

**ARGONET SRL**

**INTRODUZIONE A:**

# **API Platforms**

1.0



Prepared by Claudio Greppi

# CONTESTO

---

- La trasformazione digitale richiede la costruzione di ecosistemi interoperabili.
  - Sistemi eterogenei, gestiti da organizzazioni indipendenti, comunicano tra loro per scambiare informazioni e collegare i rispettivi processi, dando vita a processi totalmente nuovi.
  - Si è assistito negli ultimi anni alla nascita e crescita di una vera e propria “API Economy” in cui aziende ed enti hanno progressivamente reso disponibili i propri dati e servizi mediante API, per consentire a sviluppatori di terze parti di creare nuovi nuovi servizi e nuovi canali di vendita e promozione

# DEFINIZIONE

---

- Una piattaforma API è una Organizzazione che mette in relazione due o più realtà indipendenti, attraverso un servizio (Application Programming Interface - API), consumabile da applicativi software, che rende possibile l'interazione automatizzata fra queste realtà.
- L'Organizzazione in questo modo espone dati e/o funzioni che diventano la modalità principale con cui i soggetti terzi interagiscono con l'organizzazione.

# ECOSISTEMA

## Una delle idee alla base dell'approccio "API platform":

- abilitare e supportare talenti e forza lavoro esterna all'organizzazione;
  - creare un Ecosistema a valore aggiunto consentendo al mondo di "creare ciò che il mondo vuole"

## Esempio - Facebook:

- gli sviluppatori in tutto il mondo usano le API Facebook per creare nuovi servizi (es giochi) e con questo contribuiscono alla crescita di valore (e dati) di Facebook.

## Vantaggio competitivo:

- canale di comunicazione; rafforzamento del brand
- possibilità di raccogliere un grande volume di dati

# ANATOMIA DI UNA API (REST)

---

- Una API può essere vista come un insieme di punti di interscambio, che possono essere contattati da sistemi terzi con regole ben definite. Questi punti di interscambio (endpoint) sono indirizzi web che possono essere acceduti con il protocollo standard del web: http (o meglio https).
- Il codice che implementa le API traduce il modello dati interno del software – spesso complesso e di competenza degli esperti della singola piattaforma - in operazioni di lettura e scrittura aperte all'esterno secondo un modello opportunamente pensato per gli utilizzatori terzi

# ANATOMIA DI UNA API (REST)

Esempio 1/2

A titolo di esempio, un endpoint potrebbe restituire un elenco di eventi memorizzati in un sistema gestionale di eventi.

Un qualsiasi sistema esterno, interessato a ricevere l'elenco degli eventi, potrebbe fare nel nostro esempio una richiesta http di questo tipo [http GET]:

<https://gestionale-eventi.it/eventi>

Questa richiesta http(s) identifica:

- il sistema da contattare: <https://gestionale-eventi.it/>
- l'operazione da eseguire: eventi

# ANATOMIA DI UNA API (REST)

Esempio 2/2

La richiesta del metodo "eventi" della API comporterà una operazione da parte del gestionale; più precisamente eseguirà della logica interna al gestionale, ad esempio eseguirà una query sul database degli eventi, e produrrà una risposta, in forma testuale, in formato XML o JSON.

Ad esempio:

```
<?xml version="1.0" encoding="UTF-8"?>
<eventi>
  <evento>
    <id>1</id>
    <titolo>Concerto di musica
classica</titolo>
    <data>2018-02-15T10:00:00</data>
  </evento>
</eventi>
```

# ANATOMIA DI UNA API (REST)

Documentazione

- La API renderà pubblica la documentazione delle operazioni disponibili, ad esempio documenterà:
- l'esistenza dell'endpoint `"/eventi"` e
  - il formato dei dati restituiti dall'endpoint, nonché il formato dei dati da inviare alla API quando l'operazione richiesta non è di lettura, come nell'esempio precedente, ma comporta invece l'invio di dati dal sistema terzo verso l'API; ad esempio un sistema terzo abilitato a inviare nuovi eventi al gestionale.

Open API (<https://www.openapis.org/>) si sta affermando come formato standard per la documentazione delle API.

# ANATOMIA DI UNA API (REST)

Documentazione - esempio SWAGGER

## pet Everything about your Pets

**POST** /pet Add a new pet to the store

**PUT** /pet Update an existing pet

**GET** /pet/findByStatus Finds Pets by status

**GET** /pet/findByTags Finds Pets by tags

**GET** /pet/{petId} Find pet by ID

**POST** /pet/{petId} Updates a pet in the store with form data

**DELETE** /pet/{petId} Deletes a pet

**POST** /pet/{petId}/uploadImage uploads an image

# PRINCIPALI FUNZIONI DI UNA API PLATFORM

---

Slide 1 di 4

Analisi, progettazione, e testing

- Tool di progettazione
- Prototipazione e sviluppo
  - Documentazione
  - Test

# PRINCIPALI FUNZIONI DI UNA API PLATFORM

---

Slide 2 di 4

Erogazione (runtime):

- Autenticazione e autorizzazione
- Logging, analytics, monitoraggio
  - Gestione quote e limiti
  - Caching
- Trasformazione e aggregazione dati

# PRINCIPALI FUNZIONI DI UNA API PLATFORM

---

Slide 3 di 4

Supporto agli sviluppatori (esterni):

- Portale per sviluppatori
- Documentazione delle API
- Richiesta di chiavi di accesso

# PRINCIPALI FUNZIONI DI UNA API PLATFORM

---

Slide 4 di 4

Monetizzazione (scenario avanzato):

- contract & sla management
- monetizzazione dell'uso collegata alla gestione di quote

# DEPLOYMENT

---

Esistono diverse tipologie di deployment per le API platform, disponibili o meno a seconda delle scelte commerciali del produttore:

- In cloud (PaaS)
- On premise (incluso private cloud)
  - On-premise e in cloud (Ibrido)

# CONTATTI

WWW.ARGONET.IT