



FRIND

Ping your Friends and Share your events



PERCHE' NASCE FRIND?

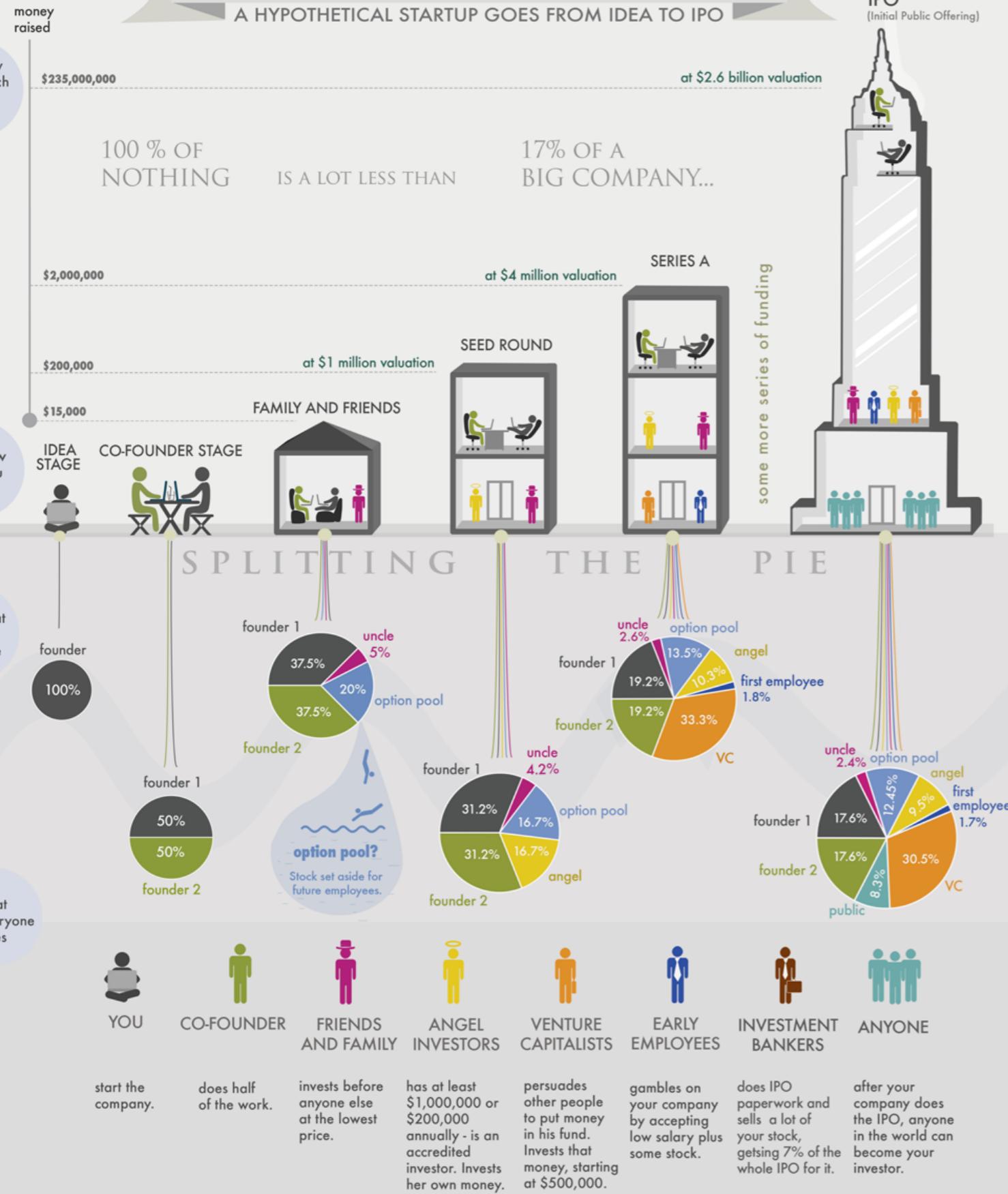
- Idea avuta durante un viaggio
- Da una semplice intuizione ad un idea completa di app
- Da sempre la voglia di creare qualche cosa di importante

HOW STARTUP FUNDING WORKS

A HYPOTHETICAL STARTUP GOES FROM IDEA TO IPO

BY ANNA VITAL

IPO
(Initial Public Offering)



FINANZIARE UNA STARTUP

Ogni progetto di questa portata ha necessità di essere finanziato perché si dispongano le giuste risorse che permettano la realizzazione in tempi brevi.

Per Frind abbiamo scelto di seguire lo stile di finanziamenti tipico della Silicon Valley, grazie al quale si da la possibilità a nuovi investor di entrare nel progetto ma allo stesso tempo mantiene un sistema bilanciato di quote.

In questo sistema entrano in gioco termini come **seed**, **round** di finanziamento, **business angel**, **venture capitalist** e **option pool**.

Un esempio è quello allegato in figura

DI COSA SI HA BISOGNO PER UN APP DI QUESTO TIPO?

- Server di messaggistica istantanea (**XMPP Server**)
- Storage illimitato per immagazzinare risorse
- Server per invio di **notifiche push** verso (Apple APNS, Android GCM)
- Gateway **SMS** per invio messaggi di verifica utente
- Istanze multiple di macchine **Linux Based** (Api, Web, Server, ecc...)
- Bilanciatori di carico per gestire le richieste

AMAZON AWS

revolutionary cloud system



COSA È AMAZON AWS E PERCHÉ È STATO SCELTO

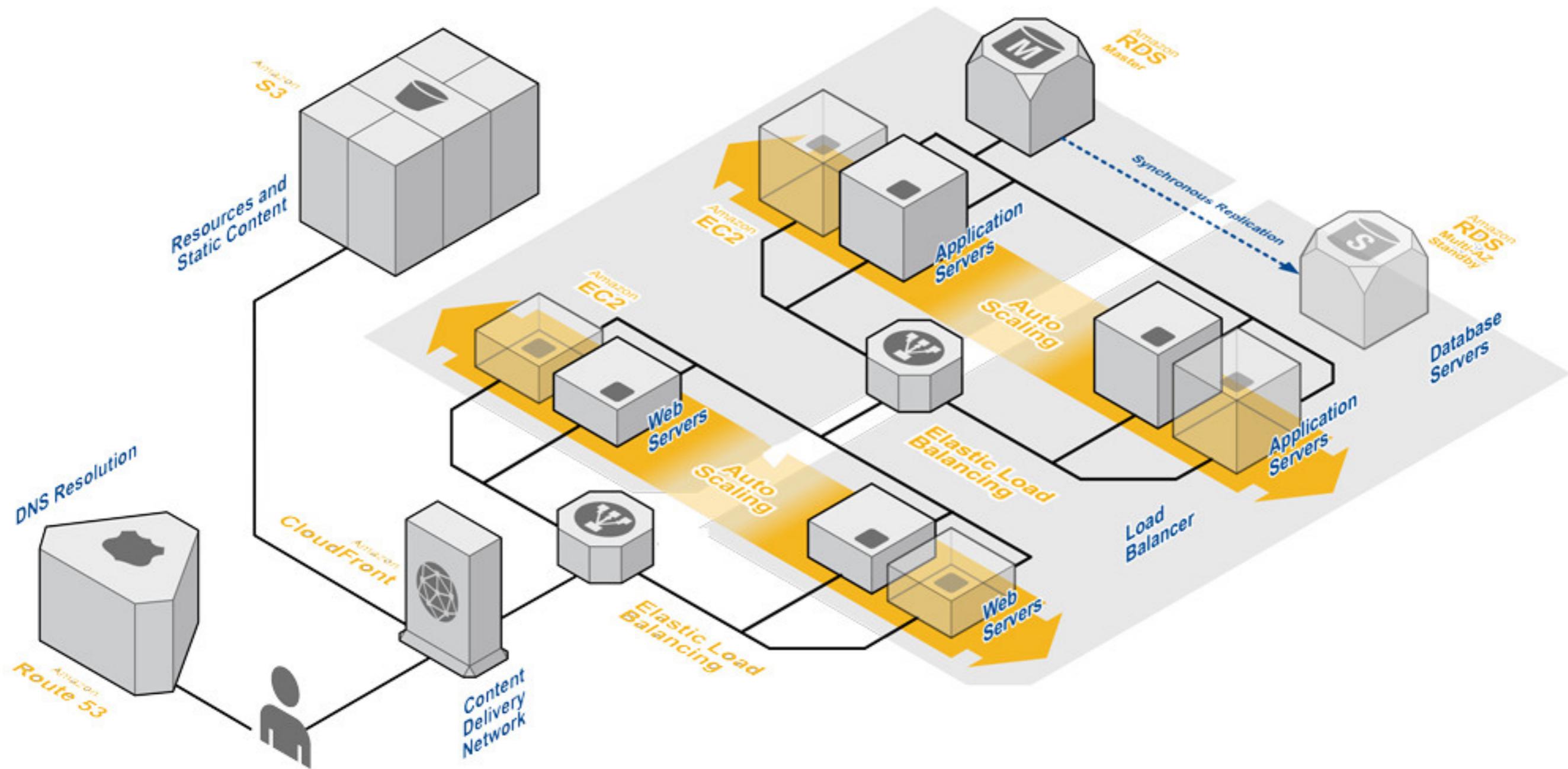
- Amazon non è solo un e-commerce ma molto di più
- **Amazon Web Services** è una piattaforma che offre servizi di cloud computing affidabili, scalabili ed economici, l'account è gratuito e si pagano solo i servizi usati
- Con **Autoscaling** ed **healt check** di immagini EC2 si ha un gran risparmio in termini economici, esempio pratico per picchi di traffico
- Affidabilità, sicurezza e risorse illimitate i punti chiave di AWS.

“

Want to increase innovation? Lower
the cost of failure!

– *Danilo Poccia cita Joy Ito*
Technical Evangelist and Solutions Architect

UNA POTENTE INFRASTRUTTURA PER UN SOLIDA RETE



QUALI SERVIZI DI AMAZON ABBIAMO UTILIZZATO

- **Amazon Route 53** - Sistema di DNS distribuito, rapido ed efficace, risponde dal punto più vicino a chi richiede
- **Amazon EC2** - Macchine linux che si attivano e disattivano su necessità, con caratteristiche configurabili
- **Amazon Cognito** - Con Amazon Cognito puoi permettere agli utenti di eseguire accessi e registrarsi sulle tue applicazioni Web e per dispositivi mobili. Con Amazon Cognito, potrai autenticare gli utenti tramite provider di identità social come Facebook, Twitter e Amazon, oppure tramite la tua soluzione di gestione di identità
- **Amazon S3** - Potente sistema di storage illimitato
- **Amazon SNS** - Sistema per la gestione delle notifiche push
- **Amazon Aurora** - Un database 100% mysql compatibile con capacità cloud

UN INFINITÀ DI SERVIZI A PREZZI RIDOTTI

Amazon Web Services

Compute

-  **EC2**
Virtual Servers in the Cloud
-  **EC2 Container Service**
Run and Manage Docker Containers
-  **Elastic Beanstalk**
Run and Manage Web Apps
-  **Lambda**
Run Code in Response to Events

Storage & Content Delivery

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System** PREVIEW
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Snowball**
Large Scale Data Transport
-  **Storage Gateway**
Hybrid Storage Integration

Database

-  **RDS**
Managed Relational Database Service
-  **DynamoDB**
Managed NoSQL Database
-  **ElastiCache**
In-Memory Cache
-  **Redshift**
Fast, Simple, Cost-Effective Data Warehousing
-  **DMS**
Managed Database Migration Service

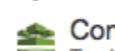
Networking

-  **VPC**
Isolated Cloud Resources
-  **Direct Connect**
Dedicated Network Connection to AWS
-  **Route 53**
Scalable DNS and Domain Name Registration

Developer Tools

-  **CodeCommit**
Store Code in Private Git Repositories
-  **CodeDeploy**
Automate Code Deployments
-  **CodePipeline**
Release Software using Continuous Delivery

Management Tools

-  **CloudWatch**
Monitor Resources and Applications
-  **CloudFormation**
Create and Manage Resources with Templates
-  **CloudTrail**
Track User Activity and API Usage
-  **Config**
Track Resource Inventory and Changes
-  **OpsWorks**
Automate Operations with Chef
-  **Service Catalog**
Create and Use Standardized Products
-  **Trusted Advisor**
Optimize Performance and Security

Security & Identity

-  **Identity & Access Management**
Manage User Access and Encryption Keys
-  **Directory Service**
Host and Manage Active Directory
-  **Inspector**
Analyze Application Security
-  **WAF**
Filter Malicious Web Traffic
-  **Certificate Manager**
Provision, Manage, and Deploy SSL/TLS Certificates

Analytics

-  **EMR**
Managed Hadoop Framework
-  **Data Pipeline**
Orchestration for Data-Driven Workflows
-  **Elasticsearch Service**
Run and Scale Elasticsearch Clusters
-  **Kinesis**
Work with Real-Time Streaming Data
-  **Machine Learning**
Build Smart Applications Quickly and Easily

Internet of Things

-  **AWS IoT**
Connect Devices to the Cloud

Game Development

-  **GameLift**
Deploy and Scale Session-based Multiplayer Games

Mobile Services

-  **Mobile Hub**
Build, Test, and Monitor Mobile Apps
-  **Cognito**
User Identity and App Data Synchronization
-  **Device Farm**
Test Android, iOS, and Web Apps on Real Devices in the Cloud
-  **Mobile Analytics**
Collect, View and Export App Analytics
-  **SNS**
Push Notification Service

Application Services

-  **API Gateway**
Build, Deploy and Manage APIs
-  **AppStream**
Low Latency Application Streaming
-  **CloudSearch**
Managed Search Service
-  **Elastic Transcoder**
Easy-to-Use Scalable Media Transcoding
-  **SES**
Email Sending and Receiving Service
-  **SQS**
Message Queue Service
-  **SWF**
Workflow Service for Coordinating Application Components

Enterprise Applications

-  **WorkSpaces**
Desktops in the Cloud
-  **WorkDocs**
Secure Enterprise Storage and Sharing Service
-  **WorkMail**
Secure Email and Calendaring Service

XMPP

*Extensible Messaging and
Presence Protocol*



COSA È E PERCHÈ È STATO SCELTO?

- Necessità di realizzare una **chat interna** in stile whatsapp, con migliaia di messaggi immediati e veloci
- XMPP precedentemente noto come **Jabber** è un insieme di protocolli di messaggistica e presenza basati su **XML**, nel 2003 già usato da oltre 10 milioni di persone in tutto il mondo
- Pur trattandosi di un sistema aperto, vi sono una serie di estensioni definite tramite degli **XEPS**, una specie di **RFC** dedicati. Questi XEPS definiscono moltissime estensioni come ad esempio: invio ultimo accesso, richiesta di amicizia, ecc... client diversi che rispettano gli XEPS riescono a comunicare correttamente tra loro
- La popolarità di XMPP raggiunge il suo culmine nel 2009 quando viene utilizzato da WhatsApp il quale però ne farà una sua personalizzazione chiamata funXMPP

JAVA VS ERLANG = OPENFIRE VS EJABBERD

Come detto in precedenza XMPP è un insieme di protocolli non un software, ci sono in commercio 2 software principali che implementano questi protocolli e forniscono un server vero e proprio: OpenFire ed Ejabberd, il primo scritto in Java il secondo scritto in Erlang.



XMPP E IL PROBLEMA DELLE RETI MOBILE

XMPP è un protocollo datato che pur se funzionale è stato creato in un periodo in cui gli smartphone e le reti mobile erano quasi assenti, per questo motivo ci sono parecchi problemi da affrontare quanto si utilizza in queste infrastrutture, ne citiamo uno per tutti: **Le cadute momentanee di segnale.**

A: Invia un messaggio a B che momentaneamente sembra connesso

B: Trovandosi in un punto con poco segnale perde la connessione 3G poco prima che il messaggio sia recapitato, trattandosi di una connessione TCP questa chiusura non viene rilevata, quindi il messaggio per il server è consegnato ma in realtà andrà perso infatti al momento in cui B torna online non gli sarà recapitato nulla

XEP 0198 - STREAM MANAGEMENT

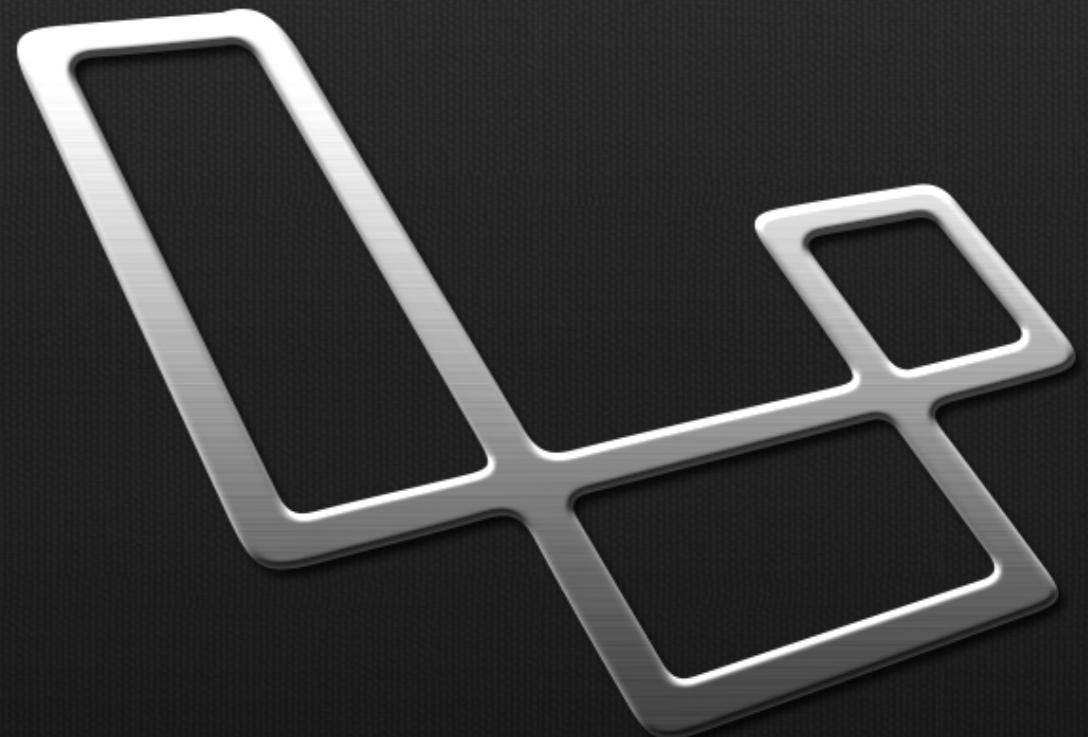
Come metodo per ovviare a quanto detto in precedenza è stato introdotto un nuovo XEP, il 0198, chiamato Stream Management, in parole semplici viene introdotto un nuovo livello di controllo, per ogni messaggio inviato dal server ad un client, questo resta in attesa di un messaggio di ack:

```
C: <enable xmlns='urn:xmpp:sm:3' />  
    <!-- Client sets outbound count to zero. -->  
  
C: <message from='laurence@example.net/churchyard'  
    to='juliet@example.com'  
    xml:lang='en'>  
    <body>  
        I'll send a friar with speed, to Mantua,  
        with my letters to thy lord.  
    </body>  
    </message>  
  
    <!-- Note that client need not wait for a response. -->  
  
S: <enabled xmlns='urn:xmpp:sm:3' />  
  
    <!--  
        Server receives enable, and responds,  
        setting both inbound and outbound counts  
        to zero.  
  
        In addition, client sets inbound count to zero.  
    -->  
  
C: <r xmlns='urn:xmpp:sm:3' />  
  
S: <a xmlns='urn:xmpp:sm:3' h='1' />
```

Se il messaggio di ack non viene ricevuto, il server provvederà a rinviarlo! Dato che OpenFire non supporta Stream Management, questo è stato il motivo che ci ha fatto scegliere Ejabberd, anche se scritto in Erlang un linguaggio molto più complesso da gestire.

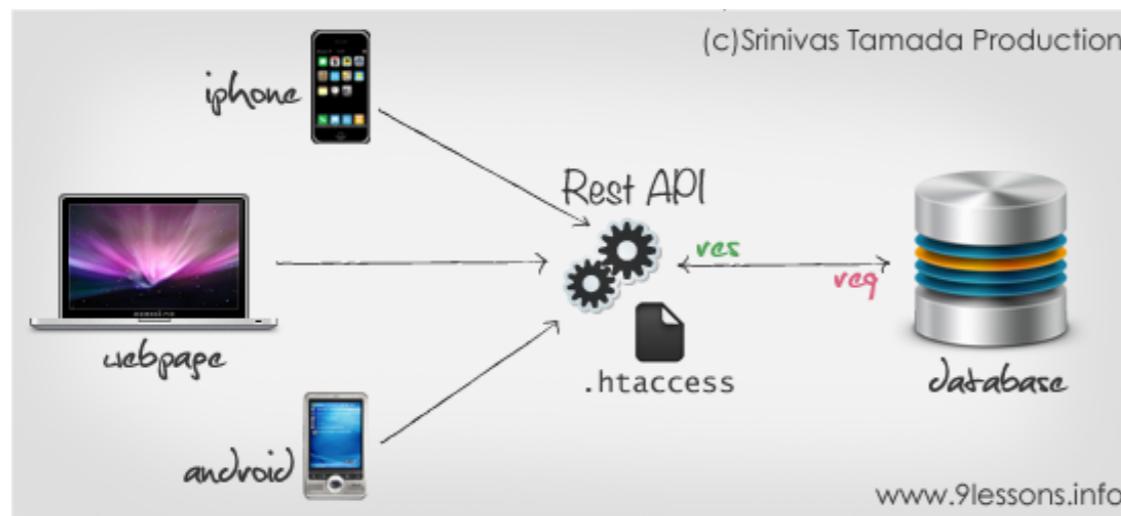
LARAVEL FRAMEWORK

web service e autenticazione



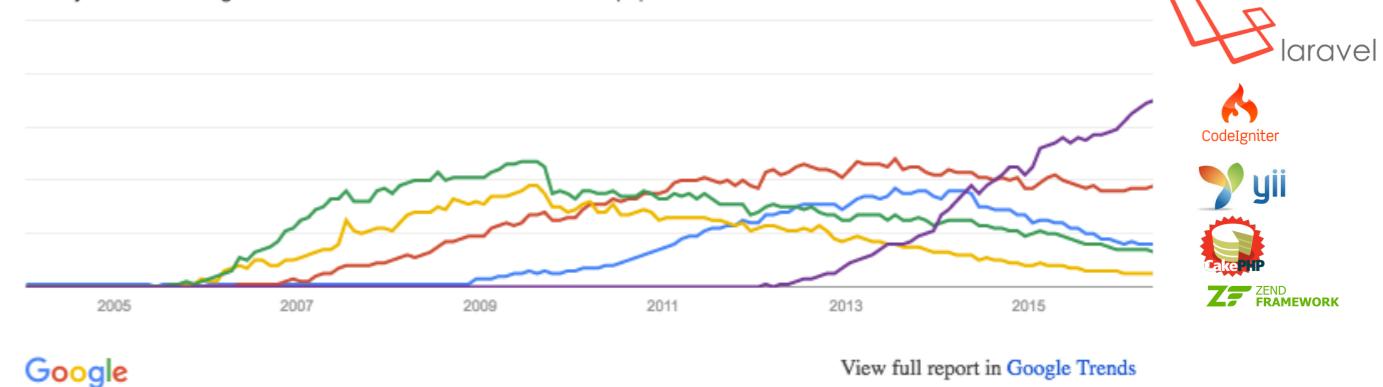
WEB SERVICE CON LARAVEL FRAMEWORK

Ogni applicazione che ha la necessità di scambiare dei dati ed operare con un server remoto necessita di un sistema di API.



Interest over time. Web Search. Worldwide, 2004 - present.

yii codeigniter zend framework cakephp laravel



Questo web service può essere realizzato in vari linguaggi e con varie piattaforme, noi abbiamo scelto PHP e Laravel Framework.

Laravel è un nuovissimo framework che prende il meglio delle tecnologie di altri sistemi fino ad ora conosciuti, utilizza la programmazione ad oggetti delle nuove versioni di PHP e il pattern MVC ormai noto e diffuso in molti linguaggi.

Laravel è un framework PHP che negli ultimi anni ha avuto un discreto successo. Insieme a Symfony rappresenta una delle soluzioni più consolidate del momento per lo sviluppo di applicazioni Web in PHP.

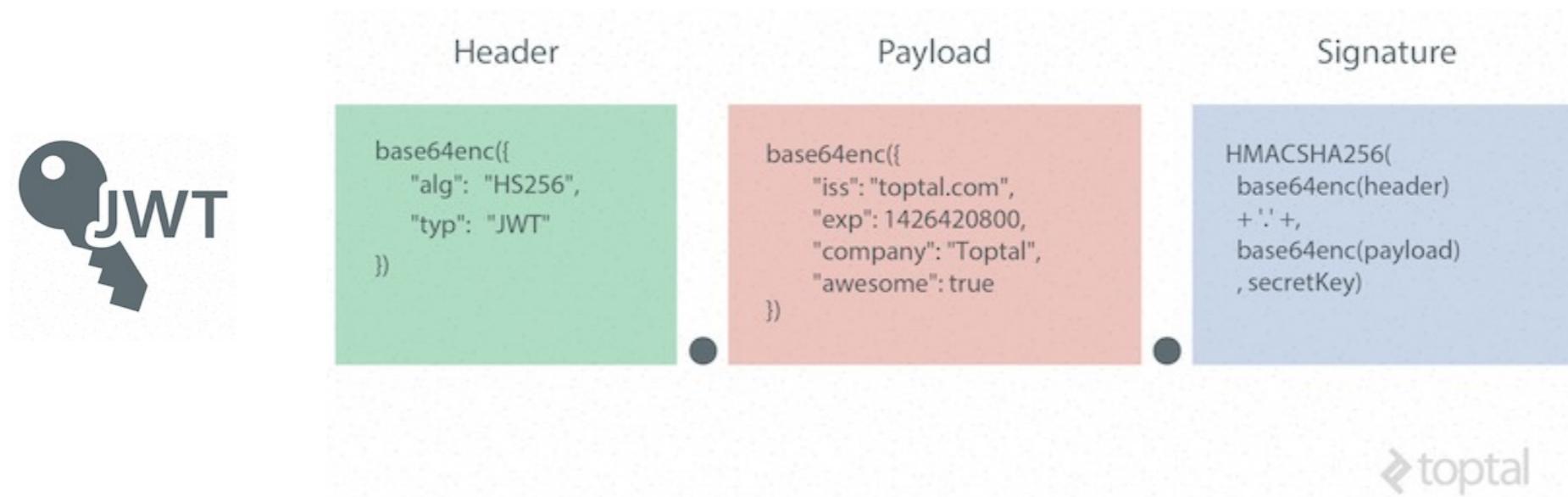
Oltre alla potenza ed alle moltissime librerie già incluse nel progetto, utilizza l'ormai affermato sistema di dipendenze "composer" che permette l'aggiunta di nuovo codice un operazione molto semplice.

AUTENTICAZIONE TRAMITE JWT

Per la fase di autenticazione delle chiamate che le app fanno al web service abbiamo scelto la tecnologia JWT (JSON Web Token).

Il JWT è uno standard JSON web Token per regolare le richieste http tra 2 parti, in questo caso app e webservice:

- l'applicazione effettua inizialmente una richiesta al web service, nella quale inserisce delle credenziali; il server controlla tali credenziali e, se tutto va bene, rilascia un token;
- l'applicazione memorizza localmente questo token; per tutte le richieste successive, l'applicazione rimanderà al server il token rilasciato;
- il server verificherà il token ad ogni richiesta prima di procedere



Il token è composto da 3 parti:

Header: l'intestazione. Contiene delle informazioni sul token stesso e su come è stato criptato;

Payload: il corpo vero e proprio che contiene dei dati "variabili" in base al contesto;

Signature: hash calcolato da un algoritmo descritto nell'header (HS256 in questo caso), usando una parola chiave nota SOLO al server. Le stringa che viene "hashata" è la semplice concatenazione delle versioni Base64 di header e payload, divise da un punto.

COME VIENE GERATO IL TOKEN

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpvag4gRG91IiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMhrHDcEfxjoYZgeF0NFh7HgQ
```

Decoded

EDIT THE PAYLOAD AND SECRET (ONLY HS256 SUPPORTED)

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

PAYOUT: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

VERIFY SIGNATURE

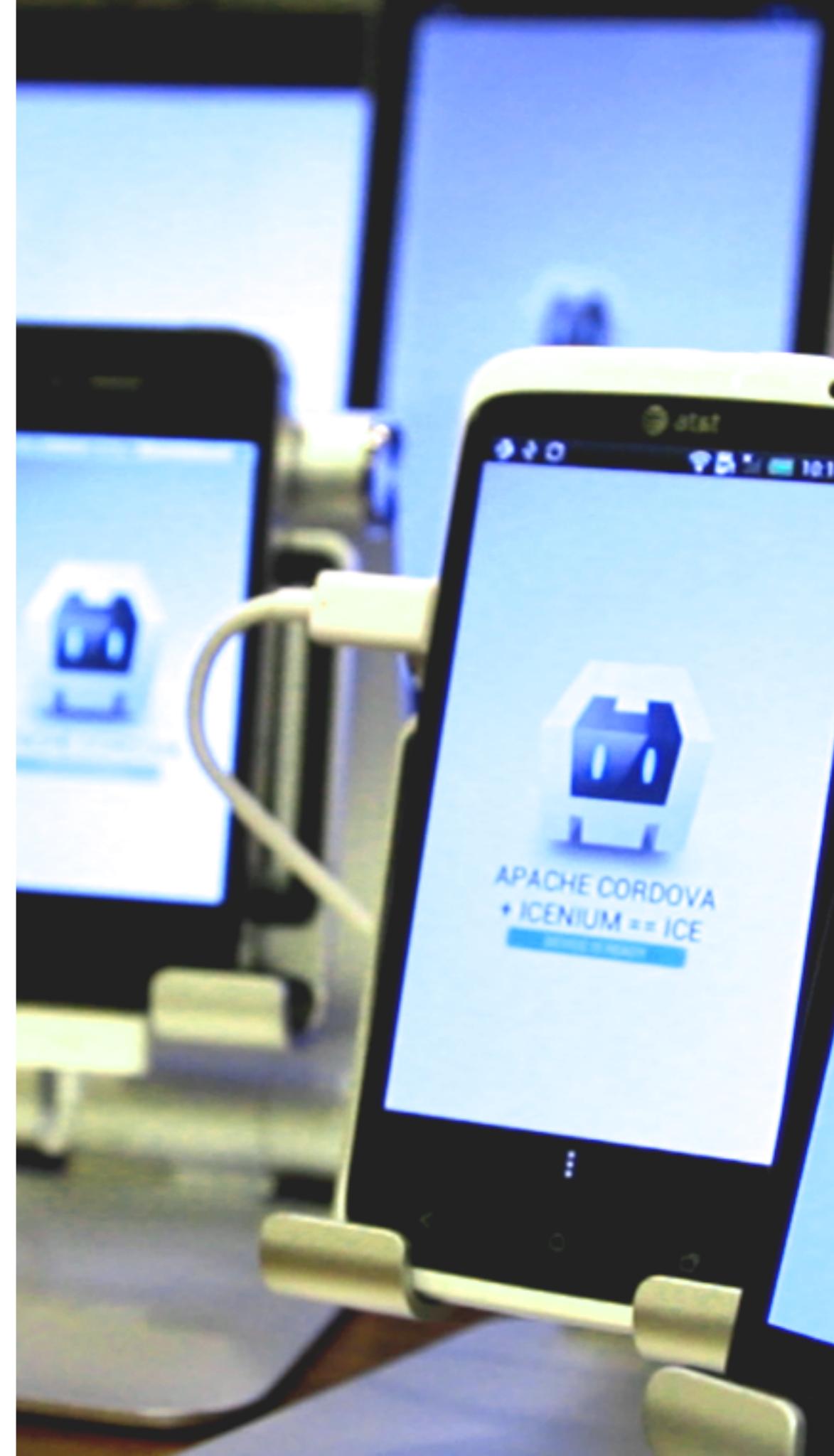
```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
)  secret base64 encoded
```

Tale token può essere **salvato** senza problemi in **locale**. La procedura di verifica, da parte del server, si occupa di controllare e verificare l'hash. Operazione che può effettuare solo il server, in quanto possiede la parola chiave segreta.

Al fine di aumentare la sicurezza, ogni token rilasciato dal server scadrà dopo 60 minuti, a questo punto si dovrà richiedere un nuovo token, a patto che si conosca una chiave segreta chiamata “refresh token” unica per ogni utente.

IBRIDO VS LINGUAGGIO NATIVO?

scegliere la miglior soluzione



IBRIDO VS NATIVO

Con l'aumento dell'utilizzo di smartphone si sono moltiplicati gli sviluppatori di App ed allo stesso tempo anche gli strumenti con i quali è possibile sviluppare. Principalmente identifichiamo due strade principali:

- **WebApp:** Sono veri e propri progetti web, programmati con linguaggi tipici del web (HTML/ Javascript) Queste applicazioni non sono disponibili negli store. La loro interfacce cercano di simulare la grafica di app native.
- **Applicazioni Ibride:** Sono applicazioni scritte in un linguaggio cross-platform (in genere Javascript + HTML5) come le web app, che però possono essere incapsulate tramite Web View nel linguaggio nativo di una certa piattaforma. Si possono quindi scaricare dagli store di riferimento ed installare sui rispettivi dispositivi: hanno un discreto accesso al sistema operativo anche se non sono state scritte interamente nel linguaggio specifico della piattaforma.



APACHE
CORDOVA™



PhoneGap



Xamarin

- **Applicazioni native:** Sono applicazioni create appositamente per un certo sistema operativo mobile, scritte con il linguaggio della piattaforma su cui andranno a girare e installabili sul dispositivo. In questo caso le due app sono viste come entità singole e vanno scritte separatamente



PERCHÈ ABBIAMO SCELTO IL NATIVO

Sia le applicazioni ibride che le web app hanno raggiunto oggi un ottimo livello di configurazione ed accesso al sistema, permettono di creare ottimi progetti in tempi brevi, ma, non scrivendo direttamente il codice nativo, si ha sempre un passaggio intermedio che è quello della compilazione in nativo per le app ibride o delle chiamate al sistema per le web app, alla luce di questo abbiamo scelto il linguaggio nativo, principalmente per tre ragioni:

- **MIGLIORI PERFORMANCE**

Avendo pieno accesso al sistema e scrivendo il codice direttamente possiamo analizzare ed ottimizzare le prestazioni fino al singolo componente. Maggiore velocità di esecuzione.

- **ACCESSO COMPLETO AL SISTEMA**

Sfruttando il linguaggio nativo abbiamo accesso ad ogni remota funzionalità del dispositivo, senza limiti e con il massimo delle prestazioni.

- **UTILIZZO DI EDITOR UFFICIALI**

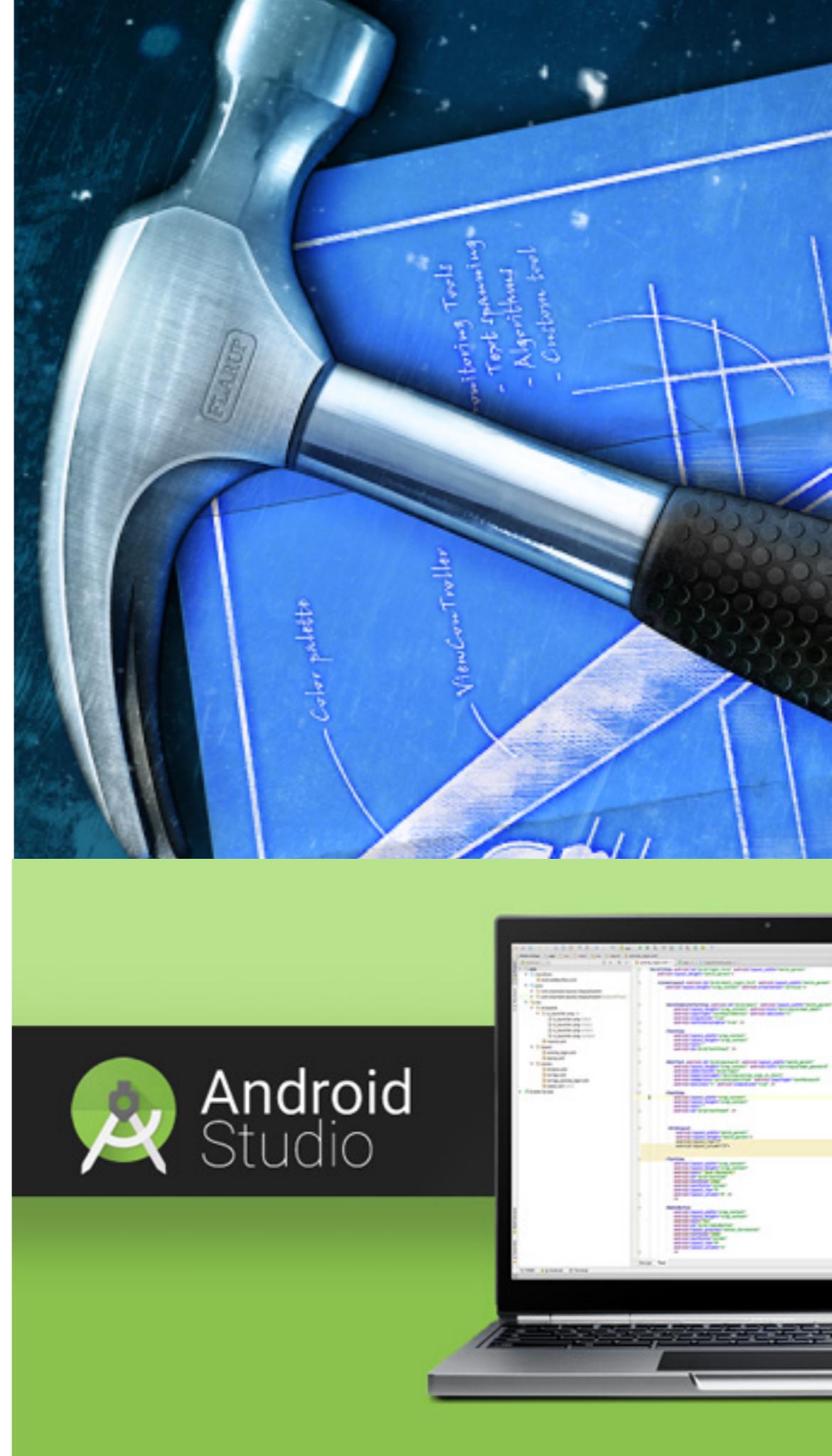
Lavorando nel nativo si utilizzano gli editor ufficiali delle case “Android Studio” ed “Xcode”, ed a tutta una serie di strumenti per debug, analisi e quant’altro molto avanzati.

- **MINORI PROBLEMATICHE CON AGGIORNAMENTI DI SISTEMA**

In nativo, ogni aggiornamento è direttamente disponibile, se si utilizzano framework di terze parti si è sempre legati ai loro tempi di aggiornamento, ed eventuali problemi di compatibilità.

STRUMENTI PER SVILUPPO NATIVO

le risorse utili



STRUMENTI UTILI NELLO SVILUPPO IOS

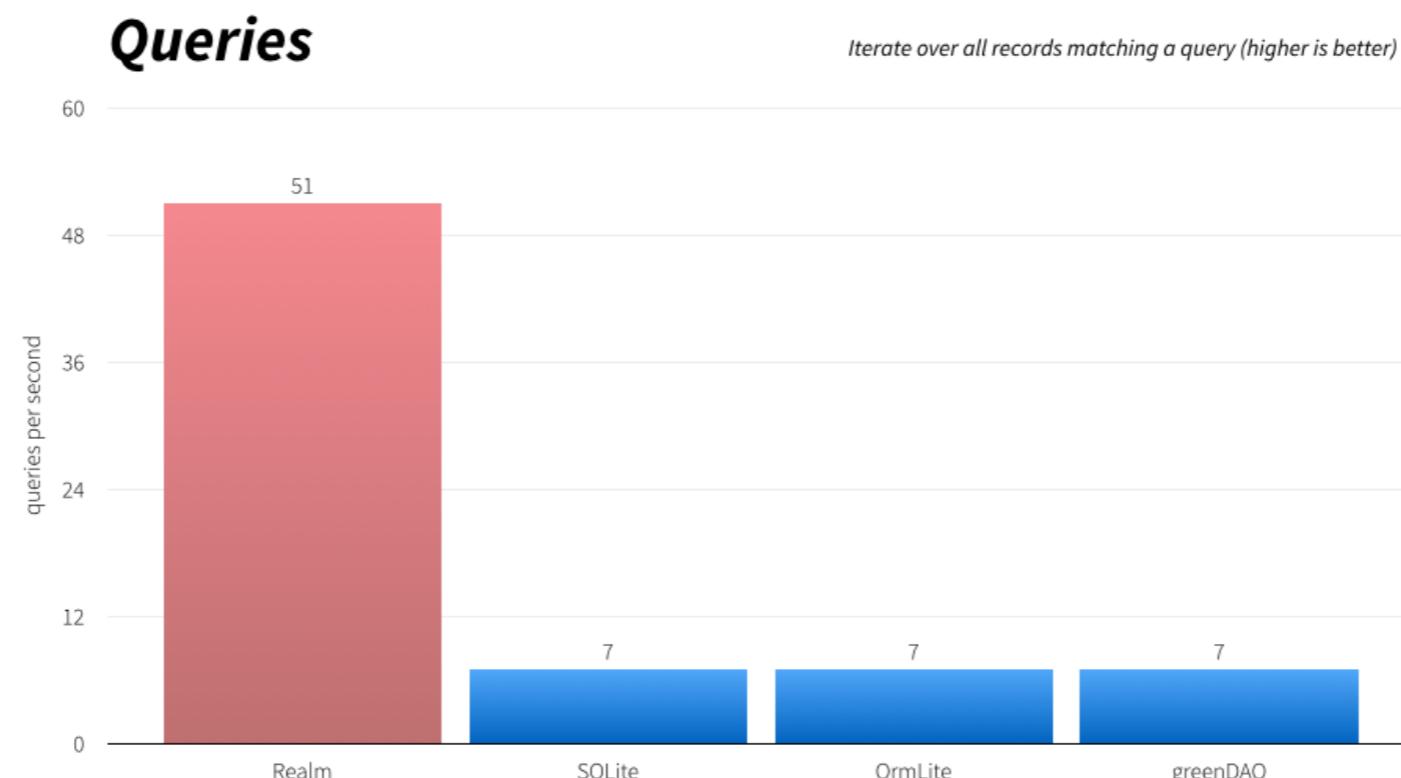
Durante la fase di sviluppo, abbiamo utilizzato una serie di software e librerie che migliorano lo sviluppo e le performance e che sono degne di nota:

- **Cocoapods / Gradle**

Ormai affermato nel mondo IOS cocoapods è un sistema di gestione delle dipendenze, il suo analogo nel mondo Android si chiama Gradle. Un sistema di dipendenze permette con poche righe di codice di aggiungere librerie di terze parti al nostro progetto e mantenerle sempre aggiornate, questo permette anche di risparmiare tempo nel riscrivere librerie già scritte da altri

- **Realm.io**

Per problemi soprattutto di performance ci siamo resi conto che il coredata, sistema interno di IOS per il salvataggio temporaneo dei dati in alcune situazioni di stress, come ad esempio nel nostro caso la sincronizzazione della rubrica risultava poco performante, per questo abbiamo optato per un nuovo database non relazione chiamato realm.



PROSPETTIVE FUTURE

cosa ci aspettiamo



PROSPETTIVE FUTURE

La nostra App si trova ad un ottimo stato di avanzamento, quasi in dirittura di arrivo nel mondo IOS e in pieno sviluppo nel mondo Android.

A breve inizieremo con la prima fase di beta test con gli **early adopters** e già da qui ci aspettiamo tanti feedback e soprattutto tanti spunti per migliorare e risolvere ogni eventuale problema.

DIVENTA UN EARLY ADOPTER

Se sei interessato ad aiutarci in questa fase, iscriviti al sito

www.frind.it

Quello intrapreso è solo l'inizio di quello che crediamo sarà un bel successo e soprattutto una bellissima esperienza professionale, speriamo di trovare presto altri imprenditori interessati a sostenere il nostro progetto e creare in Umbria una piccola Silicon Valley Tecnologica.

GRAZIE A TUTTI

stay tuned