

Tecnologie Cloud e Mobile

Lez. 01

Giuseppe Psaila

Università di Bergamo

giuseppe.psaila@unibg.it

Stack dei protocolli di rete: Modello ISO/OSI

- Livello Applicazione
- Livello Presentazione
- Livello Sessione
- Livello Trasporto
- Livello Rete
- Livello Collegamento Dati
- Livello Fisico

I sistemi Web

- Si basano su alcuni protocolli aperti
- che consentono di realizzare un'infrastruttura di comunicazione **aperta**.
- Chiunque può entrare in rete, pubblicare un servizio Web, accedere ad un servizio web

Modello ISO/OSI e Web

- | | |
|-----------------------------|---------------------------|
| • Livello Applicazione | HTML |
| • Livello Presentazione | UNICODE/UTF-8 |
| • Livello Sessione | HTTP/HTTPS |
| • Livello Trasporto | TCP |
| • Livello Rete | IP |
| • Livello Collegamento Dati | CSMA/CD, 802.11, LTE, ... |
| • Livello Fisico | |

OK, ma le risorse computazionali?

Per realizzare un Sistema web serve:

- Storage (un file system)
- Un DBMS e i database
- Un server sul quale gira un Web Server (o HTTP Server, software che gestisce le chiamate HTTP)

Dove li mettiamo?

- Su una o più macchine (server) collegate ad Internet

Problemi dei Server in casa?

- Rischio rottura
- Rischio furto (ambienti protetti/blindati)
- Surriscaldamento (quindi, ambienti condizionati)
- Costi energetici elevati
- Risorse umane dedicate
- Obsolescenza delle macchine
- Collegamento ad Internet costoso
- Scarsa scalabilità

In principio, l'hosting

- Un Provider fornisce risorse computazionali sulle sue machine
- fa Hosting del sito
- fornisce un collegamento ad Internet veloce
- si accolla i costi di obsolescenza e aggiornamento del software di base

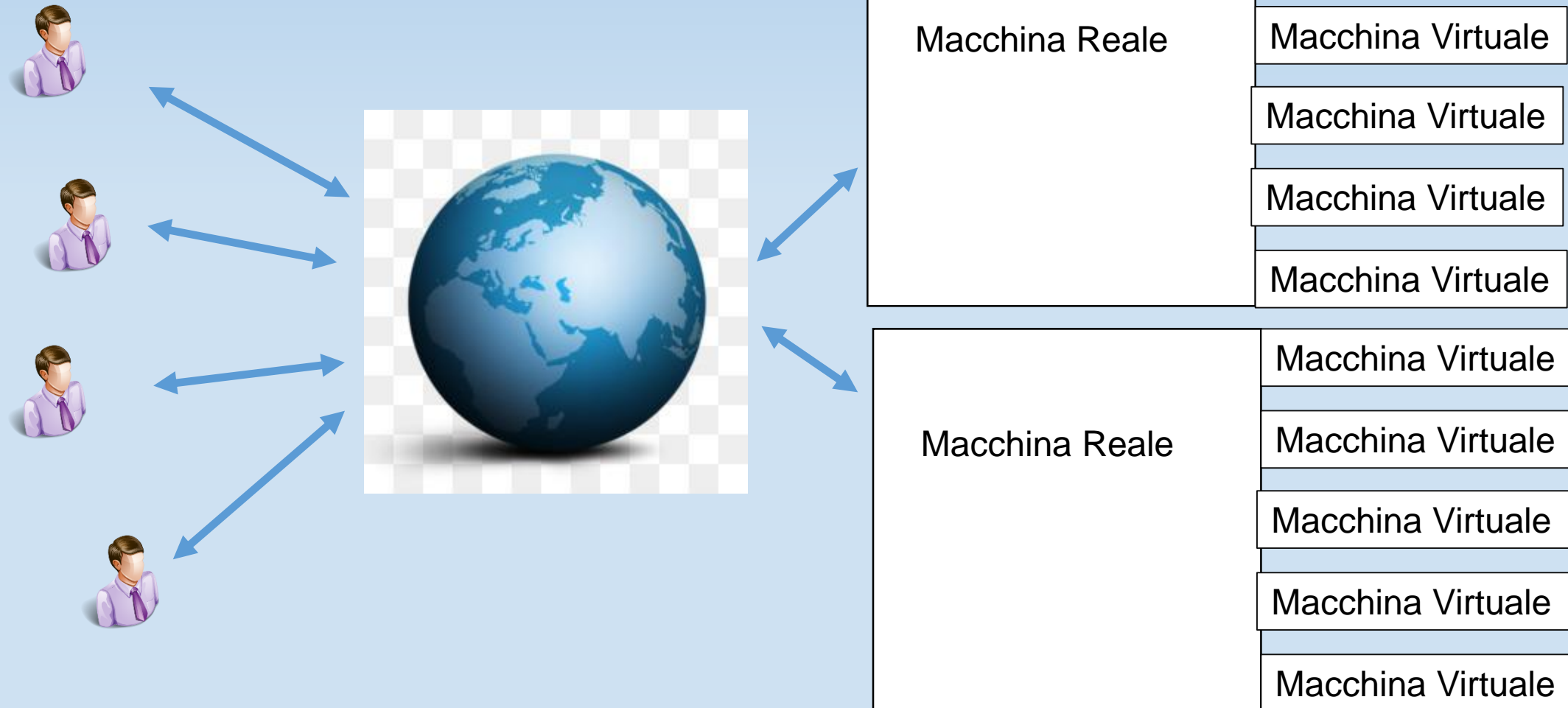
Macchine Virtuali

- Macchina Virtuale
vi sono programmi che consentono di creare una **macchina virtuale**, cioè un sistema di elaborazione completo ma virtuale
- Sulla stessa macchina fisica possono essere emulate molteplici macchine virtuali
 - Vantaggio: la macchina reale viene sfruttata di più, ripartendola su più macchine virtuali

Ripartizione sulle Macchine Virtuali

- Soluzione tradizionale:
la ripartizione delle risorse è fissa (per es.,
numero di processori, tempo di CPU, storage)

Hosting con Macchine Virtuali



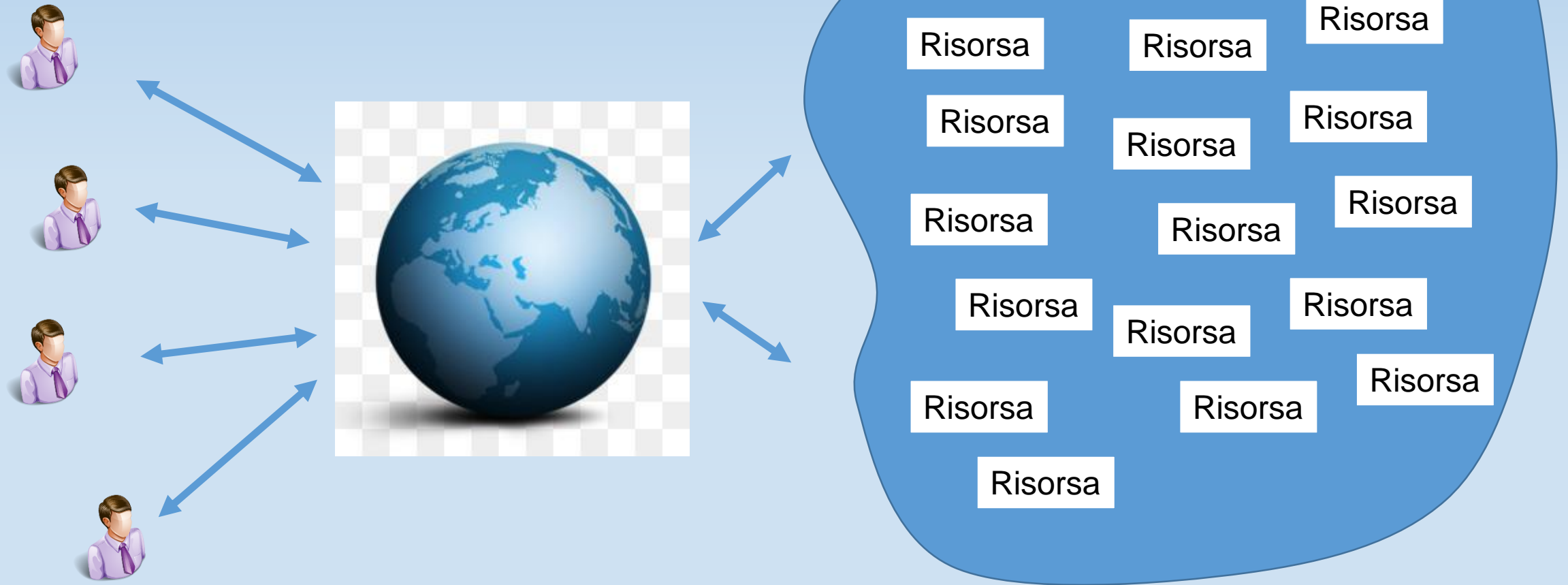
Ripartizione sulle Macchine Virtuali

- Soluzione del mondo cloud: viene riconfigurata dinamicamente, in base alle necessità
- Modello Pay per Use: si paga per quanto si usa (o si impegna)

Quindi, perché Cloud?

- Perché si passa da una soluzione di hosting, dove le macchine virtuali sono ospitate sempre sulla stessa macchina fisica
- ad una soluzione dove l'infrastruttura di elaborazione ridistribuisce le macchine virtuali sulla «Nuvola di Server»
- «Nuvola di Server» = «Cloud Computing»

Cloud Computing

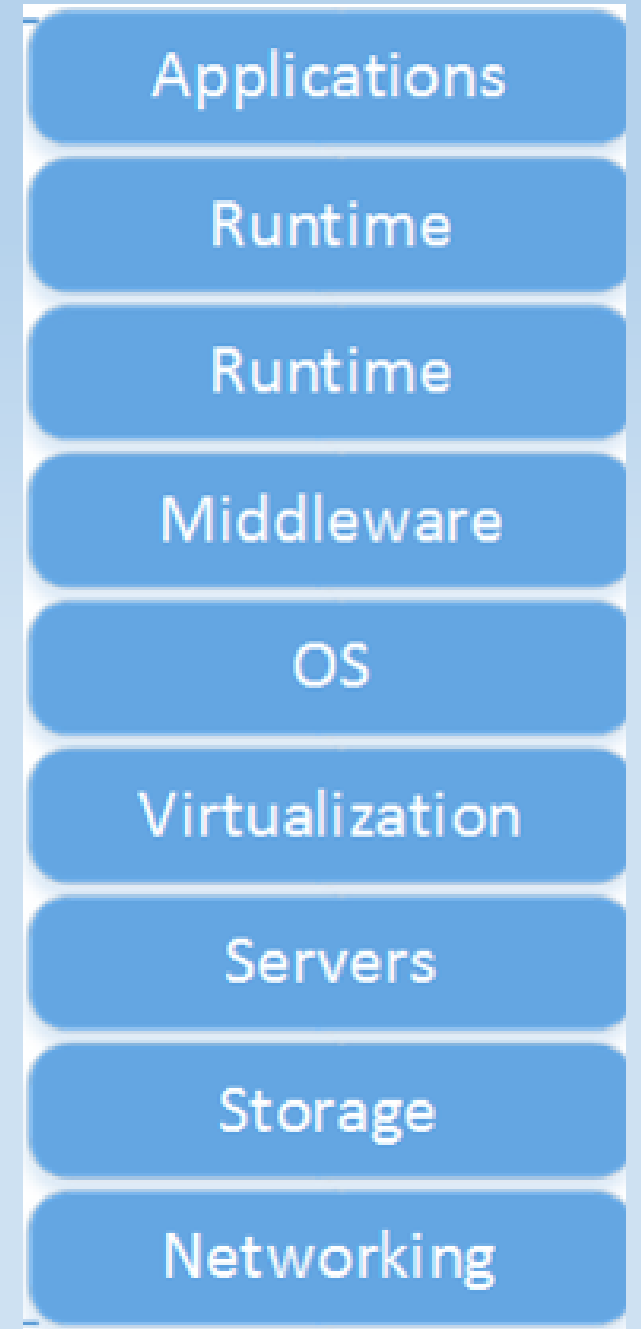


I Layer del Cloud Computing

Un altro stack: software

Un'applicazione si appoggia su molti strati di software sottostanti

- Storage
- Networking
- Sistema operativo
- Middleware
- Servizi tra S.O. e Applicazione (es., DBMS)



Un altro stack: software

Quindi, lo stack software potrebbe essere suddiviso in layer:

- Infrastruttura
- Piattaforma
- Software Applicativo

Applications

Runtime

Runtime

Middleware

OS

Virtualization

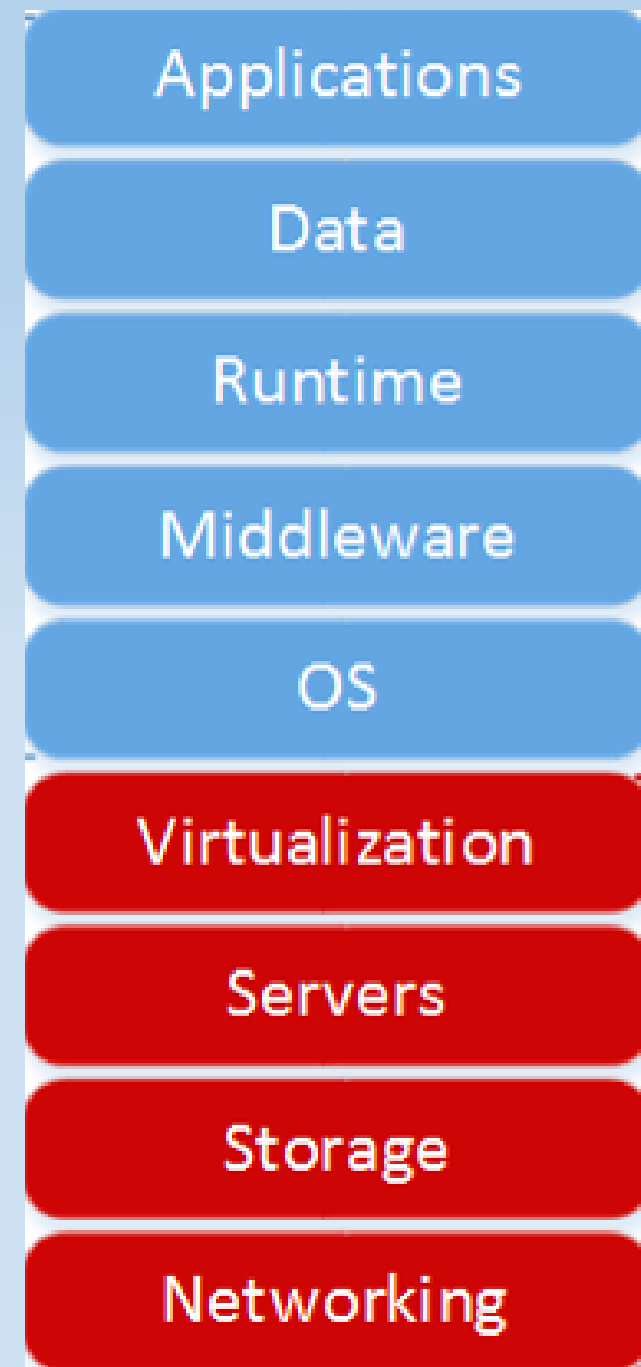
Servers

Storage

Networking

Infrastruttura

- Si intende il sistema di elaborazione vero e proprio, virtualizzato
- Il sistema cloud fornisce quindi l'infrastruttura di elaborazione, che il cliente poi gestisce
- **IaaS: Infrastructure as a Service**



Piattaforma

- Si intende la piattaforma sulla quale si sviluppano e si eseguono i programmi e i sistemi informativi
- Comprende gli strumenti di sviluppo, di testing e di deploy e tutti gli strumenti di integrazione
- **PaaS: Platform as a Service**

Applications

Data

Runtime

Middleware

OS

Virtualization

Servers

Storage

Networking

Software

- Si intende il software applicativo usato dagli utenti, che si avvale degli strati sottostanti
- **SaaS: Software as a Service**

Applications

Data

Runtime

Middleware

OS

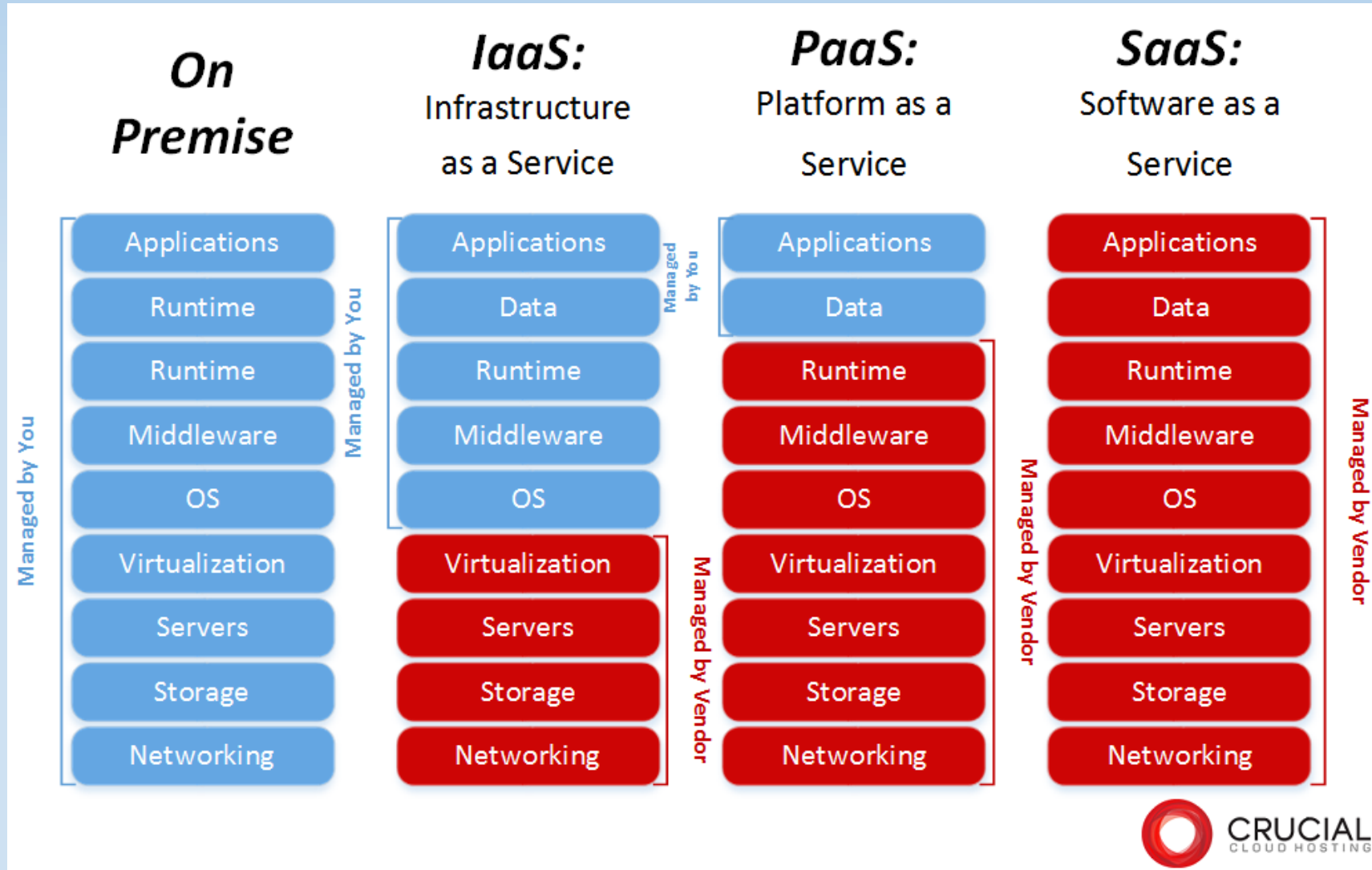
Virtualization

Servers

Storage

Networking

Visione di Insieme



Ma perché «as a service»?

IaaS: Infrastructure as a Service

- Il cliente non possiede più l'infrastruttura
- L'infrastruttura è posseduta dal Provider
- Per il cliente, l'infrastruttura è un servizio fornito dal Provider

Ma perché «as a service»?

PaaS: Platform as a Service

- Il cliente non possiede né l'infrastruttura né la piattaforma di sviluppo ed esecuzione
- L'infrastruttura e la piattaforma sono possedute dal Provider
- Per il cliente, la piattaforma (e al di sotto, l'infrastruttura) è un servizio fornito dal Provider

Ma perché «as a service»?

SaaS: Software as a Service

- Il cliente non possiede né l'infrastruttura, né la piattaforma, né il software
- Il software è posseduto dal Provider
- Per il cliente, il software (e al di sotto, l'infrastruttura e la piattaforma) è un servizio fornito dal Provider
- Il cliente **non deve installare il software**

Approccio «as a service»

Che cosa serve all'utente finale,

- Possedere il software?
- Il servizio svolto dal software?

All'utente finale interessa il servizio

Approccio «as a service»

Vantaggi

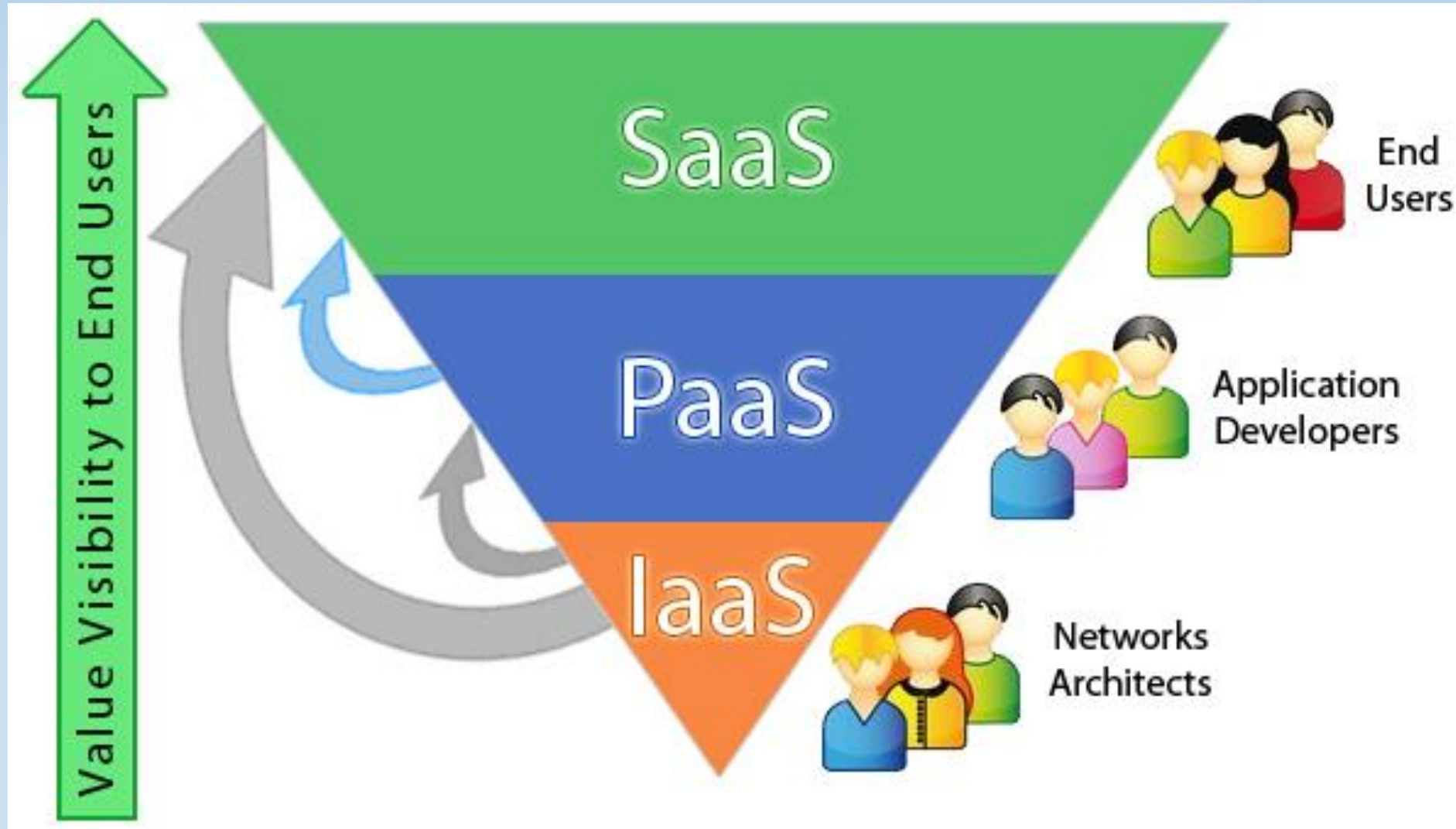
- Non deve comprare i server (la ferraglia)
- Non deve comprare il software di base/applicativo
- Non deve occuparsi di installare gli aggiornamenti

Approccio «as a service»

Chi è l'utente finale

- Se parliamo di un sistema informativo aziendale, potrebbe essere
 - Un dipendente dell'azienda
 - Un cliente dell'azienda
 - Un potenziale cliente che visita il sito e cerca informazioni

Layer e Utenti



Esempi Famosi

- SaaS
 - La suite Google
 - Microsoft Office 365
 - Overleaf (ambiente di scrittura Latex)
 - Salesforce.com

Esempi Famosi

- PaaS
 - Microsoft Azure
 - Google Cloud Platform
 - Amazon Web Service
 - IBM
 - Heroku
 - Force.com (la piattaforma di Salesforce.com)

Esempi Famosi

- IaaS
 - HP
 - IBM
 - IBM AS400 in Cloud

Storage as a Service

- Il servizio è lo storage
 - Dropbox
 - Google Drive
 - Apple iCloud
 - Microsoft OneDrive

Salesforce.com



- Si tratta di un prodotto totalmente a sé
- È il primo sistema informativo aziendale erogato da sempre in modalità cloud (dal 1998)
- Di base, fornisce un CRM (Customer Relationship Management) molto completo e altamente personalizzabile ed estendibile

Force.com



- È la piattaforma di Salesforce.com
 - Database a oggetti senza gerarchie
 - Linguaggio di programmazione Apex (derivato da Java), tradotto in Byte-code
 - Formato XML specifico per pagine web, con pattern Model-View-Controller
 - Interfaccia *Lightning* basata sul pattern Model-Controller-Controller-view e JavaScript
 - Web Services, JSON e WSDL integrati

Architettura Multi-tenant



- Salesforce.com ripartisce le risorse in modo dinamico
- Una server farm gestisce i sistemi informativi di molti clienti (detti ORG)
- Le routine possono essere eseguite su un qualsiasi server
- Si perde completamente il concetto di macchina

L'Infrastruttura?



- L'infrastruttura ovviamente c'è, ma non si vede
- È completamente mascherata al di sotto della piattaforma

Quali tecnologie?

- Le tecnologie usate nel mondo del cloud sono molteplici
- Ma, di fatto, sono tecnologie che provengono da mondi noti:
 - Web: HTML, JavaScript, XML
 - Web Services: REST, SOAP; WSDL
 - Database: Relazionali, NoSQL

Dettagli del Corso

Argomenti Trattati

- JavaScript: il linguaggio di programmazione client-side delle pagine web
- JSON: il nuovo formato di rappresentazione dei dati per il web
- AJAX: il meccanismo di comunicazione delle pagine web con i server
- XML: il formato di rappresentazione dei dati alla base di molte applicazioni

Argomenti Trattati

- JQuery: un framework per potenziare JavaScript (e usare AJAX facilmente)
- Node.js: JavaScript passa sul lato server
- MongoDB e i NoSQL databases: il più famoso esempio di JSON storage
- I Web Services: REST, SOAP e WSDL
- Python: il linguaggio di programmazione per la Data Science

Argomenti Trattati

- L'approccio a Micro-Servizi
- Map-Reduce: parallelizzare applicazioni usando primitive semplici, senza occuparsi dell'esecuzione
- Block-Chain: diverse soluzioni per diversi problemi applicativi (panoramica)

Esercitazioni

- Esercitazioni pratiche in laboratorio
- Useremo Amazon Web Services
- Tenute dal Ing. Mauro Pelucchi

- Obiettivo: toccare con mano queste tecnologie
- Al fine di svolgere elaborati/progetti

Modalità d'Esame

- In presenza
con quiz e codici da completare
- Progetto/Elaborati
facoltativo: consente di completare il voto
Valutato dall'Ing. Mauro Pelucchi

Mobile

Applicazioni Mobili

- Le applicazioni mobili sono dette semplicemente «app»
- Difficile trattare nel dettaglio anche questo argomento
- Tuttavia, gli argomenti del corso tornano utili anche per questo

Tipologie di App

- Applicazioni Native (Native Applications)
- Applicazioni Ibride (Hybrid Applications)
- Web App (Embedded Web Applications)

Applicazioni Native

- Sono scritte nello specifico linguaggio di programmazione nativo della piattaforma mobile
- Ovviamente, ogni piattaforma ha un suo linguaggio di programmazione
- La stessa App va completamente riscritta, per passare da una piattaforma ad un'altra

Applicazioni Native Android

- Linguaggio di programmazione: Java
- Ambiente di sviluppo: Android Studio (in origine Eclipse)
- Target intermedio: Byte-code
- Target finale: Dalvik Virtual Machine, con evoluzione del Byte-code a 16 bit
- Distribuzione attraverso Google Play Store
- Tutto il processo di pubblicazione è gestito via Web

Applicazioni Native Android

- Recensente, Google ha introdotto un nuovo linguaggio di programmazione per sviluppo nativo
- **Kotlin**
- Per certi versi, è una via di mezzo tra Java e JavaScript
- Più snello di Java con tipizzazione dinamica
- Che viene tradotto in Byte-code

Applicazioni Native iOS

- Linguaggio di programmazione: Objective C
- Ambiente di sviluppo: XCODE su Mac
- Distribuzione attraverso iTunes Store
- L'app viene preparata da XCODE, che la invia ad iTunes Store
- Non si possono pubblicare App senza Mac

Storia di Objective C

- All'inizio degli anni '80
- vennero proposte due estensioni a oggetti del C
 - C++
 - Objective C
- Il «vincitore» è stato il C++ (come estensione ufficiale del C)
- Ma Objective C non scomparve, rimase nascosto fino alla fine degli anni 2000

Storia di Objective C

- È ricomparso in modo deciso con i primi smartphone di Apple
- L'ambiente di sviluppo comprende molte librerie, dove molte funzioni hanno il prefisso «NS»
- Che cosa indica «NS»?
- NextStep, il sistema operativo sviluppato dalla società Next fondata da Steve Jobs
- Quando venne messo alla porta dalla Apple

Storia di Objective C

- Nel 1985, Steve Jobs viene cacciato dalla Apple (che aveva fondato)
- Fonda NexXT per lo sviluppo di workstation grafiche ad alte prestazioni (chiamate NextStep)
- NeXT acquisisce la licenza di Objective C e sviluppa il sistema operativo con Objective C
- Quindi, Objective C si è sviluppato nella nicchia di NeXT, per software ad alte prestazioni

Storia di Objective C

- Nel 1997, ripreso il controllo della Apple, Objective C arriva in Apple insieme a Steve Jobs
- Objective C rimane nascosto all'interno di Apple, fino a che non c'è stata l'esplosione degli smartphone Apple.

Applicazioni Native iOS

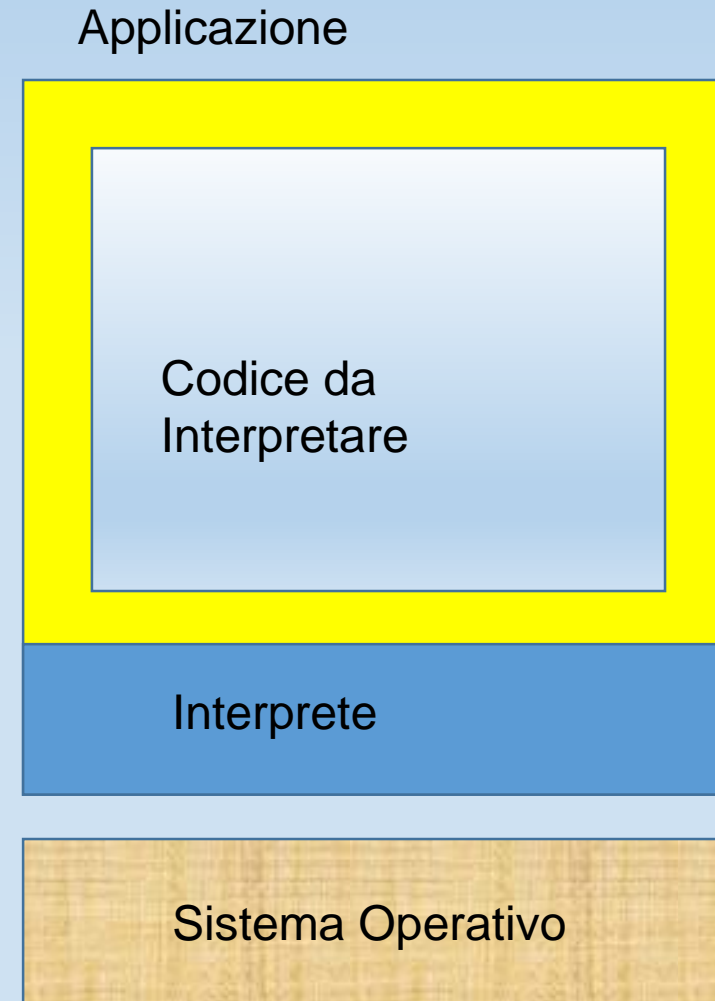
- Apple ha introdotto un nuovo linguaggio di programmazione per iOS
- Swift
- Progettato per attirare i programmatori JavaScript
- Molto più semplice e stabile di Objective C
- Anch'esso basato sulla tipizzazione dinamica

Applicazioni Ibride

- Come evitare di riscrivere le applicazioni per passare da iOS ad Android e viceversa?
- Idea: usare un altro linguaggio di programmazione
- Ma come eseguire questa app?
- Due possibilità:
 - Interpretazione
 - Traduzione

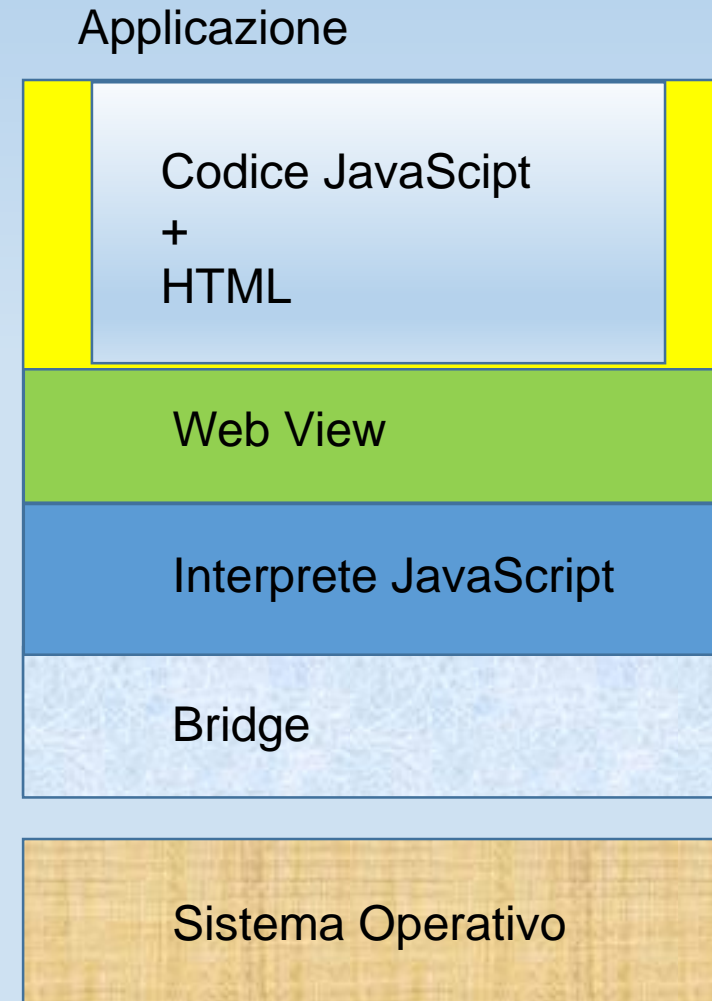
Applicazioni Ibride Interpretate

- Il linguaggio viene interpretato all'interno di uno scheletro base



Idea: JavaScript + HTML

- HTML per il layout grafico
- JavaScript per la parte dinamica
- Web View: componente per visualizzare HTML
- Interpretare JavaScript
- Bridge: libreria per consentire al codice di accedere al sistema operativo



Cordova / PhoneGap

- Apache Cordova: il framework che integra il bridge e l'interprete JavaScript
- Versioni principali:
 - Per iOS
 - Per Android
- Le varie versioni forniscono la stessa interfaccia (più o meno) con i diversi sistemi operativi.
- Apache Cordova è gratuito

Cordova / PhoneGap

- Adobe PhoneGap:: un servizio di Adobe per sviluppare applicazioni ibride basate su Apache Cordova
- Il sistema genera i build per le diverse piattaforme on-line

Microsoft Xamarin

- Questo strumento adotta il secondo approccio
- L'applicazione viene scritta in C# (linguaggio proprietario di Microsoft)
- Xamarin traduce il codice C#
- in Java, per Android
- nel codice binario per iOS

Vantaggi/Svantaggi

- Apache Cordova consente di adattare le competenze dei programmatori web
- Ma l'esecuzione risulta generalmente lenta
- Xamarin richiede di acquisire competenze specifiche (se non si conosce già C#)
- C# è un linguaggio più complicato di JavaScript
- Le applicazioni sono molto veloci in esecuzione

React Native

- Esiste una terza via: il framework React Native
- Ideato e spinto da Facebook
- Il codice REACT-native viene tradotto in Java (per Android) e in Objective C (per iOS)
- Ma si usa un unico linguaggio di programmazione
- Quindi, un'app React-Native non è una web view

React Native

- Il linguaggio di programmazione è JavaScript
- Ma non viene interpretato
- Fornisce un framework per gestire le view
- Fornisce librerie per gestire le risorse del dispositivo

Web Application

- L'app che si installa sul dispositivo mobile contiene solo una Web View
- Le schermate sono pagine HTML + JavaScript che vengono recuperate dal server
- Vantaggi: l'app è di fatto un'applicazione web, gestibile con specifici fogli di stile dal sito web
- Svantaggi: necessità del collegamento ad Internet

App Mobili e Resto del Mondo

- Le App devono comunicare con i server di appoggio
- I server forniscono API (Application Programming Interface) basate sui web service
- Quindi, si ha una visione integrata: sono le stesse tecnologie
- Progettazione integrata: separare l'interfaccia dai servizi
- Diverse interfacce usano gli stessi servizi

Approccio Integrato

- Tanti componenti diversi
- Ognuno con un ruolo ben preciso

