UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

Stili architetturali

Stili architetturali (o pattern architetturali)

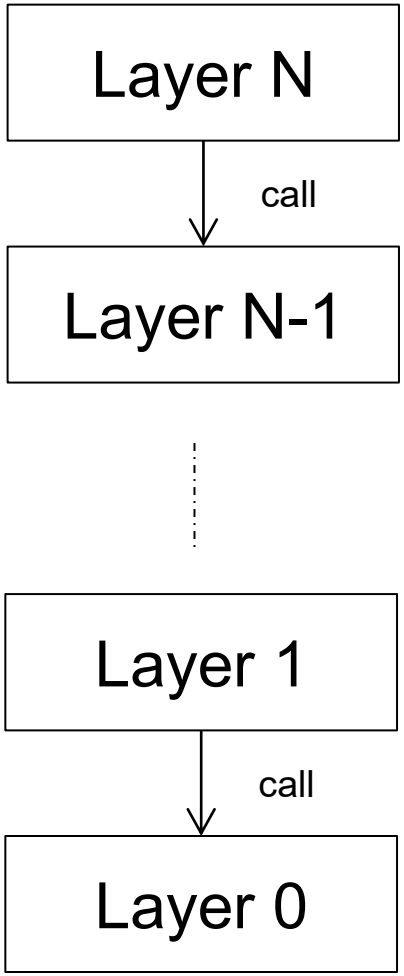
- Un modello generale di architettura che può essere usato come punto di partenza per il system design
  - L'architetto personalizza e aggiunge dettagli sulla base dei requisiti

Stili architetturali software

- Layered
- Client-Server
- N-tier
- REST
- Model-View-Controller (MVC)
- Model-View-Presenter (MVP)
- Pipe and filter
- Repository
- Blackboard
- Peer-to-Peer
- Publish-Subscribe
- Plugins
- Microservices
- Serverless
- …

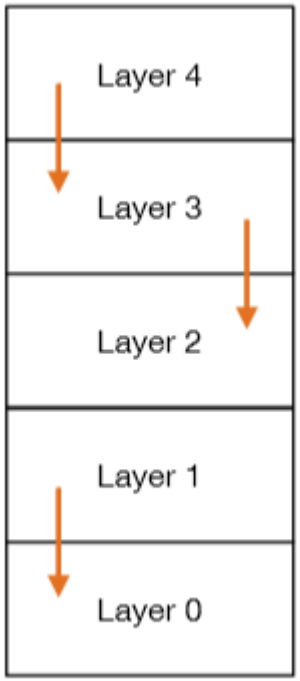
Layered

- Ogni strato espone un'interfaccia (API) che è utilizzata da un componente dello strato superiore
  - Ogni strato corrisponde a un livello di astrazione che nasconde i suoi segreti
- Vantaggi: le modifiche ai livelli superiori non si propagano ai livelli inferiori
- Svantaggi: prestazioni degradate in presenza di molti strati



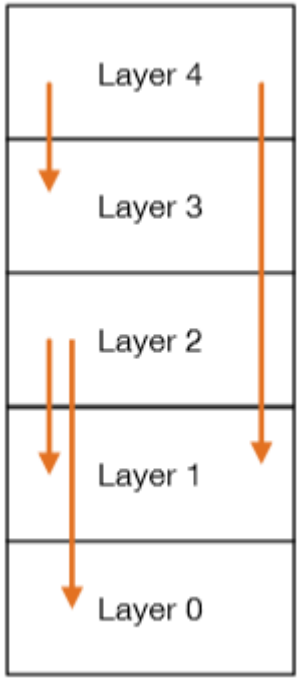
Stratificazione stretta

- Si possono invocare solo operazioni dello strato immediatamente inferiore
- Es. Modello ISO/OSI



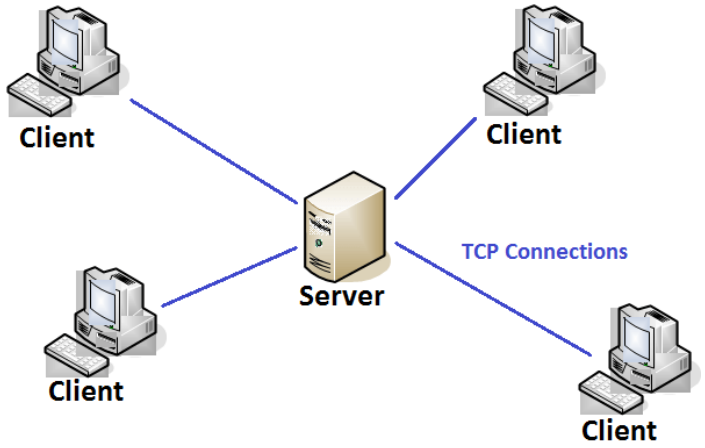
Stratificazione lasca

- Si possono invocare operazioni di qualsiasi strato inferiore
- Es. TCP/IP stack



Client-server

- Variante dello stile layered per sistemi distribuiti:
  - strati su 2 tipi di nodi: client e server
- Il client inizia la comunicazione spedendo una richiesta al server

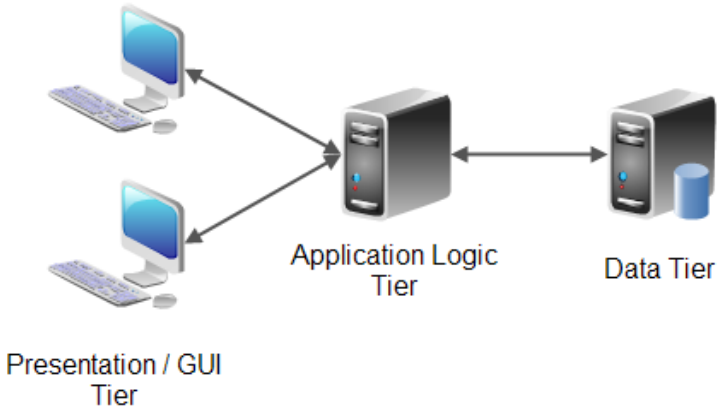


- Il server è in attesa di essere contattato e risponde alle richieste
  - Il server gestisce le risorse condivise (es. database)
- Vincoli: nessuna comunicazione diretta tra client

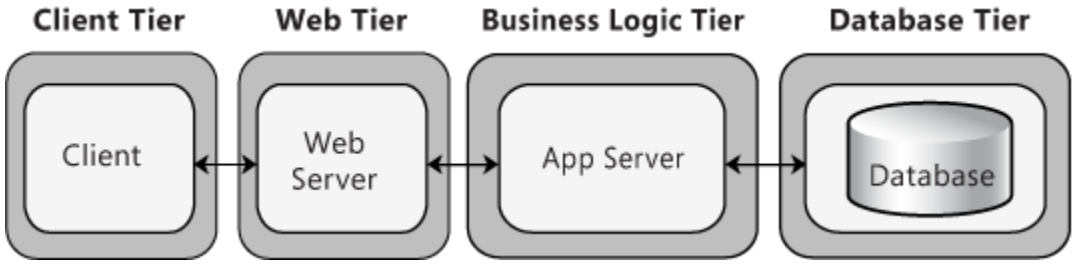
N-tier

- Variante dello stile layered per sistemi distribuiti:
  - strati localizzati su N nodi

3-tier per applicazioni gestionali



4-tier per applicazioni web



REpresentational State Transfer (REST)

- API accessibile mediante i metodi di HTTP

  POST, GET, PUT, DELETE
  equivalgono a

  CREATE, READ, UPDATE, DELETE
  applicati a una risorsa o a una collezione di risorse

- Risorse identificate da URI

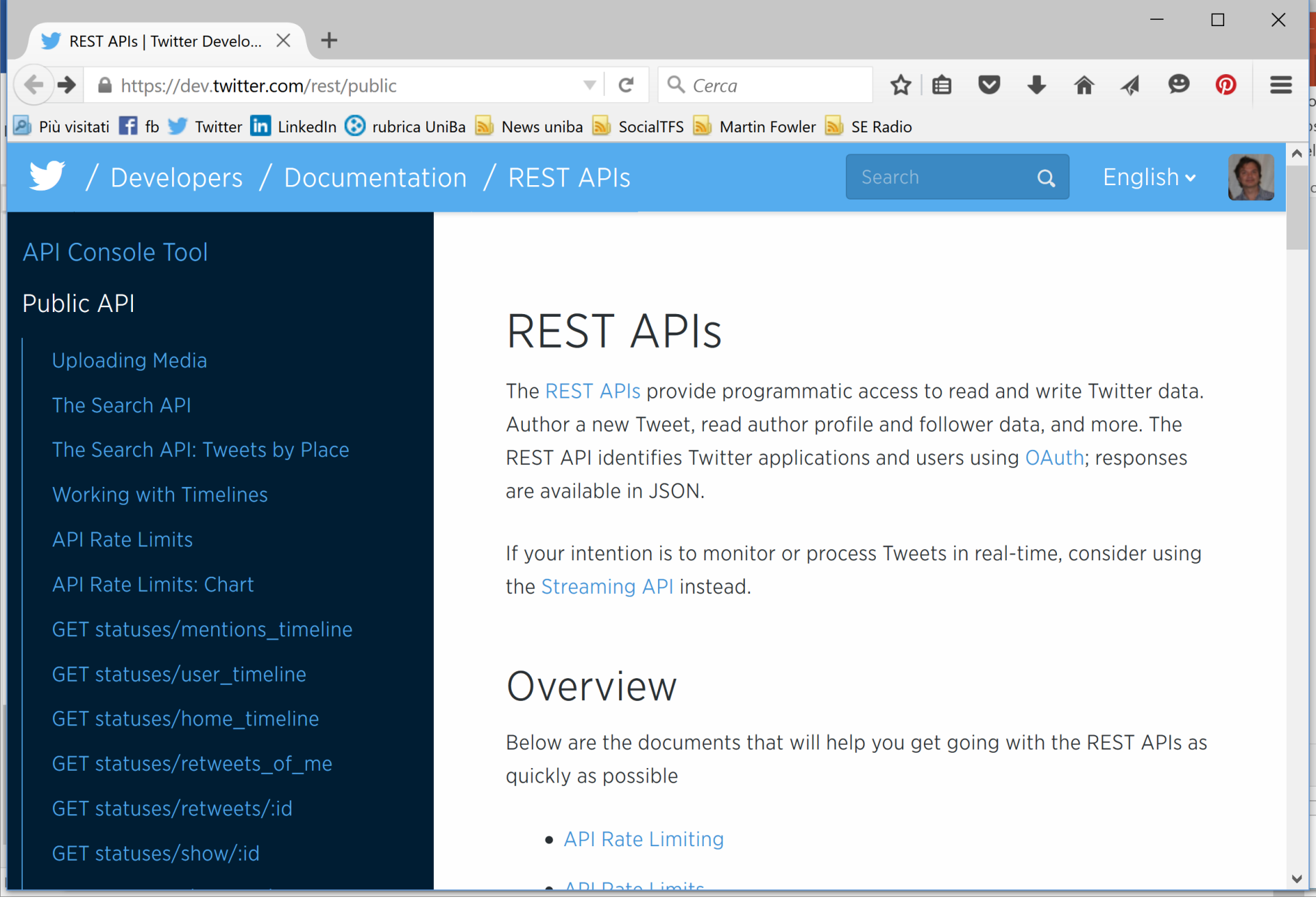
  http://example.com/customers/

  http://example.com/orders/2007/10/

  http://example.com/products/4554

- Stato delle risorse restituito in un formato negoziabile tra client e server
  - JSON, XML

Roy Fielding, *Architectural Styles and the Design of Network-based Software Architectures,* PhD thesis, 2000

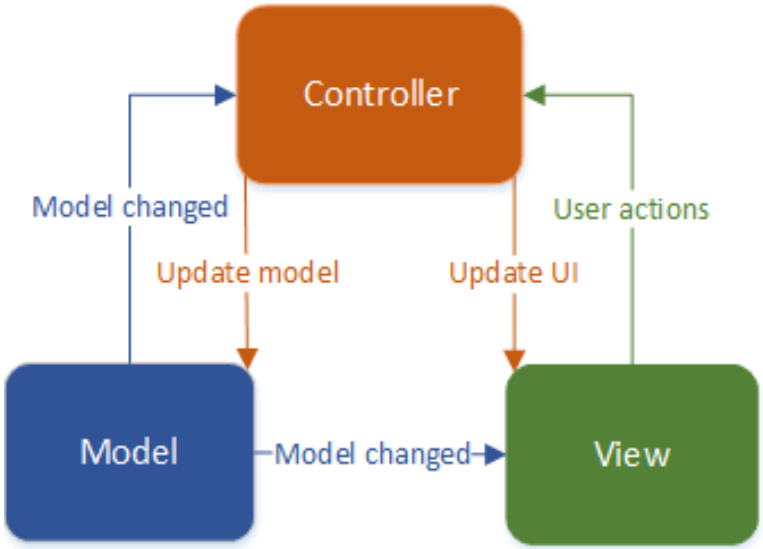

REST APIs | Twitter Develo...
https://dev.twitter.com/rest/public
Cerca
Più visitati
fb
Twitter
LinkedIn
rubrica UniBa
News uniba
SocialTFS
Martin Fowler
SE Radio
/ Developers / Documentation / REST APIs
Search
English
API Console Tool
Public API
Uploading Media
The Search API
The Search API: Tweets by Place
Working with Timelines
API Rate Limits
API Rate Limits: Chart
GET statuses/mentions_timeline
GET statuses/user_timeline
GET statuses/home_timeline
GET statuses/retweets_of_me
GET statuses/retweets/:id
GET statuses/show/:id
REST APIs
The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are available in JSON.
If your intention is to monitor or process Tweets in real-time, consider using the Streaming API instead.
Overview
Below are the documents that will help you get going with the REST APIs as quickly as possible
API Rate Limiting

about or private to the currently authenticated user.

Supported Operations	list	insert	update	delete
activity	✓	✓	⊘	⊘
caption	✓	✓	✓	✓
channel	✓	⊘	⊘	⊘
channelBanner	⊘	✓	⊘	⊘
channelSection	✓	✓	✓	✓
comment	✓	✓	✓	✓
commentThread	✓	✓	✓	⊘
guideCategory	✓	⊘	⊘	⊘

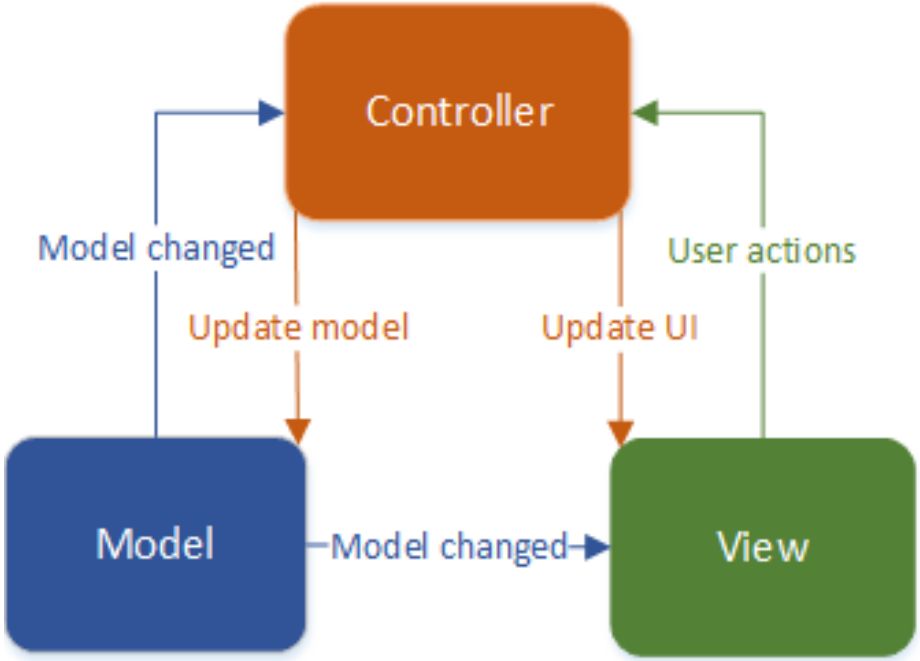
Model-View-Controller (MVC)

- Per sistemi interattivi
- Separa la logica di presentazione dei dati (View) dalla logica di business (Model)
- Implementato da OO framework
  - Spring per Java
  - Laravel per PHP
  - Rails per Ruby
  - Django per Python
  - ASP.NET MVC per C#
  - AngularJS per JavaScript



MVC: i tre sottosistemi

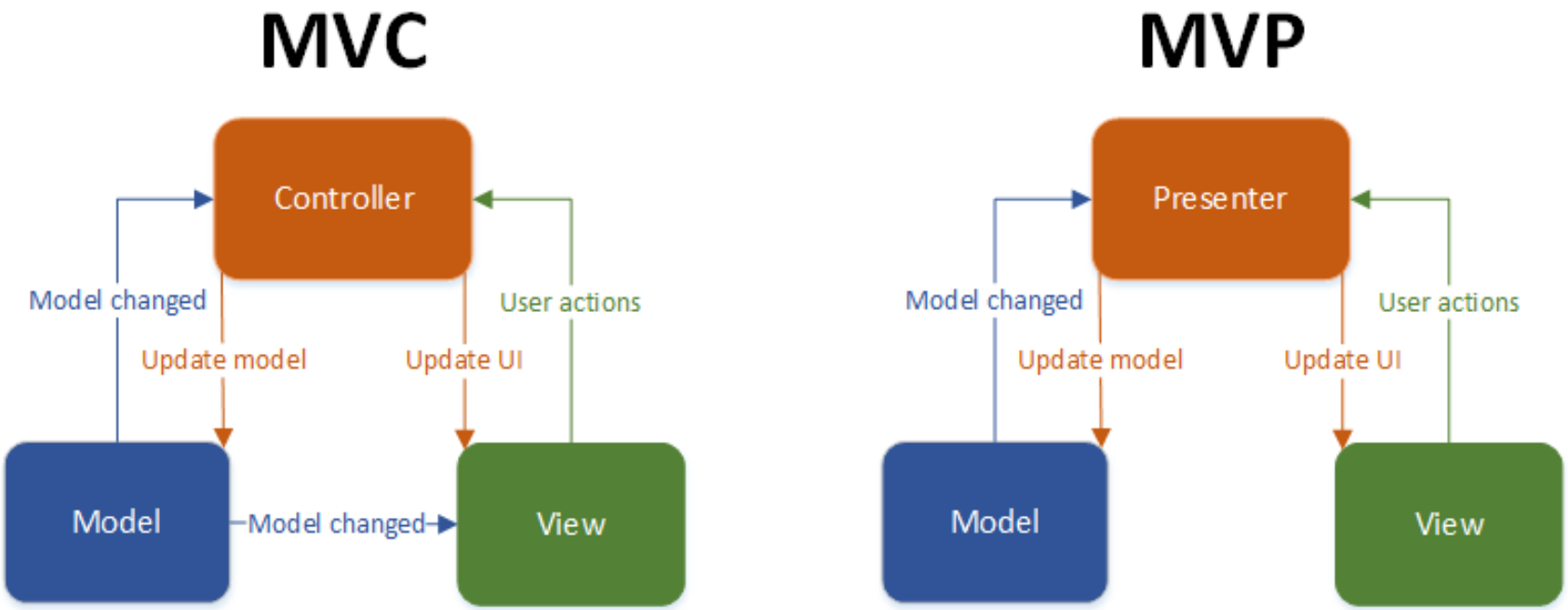
- Model
  - Include oggetti che rappresentano la conoscenza del dominio dell'applicazione
  - Include i metodi per l'accesso ai dati
- View
  - Include la rappresentazione visuale dei dati
  - Sono possibili viste multiple
- Controller
  - Gestisce la sequenza di interazioni con l'utente
  - Accetta l'input e lo trasforma in comandi per il modello o la vista
    - Es. Elaborazione richieste REST
  - Media tra gli oggetti del Model e le viste (View) che li rappresentano



Model-View-Presenter (MVP)

- Variante di MVC in cui
  - la View non sa niente del Model
  - il Model non notifica eventi alle viste

di fatto Entity-Boundary Control



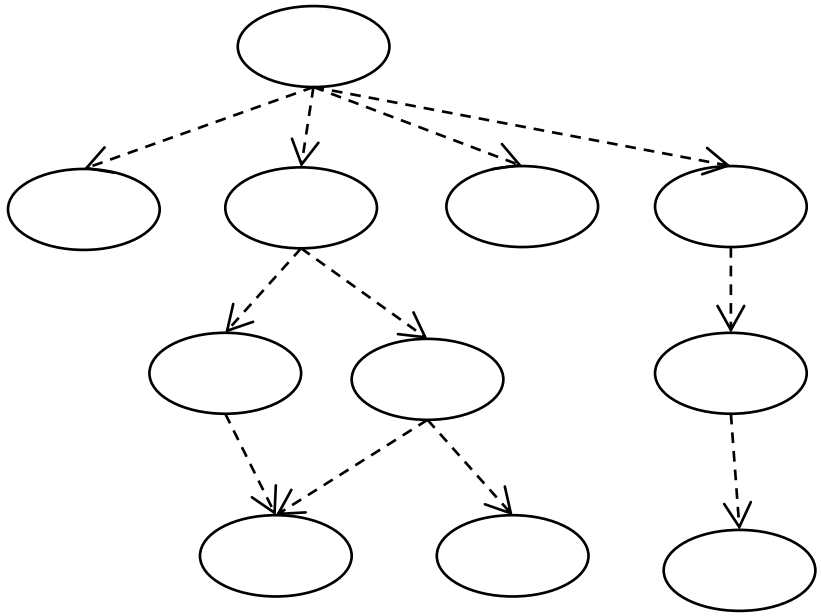
Pipe and filter

- I sottosistemi elaborano i dati ricevuti in input e mandano l'output in input ad altri sottosistemi
  - **Filter** è un sottosistema (inclusi programmi indipendenti)
  - **Pipe** è un'associazione tra sottosistemi (realizzata dal sistema operativo o da un tool)
  - **Pipeline** è una sequenza lineare di filtri
- Es. UNIX shell
  - cat esami.txt | grep "superato" | wc -l

Microservice architecture

A collection of services in which

- each service is an **independent unit of deployment**
- each service **communicates** with other services **through service interfaces**
  - usually, **REST**ful **API**s
- each service provides a small amount of functionality
- the total functionality of the system is derived from composing multiple services



A user request may activate tens of services (e.g. 70 in LinkedIn)