

italianaSoftware 



API Market



italianaSoftware nasce ed opera per innovare e rivoluzionare il mondo dell'IT portandolo verso soluzioni distribuite a microservizi.

Facciamo questo contribuendo alla crescita di Jolie, il primo linguaggio per microservizi, ed offrendo una suite di prodotti per la gestione di sistemi distribuiti orientati ai microservizi.



Jolie



*Jolie è la nostra tecnologia,
immaginata, formalizzata, sviluppata e fatta crescere in Italia.*

Jolie è un linguaggio di programmazione ed è open source.

*Jolie introduce un nuovo paradigma di programmazione,
quello orientato ai servizi.*

Jolie è un progetto internazionale



I microservizi rivoluzioneranno ogni aspetto riguardante la produzione di software: dalla progettazione allo sviluppo fino fino alla messa in esecuzione



Chi usa microservizi oggi estensivamente?

The Netflix logo, consisting of the word "NETFLIX" in white, bold, sans-serif capital letters with a black drop shadow, centered on a solid red rectangular background.The Amazon logo, featuring the word "amazon" in a lowercase, bold, sans-serif font, with a curved orange arrow underneath that starts under the 'a' and ends under the 'z'.

Il successo delle architetture a microservizi ottenuto per le infrastrutture IT di aziende pioniere ed innovative, si propagherà velocemente all'intero settore. La progettazione, lo sviluppo e la composizione di API offerte da microservizi diventerà uno standard de facto.



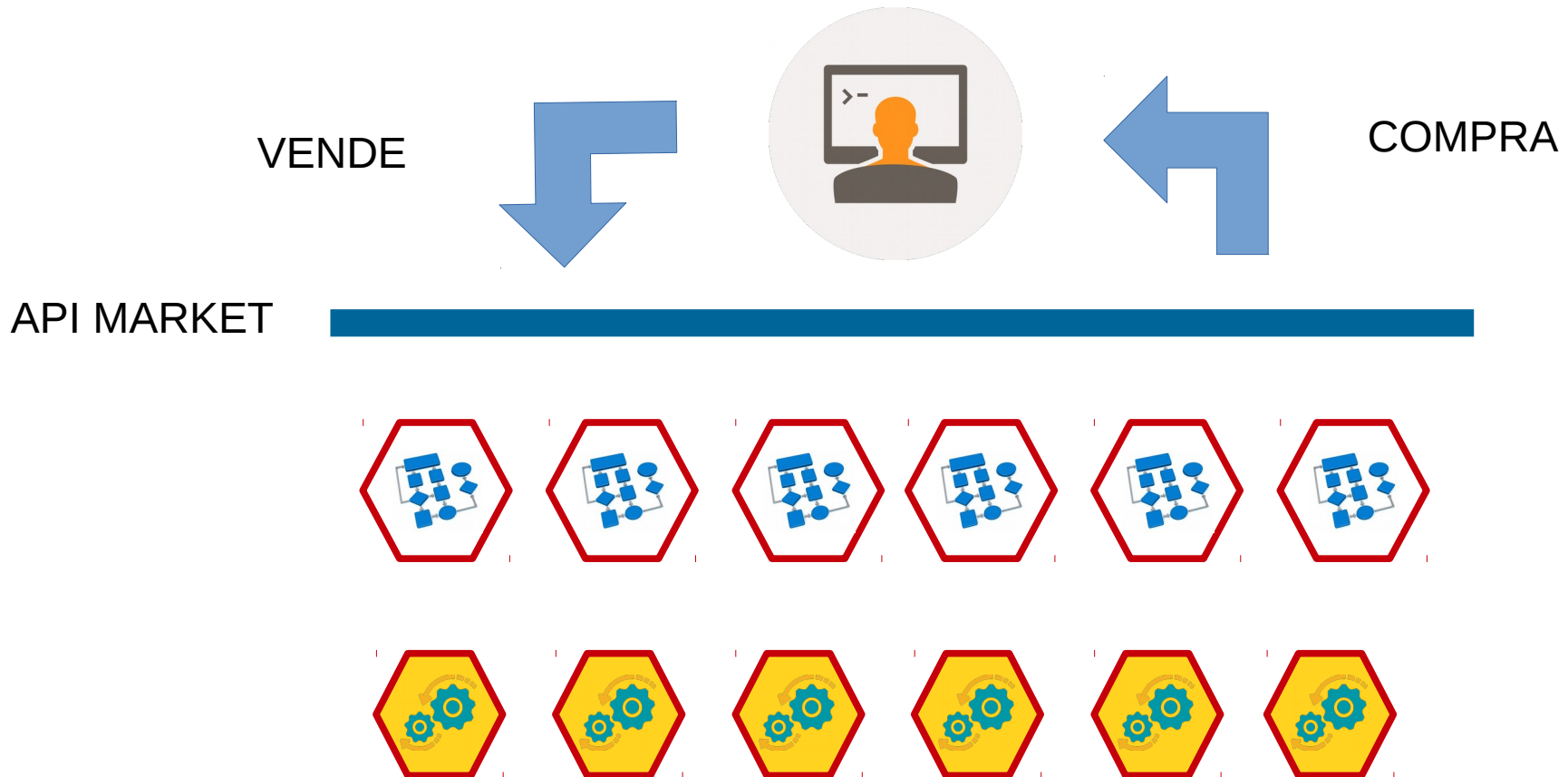
I microservizi si possono comporre



*I microservizi potranno facilmente essere composti uno con l'altro al fine di creare nuovi microservizi, e ben presto diventerà rilevante **poter acquistare e comporre API già esistenti** senza doverle implementare nuovamente*



Il progetto – un API market per microservizi



Capitolato

Obiettivi minimi:

Creare un'applicazione web che sia in grado di:

- **Registrare le API di un microservizio** attraverso la registrazione e la pubblicazione della sua interfaccia
- **Permetta di consultare le API** e la documentazione relativa
- **Permetta di associare diverse API key d'uso** per ogni API registrata nell'applicativo e per ogni utente che ne faccia richiesta
- **Permetta di monitorare l'utilizzo delle API** da parte di utenti esterni in base alle API key rilasciate
- **Blocchi le chiamate di utenti in possesso di API key** scadute e/o non valide
- **Permetta di visualizzare i dati tecnici d'uso delle singole API**



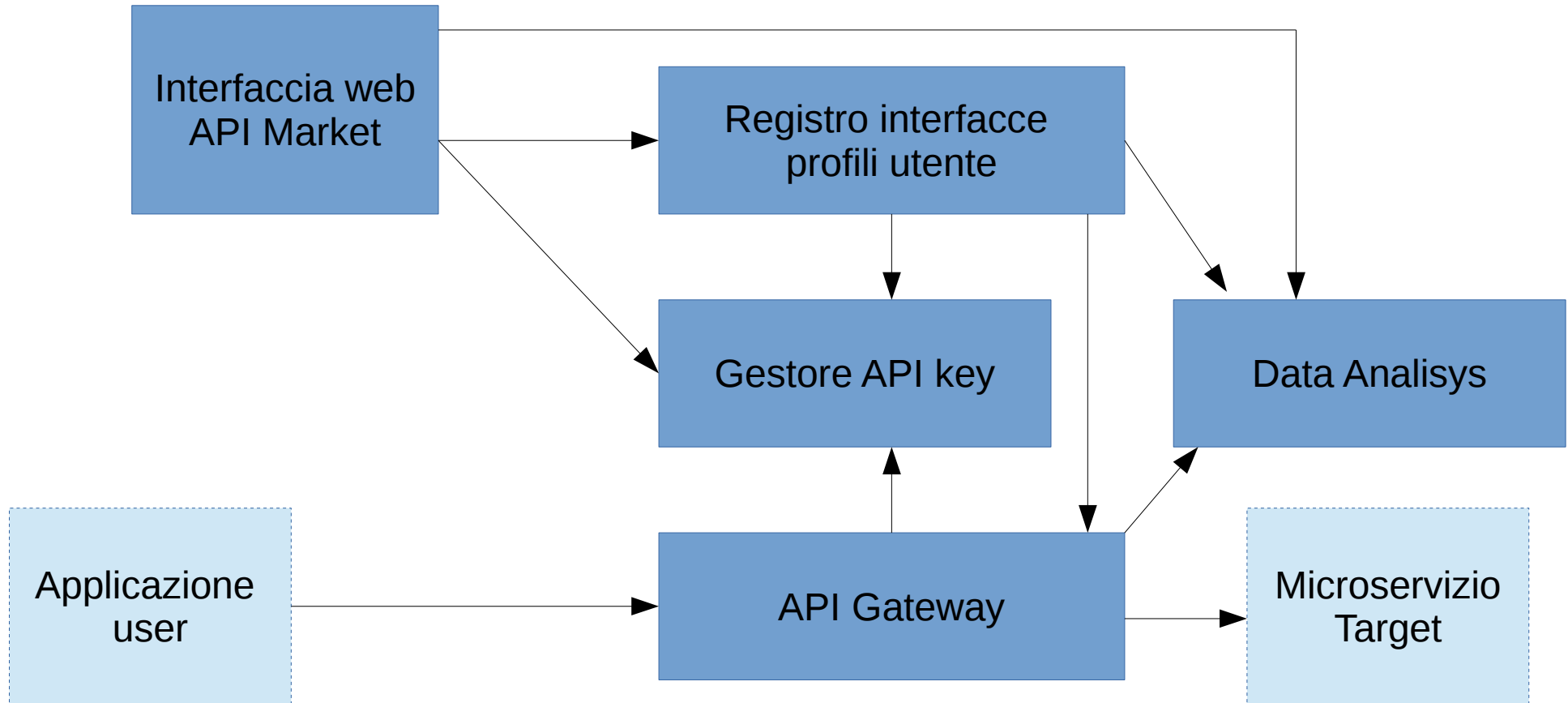
Capitolato

Obiettivi estesi:

- ***Permettere di definire diverse policy di vendita delle singole API in base al tempo, in base ai dati scambiati, in base al numero di chiamate, ecc.***
- ***Permettere di definire un Service Level Agreement (SLA) per ciascuna API rilasciata al netto dei ritardi introdotti dall'API Gateway***
- ***Permetta di monitorare il rispetto delle SLA definite al momento del rilascio***
- ***Permettere di gestire una moneta virtuale per la compravendita delle API da parte degli utenti***
- ***Permetta di confrontare i dati tecnici delle API tra loro***
- ***Permetta una gestione social dei profili utenti: rating delle API, commenti delle API***



Architettura di massima



Il sistema dovrà essere realizzato con un'architettura a microservizi



Modalità di esecuzione del progetto

Il progetto può essere realizzato da uno o più gruppi. Qualora sia affrontato da un unico gruppo, solamente gli obiettivi minimi saranno richiesti. In caso invece sia affrontato da più gruppi dovranno essere considerati anche gli obiettivi estesi, in ogni caso le specifiche di design iniziali dovranno essere condivise fra tutti i partecipanti al fine di rendere il progetto finale integrabile ed omogeneo.

Durante il progetto il team di italianaSoftware si rende disponibile a confrontarsi con i gruppi di lavoro

Il team di italianaSoftware offre altresì una formazione minima necessaria sull'utilizzo della tecnologia Jolie



Tecnologie

Suggeriamo di utilizzare Jolie:

- *per la rappresentazione delle interfacce (in particolar modo per quelle da registrare)*
- *e per la creazione dell'API Gateway (in particolarità c'è la possibilità di lavorare su un Gateway REST che sfrutta swagger già esistente e che è possibile estendere per gestire le policy di verifica delle API key)*

Le componenti web possono essere realizzate utilizzando javascript, HTML, css3 (è possibile utilizzare framework come Bootstrap)

A livello di Database possono essere utilizzati sia DB NoSQL che database SQL.

Per le altre componenti si lascia libertà di scelta



Materiali da consegnare

Prima dell'inizio della fase di sviluppo:

- *Per ogni servizio:*
 - *Descrizione del microservizio e delle singole API*
 - *Interfaccia delle API*
 - *Schema Design relativo all'eventuale base di dati associata*
- *Per il sistema:*
 - *Architettura completa dei microservizi utilizzati e delle loro connessioni*
 - *Sequence Chart Diagram delle interazioni che prevedono il coinvolgimento di più microservizi*
 - *Algoritmi delle policy per l'utilizzo delle API*
 - *Algoritmo di generazione delle API key*

Dopo la fase di sviluppo:

- *Codice depositato in un repository git*
- *Breve report tecnico che evidenzia gli aspetti positivi e gli aspetti negativi di un'architettura a microservizi*



Bibliografia

- *“Microservices” - Martin Fowler*
[<http://www.martinfowler.com/articles/microservices.html>]
- *“Jolie – the first language for microservices” [<http://www.jolie-lang.org>]*
- *“Swagger” - [<http://swagger.io/swagger-ui/>]*
- *“Jester” - Jolie REST router [<https://github.com/jolie/jester>]*



italianaSoftware

Contatti



Claudio Guidi
cguidi@italianasoftware.com