WP2 – Infrastruttura Data-Lake

Revisione Bozza

MAPS | Multidimensional Area Place-based System

Incontro bi-settimanale tech team · 2 marzo 2026

Agenda

Cosa presentiamo

1. Scope e deliverable WP2
2. Architettura Medallion (Bronze → Silver → Gold)
3. Stack tecnologico
4. Architettura cloud
5. Sfide tecniche specifiche
6. Stato attuale della bozza

Cosa vogliamo discutere

- Decisioni architetturali aperte
- Deployment: managed vs self-hosted
- Priorità MVP Fase 1a
- Feedback sul documento

WP2 – Scope e Deliverable

Task 2.1 · M9–M10

Architettura tecnica

- ✓ D2.1.1 – Progettazione Tecnica Data-Lake
- ✓ D2.1.2 – Cloud Solution Design
- ✓ D2.1.3 – Specifiche Hosting

Deadline: 31/03/2026

Task 2.2 · M10–M12

Pipeline ETL

- ◯ D2.2 – Script ETL (Python/SQL)

Deadline: 31/05/2026

Task 2.3 · M10–M12

Validazione

- ◯ D2.3 – Report di Validazione ETL

Deadline: 31/05/2026

Oggi presentiamo la bozza dei tre deliverable del Task 2.1 — architettura, cloud, hosting. D2.2 e D2.3 saranno completati nella fase di sviluppo.

D2.1.1 – Requisiti

FUNZIONALI

RF1 – Acquisizione eterogenea
180+ dataset da ISTAT, Ministeri, registri pubblici. Formati: CSV, XLSX, PDF, HTML.

RF2 – Gestione serie storiche
2010–2025, ~8.000 comuni. Gestione gap COVID e ~100 fusioni/scissioni comunali.

RF3 – Standardizzazione
Normalizzazione codici ISTAT, riferimento spaziale EPSG:32632, distinzione tra valori mancanti e null semantici.

RF4 – Architettura multi-layer
Raffinamento progressivo del dato: grezzo → pulito → pronto per le applicazioni.

NON FUNZIONALI

RNF1 – Completezza ≥95% dei comuni per dataset prioritari

RNF2 – Accuratezza ≥99,9% dei record con codice ISTAT valido

RNF3 – Tempestività Pipeline prioritarie completate entro 24h

RNF4 – Lineage Tracciabilità completa per ogni record, fonte → Bronze → Silver → Gold

RNF5 – Scalabilità Fino a 1TB in 3 anni, throughput ≥100 record/sec

RNF6 – Governance Catalogo completo dei dataset, dashboard qualità real-time, log di audit

D2.1.1 – Vincoli Architetturali

V1 – Open source self-hosted

Nessun servizio cloud gestito (BigQuery, Azure, AWS esclusi). Stack completo: PostgreSQL, PostGIS, Prefect, OpenMetadata, DuckDB. Indipendenza dal vendor e costi prevedibili.

V3 – Standard territoriali

PostGIS come standard de facto per dati spaziali in PostgreSQL. Sistema di riferimento EPSG:32632 (WGS84 / UTM zona 32N), compatibile con tutti i dataset geografici nazionali italiani.

V2 – Adeguatezza alla scala

Volumi dati ~10^4 righe × ~10^3 colonne. Infrastrutture enterprise (BigQuery, Spark) sarebbero eccessive — costi e complessità ingiustificati per i carichi di lavoro effettivi.

V4 – Conformità FAIR

I dataset rilasciati devono essere Findable (metadati strutturati), Accessible (CC-BY/CC0), Interoperable (GeoJSON, GeoParquet, CSV), Reusable (DOI per citabilità scientifica).

Dati questi vincoli, la soluzione tecnica adottata per RF4 è il **pattern Medallion** (Bronze → Silver → Gold), dettagliato nel deliverable D2.1.1.

Architettura Medallion

```
FONTI ESTERNE → BRONZE → SILVER → GOLD → APPLICAZIONI
```

Bronze

Archivio immutabile

- File system locale
- File originali, nessuna trasformazione
- Write-once, mai modificabile
- Checksum integrità SHA-256
- File JSON di metadati sidecar

Silver

Dati puliti e validati

- Schema PostgreSQL `silver`
- Schema EAV (Entity-Attribute-Value)
- SCD Type 2 per variazioni comunali
- Validazione con Great Expectations
- Unica fonte di verità

Gold

Dati pronti per le applicazioni

- PostgreSQL + PostGIS
- Tabelle wide denormalizzate
- Geometrie spaziali pronte per le query
- Alimenta API, dashboard, algoritmi SLO
- DuckDB per analytics ad-hoc

Silver Layer – Perché EAV?

Il problema: 180+ dataset con strutture eterogenee — lo schema tradizionale richiede centinaia di colonne e migrazioni continue. **La soluzione:** ogni osservazione diventa una riga `(comune, attributo, valore)`.

Schema tradizionale (problematico)

codice_istat	popolazione	asili_nido	ha_ospedale	ato_gas	…
001001	45230	12	true	?	…
001002	8420	2	false	?	…

Ogni nuovo dataset richiede nuove colonne → ALTER TABLE

Schema EAV (adottato)

entity	attribute	value	valid_from	source
001001	popolazione	45230	2024-01-01	ISTAT
001001	asili_nido	12	2024-01-01	ISTAT
001001	ha_ospedale	true	2015-01-01	MinSalute
001002	popolazione	8420	2024-01-01	ISTAT

Nuovi dataset = nuove righe. Lo schema rimane stabile.

Trade-off principale: query più complesse (PIVOT con CASE WHEN), ma la complessità è confinata al layer di trasformazione Silver → Gold, invisibile agli utenti dei data mart.

Sfide Tecniche

S1 – Nessuna geolocalizzazione puntuale

La maggior parte dei servizi (scuole, ospedali, tabaccherie) non dispone di coordinate precise. Approccio: presenza/assenza a livello comunale. Le isocroni operano su distanze comune-comune.

S2 – Variazioni dei confini amministrativi

~100 fusioni/scissioni comunali nel 2010-2025. Soluzione: tabella `comuni_variazioni` con SCD Type 2 e colonne `valid_from` / `valid_to`.

S3 – Gap temporali COVID

Discontinuità metodologiche nel 2020-2021. Flag `covid_affected` sui dataset impattati. Opzioni: imputazione statistica o esclusione esplicita dei periodi interessati.

S4 – Eterogeneità dei formati

207 dataset tra CSV, XLSX, HTML, PDF. Stack di estrazione multi-layer: Docling (PDF, 97,9% accuratezza), Pandas (CSV/Excel), BeautifulSoup (HTML).

Stack Tecnologico

Componente	Tecnologia	Ruolo
Storage primario	PostgreSQL 17 + PostGIS 3.5	Dati master, operazioni spaziali
Orchestrazione	Prefect 3.x	Scheduling, monitoraggio, ETL
Analytics	DuckDB 1.x	Query federate da PostgreSQL
Governance interna	OpenMetadata 1.x	Catalogo, lineage, qualità
Open data	CKAN 2.11	Catalogo pubblico (DCAT-AP_IT)
Qualità dati	Great Expectations 1.x	Validazione, anomaly detection

Principi guida

- **Open source self-hosted**: nessun vendor lock-in, costi prevedibili
- **Adeguatezza alla scala**: ~10^4 righe × ~10^3 colonne. BigQuery è eccessivo.
- **Standard territoriali**: PostGIS è lo standard de facto per i dati spaziali
- **Conformità FAIR**: CC-BY, GeoJSON/GeoParquet, DOI

D2.1.2 – Cloud Solution Design

Provider: DigitalOcean

- Prezzi mensili fissi — nessuna fatturazione variabile
- Kubernetes gestito (DOKS) — control plane gestito da DO
- Curva di apprendimento più semplice rispetto agli hyperscaler AWS/GCP/Azure

Perché Kubernetes invece di Docker Compose

- Infrastruttura dichiarativa, self-healing
- Isolamento dei workload per servizio (limiti CPU/memoria)
- Supporta la crescita del progetto senza migrazione architetturale

PostgreSQL gestito vs self-hosted

- Gestito: backup automatico, failover, PITR — zero overhead operativo
- Self-hosted (operatore CNPG): controllo totale, costo inferiore, maggiore carico operativo
- D2.1.2 presenta entrambe le opzioni — decisione rinviata a D2.1.3

Stima costi D2.1.2 (PostgreSQL gestito)

Voce	Costo/mese
DOKS control plane	$12
3 nodi (4 vCPU / 8 GB)	$192
PostgreSQL gestito	$80
Block storage (150 GB)	$15
Load Balancer	$12
Totale	**~$311/mese**

Deployment MVP attuale: server op-linkurious con Docker Compose. Migrazione a DOKS prevista per la fase di produzione.

D2.1.3 – Specifiche Hosting

IaC: Pulumi (Python)

- Cluster completamente provisioning con `pulumi up --stack maps`
- Due node pool: `general-purpose` (2× 2vCPU/4GB, fisso) per componenti di sistema; `workloads` (1–3× 4vCPU/8GB, autoscaling) per Prefect, PostgreSQL, OpenMetadata, CKAN

Database: CloudNative PostgreSQL (CNPG)

- Operatore self-hosted — risolve la domanda aperta di D2.1.2
- 2 istanze, replica sincrona, failover automatico
- Backup WAL su S3, point-in-time recovery (retention 30 giorni)

Segreti: AWS Parameter Store + External Secrets Operator

- Schema path: `/maps/{service}/{parameter}`
- Sincronizzazione a Kubernetes Secrets ogni ora

Stima costi D2.1.3 (CNPG self-hosted + autoscaling)

Voce	Costo/mese
DOKS control plane	$12
Pool general-purpose (2 nodi)	$48
Pool workloads (min 1 nodo)	$48
Pool workloads (max 3 nodi)	$144
Block storage (200 GB)	$20
Load Balancer	$12
Backup Spaces	$5
Totale (idle)	**~$145/mese**
Totale (picco ETL)	**~$241/mese**

$70–166/mese in meno rispetto alla stima D2.1.2 grazie a CNPG self-hosted + autoscaling.

MVP Fase 1a – Dataset Critici

5 dataset minimi per avviare gli algoritmi SLO (WP3):

Dataset	Fonte	Formato	Frequenza	Priorità
Confini comunali + metadati	ISTAT	Shapefile/JSON	Annuale	Alta
Popolazione residente	ISTAT	CSV	Annuale	Alta
Matrici pendolarismo	ISTAT	CSV	Decennale	Alta
Accessibilità infrastrutture di trasporto	ISTAT	XLSX	Spot	Media
Strutture sanitarie	Min. Salute	Excel	Annuale	Media

≥95%
copertura comunale per dataset prioritari

≥99,9%
record con codice ISTAT valido

<24h
latenza per le pipeline prioritarie

Stato della Bozza

Completato (Task 2.1)

- **D2.1.1** – Architettura Data-Lake
  - Pattern Medallion con motivazioni
  - Schema EAV Silver dettagliato
  - Data mart Gold (SQL completo)
  - Progettazione governance e lineage
  - Best practice (idempotenza, evoluzione schema)
- **D2.1.2** – Cloud Solution Design
  - Selezione provider (DigitalOcean)
  - Dimensionamento cluster e stime costi
  - PostgreSQL gestito vs self-hosted
- **D2.1.3** – Specifiche Hosting
  - Specifiche server dettagliate
  - Configurazione dei servizi

Da sviluppare (Task 2.2 / 2.3)

- **D2.2** – Script ETL
  - Struttura multi-worker-pool definita
  - Pattern pipeline comune documentato
  - Implementazione: M10–M12
- **D2.3** – Report di Validazione
  - Criteri di accettazione definiti
  - Suite di test strutturata
  - Esecuzione: dopo l'implementazione

Domanda: la bozza attuale è sufficiente per la revisione formale dei deliverable D2.1.x entro il 31/03/2026?

Domande Aperte – Progetto & Team

DOMINIO & IDENTITÀ

Q1 – Qual è il dominio definitivo? D2.1.3 usa `maps.gransassotech.it` come placeholder.

Q2 – Nome ufficiale del progetto: "MAPS" o "GST-MAPS"? Influisce sul catalogo CKAN e sui metadati DOI Zenodo.

Q3 – Esiste già un account DigitalOcean dedicato a MAPS, o va creato?

Q4 – Account AWS per Route53: dedicato a MAPS, o si usa l'account DEPP/Openpolis esistente?

TEAM & CONTINUITÀ

Q5 – Chi gestisce l'infrastruttura dopo la chiusura del progetto (maggio 2027)? I costi DigitalOcean sono ~$300/mese in corso.

Q6 – Dataset e piattaforma web rimarranno accessibili pubblicamente dopo la chiusura? Esiste un piano di sostenibilità?

Q7 – Chi realizza il frontend (sito scroll-telling + visualizzazione dati)? È previsto un contratto separato?

Domande Aperte – Tecniche & Processo

DECISIONI TECNICHE

Q8 – Database definitivo: DigitalOcean gestito o self-hosted via CloudNativePG? D2.1.2 presenta entrambe le opzioni senza una decisione finale.

Q9 – CKAN è confermato come catalogo open data pubblico? Richiede setup e manutenzione non banali — sono allocate le risorse?

Q10 – D2.1.3 risolve la questione managed vs self-hosted a favore di CloudNative PostgreSQL (CNPG): costo inferiore ($145–241/mese vs $311), pieno controllo PostGIS, maggiore carico operativo. Il team conferma questa scelta?

DELIVERABLE & PROCESSO FORMALE

Q11 – Il Task 2.2 (sviluppo ETL) inizia a M10, ma richiede un'infrastruttura operativa: PostgreSQL, Prefect, storage Bronze. Chi la predispone, e quando? Nel piano non esiste un task dedicato esplicitamente al deployment dell'infrastruttura prima dell'avvio dello sviluppo.

Q12 – I D2.1.x devono essere formalmente inviati al MUR entro il 31/03/2026, o è una scadenza interna con una fase di accettazione GST?

Q13 – Esiste un formato standard richiesto dal bando FRES 2? Template, copertina, firma, protocollo di invio?

Prossimi Passi

Entro il 31/03

- Revisione interna D2.1.x
- Chiusura feedback odierni
- Finalizzazione documento

M10–M11 (Mar–Apr)

- Avvio sviluppo pipeline ETL
- 5 dataset critici MVP
- Setup Prefect + PostgreSQL

M12 (31 maggio)

- D2.2 – Script ETL completati
- D2.3 – Report di validazione
- Consegna a WP3 (SLO)

Bozza completa: `deliverables/docs/it/wp2/`