della IBM),
- **Graphite** (per monitoring di sistema),
- **RawGraph**,
- **Vega** (legge anche file JSON) e **Vegas** (librerie Scala),
- **R Shiny** (per visualizzazione statistica interattiva con R),
- **Google Data Studio** (anche per dashboard dinamiche),
- **Infogram** (sviluppato anche per dashboard e mappe),
- **TimeLineJS**(per sequenze temporali),
- **Breeze-viz**(Libreria Scala).

Altri software degni di nota:

- a pagamento: **FineReport**, **Fusion Charts**, **PowerBi**, **QlikSense**, **QlikView**, **Tapclicks**, **Chartio**, **ClicData**, **Visme**, **Klipfolio** (include una sezione sulle dashboard), **Looker**, **Splunk**, **Sisense**, **Zoho Analytics** (Business Intelligence e data analytics), **Gnuplot**;
- specializzati in creazione e uso di mappe: **Leaflet**, **ArcGIS**, **CARTODB**, **Openheatmap**;
- librerie PYTHON interessanti: **seaborn**, **ggplot**, **plotnine**, **Bokeh**, **Pandas**;
- altre librerie di R: **Ggplot2**, **Base graphics**, **Gridgraphics**, **Lattice**, **ggvis**, **Rgl** (possiede anche plot 3D interattivi);
- altre librerie e API Java sono: **Java2D**, **Java3D**, **Prefuse**, **Java advancedimaging**, **VisAD**.

7.3 Esempi di Siti Web

Qui di seguito riportiamo alcuni esempi di siti web, particolarmente sviluppati, che descrivono alcuni fenomeni usando diversi KPI, anche attraverso *dashboard* con valori dinamici.

7.3.1 OECD

Il sito dell'OECD (Organizzazione per la cooperazione e lo sviluppo economico) tenta di misurare il benessere in differenti regioni attraverso 10 indicatori, mostrati contemporaneamente con diversi colori, con punteggi che vanno da 0 a 10 e anche con il confronto con città simili (si veda nella seguente figura⁴⁸).

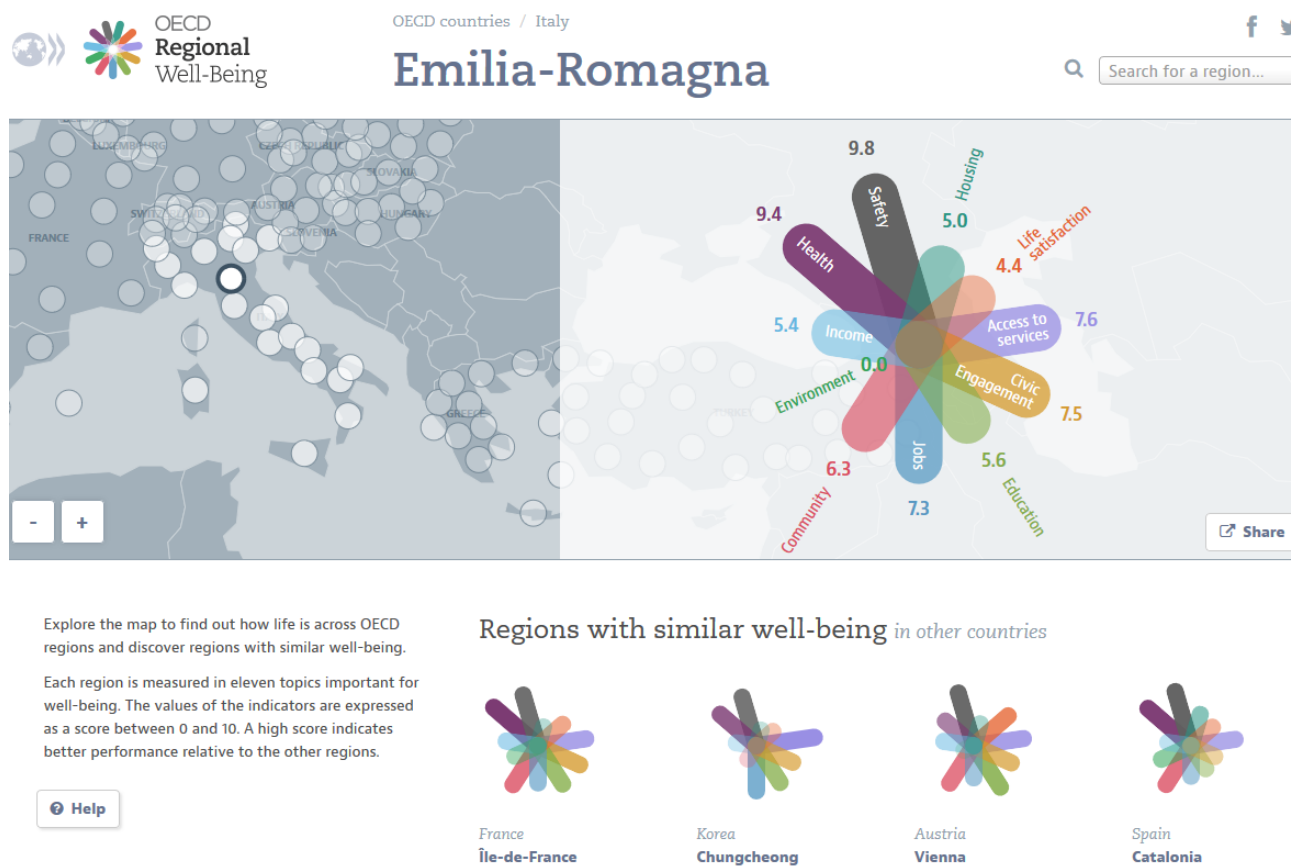


Figura 26. OECD - Benessere in Emilia-Romagna e confronto con altre città

⁴⁸ <https://www.oecdregionalwellbeing.org/ITH5.html>

7.3.2 EIGE

Il sito dell'EIGE⁴⁹(European Institute for Gender Equality) mostra il Gender Equality attraverso l'analisi di sette indicatori e contemporaneamente confronta l'Italia con la performance di altri paesi (si veda l'esempio⁵⁰ nella seguente figura).

Gender Equality Index

[View countries](#) [Compare countries](#) [Thematic Focus](#) [About Index](#) [Publications](#) [Conference 2019](#)

Index score for  Italy ▼ for 2019 ▼

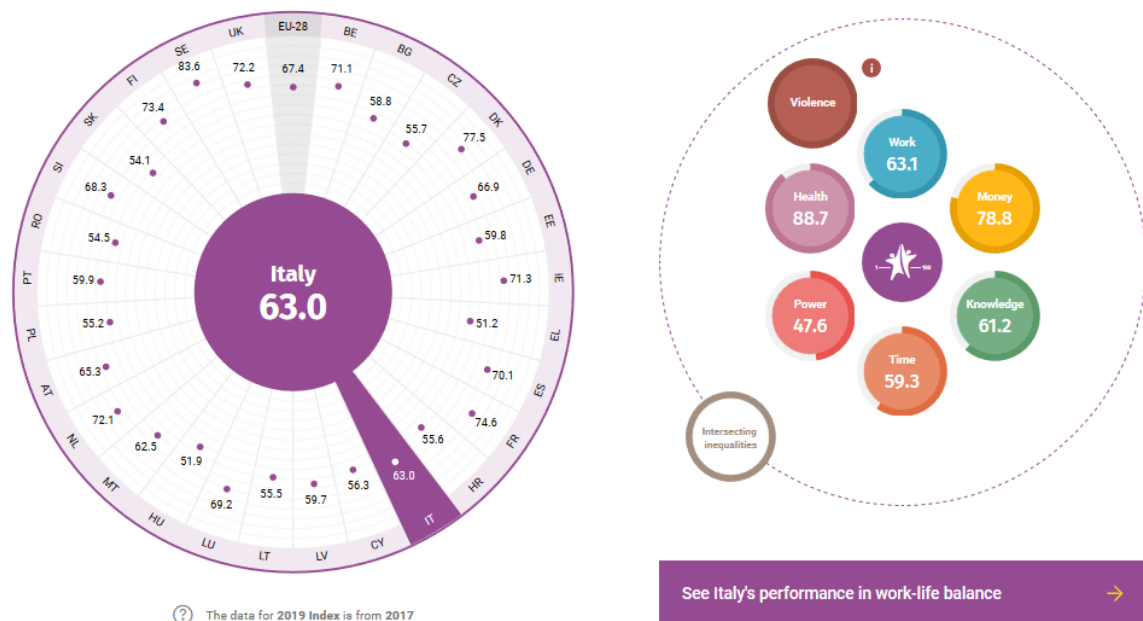


Figura 27. EIGE- Gender Equality Index

⁴⁹<https://eige.europa.eu/>

⁵⁰<https://eige.europa.eu/gender-equality-index/2019/IT>

7.3.3 Flightradar24

Alcuni siti web hanno l'obiettivo di monitorare il traffico in tempo reale.

Si veda come esempio il monitoring del traffico aereo del sito di Flightradar24 nella seguente figura⁵¹, che sfrutta la connettività della rete.



Figura 28. Flightradar24 - Monitoring del traffico aereo

⁵¹<https://www.flightradar24.com/44.09.11.13/7>

7.3.4 Singapore

Alcuni siti di Open Data riguardanti le città incorporano già la visualizzazione dei dati, come per esempio quello di Singapore⁵², che mostra cinque grafici chiave per descrivere e sintetizzare le performances di diversi settori (economia, istruzione, ambiente, ecc.).

Si veda l'esempio nella seguente figura, che descrive il settore ambiente.

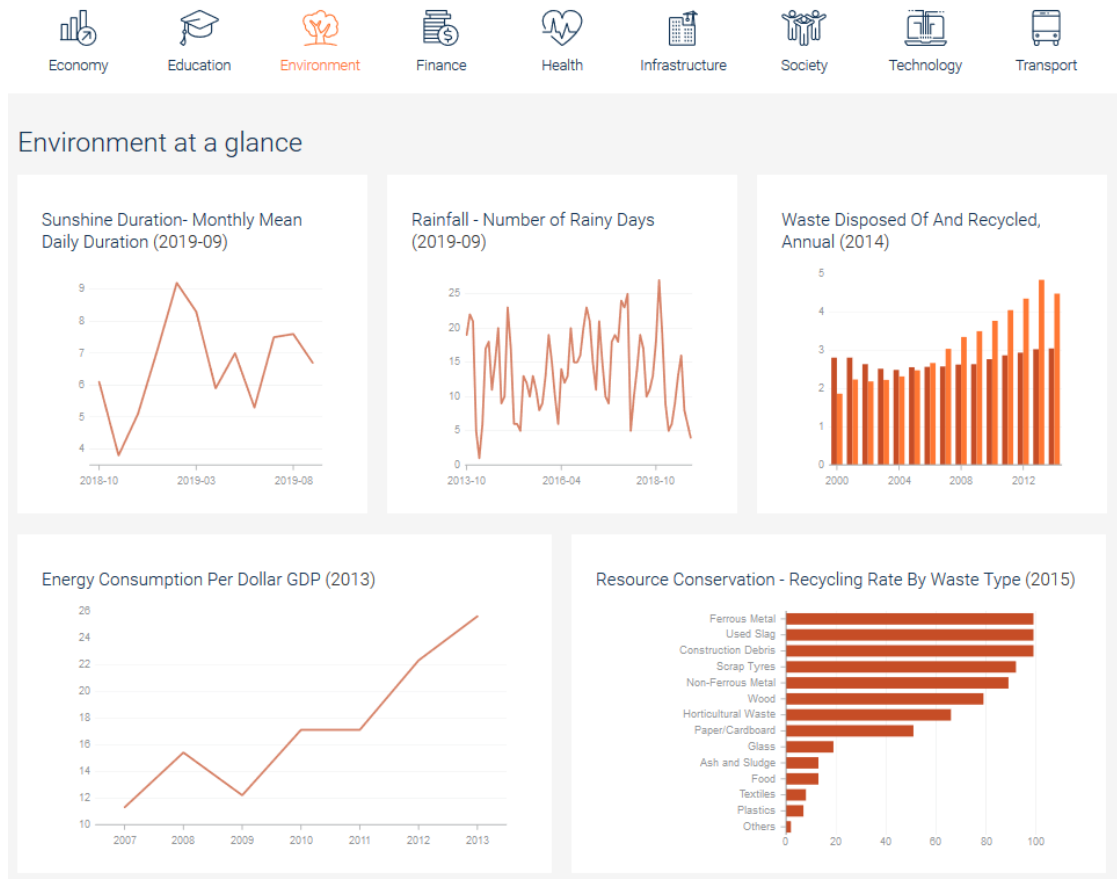


Figura 29. Singapore - Open Data web site

⁵² <https://data.gov.sg/>

7.3.5 EUROSTAT

Un sito estremamente ricco per quanto riguarda la visualizzazione dati con grafici interattivi è sicuramente quello di Eurostat. Il sito raccoglie una gran mole di dati dei paesi dell'Unione Europea, con sezioni interamente dedicate alla statistica illustrata⁵³ e alla visualizzazione di dati⁵⁴.

Eurostat inoltre ha promosso diversi seminari e webinar⁵⁵ sulla visualizzazione di dati, in particolare evidenziando l'uso dei tools D3.js e QlikSense⁵⁶.

Un esempio di visualizzazione dati applicato alle città europee⁵⁷ viene mostrato in figura.

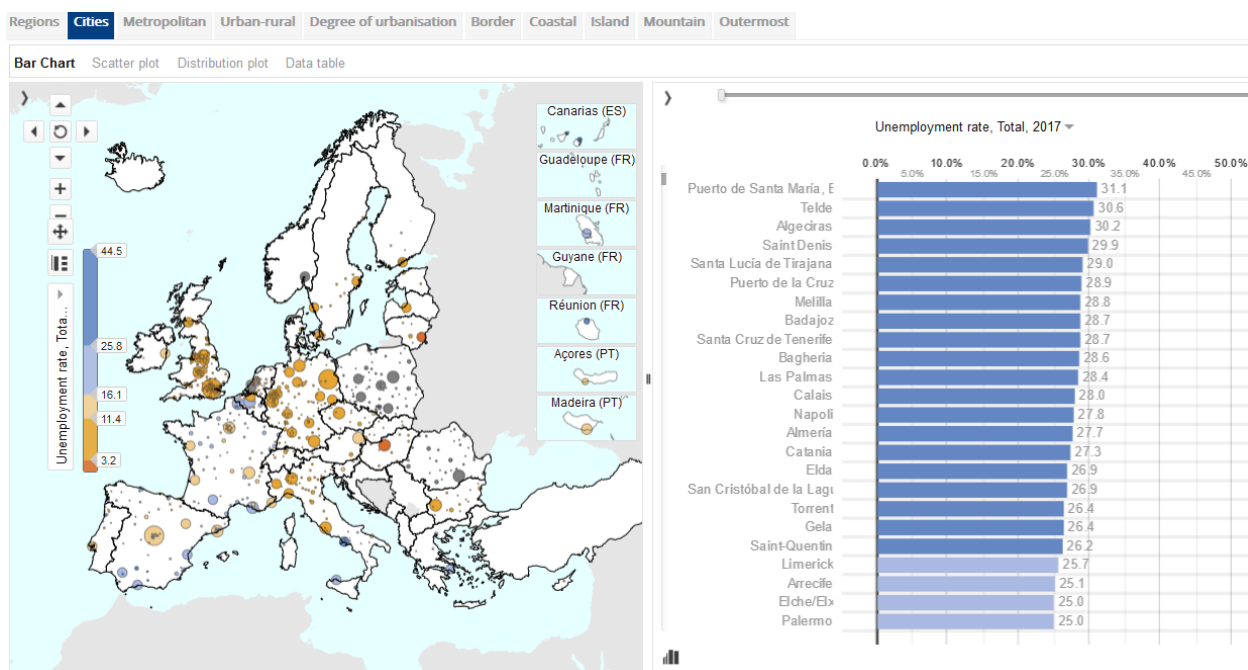


Figura 30. Eurostat- Tasso di disoccupazione nelle città europee

⁵³ <https://ec.europa.eu/eurostat/web/energy/statistics-illustrated>

⁵⁴ <https://ec.europa.eu/eurostat/help/first-visit/tools>

⁵⁵ <https://data.europa.eu/euodp/en/visualisation-home>

⁵⁶ <http://data.europa.eu/euodp/en/node/8011>

⁵⁷ <https://ec.europa.eu/eurostat/cache/RCI/#?vis=city.statistics&lang=en>

7.3.6 ISTAT

Anche il sito dell'ISTAT⁵⁸ è ricchissimo per quanto riguarda la visualizzazione dati per mezzo di grafici interattivi, mostrando mappe dinamiche, serie storiche, confronti con altri paesi europei e descrivendo diversi fenomeni (popolazione, salute, economia) con una pluralità di infografiche.

ISTAT usa diversi strumenti:

- Librerie di JavaScript (Highcharts, D3),
- Piattaforme di Business Intelligence/Visual Analytics (MicroStrategy, Tableau)⁵⁹ e
- software di mappe (OpenStreetMap).

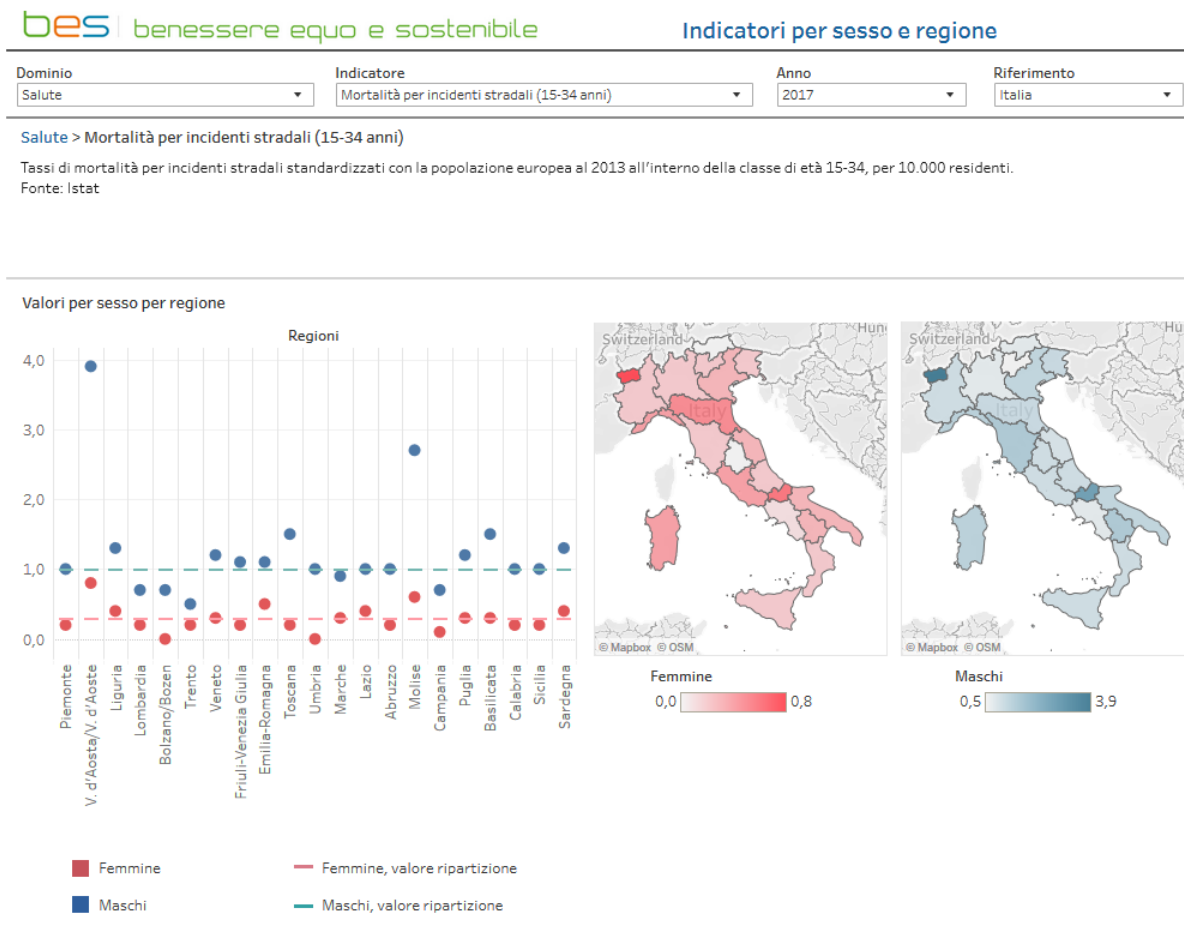


Figura 31. ISTAT - Mortalità di incidenti stradali 15-34 anni

⁵⁸ <https://www.istat.it/it/dati-analisi-e-prodotti/visualizzazioni>

⁵⁹ <https://www4.istat.it/it/files/2015/05/Annunziata-Fiore.pdf>

7.3.7 OPEN DATA Campania

Un sito italiano interessante sugli Open Data è quello della regione Campania⁶⁰, che permette all'utente di visualizzare tutti i dati disponibili (generalmente lasciati come dati grezzi .csv, .json, ecc.) lasciando la scelta delle variabili da rappresentate, le etichette e il tipo di grafico.

Un esempio⁶¹ viene mostrato nella seguente figura.

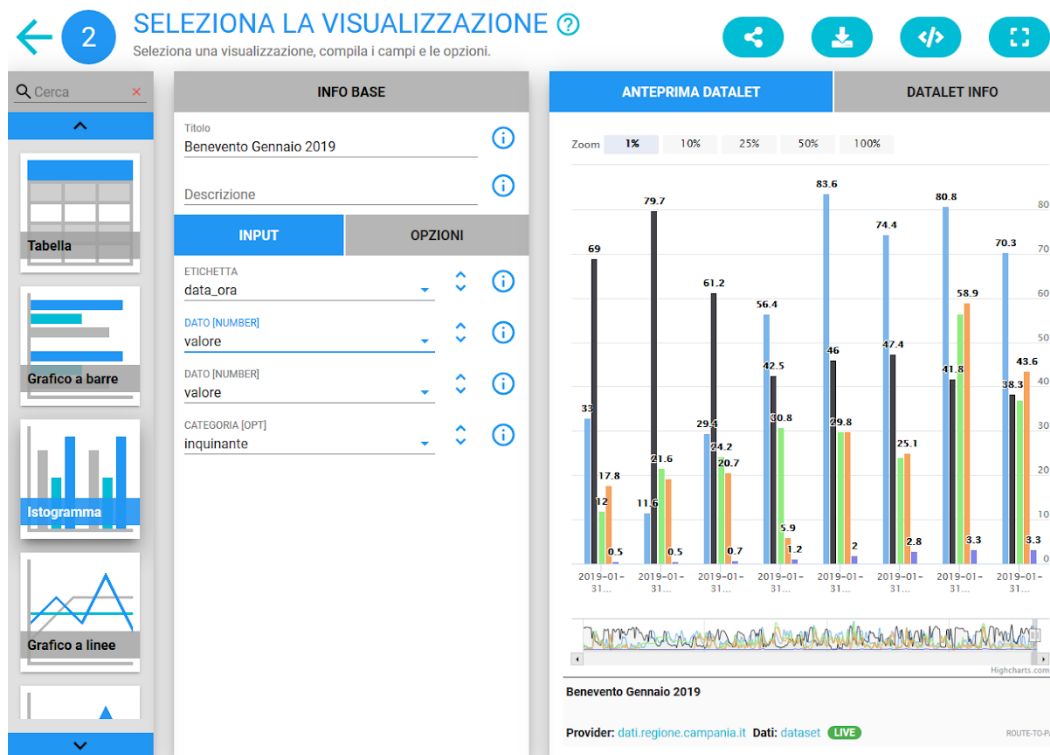


Figura 32. Open Data Campania - Emissioni inquinanti a Benevento a Gennaio

⁶⁰ <https://dati.regione.campania.it/opendata>

⁶¹ <https://dati.regione.campania.it/catalogo/datasetdetail/dati-qualita-dell-aria-2019-benevento>

7.3.8 Bristol

Interessante è anche il sito degli Open Data della città di Bristol⁶², che fornisce diverse visualizzazioni di dati riguardanti diversi settori e mostra alcune dashboard della città.

Un esempio⁶³ è mostrato nella figura seguente.

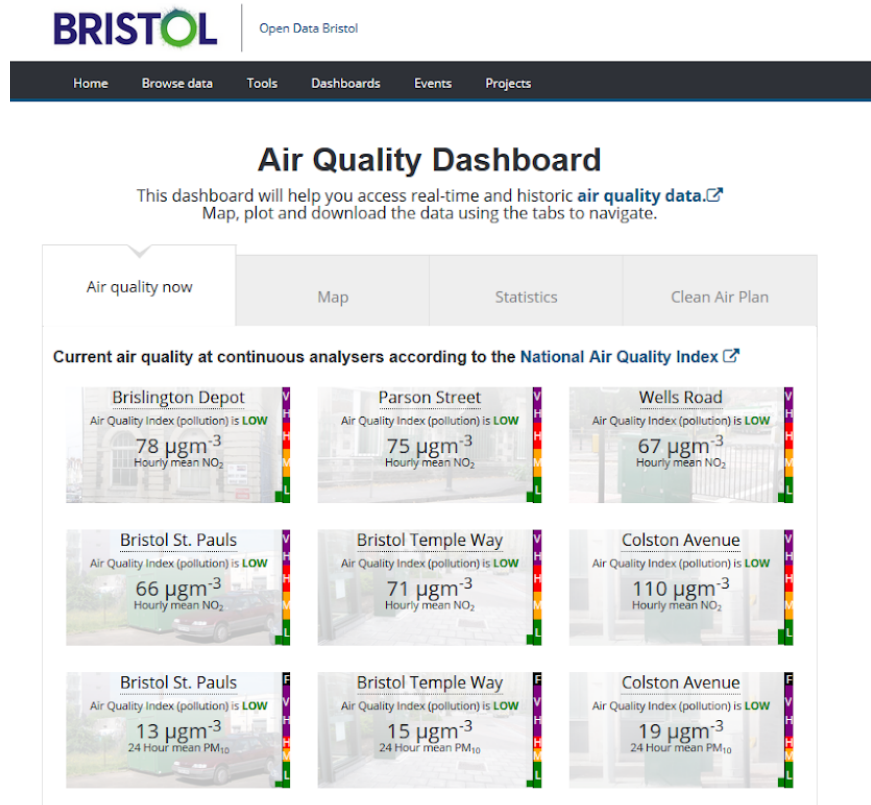


Figura 33. Bristol - Qualità dell'aria

⁶²<https://opendata.bristol.gov.uk/pages/homepage/>

⁶³<https://opendata.bristol.gov.uk/pages/air-quality-dashboard-new/#air-quality-now>

7.3.9 Comune di Bologna

Una mappa comunale interattiva particolarmente ricca di dati è la Open Map del sito del Comune di Bologna⁶⁴, collegato a 56 differenti dataset (dati su scuole, farmacie, colonnine di ricarica per veicoli elettrici, singoli alberi, ecc.).

La visualizzazione di questi dataset può essere attivata e disattivata dall'utente.

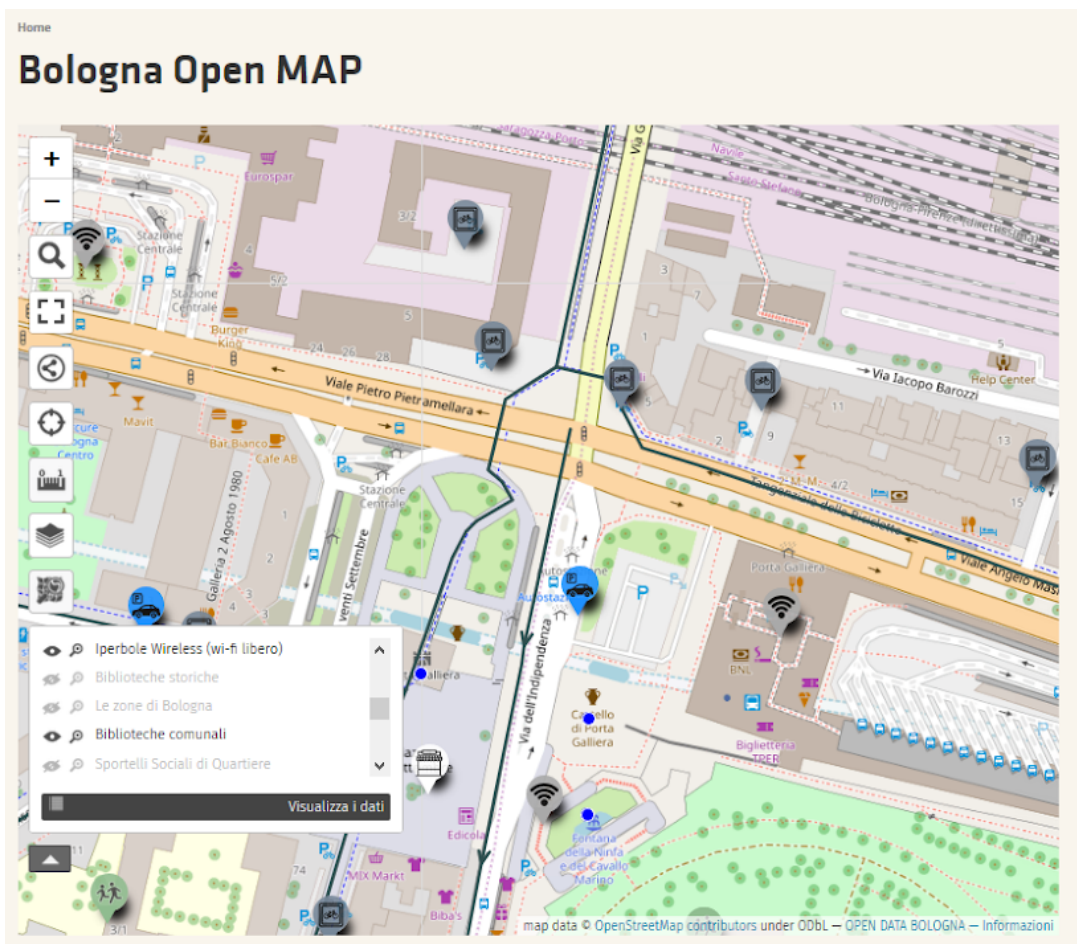


Figura 34. Comune di Bologna - Mappa interattiva

⁶⁴ <http://dati.comune.bologna.it/bolognaopenmap>

7.3.10 Comune di Firenze

Un altro esempio di applicazione di Snap4city è una dashboard nel comune di Firenze che mostra gli eventi culturali della città, i parcheggi liberi, le condizioni meteo, le linee di trasporto, assieme agli hashtag più utilizzati nei social networks.

Si veda come esempio⁶⁵ la seguente figura.



Figura 35. Comune di Firenze - Dashboard

⁶⁵ https://dashboard.km4city.org/dashboardSmartCity/view/index.php?iddashboard=NjQ=&nome_dashboard=SocialAspects2

8 Conclusioni

In questa linea di attività sono state poste le fondamenta dell'intero "Framework per la governance dei dati urbani energetici" con particolare focus sulla piattaforma di recupero dati su scala nazionale: la inter-SmartCityPlatform (iSCP).

Nel capitolo 1 è stata presentata l'architettura di riferimento e una prima definizione della piattaforma inter-SmartCityPlatform per il recupero dei dati energetici su scala nazionale.

Nel capitolo 2 è stata trattata, con un livello di dettaglio maggiore, la comunicazione interoperabile e quali sviluppi software si dovranno rendere necessari.

Nel capitolo 3 è stato compiuto con successo lo studio di fattibilità per l'integrazione al framework di altre piattaforme nazionali (p.es. PELL). Ne seguirà dunque un'implementazione.

Nel capitolo 4 viene riportata un'azione mirata di consolidamento dell'interfaccia della SCP (SCP-GUI) per predisporre il prototipo alla successiva evoluzione su scala nazionale.

Nel capitolo 5 è stata individuata la metodologia di adesione alla piattaforma inter-city (iSCP) ed è stata fatta una prima ipotesi di percorso di adesione alla inter-SCP in sette passi graduali.

Nel capitolo 6 è stato realizzato un ampio stato dell'arte sulle tecnologie per la visualizzazione dati, necessario per compiere una scelta consapevole per gli sviluppi futuri relativi alle piattaforme ICT in ambito Smart City che ENEA sta portando avanti.

In conclusione, la linea di attività corrente ha raggiunto tutti i risultati prefissati ed è andata oltre, effettuando lo studio dello stato dell'arte delle interfacce di reportistica, in modo da preparare nel modo migliore le basi per la progettazione e sviluppo che avverranno nella prossima annualità (linea di attività 20, annualità 2020).

9 Riferimenti bibliografici

1. C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, M. Paolucci, “Analysis and assessment of a knowledgebasedsmart city architectureproviding service APIs”, Future Generation Computer Systems Volume 75 (2017), pp 14–29. Elsevier
2. [Fernanda Viegas and Martin Wattenberg "How To Make Data Look Sexy", (April 19, 2011). CNN.com. Archived from the original on May 6, 2011.
https://web.archive.org/web/20110506065701/http://articles.cnn.com/2011-04-19/opinion/sexy.data_1_visualization-21st-century-engagement?_s=PM%3AOPINION
3. “Financial Times Visual Vocabulary”, <https://github.com/ft-interactive/chart-doctor/tree/master/visual-vocabulary>
4. VijayGupta, “An Analysis of Data Visualization Tools”, International Journal of Computer Applications (0975–8887), Volume 178 – No. 10 (May 2019).
5. Mohammed Kemal, “Data Visualization Tools In Action Choosing a Visualization Software”, Technical report (April 2019), DOI: 10.13140/RG.2.2.11690.26560.
6. [Paolo Nesi, Michela Paolucci, “Supporting Living Lab with Life Cycle and Tools for Smart City Environments (S)”, Computer Science, Engineering, Published in DMSVIVA 2018, DOI:10.18293/seke2018-019.