

UniCO

Progettazione dei metodi di
pubblicazione dei dati aperti

Sommario

SOMMARIO	1
OBIETTIVI DELLA PROGETTAZIONE	2
ANALISI DEI REQUISITI	2
LA DISTRIBUZIONE DKAN	2
IMPLEMENTAZIONE DEL MULTI-ENTE	3
SOLUZIONE MULTI-SERVER	4
SOLUZIONI MULTI-SITE E MULTI-DOMAIN	4
SOLUZIONE MULTI-TENANT	6
ALBERO DELL'AMMINISTRAZIONE TRASPARENTE	6
DESCRIZIONE DEL SISTEMA	9
STRUTTURA DEL SISTEMA	9
ARCHITETTURA DI SISTEMA	11
USER ACCESS AND PERMISSION	12
MULTI TENANT BACKEND	12
DATA SELECTION AND VIEW INTERFACE	12
INTERFACCE SERVIZI MULTICANALE	12
DATA INTERFACE LAYER	12
LEGACY INTERFACE LAYER	13
DESCRIZIONE DELLE INTERFACCE	13
DATASET REST API	13
DATASTORE API	13
CKAN DATASET API	13
OPEN DATA API	13

Obiettivi della progettazione

Il progetto *Uni.C.O. - Unione Comuni Opendata* prevede il riutilizzo e l'implementazione della "buona pratica" della Città Metropolitana di Genova relativa ai processi di apertura dei dati per la loro diffusione a cittadini, enti e portatori di interessi. La pubblicazione degli "open data" è veicolata attraverso l'albero dell'Amministrazione Trasparente (<http://dati.cittametropolitana.genova.it/>) utilizzando il Content Management System open-source DRUPAL (<http://www.drupal.org>, versione 7.x).

L'obiettivo del progetto è quello di estendere la buona pratica agli Enti del partenariato consentendo di ottenere un modello di trasparenza e comunicazione, aggiornato ed in linea con la normativa.

L'estensione prevede una reingegnerizzazione del sistema al fine di renderlo multi-ente e con strutture native open data secondo la logica CKAN/DKAN.

A fronte dei requisiti del progetto, evidenziati anche nelle riunioni con i responsabili dell'Area Metropolitana di Genova, la progettazione è consistita nei seguenti passi:

- analisi dei requisiti e delle necessità del committente
- valutazione della distribuzione DKAN per la gestione degli opendata
- valutazione delle modalità di implementazione del multi-ente
- valutazione della modalità di implementazione del front-end (albero amministrazione trasparente)

Nel seguito vengono descritte le risultanze dei passi precedenti.

Analisi dei requisiti

L'analisi dei requisiti è partita dal fatto che era stata già individuata, in fase di stesura del progetto, la piattaforma da utilizzare per reingegnerizzare il sistema. Il requisito stesso prevedeva cioè la conoscenza della distribuzione DKAN e del CMS Drupal sotteso. Tale requisito è ampiamente soddisfatto dalla mia esperienza professionale sull'utilizzo e la configurazione di molteplici portali sviluppati con Drupal, ivi compresa una distribuzione DKAN.

Inoltre è stato analizzato il requisito relativo alla distribuzione della soluzione ai comuni dell'Unione Valle Scrivia. L'obiettivo è quello di non creare tante installazioni uguali, una per comune, in modo da evitare di dover mantenere ogni singola installazione, dato anche il fatto che la manutenzione è a carico della Città Metropolitana.

La distribuzione DKAN

DKAN è una piattaforma di open data basata sulla community, gratuito e open source che permette alle organizzazioni e agli individui di pubblicare e utilizzare informazioni strutturate.

DKAN (<https://getdkan.org/>) è una suite completa di funzionalità di catalogazione, pubblicazione e visualizzazione che consente ai governi, alle organizzazioni non profit e alle università di pubblicare facilmente i dati al pubblico. DKAN è gestito da CivicActions.

DKAN è una distribuzione del CMS Drupal (versione 7) basato su CKAN (<https://ckan.org/>), il primo portale open source di dati aperti, ampiamente adottato dalle organizzazioni governative e non, in tutto il mondo. CKAN è l'acronimo di Comprehensive Knowledge Archive Network.

Dalla mia esperienza con Drupal e DKAN, ed anche analizzando portali analoghi a quello da sviluppare, come ad esempio il portale dei dati aperti del Comune di Genova (<http://dati.comune.genova.it/>), risulta evidente che la distribuzione ha tutte le funzionalità richieste dal progetto e soddisfa tutti i requisiti per la gestione, la condivisione e la pubblicazione degli open data.

Per verificare lo stato di mantenimento della distribuzione dal punto di vista dei moduli software e della compatibilità con gli stack LAMP più evoluti, ho creato un nuovo progetto su Pantheon (piattaforma di hosting e sviluppo di portali, <https://pantheon.io/>) in cui ho installato una versione completa di DKAN.

Lo stack LAMP dispone di PHP 7.2 e questo ha causato qualche warning PHP risolto con alcune patch mirate di alcuni moduli utilizzati dalla distribuzione.

Per il resto il funzionamento è ottimale ed è adeguato a rispondere a tutte le esigenze del progetto.

Il livello di customizzazione del tema a livello di interfaccia, potrebbe essere sufficiente per sviluppare una interfaccia pulita ed univoca per il progetto, senza interventi (o comunque con pochi interventi) a livello di codice css o di tema.

Implementazione del multi-ente

Ognuno dei comuni della Valle Scrivia, inseriti nel progetto, deve avere a disposizione un portale di back-end per inserire i dati aperti all'interno della struttura di dataset condivisa e un sito di front-end dove tali dati sono visualizzati e ricercabili.

Le possibili soluzioni vagliate sono:

- **multi-server:** ogni comune coinvolto (e la città metropolitana) dispone di un proprio web server su cui vengono ospitati il portale di gestione dei dati aperti e il sito di front-end per l'esposizione dei dati (il sito può anche essere all'interno del portale)
- **multi-site:** il sistema si basa su un unico web server su cui vengono implementati portali separati per i comuni (soluzione multi-site di Drupal)
- **multi-domain:** il sistema si basa sempre su un unico web server ed anche su un unico database, mentre è demandato al CMS (anche attraverso moduli aggiuntivi) lo smistamento su diversi domini dei siti

- **multi-tenant:** il sistema si basa su un unico web server su cui è installato un unico portale in cui le informazioni e le pagine sono raggruppate secondo la logica dei gruppi

Soluzione multi-server

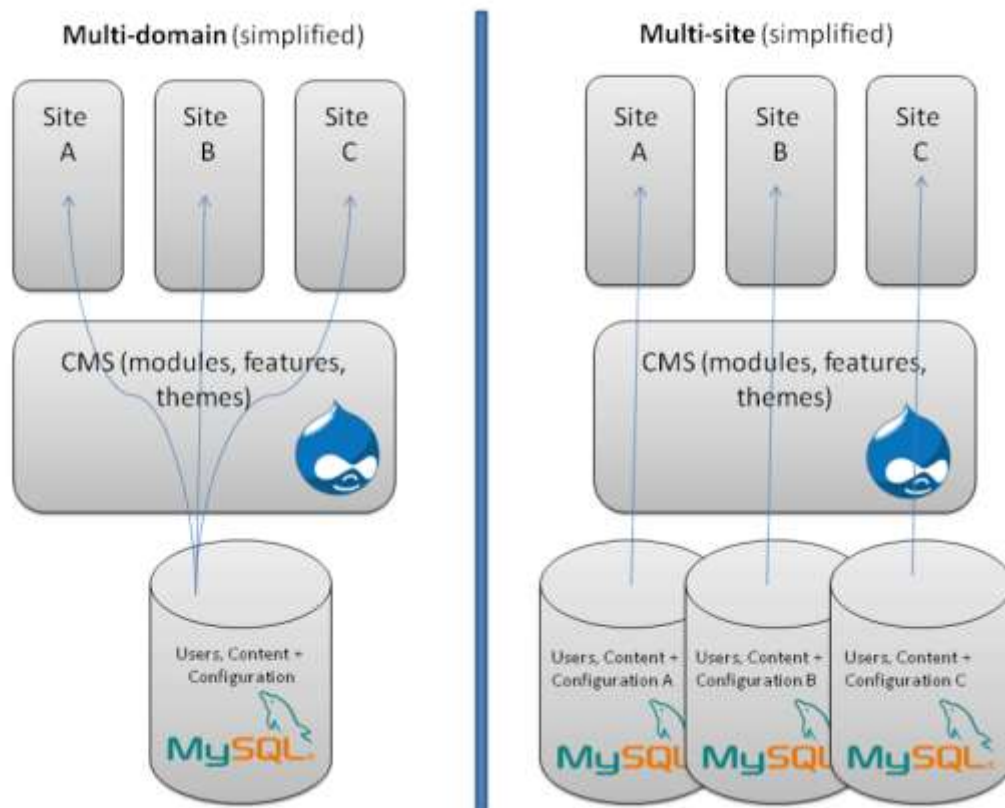
La soluzione **multi-server** ha alcuni aspetti critici:

- ogni portale richiede manutenzione sistemistica e manutenzione software
- ogni modifica effettuata al sistema, deve essere ripetuta per tutte le installazioni
- tutte le parti condivise sono ripetute per ogni installazione
- le problematiche relative alla sicurezza sono indipendenti su ogni sito, e sono quindi nel caso peggiore moltiplicate per il numero di siti

Siccome l'idea è quella di estendere ad altri comuni la possibilità di pubblicare i propri dati aperti tramite questo sistema, ogni volta bisognerebbe creare un nuovo web server e clonare la baseline del sistema.

Comunque il fatto che ogni comune abbia il proprio portale, permette una maggiore facilità di personalizzazione e assicura una completa separazione dei dati rispetto agli altri comuni.

Soluzioni multi-site e multi-domain



Schema soluzione multi-site VS multi-domain

Le soluzioni **multi-site** e **multi-domain** sfruttano la capacità di Drupal di implementare più siti nella stessa installazione. A seconda della configurazione, possono esserci database separati per i siti, oppure un database unico o ancora è possibile condividere alcune tabelle (utenti ad esempio) e separare il resto dei dati.

Dal punto di vista dell'utente di back-end la differenza tra multi-site e multi-domain è abbastanza irrilevante, mentre tecnicamente e dal punto di vista dello sviluppatore, le differenze tra i due sono rilevanti.

Queste due soluzioni hanno sicuramente il pregio di ridurre l'effort necessario per la manutenzione sistemistica e riducono anche le problematiche relative alla sicurezza, in quanto il sistema si poggia su un unico web server.

Di contro hanno alcune problematiche:

- il multi-site prevede comunque di avere tante installazioni quanti sono i siti in sottocartelle della document root (vedi schema successivo) e quindi la manutenzione software è comunque da fare singolarmente per ogni sito
- l'aggiornamento di Drupal alle versioni successive è più difficoltoso sia perché bisogna ripeterlo per tutte le installazioni ma anche perché il fatto di avere diversi database o unico database con tabelle condivise, complica notevolmente la procedura
- il multi-domain è implementato attraverso moduli di terze parti (Domain Access è il più usato) che hanno grande impatto sulla installazione di Drupal, rendendo l'intero sistema più fragile e soggetto a bug.

In generale, per la mia esperienza, sono soluzioni meno robuste e un po' più problematiche rispetto ad avere un sito unico o molti siti separati.

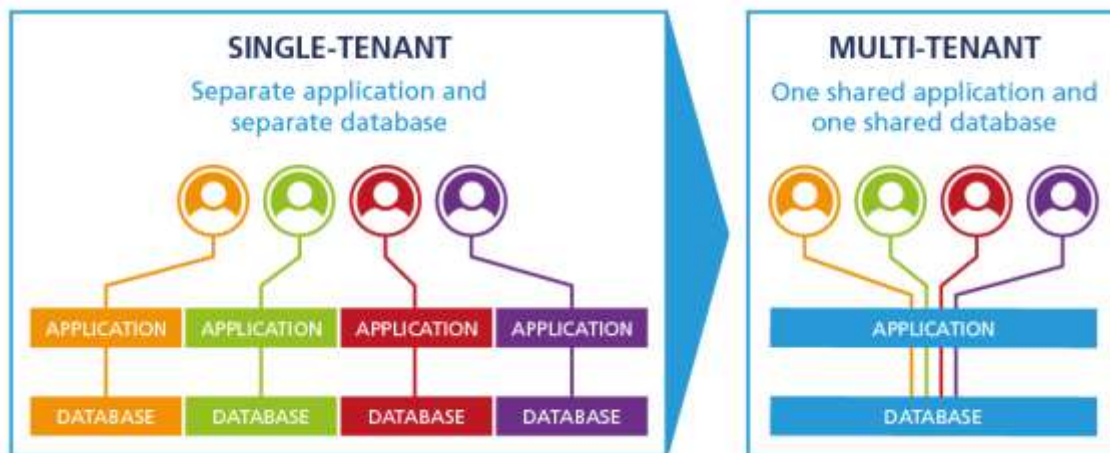
Multi-Site Structure



Name	Size
all	3 items
default	3 items
localhost.site1	4 items
sites	0 items
modules	0 items
themes	1 item
settings.php	8.7 KB
localhost.site2	4 items
sites	0 items
modules	0 items
themes	1 item
settings.php	8.7 KB

Schema struttura cartelle multi-site

Soluzione multi-tenant



Schema soluzione multi-tenant

Il modulo OG Groups permette di rendere **multi-tenant** un portale sviluppato con Drupal. Tale modulo è utilizzato da diversi anni ed è molto maturo. Lo ho utilizzato spesso in diversi contesti anche di applicazioni core per le aziende.

In sostanza con OG Groups è possibile suddividere i contenuti per gruppi, creando veri e propri mini siti in cui possono accedere solo gli utenti appartenenti al gruppo stesso (ed eventuali amministratori del sistema).

Con il meccanismo dell'audience ogni dato sarà quindi gestito solo dal proprietario e dal gruppo di appartenenza. Associando i comuni a tali gruppi arriviamo al risultato desiderato.

Il punto di attenzione per questa soluzione è dato dal fatto che la configurazione di OG Groups deve essere fatta con molta attenzione, per impedire che i dati di un gruppo siano modificabili da utenti senza i dovuti permessi.

La scelta è ricaduta sullo sviluppo del multi-tenant anche in considerazione del fatto che DKAN già prevede questa soluzione. Sfrutteremo i gruppi in DKAN per sezionare i dati e le pagine per i diversi comuni, sia dal punto di vista del back-end in cui gli utenti dei singoli comuni potranno vedere e gestire solo i propri dati, sia dal punto di vista del front-end in cui i dati di ogni comune saranno visionabili e ricercabili dal front-end separatamente.

Albero dell'amministrazione trasparente

Attualmente l'albero dei dati dell'Amministrazione Trasparente della Città Metropolitana di Genova (<https://dati.cittametropolitana.genova.it/it/content/amministrazione-trasparente>) è

una pagina di un sito che lista i link agli open data inseriti nel sito stesso (sviluppato con Drupal 7).

Uno degli obiettivi del progetto è quello di creare, per ogni comune, tale pagina (oltre ad aggiornare quella della Città Metropolitana di Genova), in cui i dati sono presi automaticamente dagli open data inseriti in DKAN.

Le opzioni di cui abbiamo discusso sono:

- creare un mini sito (di una sola pagina) di front-end per comune disaccoppiandolo dal DKAN di back-end
- creare un unico sito di front-end, con una pagina dedicata ad ogni comune, disaccoppiandolo dal DKAN di back-end
- utilizzare DKAN anche come front-end

Le prime due soluzioni prevedono il decoupling (disaccoppiamento) di Drupal, pratica che si sta sempre più consolidando in questi ultimi mesi.



Drupal disaccoppiato VS tradizionale

Uno dei trend più seguiti dalla community degli sviluppatori web negli ultimi anni è sicuramente quello del “Decoupled CMS”, ossia la possibilità di separare il frontend dal backend per rendere disponibili i contenuti su più canali e nei formati più opportuni.

L’evoluzione dei siti web tradizionali verso web applications interattive e la necessità di accedere ai dati nelle modalità più disparate hanno reso necessario un approccio “disaccoppiato” dove al front-end web tradizionale vengono affiancati più front-end. In alcuni casi, addirittura, il front-end web nativo cessa di essere un requisito, diventando secondario rispetto a un framework del tipo “client side” che permette agli sviluppatori front-end un controllo maggiore sulla presentazione e sulle interazioni con i dati.

In Drupal 8 (ultima versione disponibile di Drupal), il disaccoppiamento è quasi nativo, visto che nel core sono presenti i moduli che permettono di interrogare e aggiornare i dati attraverso web service.

In Drupal 7 invece per poter operare il disaccoppiamento è utile (necessario, se non si vuole sviluppare a mano le interfacce) installare una serie di moduli (Services in particolare) che permettono di creare degli end point accessibili per interrogazioni ed aggiornamenti.

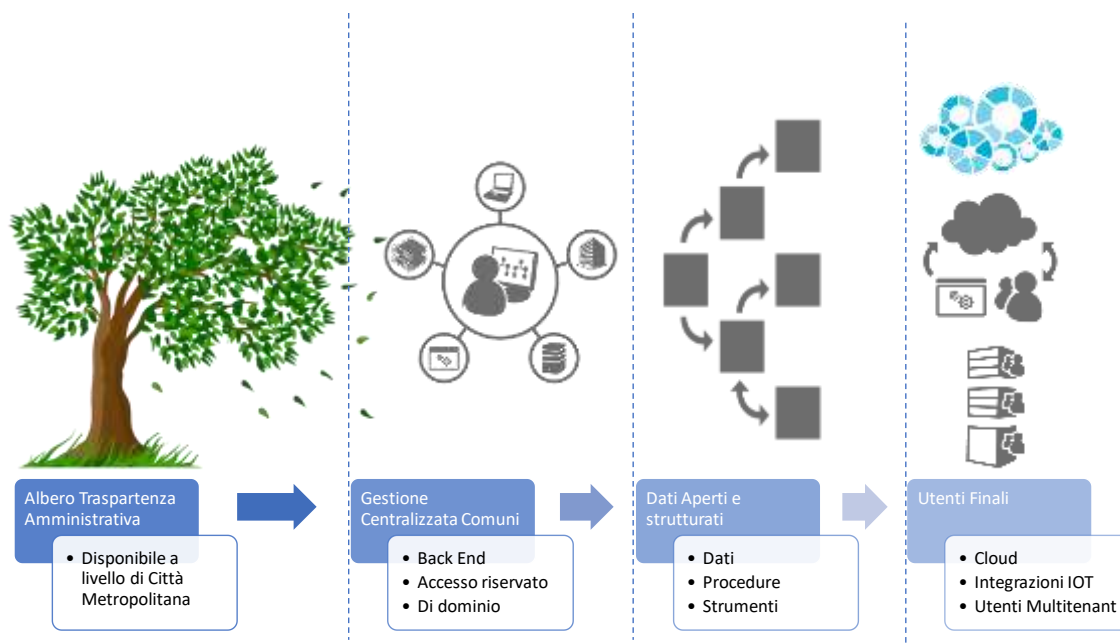
DKAN ha già questa parte installata, perché permette di condividere i propri dataset attraverso delle API.

In una prima fase abbiamo concordato comunque di procedere con la soluzione che prevede di utilizzare DKAN anche per il front-end, in modo da sfruttare il tema di DKAN (configurabile per le nostre esigenze) per creare le pagine custom con gli alberi dei dati dell'Amministrazione Trasparente dei comuni, e non implementare altri siti di front-end.

Questa scelta sarà valutata ed eventualmente modificata, alla luce dei risultati che si otterranno.

Descrizione del sistema

In questo capitolo illustro il sistema che andrò ad implementare.



Schema funzionale

Da un punto di vista funzionale, il sistema deve rendere disponibile l'albero della Amministrazione trasparente agli utenti finali che navigano i siti dei Comuni coinvolti.

Come mostrato nello schema qui sopra, questo è realizzato attraverso la gestione della struttura dell'albero a livello di Città Metropolitana, poi ogni comune accede alla gestione dei propri dati aperti, organizzati attraverso le categorie dell'albero, tali dati sono processati, manipolati e resi disponibili in diverse modalità all'utente finale.

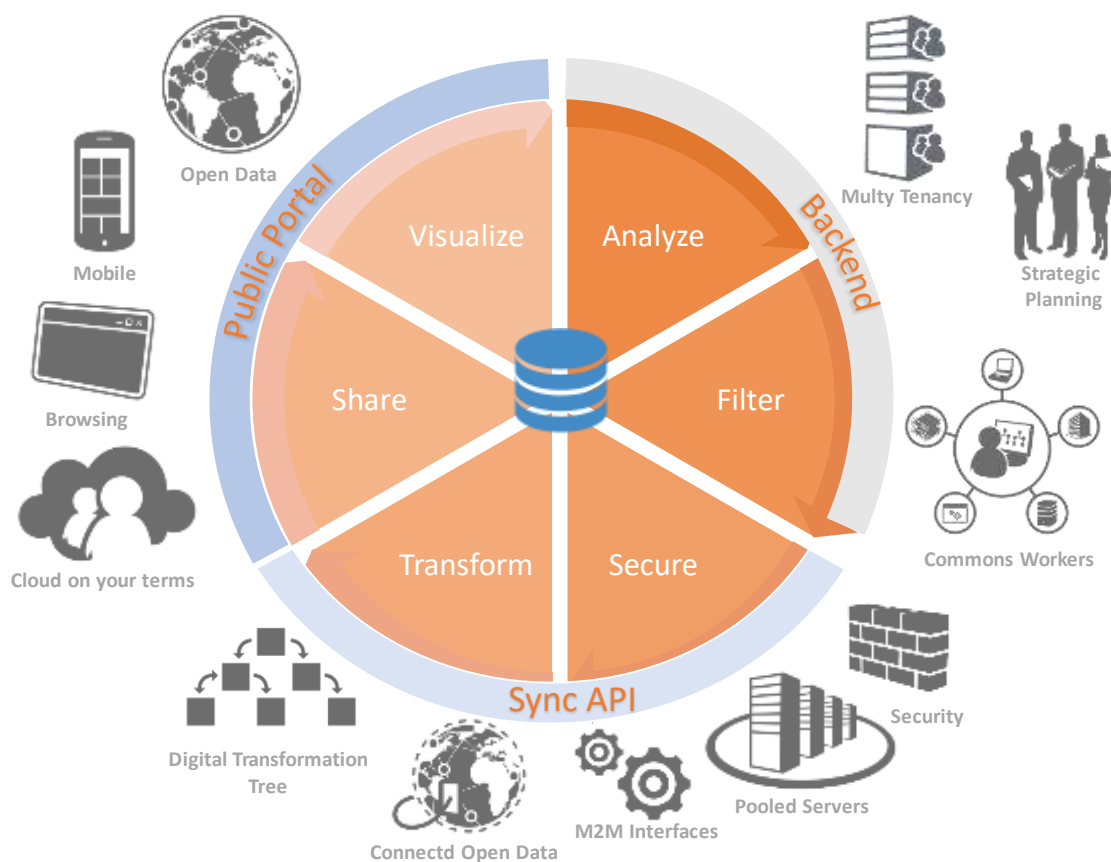
Struttura del sistema

Il sistema sarà centrato su una installazione DKAN che fornirà sia la parte di back-end per i singoli Comuni (attraverso il meccanismo degli OG Groups), sia la parte di front-end con le pagine custom degli Alberi dell'Amministrazione Trasparente per ogni Comune.

La parte di back-end sarà accessibile solo dagli utenti abilitati, e gli utenti dei singoli comuni saranno abilitati ad operare solo sui dati del proprio comune.

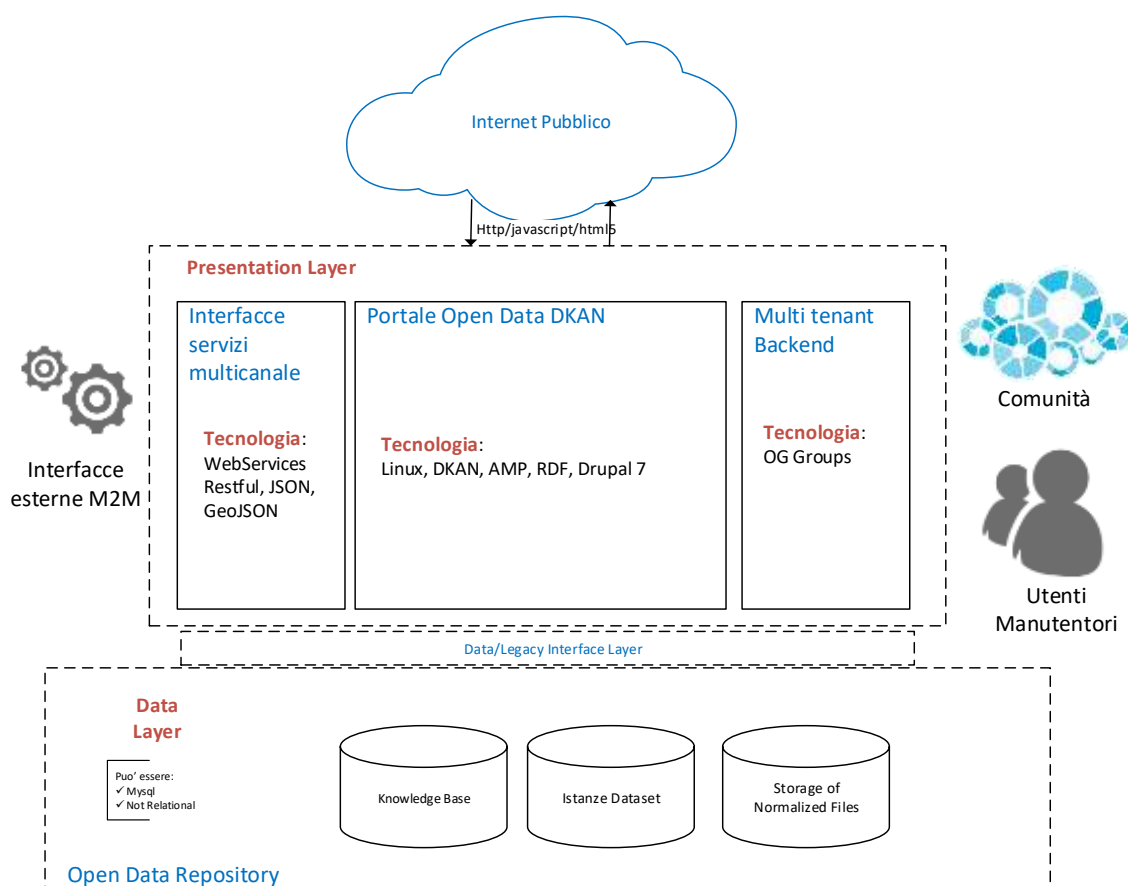
Il front-end, composto dalla pagina custom con l'albero dell'Amministrazione Trasparente (parametrica, nell'url sarà presente il comune tramite id del gruppo o altro meccanismo), sarà pubblico e linkato nei singoli istituzionali dei singoli comuni.

Il sistema esporrà una serie di API per permettere ad agenti esterni di fruire degli open data. Le API sono già previste in formato DCAT ed in altri formati standard, oltre a quelle dedicate all'interfacciamento con i dati e portali CKAN.



Schema generale ad alto livello

Architettura di sistema



Schema dell'infrastruttura

Livello di presentazione:

- User Access and permission
- Multi tenant backend
- Data Selection and View Interface
- Interfacce servizi multicanale
- Front end

Business Logic:

- Data Interface Layer
- Legacy interface Layer

Il livello di presentazione sarà interamente gestito tramite DKAN, personalizzato per le esigenze di progetto. Anche il front end sarà gestito direttamente con pagine custom all'interno di DKAN.

Il livello di Business Logic in questo contesto si compone solo dei componenti per astrarre le connessioni verso il db e verso altre sorgenti di dati.

User Access and permission

Tramite questa interfaccia di Back-End gli utenti con privilegi di amministrazione del sistema potranno definire o revocare accessi e permessi ai nuovi utenti. Sarà possibile definire privilegi di accesso a diversi ruoli utenti ed assegnare l'audience ai dati di differenti gruppi (Comuni) in base ad un meccanismo denominato Organic Group.

Multi tenant backend

Ogni utente dei Comuni, una volta autenticato nel sistema, accederà ai dati del portale relativi al proprio Comune. Potrà inserire nuovi dati aperti (e modificare gli esistenti), che saranno automaticamente assegnati al proprio Comune e saranno quindi visibili nel front-end relativo al Comune.

Data Selection and View Interface

Il sistema sarà in grado di gestire e organizzare i dataset ed i dati aperti. Tutti i contenuti generati nel tempo, potranno essere acceduti, visualizzati e gestiti in base alla logica di accesso e ai permessi dell'utente collegato. Attraverso la medesima interfaccia potranno essere acceduti anche i dati storicizzati su Database per esigenze operative. Sarà possibile definire criteri di ricerca a piacere ed esporre filtri di interrogazione sulla base delle esigenze degli utenti e degli uffici operativi.

Interfacce servizi multicanale

DKAN comprende una serie di API per consentire di comunicare con applicazioni esterne:

- Dataset REST API
- Datastore API
- CKAN Dataset API
- Open Data API

Le interfacce saranno descritte nel dettaglio nel prossimo capitolo

Data Interface Layer

E' la componente di interfaccia che consente di effettuare la connessione e gestire i flussi da e verso i differenti repository al fine di astrarre la Business Logic dal Data Layer. Funziona come

livello di astrazione per l'accesso ai dati e rende il sistema configurabile per essere interfacciato in futuro con differenti gestionali aventi DB engine differenti.

Legacy interface Layer

E' la componente di integrazione che astrae la connessione e il recupero di dati da sistemi non db. In questo caso potrebbe rendersi necessario il recupero dati anche da file system.

Descrizione delle interfacce

DKAN comprende una serie di API per consentire di comunicare con applicazioni esterne:

- Dataset REST API
- Datastore API
- CKAN Dataset API
- Open Data API

Dataset REST API

L'API REST di Dataset DKAN utilizza il modulo Services per creare un endpoint CRUD all'indirizzo `api/dataset/[nodo]`. Per impostazione predefinita, questo endpoint fornisce pieno accesso CRUD ai contenuti di un sito Web e accesso limitato agli utenti (per consentire l'autenticazione).

Datastore API

DKAN offre un Datastore API come endpoint personalizzato per il modulo Drupal Services. Questa API è progettata per essere il più compatibile possibile con l'API di Datastore CKAN.

CKAN Dataset API

DKAN fornisce una serie di API che sono progettate per fornire informazioni di catalogo e dataset pubblico, di sola lettura, nonché gli aggiornamenti che consentono agli osservatori di monitorare le modifiche. Tutte le API sono fornite tramite il modulo Open Data Schema Map.

Open Data API

DKAN supporta API per i due principali standard dei dati aperti: Project Open Data e DCAT-AP. Entrambi sono forniti e sono configurabili attraverso il modulo Open Data Schema Map.

DKAN fornisce il supporto completo e il mapping per Project Open Data (<https://project-open-data.cio.gov/>), nelle sue varianti federali e non federali, con un endpoint di tipo `data.json`.

DKAN fornisce anche gli endpoint e un mapping configurabile dei campi per DCAT-AP, il profilo di applicazione per portali dei dati aperti in Europa, sviluppato dalla Commissione europea.

DCAT-AP si basa sul Data Catalog Vocabulary (DCAT), ma fornisce definizioni più rigorose di cataloghi, set di dati, distribuzioni e altri oggetti. Attraverso Open Data Schema Map, DKAN fornisce sia un endpoint di catalogo, sia endpoint dei singoli RDF per ogni dataset.