

Sistema informativo

Un **sistema informativo** è un insieme strutturato di procedure e di risorse umane e materiali finalizzate alla raccolta, archiviazione, elaborazione e comunicazione di dati al fine di ottenere le **informazioni** necessarie ad un'organizzazione per la gestione delle proprie attività.

Sistema informativo

E' necessario disporre di adeguate informazioni per due motivi:

operativo: riguarda la gestione dei dati necessari all'esercizio delle attività di un'organizzazione e da cui derivano informazioni di servizio (es. per un'azienda: fatturazione delle merci e paghe dei propri dipendenti - es. per un comune: anagrafe dei cittadini).

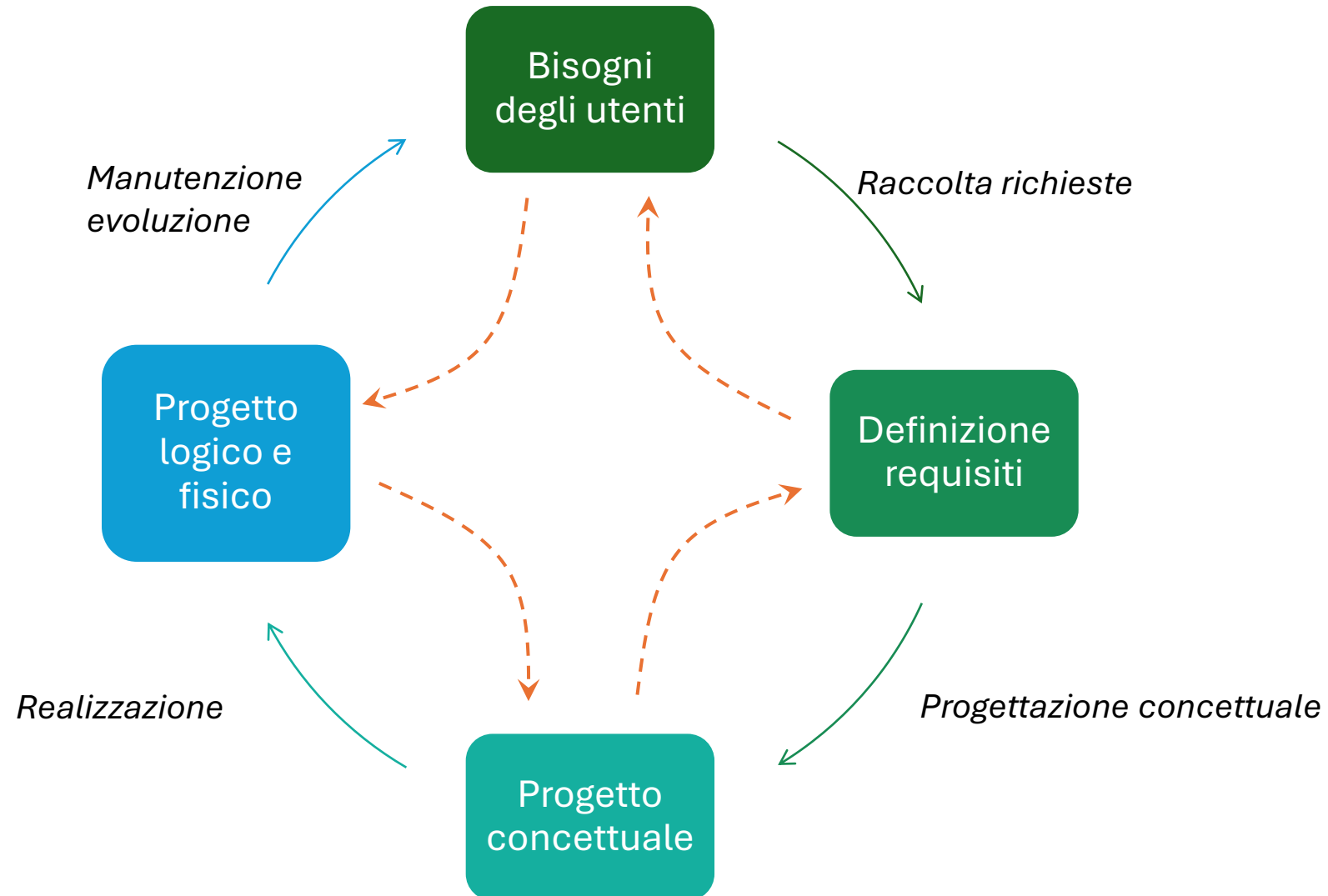
decisionale: riguarda la gestione di informazioni di governo finalizzate a prendere decisioni relative ad attività di programmazione controllo e valutazione(es. per un'azienda: decidere il prezzo di un prodotto - es. per un comune: potenziare alcuni servizi).

Sistema informatico

Raccogliere archiviare ed elaborare dati per gestire e comunicare informazioni è una necessità che ha sempre caratterizzato le attività umane. Il concetto di sistema informativo è indipendente dagli strumenti utilizzati per trattare le informazioni che esso gestisce. Nel tempo ne sono stati utilizzati tra i più disparati: dalle incisioni su pietra ai più moderni computer.

Un **sistema informatico** è il sottoinsieme di un sistema informativo dedicato al trattamento automatico di informazioni derivanti dalla gestione di dati archiviati in formato digitale.

Ciclo di vita di un sistema informatico



Gestione dei dati

File system

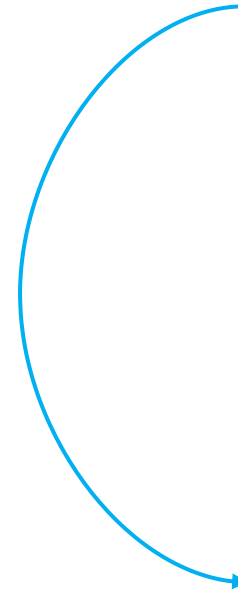
Componente del sistema operativo che gestisce l'organizzazione e l'accesso ai file su memoria di massa. Un file system fornisce la mediazione fra il modello logico dei file presentato all'utente e l'accesso fisico necessario per leggere e scrivere sul supporto di memorizzazione tipicamente un file system presenta un'organizzazione gerarchica dei file.

DataBase Management System

Sistema software in grado di gestire grandi collezioni di dati integrate, condivise e persistenti in modo efficiente ed efficace assicurando affidabilità e privacy.

Database

Collezione di dati gestita da un database.

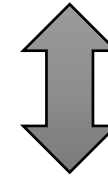
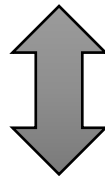
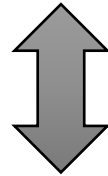


Approccio file system

*software
applicativo*



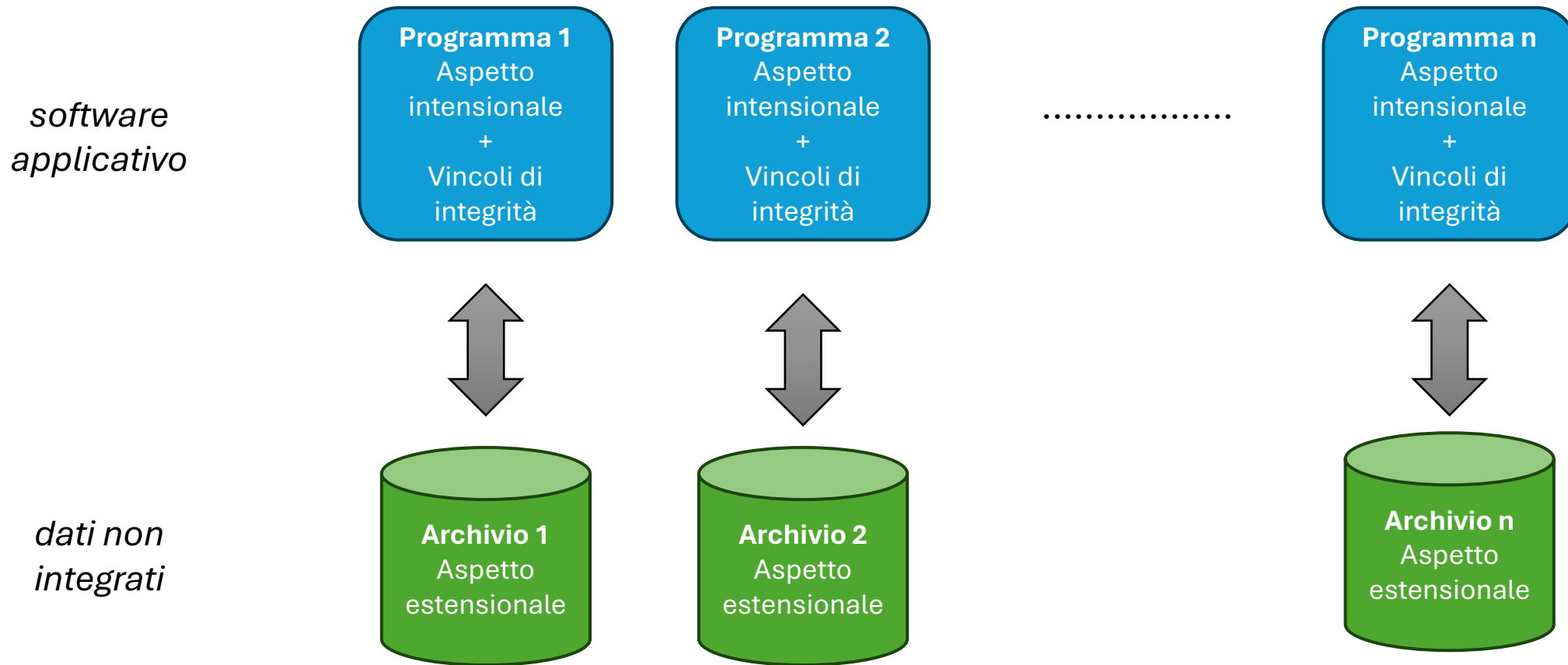
.....



*dati non
integrati*



Approccio file system

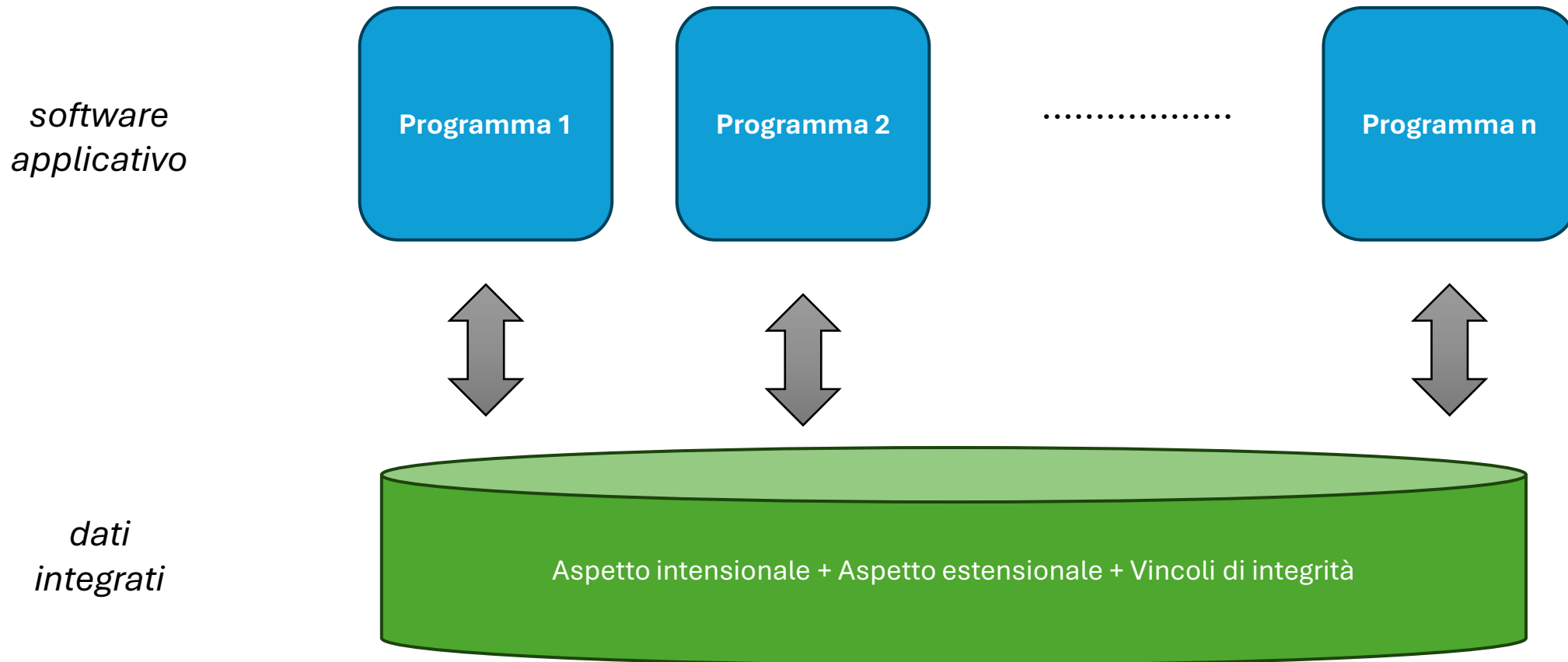


duplicazione?

significato degli archivi?

modifica della struttura dei dati?

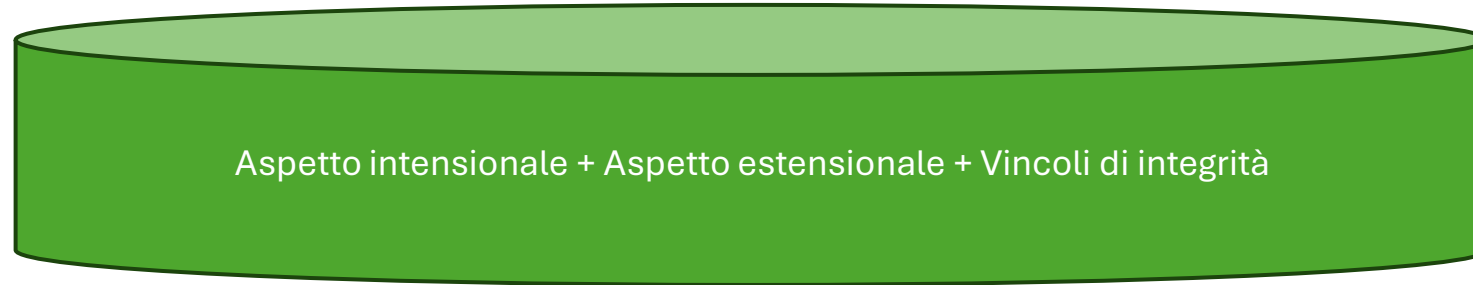
Approccio DBMS



*Integrazione dei dati...
indipendenza logica e fisica...
integrità....*

Approccio DBMS

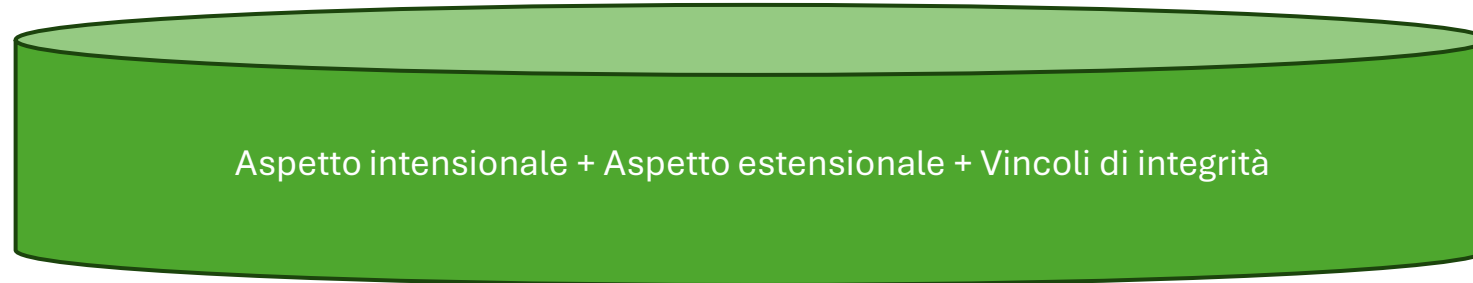
*dati
integrati*



Integrazione dei dati: un DB è un insieme di dati strutturati e permanenti memorizzati senza ridondanze superflue insieme al loro aspetto intensionale, ai legami che li caratterizzano e ai vincoli di integrità a cui debbono soddisfare in modo tale da rendere chiaro il modello della realtà informatizzata.

Approccio DBMS

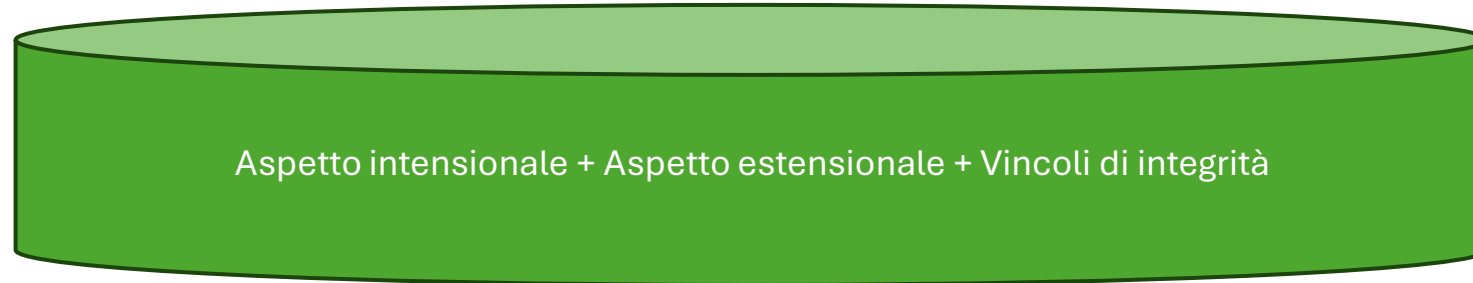
*dati
integrati*



Indipendenza logica: *i programmi applicativi non contengono la definizione della struttura dei dati, ma fanno riferimento ad essa attraverso i nomi dei campi che la caratterizzano: la struttura logica dei dati può essere modificata senza dover intervenire sul codice applicativo.*

Approccio DBMS

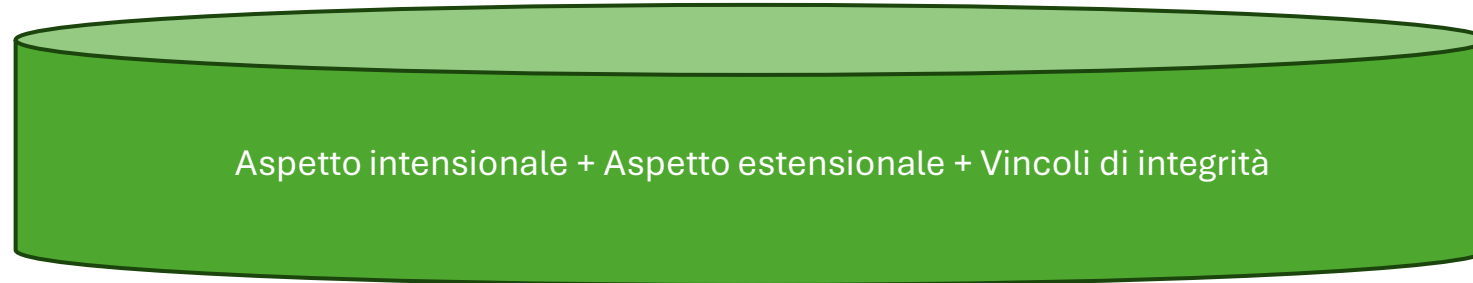
*dati
integrati*



Indipendenza fisica: la definizione della struttura dei dati astrae dalla sua implementazione fisica permettendo la modifica di quest'ultima in modo trasparente rispetto alla prima e ai programmi applicativi

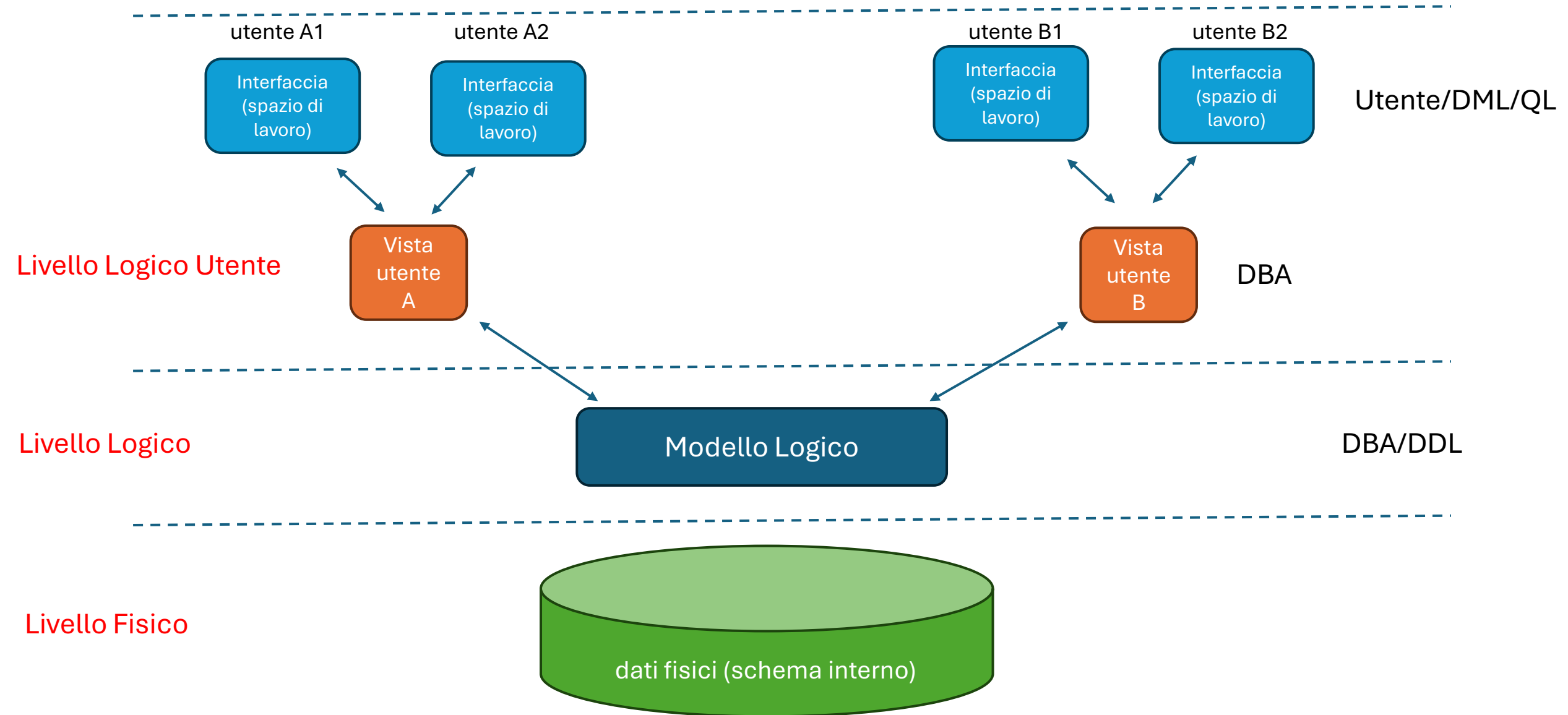
Approccio DBMS

*dati
integrati*



Integrità: è il DBMS e non le procedure di gestione a verificare che i dati vengano inseriti e modificati nel rispetto dei vincoli di integrità preventivamente fissati.

Architettura di un DBMS



Database e transazioni

Un aspetto qualificante di un DBMS è dato dal numero di "utenti" (applicazioni) che possono accedere simultaneamente a una base di dati. I DBMS attuali sono in genere "*multiutente*" e supportano accessi concorrenti al database.

Le operazioni di interrogazione dei dati in multiutenza comportano problematiche di tipo prestazionale risolte dal fatto che il DBMS utilizza sofisticate tecniche di ottimizzazione delle interrogazioni; anche la scelta adeguata dell'hardware influisce su questo tipo di prestazione.

Le operazioni (inserimento, modifica, eliminazione) che trasformano le basi di dati devono evitare che l'interazione contemporanea di più "utenti" con la base di dati induca il rischio potenziale che si possa determinare uno stato incoerente dei dati; esistono per questo degli opportuni meccanismi, come per esempio le transazioni, che gestiscono queste situazioni potenzialmente invalidanti.

Database e transazioni

incoerenza dei dati.....

Ridondanza dei dati, cioè gli stessi dati compaiono più volte all'interno dell'archivio

Incongruenza (portata dalla ridondanza), nel caso in cui lo stesso dato sia aggiornato in un archivio e non in un altro;

Inconsistenza (portata dall'incongruenza), cioè i dati a disposizione non sono più affidabili, perché non si sa in modo certo quale dei diversi valori sia quello corretto.

Database e transazioni

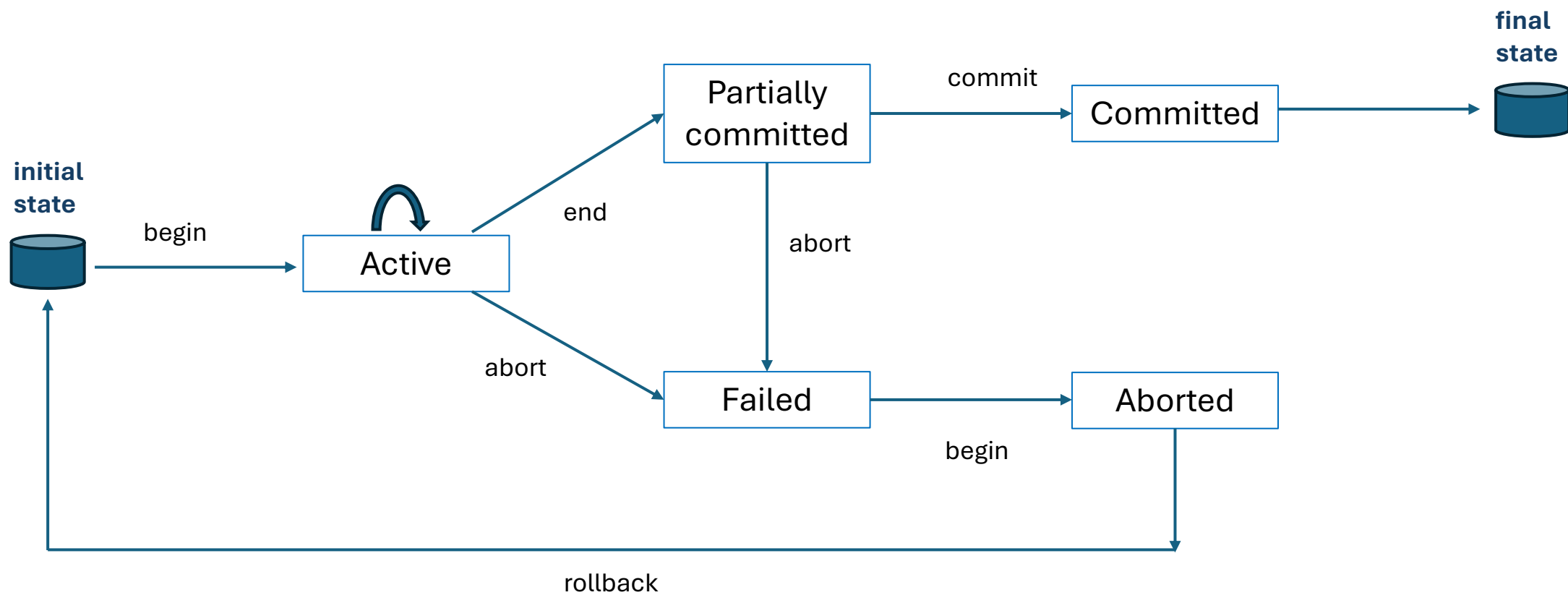
Transazione

Unità logica di elaborazione nel corso della quale una base di dati viene trasformata da uno stato iniziale consistente ad uno stato ad uno stato finale consistente.

Un'unica transazione può comprendere una o più operazioni di modifica dei dati e deve automaticamente ripristinare il suo stato iniziale se c'è anche una sola di tali operazioni che non ha successo.

Un'applicazione può essere vista come una sequenza di transazioni intervallate dall'esecuzione di operazioni non relative ai dati.

Database e transazioni



A.C.I.D.

le quattro proprietà fondamentali che deve soddisfare un DBMS che implementa le transazioni.....

Atomicity (atomicità)

Le operazioni di una transazione sono eseguite in modo "atomico": tutte o nessuna

Consistency (consistenza)

La transazione trasforma il database da uno stato consistente a un altro stato consistente

Isolation(isolamento)

Una transazione è indipendente dalle altre (transazioni concorrenti non si influenzano)

Durability (persistenza)

Registrazione permanente dell'effetto della transazione dopo il commit

...altre caratteristiche importanti dei DBMS

Dimensione

Devono prevedere una gestione dei dati in memoria secondaria, con architetture molto articolate e sofisticate.

Condivisione

Applicazioni(utenti) devono poter accedere, secondo opportune modalità, a dati comuni (*controllo di concorrenza*)

Persistenza

I dati permangono nel tempo, anche dopo la terminazione dell'applicazione

Affidabilità

Capacità di conservare intatto il contenuto della base di dati in caso di malfunzionamento hardware o software, in caso di evento catastrofico o in caso di errore umano(*backup e recovery*).

...altre caratteristiche importanti dei DBMS

Privatezza

Ciascun utente, opportunamente riconosciuto, viene abilitato a svolgere solo determinate azioni sui dati (*autorizzazione*).

Efficienza

Insieme di risorse (tempo e spazio) accettabile da tutti gli utenti

Efficacia

Le attività degli utenti devono essere sempre produttive

Utenti e progettisti

Database Administrator (DBA)

Responsabile della progettazione, controllo e amministrazione della base di dati (progetto, esigenze degli utenti, prestazioni, affidabilità, autorizzazioni, tecnologia)

Progettisti e programmatori di applicazioni

Definiscono e realizzano i programmi che accedono alla base di dati (DML + Java, C#, Python, php, Front-End,...)

Utenti

Utilizzano la base di dati per le proprie attività

Concludendo

- I DBMS permettono di considerare i dati come una risorsa comune di un'organizzazione, a disposizione di tutte le sue componenti (con opportune forme di controllo)
- La base di dati fornisce un modello unificato e preciso della parte del mondo reale di interesse per l'organizzazione, utilizzabile nelle applicazioni attuali e, con possibili estensioni, in applicazioni future
- Con l'uso di un DBMS è possibile un controllo centralizzato dei dati
- La condivisione permette di ridurre ridondanze e inconsistenza
- L'indipendenza dei dati, caratteristica fondamentale di DBMS favorisce lo sviluppo di applicazioni più flessibili e facilmente modificabili
- Applicazioni con uno o pochi utenti senza necessità di accessi concorrenti e relativamente stabili nel tempo possono essere realizzati più proficuamente con file ordinari piuttosto che con DBMS