STANDARD E PATTERNS DATA LAKE

Architettura AWS Data Lake

Documento:	Standard e Patterns Data Lake
Titolo:	Standard e Patterns Data Lake
Nome:	
Autore:	*Manuel Zini, Antonio Cappuccio, Paolo Simoni, Mario Marfella*
Revisore:	*Manuel Zini, Leonardo Boretti, Mario Marfella*
Data creazione:	*20/07/2020*
Data ultima modifica:	*04/11/2020*
Versione:	*1.6. Candidate*

Precedenti documenti:

Indice

1. Premessa

Il presente documento traccia le regole generali di integrazione nell'ambito del progetto Data Lake.

Esempio	Argomento da chiarire, punto aperto
Esempio	Da completare
Esempio	Integrazione rispetto a versione precedente

2. Introduzione

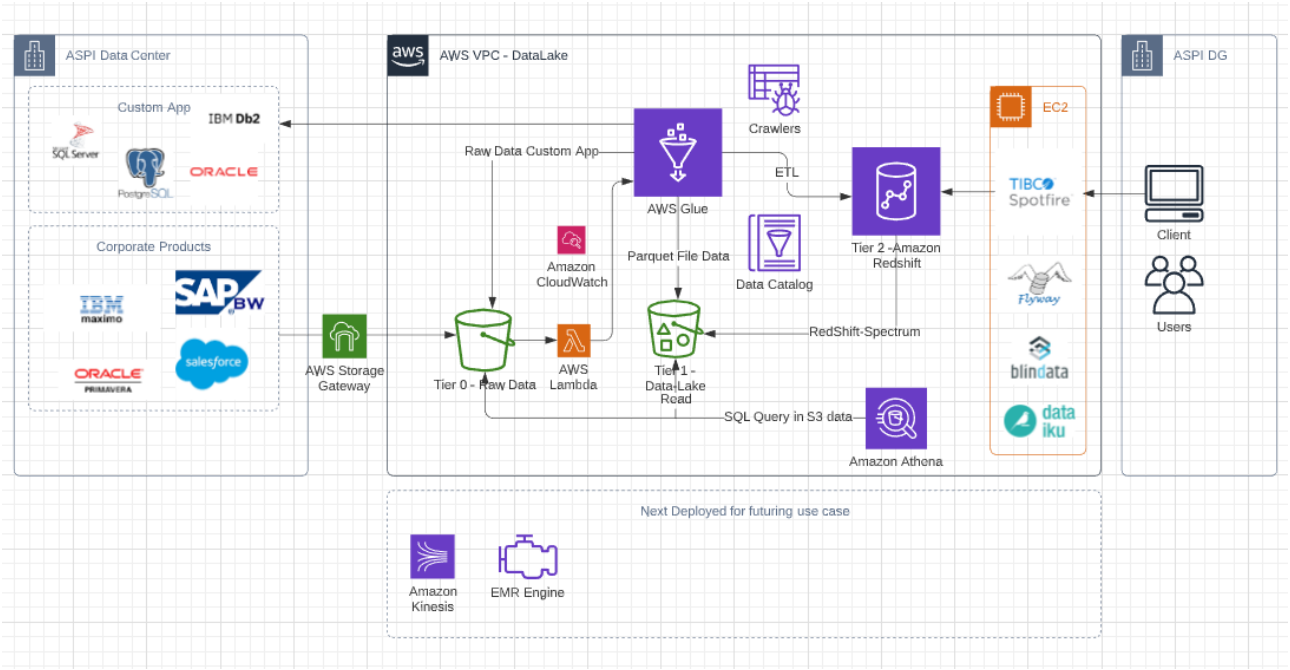
Il presente documento descrive e raccoglie gli standard adottati nella costruzione del sistema Data Lake di ASPI oltre ad una definizione puntuale dei design patterns utilizzati nella creazione dei flussi di integrazione ETL.
Ciascun flusso di alimentazione di uno dei Tier che compongono il Data Lake o ciascun flusso in uscita da questi o ciascun metodo di accesso/fruizione del dato presente in uno dei Tier sarà realizzato seguendo uno o più dei pattern descritti in questo documento.
Sono definiti in questo documento anche i file excel standardizzati per la raccolta delle sorgenti, delle mappature e delle tabelle datamart, tali documenti costituiranno documentazione del funzionamento dei flussi, input per i team che sviluppano e documentazione ad integrazione del sistema BlinData.

2.1. Utilizzo del documento

Il presente documento censisce standard generali e pattern di integrazione. Per ciascuna problematica che si presenti il richiedente di un nuovo flusso dati o in generale di un qualsiasi tipo di utilizzo dell'infrastruttura Data Lake farà riferimento al presente documento per identificare una soluzione disponibile. Solo qualora la soluzione non fosse presente richiederà il censimento di una nuova soluzione che sarà progettata con il team Architetture, eventualmente implementata in termini di moduli di 'enablement', testata e poi censita nel presente documento.

3. Architetture Data Lake. Definizioni

3.1. Architettura generale



Nel dettaglio:

- **AWS S3** - "Simple Cloud Storage Service" è lo storage selezionato per la conservazione dai dati. Rappresenta il vero e proprio D-LAKE.
- **AWS Glue** è utilizzato come servizio per lo sviluppo degli ETL in modalità "Spark". Legato a Glue ci sono il Data Catalog e i Crawlers. Il primo, a partire dal Job, permette di tracciare il modello dati e il relativo flusso, il secondo permette, volendo, di "deployare" automaticamente il modello dati in Amazon RedShift.
- **AWS Lambda** è un servizio serverless per l'esecuzione di codice (python o spark) custom per esigenze specifiche non coperte direttamente da Glue
- **Amazon Athena** è un servizio che permette di interrogare i dati conservati in S3 in modalità SQL-Like
- **AWS Storage Gateway** che permette di estendere S3 come file system remoto sui sistemi ON-PREM
- **AWS CloudFormation** per il change management di infrastruttura esteso in alcune funzionalità con la aws cli
- **Amazon APPflow** Servizio per integrazioni API sorgenti SaaS esterne (es. Salesforce, Dynatrace, ServiceNow).
- Una serie di servizi custom creati in EC2 quali:
    - **Tibco Spotfire**: Software SelfService per la creazione delle dashboard dei KPI.
    - **FlyWay**: Software OpenSource per la gestione dei change della banca dati RedShift per tutti i casi custom che non prevedono l'uso dei Crawlers
    - **Blindata**: Software per il Data Lineage e Augmented Data Catalogs
    - **DataIKU**: Per la Data Exploration e Machine Learning (usato per POC)

Per gli sviluppi futuri, in base a prossimi use-case l'infrastruttura è già predisposta per:

- Amazon Kinesis: Servizio per ingestion IOT.
- Amazon EMR: Servizio Big Data.
- Amazon DynamoDB: Database Chiave-Valore per la gestione dei dati in formato Json
- AWS Lake Formation: Servizio per il supporto e lo sviluppo del Data Lake. Permette di automatizzare e standardizzare una serie di aspetti del D-LAKE compreso la security di S3.
- RedShift Spectrum: Servizio aggiuntivo di RedShift che permette di creare External Table sfruttando i file salvati in S3.
- BMC Control-M per la gestione dell'orchestrazione dei JOB.
- AWS SageMaker come alternativa a DataIku.

Al momento è stato deciso di utilizzare come REGION Francoforte. Appena i servizi saranno disponibili nella Region di Milano si valuterà una migrazione.
Oltre ai servizi indicati vengono utilizzati i servizi AWS CloudWatch per alcune metriche di monitoraggio e AWS Backup per il backup dei servizi critici.

3.2. Tier 0 – Staging – Raw Data Zone

Il Tier 0 è l'area destinata ad accogliere i dati in ingresso nella loro forma sorgente, senza modifiche rispetto al tracciato di ingresso, così come messi a disposizione dai sistemi sorgente. Tipicamente accoglie pacchetti di informazioni che rappresentano variazioni.
Il meccanismo di acquisizione processa i pacchetti disponibili nel Tier 0 per alimentare il Tier 1. Gli incrementi acquisiti non vengono eliminati per garantire la completa ricostruibilità del Tier 1. (Punto aperto, eventualmente potranno essere ricompattati ad intervalli regolari per ridurre l'occupazione, da valutare metodi di trasferimento su storage a costo inferiore).
Il Tier 0 è implementato attraverso Object Storage S3.

3.3. Tier 1 – Data Lake

Il Tier 1 è la struttura che ospita il vero e proprio 'Data Lake'. Il Data Lake conserva informazioni eterogenee, allo scopo di consentire una veloce fruizione sia per alimentare il Tier 2 (livello analitico) sia allo scopo di supportare i bisogni informativi dei Citizen Data Scientists e del Business, attraverso query dirette o collegamento a Data Science Tools od Advanced Analytics Tools. Il Tier 1 sarà interrogabile attraverso strumenti sql-like oppure attraverso tools connessi direttamente al repository S3.
Il Tier 1 contiene le informazioni acquisite dal Tier 0, generalmente con un livello di integrità, qualità e struttura non inferiore rispetto alla sorgente alimentante. Si preserva la modellazione esistente se e solo se essa è relativa alla corretta modellazione relazionale delle entità coinvolte, non è necessario preservarla se la modellazione del dato è specifica del sistema sorgente o di uno use case specifico.
In questo modo il Data Lake è in grado di supportare gli use case di bisogni non noti a priori e della sperimentazione, mantenendo una informazione non meno strutturata della fonte di origine, quindi non perdendo informazione nella trasformazione. E' inoltre in grado di alimentare il Tier 2,

attraverso trasformazioni anche di struttura volte ad ottimizzare per la specifica funzione analitica richiesta.
Il Tier 1 è completamente ricomputabile dal Tier 0. (punto aperto).
Il Tier 1 è implementato attraverso Object Storage S3.

3.4. Tier 2 – Analytical Data Zone

Il Tier 2, o livello analitico, contiene dati in strutture costruite ad-hoc per supportare funzioni analitiche, ottimizzate per lo specifico uso.
Il Tier 2 supporta lo Use Case relativo a bisogni noti, derivati eventualmente dalle sperimentazioni condotte dai Data Scientist e dal Business sul Tier 1 e frutto di analisi specifiche volte alla realizzazione di reports, dashboards, algoritmi etc.
La modellazione dei dati nel Tier 2 è specifica dello use case ed anche dello strumento con cui saranno fruiti, tra le varie ottimizzazioni da considerare la riduzione della profondità storica, con una precisa finalizzazione allo use case.
Il Tier 2 è completamente ricomputabile dal Tier 1.
In ottica evolutiva il Tier 2 potrà ospitare anche modellazioni più raffinate e generali dei dati, una volta che i prototipi realizzati nelle fasi di sperimentazione abbiano raggiunto un livello di maturità adeguato.
Il Tier 2 è implementato attraverso Database Redshift.

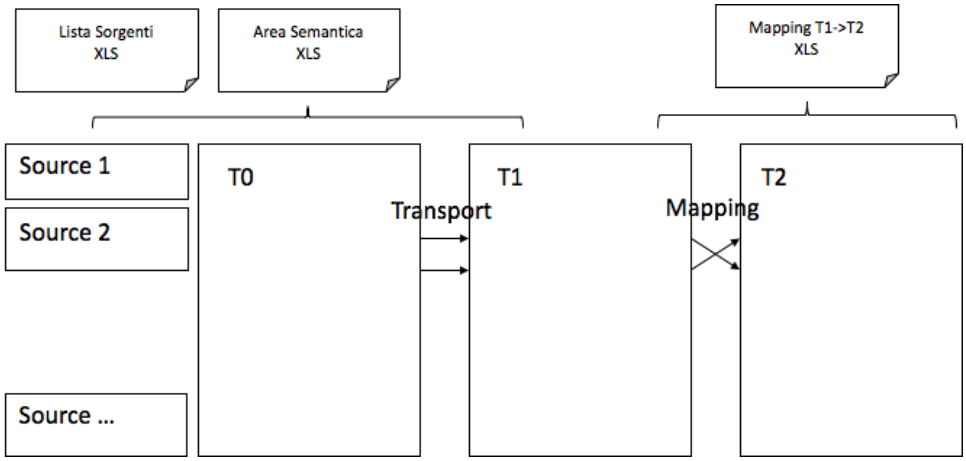
4. Standard generali

4.1.1. Note generali sul change management e annotazioni di versionamento e di release.

Ciascun documento descrittivo (ad es. i file excel che descrivono i flussi) riporta una propria versione che rappresenta un successivo avanzamento di revisione del documento. Tali versioni sono indicate con la lettera V ed un numero progressivo indicante la revisione successiva.
Ciascun elemento soggetto a rilascio (es. codice sorgente, binari, file YAML o di configurazione, DDL di tabelle, excel di documentazione etc.) viene marcato con una indicazione di release, che rappresenta una unità di rilascio coerente. Tali unità di rilascio sono indicate con la lettera R e indicano anche un numero di release major e minor (es. R1.1). Tale indicazione è da riportarsi in tutti gli strumenti di gestione e documentazione del change (es. GitLab, FlyWay etc., release notes etc.) allo scopo di avere sempre a disposizione tutti i componenti appartenenti ad un rilascio nella giusta versione allo scopo di coordinamento delle installazioni, dei rollback o dei troubleshooting.
I nomi che rappresentano oggetti software (es. file sorgente, descrizioni workflow, nomi tabelle etc.) non contengono usualmente riferimenti a release o versioni. Unica eccezione rappresentata dal nome dell'area semantica che può contenere il concetto di EDIZIONE (si veda paragrafo area semantica).

4.1.2. Documentazione flusso di acquisizione

Ciascun insieme di flussi ETL di acquisizione viene documentato attraverso un insieme di file Excel. Si definiscono tre tipologie di documenti/flussi Excel: lista sorgenti, acquisizione area semantica e mappatura su Tier 2. I tre documenti hanno standard dei nomi distinti. Ciascun documento excel avrà associata una versione, legata alla revisione del documento.
Si utilizzerà la lettera V per indicare la versione del file excel stesso
I documenti excel per le versioni pronte per il rilascio saranno gestiti su GitLab all'interno dei progetti corrispondenti. Le release saranno marcate come tag sugli excel esattamente come per il resto del codice sorgente.



4.1.3. Documentazione Lista Sorgenti

Il documento contiene una lista preliminare delle sorgenti ed elenca, in relazione al tipo di sistema sorgente, le tabelle da acquisire, i metodi di web service da chiamare, i file etc. Il documento non descrive il dettaglio dei campi ma solo la lista delle entità da acquisire a T0 e T1.
Tale documento è un input iniziale per la realizzazione di ETL di trasporto fino al T1 e per la compilazione dell'excel di documentazione area semantica. Ha la stessa granularità del documento di area semantica e cioè sistema/area semantica. Il nome del template è **Template_Lista_Sorgenti_Excel_vxx.xlsx.**

Il nome di ciascun file excel sarà:

- "LS" (Lista Sorgenti)
- Macro Area Semantica (es. Insieme coerente di aree semantiche, ad es. Applicazione sorgente del dato)
- Area Semantica (sottoinsieme semanticamente coerente di entità, ad es. Modulo applicativo)
- Versione file

Separati da "_"

Es. **LS_AUTOST_TopologiaRete_Vxx**
Es. LS_TIS_Percorrenze_Vxx

4.1.4. Documentazione area semantica

Il documento che descrive il flusso di acquisizione dell'area semantica descrive tutte la tabelle o più in generale le fonti dati sorgente da acquisire su Tier 0 e successivamente consolidare su Tier 1. Il documento contiene appunto tutte le strutture dati da acquisire all'interno di un'area semantica e di conseguenza all'interno di un singolo workflow di acquisizione. Il documento contiene anche informazioni sulle chiavi primarie, di business (naturali) ed eventuali legami relazionali. Da notare che il documento descrive mappature tra sorgenti e T0/T1 e non è un documento descrittivo delle entità. Può essere però utilizzato per pre-popolare i metadati di un sistema di catalogo.

Il documento ha la granularità del sistema/area semantica che corrisponde ad un insieme di entità trasferite che siano omogenee dal punto di vista semantico. Il documento excel corrisponde ad uno ed un solo workflow. Il nome del template è **Template_AreaSemantica_Excel_Acq_T0_T1_vxx.xlsx.**

Il nome di ciascun file excel sarà:

- "AS" (Area Semantica)
- Macro Area Semantica
- Area Semantica
- Versione file

Separati da "_"

Es. A**S_AUTOST_TopologiaRete_Vxx**
Es. AS_TIS_Percorrenze_Vxx

4.1.5. Documentazione tabelle Tier 2 e mappatura tra Tier 1 -> Tier 2.

Il documento descrive le tabelle definite sul Tier 2 (data mart) e le mappature necessarie per costruirle a partire dal Tier 1.
Il documento ha granularità relativa agli obiettivi realizzativi per cui i mapping su T2 vengono realizzati, e quindi una granularità definita dal progetto. Nel caso del progetto KPI il documento sarà tipicamente organizzato per DataMart, salvo estrapolare e mantenere in documenti separati i fogli che descrivono le mappatura per entità condivise tra più KPI ad esempio nel caso di entità dimensionali. Tale suddivisione potrà meglio essere definita in fasi avanzate del progetto, quindi è pensabile che i documenti di mappatura possano essere diversamente suddivisi anche dopo la prima stesura, estrapolando o accorpando alcune sezioni.
Da notare che il documento ha lo scopo di descrivere le mappature per consentire i trasferimenti dati tra tabelle T1 e tabelle T2 e non è un documento descrittivo delle entità. Può essere però utilizzato per pre-popolare i metadati di un sistema di catalogo.
Template_Mapping_Excel_T1_T2_vxx.xlsx

Il nome di ciascun file excel sarà:

- "DMM" (Datamart e Mapping)
- DataMartName
- Source (T1)
- Target (T2)
- Versione file

Separati da "_"

Es. **DMM_DashboardTraffico_T1_T2 _Vzz**

5. Best Practices e design patterns architetturali

5.1. Design patterns architetturali

5.1.1. Categoria ORCHESTRAZIONE (WKF)

5.1.1.1. (WKF.1) ORCHESTRAZIONE JOB ETL

Problema
All'interno dell'esecuzione di un workflow devono essere eseguiti nella corretta sequenza i job che lo compongono, in serie od in parallelo. Il fallimento di un job deve causare il fallimento dell'intero workflow.
Soluzione
Trigger Event Based in Glue.
Conseguenze
NA
Implementazione
NA
Limitazioni, applicabilità e prerequisiti
NA

5.1.1.2. (WKF.1) SCHEDULAZIONE WORKFLOW ETL

Problema
Nel rispetto delle dipendenze dei workflow occorre schedularne l'esecuzione. Il workflow è una collezione di job che devono essere eseguiti in modo atomico. L'esecuzione di un workflow fallisce se fallisce anche un solo job contenuto.
Soluzione
Trigger CRON di Glue (in attesa di integrazione CTRLM).
Conseguenze
NA
Implementazione
NA
Limitazioni, applicabilità e prerequisiti
NA

5.1.2. Categoria flussi asincroni (ASN)

5.1.2.1. (ASN.1) WORKFLOW ACQUISIZIONE DA DB RELAZIONALE SU TIER 0

Problema
Occorre acquisire all'interno del data lake i dati contenuti su un DB relazionale in modo incrementale o full.
Soluzione
Si realizza un workflow di acquisizione suddiviso in singoli job. Il workflow rappresenta l'unità atomica di esecuzione. Il workflow è composto da n job. Ciascun job acquisisce

variazioni da una ed una sola tabella. Ciascun job può essere definito come incrementale o full e riporta questa informazione necessaria in fase di consolidamento.
Il workflow rappresenta dunque l'acquisizione atomica da una collezione di tabelle che corrispondono ad un'area di acquisizione semanticamente coerente, con la limitazione di appartenere tutte allo stesso sistema sorgente.
Ciascuna esecuzione di un workflow definisce un pacchetto dati e di conseguenza un codice pacchetto che identifica univocamente la singola acquisizione atomica che ha un singolo esito, una singola last_update_date, una data di inizio della preparazione e fine della preparazione.
Ciascun workflow è composto da n job. Ciascun job esegue le attività necessarie ad acquisire le variazioni provenienti da una singola tabella o l'intera tabella. Non si definiscono dipendenze tra i job che possono quindi eseguire in parallelo.
Constraint enforcement: in fase di acquisizione delle righe possono essere verificati i constraint relativi alla singola entità (tabella) ad esempio obbligatorietà, domini etc. Il comportamento in relazione al fallimento di questa verifica è configurabile. Di default il fallimento della verifica di Foreign Key, setta a NULL il valore non mandando in errore il job.
Un workflow termina correttamente solo se trasporta tutte le variazioni dal precedente istante di acquisizione fino all'istante corrente per ciascuna delle tabelle coinvolte, di conseguenza il fallimento di un singolo job determina il fallimento dell'intero workflow.

Conseguenze

…

Implementazione

Una tabella di log/packet contiene lo stato e l'esito dell'acquisizione di un workflow con il relativo codice pacchetto, oltre che la data di ultimo aggiornamento dell'acquisizione (LAST_UPDATE_DATE) che consente di fissare l'istante per la prossima acquisizione incrementale. La data di last update da utilizzarsi nell'acquisizione è quella dell'ultimo pacchetto andato a buon fine per il workflow in oggetto.
Ciascuna riga di dettaglio acquisita riporta il codice del pacchetto con la quale è stata acquisita ed un flag 'tipo operazione'.
Per il popolamento del Tier successivo si procede acquisendo i pacchetti chiusi con successo e non già acquisiti, in ordine di D_INI_PRE.
Il tipo operazione può assumere solo due valori I/U, D. I/U è il default. Il tipo operazione determina l'azione da effettuare sul Tier successivo.
I job FULL sono riconoscibili dai job incrementali attraverso una parametro di configurazione.

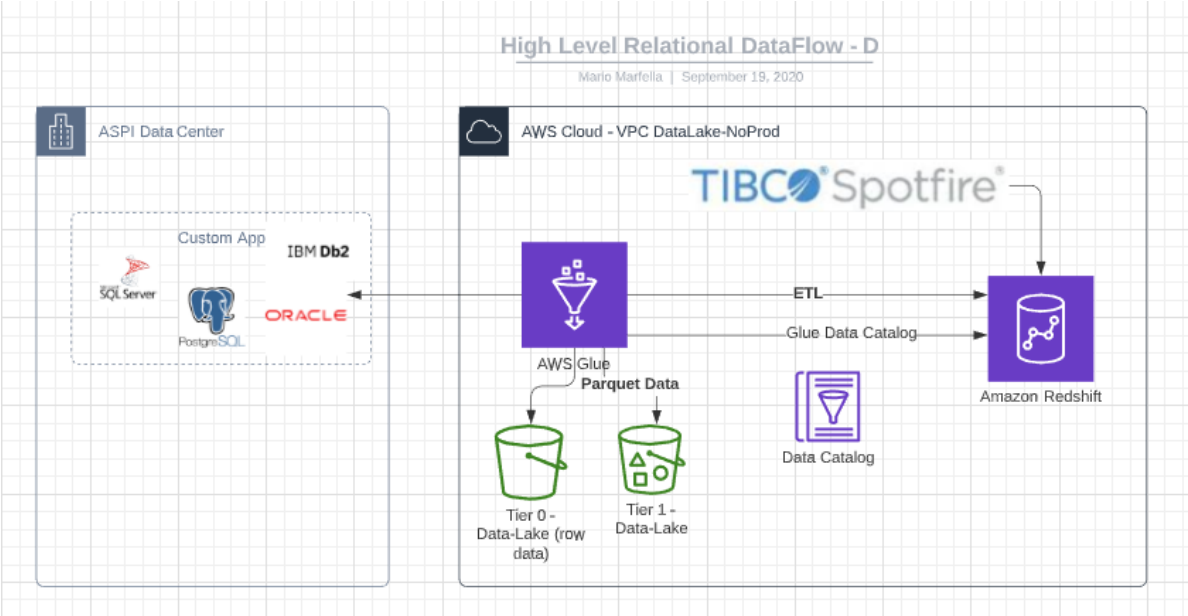
I pacchetti non andati a buon fine devono essere cancellati periodicamente da un job asincrono di clean up.

Limitazioni, applicabilità e prerequisiti

Per acquisizioni di tipo incrementale si assume che il sistema relazionale fornisca un modo per identificare le sole righe variate, tipicamente si tratta di un campo per ciascuna riga che registra il timestamp dell'ultima variazione.
Per acquisizione di tipo full il flusso acquisisce l'intera tabella.
La definizione di full o incrementale può avvenire a livello di singolo job, questa informazione è propagata al Tier successivo per permettere il corretto consolidamento.
Si assume che l'area semanticamente coerente identificata da un workflow non abbia dipendenze cicliche con altri workflow. Il grafo delle dipendenze tra le aree identificate deve essere rappresentabile come un albero.

Condizioni di blocco o fallimento:

Il fallimento di un singolo job determina il fallimento dell'intero workflow.
Il pattern è testato e validato nelle seguenti condizioni:
L'attuale implementazione delle librerie comprende la sola verifica delle Foreign Key.
Il job di clean-up non è attualmente implementato.



Esempi noti:
Flussi acquisizione traffico

5.1.2.2. (ASN.3) FLUSSI ACQUISIZIONE DA FILE INCREMENTALI O FULL A TIER 0

Problema
Occorre acquisire all'interno del data lake i dati contenuti su file in modo incrementale o full.
Soluzione
Si realizza un workflow di acquisizione suddiviso in singoli job. Il workflow rappresenta l'unità atomica di esecuzione. Il workflow è composto da n job. Ciascun job acquisisce file da uno ed un sola cartella S3.
Ciascun job del workflow può essere definito come incrementale o full.
Il workflow rappresenta dunque l'acquisizione atomica di una collezione di file che corrispondono ad un'area di acquisizione semanticamente coerente, con la limitazione di appartenere tutti allo stesso bucket di origine/sorgente.
Ciascun workflow è composto da n job. Ciascun job esegue le attività necessarie ad acquisire o tutti i file presenti nella cartella sorgente oppure, nel caso di configurazione incrementale, solo i file non ancora acquisiti.
Un workflow termina correttamente solo se trasporta tutte le variazioni dal precedente istante di acquisizione fino all'istante corrente, di conseguenza il fallimento di un singolo job determina il fallimento dell'intero workflow.
Conseguenze
…
Implementazione

Una tabella di log/packet contiene lo stato e l'esito dell'acquisizione di un workflow con i relativi codici pacchetto, oltre che le date di ultimo aggiornamento dell'acquisizione (last touch date dei file) che consente di fissare l'istante per la prossima acquisizione incrementale.
Ciascuna riga di dettaglio acquisita riporta il codice del pacchetto con la quale è stata acquisita.
Per il popolamento del Tier successivo si procede acquisendo i pacchetti chiusi con successo e non già acquisiti, in ordine di D_INI_PRE.
Il tipo operazione può assumere solo due valori I/U, D. I/U è il default. Il tipo operazione determina l'azione da effettuare sul Tier successivo.
I job FULL sono riconoscibili dai job incrementali attraverso una parametro di configurazione.

Limitazioni, applicabilità e prerequisiti
Per acquisizioni di tipo incrementale si assume che i file vengono acquisiti sulla base della touch date.
La definizione di full o incrementale può avvenire a livello di singolo job, questa informazione è propagata al Tier successivo per permettere il corretto consolidamento.
Si assume che l'area semanticamente coerente identificata da un workflow non abbia dipendenze cicliche con altri workflow. Il grafo delle dipendenze tra le aree identificate deve essere rappresentabile come un albero.
Condizioni di blocco o fallimento:
Il fallimento di un singolo job determina il fallimento dell'intero workflow.
Il pattern è testato e validato nelle seguenti condizioni:
L'attuale implementazione delle librerie comprende la sola verifica delle Foreign Key.
Il job di clean-up non è attualmente implementato.

Esempi noti:
Flussi acquisizione dati CASH PLAN

5.1.2.3. (ASN.4) WORKFLOW ACQUISIZIONE DA APP FLOW SU TIER 0

TBD

5.1.3. Sistemi di consolidamento (CONS)

5.1.3.1. (CONS.1) FLUSSI CONSOLIDAMENTO DA TIER 0 A TIER 1 CON COERENZA RELAZIONALE

Problema
Si deve popolare il Tier 1 con i dati già acquisiti su Tier 0, mantenendo le caratteristiche di qualità stabilite.
Soluzione
Per popolare il livello 1 (Tier 1) si devono consolidare all'interno del Data Lake (Tier 1) tutti i dati depositati nella RAW Data Zone e non ancora acquisiti.
Per ciascun workflow e ciascun job del Tier 0 esiste il corrispondente da Tier 0 a Tier 1.
Il workflow acquisisce tabella per tabella ed in step successivi provvede alla valorizzazione di chiavi surrogate e di legami relazionali se presenti.
Al termine dell'operazione il workflow effettua un merge delle tabelle su Tier 1 armonizzando inserimenti, update e delete nel caso di acquisizione incrementale o una rigenerazione della tabella nel caso di FULL.

Conseguenze
TBD
Implementazione
Ciascuna tabella viene acquisita separatamente, all'interno di un'area di pipeline temporanea organizzata per step.
I vari step di avanzamento sono gestiti in aree temporanee da Spark.
Ciascuno step sposta i pacchetti acquisiti da un'area alla successiva. Al termine dell'acquisizione di un intero workflow si eseguono i controlli di FK e le valorizzazioni delle PK se necessari e se specificati nel documento di mappatura.
Il fallimento di ciascuno dei job dei singoli step causa il fallimento del workflow.
Al termine con successo dei controlli si spostano le tabelle acquisite in area Data Lake.
Quando l'operazione è terminata con successo si aggiorna la tabella di log con D_FIN_PRE e STATUS=OK per il workflow id corrispondente, last update date aggiornata e codice pacchetto workflow acquisito.
Acquisizione dal tier 0 avviene prendendo tutti i pacchetti su T0 (andati a buon fine) con d_fin_pre > last_updated_date ultimo pacchetto andato a buon fine su Tier 1. In questo modo ignoriamo le righe di pacchetti falliti o ancora in corso di acquisizione.
Limitazioni, applicabilità e prerequisiti
TBD
Esempi noti

5.1.3.2. (CONS.2) FLUSSI DI MAPPATURA DA TIER 1 A TIER 2

Problema
Si deve popolare il Tier 2 con i dati già acquisiti su Tier 1, trasformando il dato in modo da rispettare il disegno delle tabelle destinazione. Il Tier 2, o livello analitico, contiene dati in strutture costruite ad-hoc per supportare funzioni analitiche, realizzazione di reports e dashboards, ottimizzate per lo specifico uso.

Soluzione
Per popolare il livello 2 (Tier 2) si trasformano i dati contenuti all'interno del Data Lake (Tier 1) conformandoli alle strutture definite nel Tier 2.
I workflow hanno la granularità derivata dalle tabelle target su Tier 2. Tipicamente i workflow saranno suddivisi in workflow che popolano le dimensioni (anagrafiche) da quelli che popolano le tabelle dei fatti (tabelle denormalizzate che registrano misure).
Il workflow acquisisce tabella per tabella di destinazione ed in step successivi provvede alla valorizzazione di chiavi surrogate e di legami relazionali se presenti.
Al termine dell'operazione il workflow effettua un merge delle tabelle su Tier 2 armonizzando inserimenti, update e delete nel caso di acquisizione incrementale o una rigenerazione della tabella nel caso di FULL.

Implementazione
Si implementa Un workflow per Area corrispondente ai workflow dei Tier precedenti per le Dimensioni, ed un workflow per una o più tabelle dei fatti. Un job per tabella destinazione nel DM, un job per ciascuna tabella dei fatti (in un WF dedicato), un job per ciascuna tabella delle dimensioni (in un WF per Area).
Limitazioni, applicabilità e prerequisiti
Il pattern è in grado di supportare le seguenti trasformazioni:

Unione tra tabelle, Join tra tabelle, Filtraggio basato su condizioni, Aggregazione e calcolo di funzioni aggregate, trasformazione di tipo dei campi, Distinct, Split tabelle basato su condizioni, calcolo funzioni basato su dati della riga. Non è adatto per trasformazioni algoritmiche complesse che richiedono implementazioni ad hoc.
Esempi noti

5.1.4. Change management e versionamento (VER)

5.1.5. Pattern di supporto (TOOL)

5.2. Progettazione flussi

5.2.1. Area semantica

L'area semantica costituisce un insieme di entità, coerenti dal punto di vista semantico, che verranno trasferite in modo atomico all'interno del data lake (l'atomicità sarà per area semantica/sistema originante, in alcuni casi può essere ulteriormente disaggregata ad esempio nel caso di gruppi di flussi a frequenza/volume molto disomogenei). L'area semantica è versionata per consentire il change e l'evoluzione. Il sistema di versionamento è ad una cifra. La cifra viene scattata quando l'area semantica subisce modifica rilevanti e non retrocompatibili. Ciascuna versione viene denominata EDIZIONE. Nel **nome** dell'area semantica si indica l'edizione con la convenzione Exx e solamente per edizioni successive alla 1.
E' possibile la coesistenza di più edizioni della stessa area semantica, per i periodi di tempo che servono all'adeguamento delle trasformazioni/mapping e/o dashboard legacy. Due edizioni di area semantica sono completamente indipendenti, e corrispondono a workflow diversi. Un area semantica senza indicazione di edizione è da considerarsi in edizione 1, dunque E1.

5.2.2. Flussi Source -> Tier 0

I flussi di acquisizione da source a Tier 0 sono organizzati per area semantica. In generale si privilegia la velocità di acquisizione su Tier 1, di conseguenza una volta identificate le sorgenti si acquisiscono le tabelle per intero.

5.2.3. Flussi Tier 0 -> Tier 1

I flussi di acquisizione da Tier 0 a Tier 1 sono speculari rispetto a quelli di acquisizione sul Tier 0. Per ciascun oggetto/entità acquisita sul Tier 0 si realizza un job di acquisizione per il Tier 1. I vincoli relazionali vengono mantenuti per i soli vincoli presenti già sulla sorgente. Questa modalità di definizione consente un approccio 'agile' al popolamento del Tier 1 che non richiede ulteriori analisi se non la definizione delle sorgenti e degli oggetti (siano essi tabelle, file, web services) da acquisire.

5.2.4. Flussi Tier 1 -> Tier 2

I flussi di acquisizione da Tier 1 a Tier 2 trasformano il dato in modo da conformarlo alle strutture di tabelle disegnate sul Tier 2. Tale disegno sarà funzionale alla fruizione del dato attraverso

report/dashboards. Tale disegno avverrà incrementalmente passando attraverso una sperimentazione effettuata direttamente sul Tier 1 fino ad una forma definitiva. Tale forma potrà essere ulteriormente raffinata attraverso opportune attività di refactor che analizzino l'intero insieme di dashboards realizzate alla ricerca di possibili 'fattorizzazioni' e riduzione quindi della ridondanza informativa in Tier 2. Tale refactor sarà retrospettivo dopo che le prime versioni di report/dashboard saranno state sperimentate con gli utenti.

5.2.5. Change Management

Per modifiche retrocompatibili ad un'area semantica si raccomanda la modifica dei flussi in essere senza la definizione di aree con diversa edizione.
In caso di cambiamenti non retrocompatibili il meccanismo del versionamento consente di lasciare intatti gli ETL preesistenti gestendo una nuova edizione parallela. Una volta pianificato l'aggiornamento dei flussi legacy la vecchia versione potrà essere rimossa.
Per ogni altro cambiamento retro compatibile si utilizza lo stesso nome di area semantica.
Ai fini della nomenclatura ogni area semantica priva della indicazione di edizione si intende alla Edizione 1.

5.3. Regole di nomenclatura

Di seguito si descrivono le regole per la nomenclatura di tutti gli oggetti coinvolti nello sviluppo di integrazioni.
Valgono le generali indicazioni riportate nel documento di standard ASPI
ITS_ST_BDL01_Rev1.9_2019_Standard_Nomenclatura_BancheDati.pdf

5.3.1. Regole per la nomenclatura delle tabelle e degli schemi.

5.3.1.1. Tier 0

5.3.1.1.1. Nome Tabella livello T0 STG

Se source DB
- Schema ID0
- Schema Source_tabella Source

Es. ID0.TISA_TTAN06_CAR_TEL_AV

Se Source xls
- Schema ID0
- TIDL
- XXX progressivo 3 cifre
- Triplette descrittive

Es. ID0.TIDL001_RFX

5.3.1.1.2. Nome Colonne tabella

Tier 0 Staging →lo stesso nome del dato source

5.3.1.2. Tier 1

5.3.1.2.1. Nome Tabella livello T1 DWH

Se source DB

- Schema ID1
- Schema Source_tabella Source

Es. ID1.TISA_TTAN06_CAR_TEL_AV

Se Source xls

- Schema ID1
- TIDL
- XXX progressivo 3 cifre
- Triplette descrittive

Es. ID1.TIDL001_RFX

5.3.1.2.1. Nome Colonne tabella

Tier 1 DWH →lo stesso nome del dato source

5.3.1.3. Tier 2

5.3.1.3.1. TBD Nome Tabella livello T2 DM

- Schema ID2
- TIDL
- D/F Dimensione/Fatto
- XXX progressivo 3 cifre
- Triplette descrittive

Es.

ID2.TDLD001_SOC_R02
ID2.TIDLF001_KM_TRF

5.3.1.3.2. Nome Colonne tabella

Tier 2 DM → triplette descrittive come da std ASPI

5.3.1.4. Colonna PK di Tier 1 e TIer2

Tripletta come Chiave Logica (se single column) o come nome tabella + "_PK"

Es: Tabella Tipo Pagamento
Chiave Logica = C_TIP_PAG
Chiave surrogata = C_TIP_PAG_PK

Es: Tabella km percorsi per Tratta Elementare
Chiave logica composta da n colonne
Chiave surrogata = C_KM_TRT_ELE_PK

5.3.1.5. Colonna FK di Tier 1 e TIer 2

Tripletta come fk logica (se single column) o come nome tabella + "_FK"

Es: Tabella Tipo Pagamento vs Tabella Società
Colonna = C_SOC_EMI
Colonna FK surrogata = C_SOC_EMI_FK

5.3.1.6. Script DDL creazione tabella T2 DM

- V
- Mjr Version"."MinVersion (per tutte le tabelle a parità di Major) (Una major per ogni deploy)
- __schema
- _nometabella
- .sql

Es. V0.14__ID2.TIDLF010_KM_TRF_MEN.sql

5.3.2. Regole per la nomenclatura dei workflow

5.3.2.1. Workflow di trasferimento/trasformazione del dato

Il meccanismo di trasferimento e trasformazione all'interno del datalake prevede che un'area semantica per sistema venga processata in modo atomico (processata completamente o non trasferita affatto). Il workflow può comprendere n job, ciascun dedicato ad una singola entità/tabella. I job all'interno di un workflow possono essere incrementali o full, in modo indipendente tra loro. Ciascun workflow trasporta ed eventualmente trasforma i dati da uno stadio al successivo (Source->T0, T0->T1, T1->T2). Ciascun workflow è realizzato secondo uno dei pattern esposti nel presente documento.

5.3.2.2. Nomenclatura Workflow Source->T0 e T0->T1

Di seguito lo standard di nomenclatura di un workflow:
Macro Area Semantica_Area Semantica_SistemaOrigine_Target

Es. AUTOST**_TopologiaRete_db2_to_T0**
Es. TIS_Percorrenze_db2_to_T0
Es. TIS_Percorrenze_T0_to_T1

5.3.2.3. Nomenclatura Workflow T1->T2 per tabelle dimensionali/anagrafica

Di seguito lo standard di nomenclatura di un workflow:
DataMartName_Macro AreaSemanticaOrigine_AreaSemanticaOrigine_SistemaOrigine_Target

Es. **DashboardTraffico_AUTOST_TopologiaRete_T1_to_T2**

5.3.2.4. Nomenclatura Workflow T1->T2 per tabelle dei fatti

Di seguito lo standard di nomenclatura di un workflow:

DataMartName_FactTable_Source_Target

Es. **DashboardTraffico_TrafficoMensile_T1_T2**

5.3.2.5. Workflow tecnici/tools

TBD

5.3.3. Nomenclatura Folder S3 del Tier 0

Tier0_Staging: contiene i dati del Tier 0 di staging, generalmente si tratta di variazioni acquisite da applicare sul Tier 1

- Inbound
  - **Sistema Origine**
    - **Macro Area coerente** XXX (macro area semantica coerente, **opzionale**)
      - **Area MMM.SSS_[Exx]** (**Area semantica**, categoria di dettaglio dell'area XXX, derivata da nome excel lista sorgenti, eventualmente versionata per consentire l'evoluzione se E>1)
        - **Tablename** (folder contenente i file che rappresentano la tabella)
        - Processing (area di lavoro per i workflow/job glue)
      - WorkflowsMetadata (area contenente metadati dei workflow, informazioni di stato di esecuzione e last update degli stati di avanzamento processazione.
    - Area coerente YYY

Es. DB2\TopologiaRete\ID0.TTAN06_CAR_TEL_AV

- Outbound
  - **Macro Area coerente XXX** (macro area semantica coerente, opzionale)
    - **Area XXX.1** (Area semantica, categoria di dettaglio dell'area XXX)
      - Processing (area di lavoro per i workflow/job glue)
    - WorkflowsMetadata (area contenente metadati dei workflow, informazioni di stato di esecuzione e bookmark degli stati di avanzamento processazione).
  - Area coerente YYY

5.3.1. Nomenclatura Folder S3 del Tier 1

Tier1_Datalake: contiene i dati consolidati per il livello datalake

- **Macro Area coerente SSS** (macro area semantica coerente, opzionale), contiene, organizzate per aree/sub-area le tabelle su S3, in formato **parquet**, del livello datalake.
    - **Area MMM.SSS_[Exx]** (**Area semantica**, categoria di dettaglio dell'area XXX, derivata da nome excel lista sorgenti ed eventualmente versionata per consentire l'evoluzione se E>1)
        - **Tablename** (folder contenente i file che rappresentano la tabella)
        - Processing (area di lavoro per i workflow/job glue)
    - WorkflowsMetadata (area contenente metadati dei workflow, informazioni di stato di esecuzione e last update degli stati di avanzamento processazione).
- Area coerente YYY

Es. TopologiaRete\ID0.TTAN06_CAR_TEL_AV
oppure
TopologiaRete_E2\ID0.TTAN06_CAR_TEL_AV

6. Appendice A. Formati Standard

6.1. Standard di formato file CSV

- Numerici:
  - i decimali sono separati dal PUNTO
  - Non sono ammessi separatori di migliaia
  - NO notazioni esponenziali

- Date:
  - date in formato "yyyy-MM-dd" (opzione Preferenziale)
  - date in formato "yyyy-MM-dd HH:mm:ss" (opzione Alternativa, altre alternative da concordare i sede di analisi)
  - timestamp in formato "yyyy-MM-dd HH:mm:ss.xxx" (Preferenziale, in alternativa il formato sarà concordato i sede di analisi)

- Separatore:
  - § (opzione preferenziale)
  - ;#; (opzione alternativa)

- Caratteri di controllo:
  - Nessuna carattere di controllo all'interno dei campi testo (es. "/r /n") (Alternativa non esportare il campo)
  - Carattere fine riga tipo Windows "/n"

- Tracciato Record:
  - Tracciato record fisso
  - Posizione colonne vincolante
  - Riga di intestazione non necessaria ma ammissibile

- Nome file:
  - Radice fissa
  - Possibile parte variabile per gestione versioni (es: dataora 20200918_1421)
  - Caratteri ammissibili lettere, numeri, e i caratteri "-" e "_"

- Encoding file:
  - UTF-8

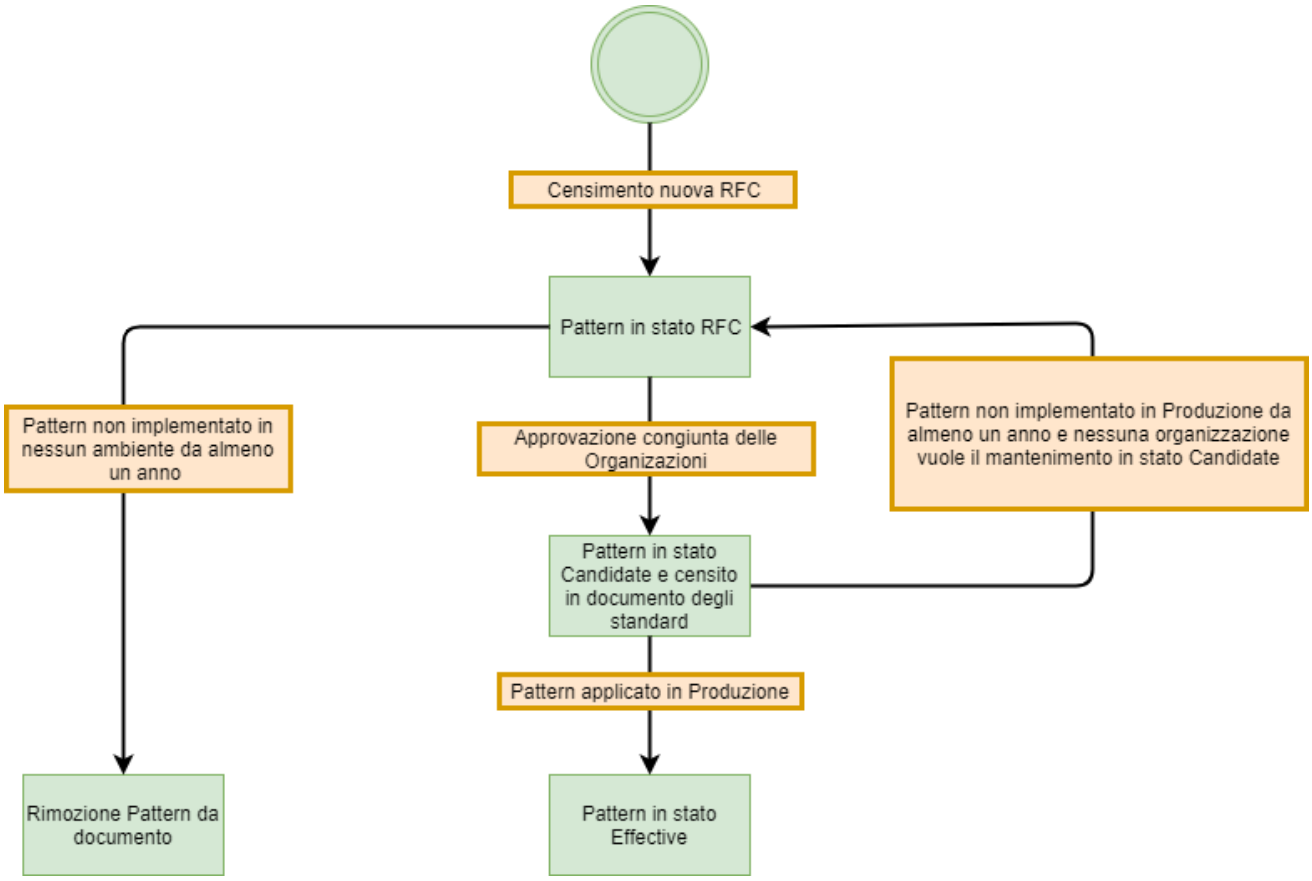
7. Appendice B. RFC

7.1. Template RFC

Data registrazione RFC
<data di registrazione della RFC>
Candidato per:
<Tipologia Pattern>
Soluzione implementata
<sì/no>
Problema
<Descrizione del problema che si intende risolvere>
Soluzione
<Soluzione al problema che si intende risolvere>
Esempio
<Esempio dell'applicazione della soluzione al problema>
Implementazione
<Modalità di implementazione>
Limitazioni di default:
Esempi noti
<Elenco di esempi di flussi che implementano la soluzione descritta operativi >

7.2. Ciclo di vita di una RFC

Il censimento di nuovi pattern avverrà attraverso la proposta di una RFC che le organizzazioni dovranno approvare congiuntamente.
Le RFC tracciano le soluzioni tecniche non promosse a standard anche se implementate sotto forma di soluzioni ad-hoc .
Di seguito viene descritto il ciclo di vita di una RFC.



1. Censimento nuova RFC per risoluzione problema riscontrato
2. Una RFC approvata congiuntamente dalle organizzazioni diventa candidate
3. I pattern candidate implementati in produzione diventano effective
4. I pattern candidate non implementati in produzione vengono riportati ad RFC dopo un anno a meno dell'esplicita richiesta delle organizzazioni di mantenerli a candidate
5. Le RFC non implementate in nessun ambiente vengono eliminate dopo un anno

7.3. Censimento RFC

7.3.1. (RFC.1) xxxxxxx

Data registrazione RFC:
Candidato per: TOOL
Soluzione implementata : No
Problema
Soluzione

Conseguenze
Implementazione
Limitazioni ed applicabilità
Esempi noti