

Sistemi Distribuiti

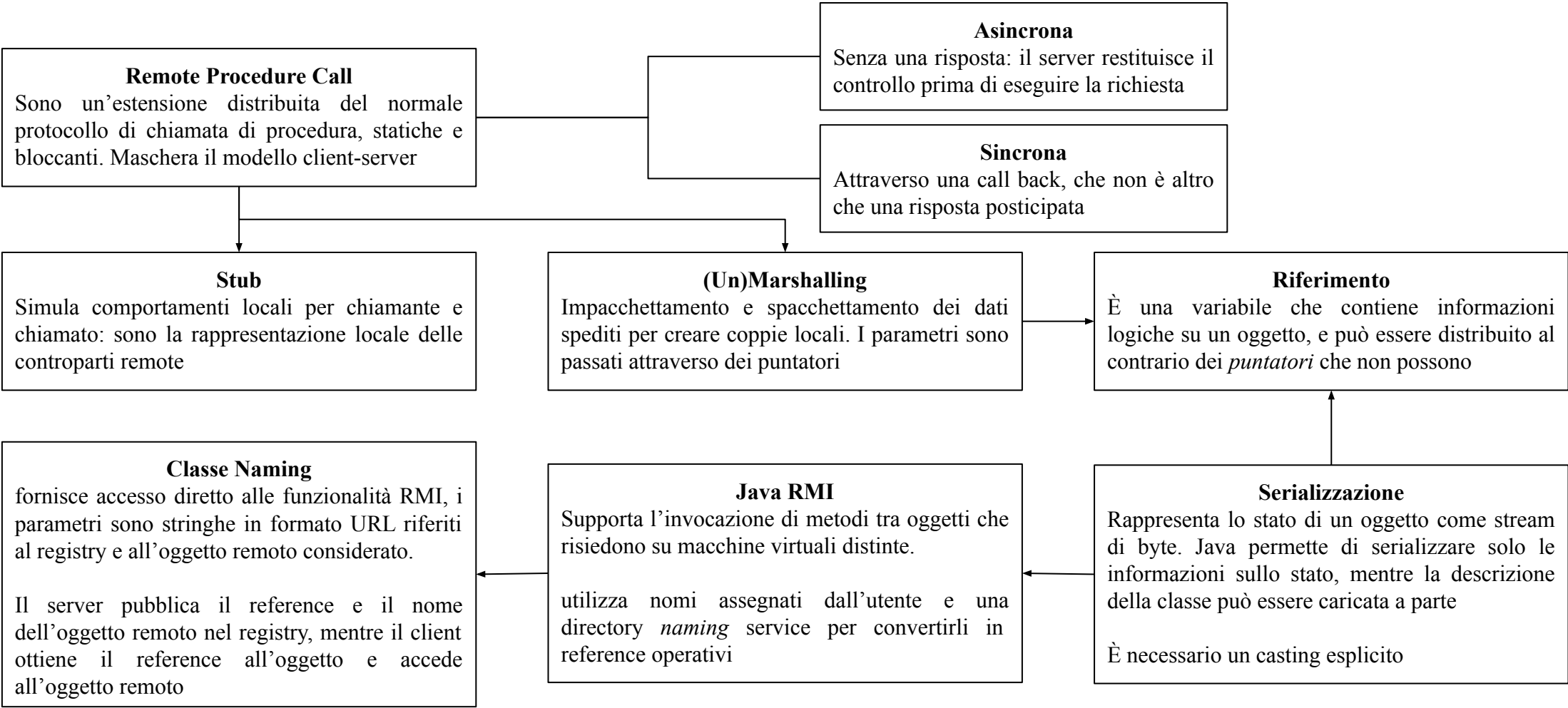
Schemi riassuntivi

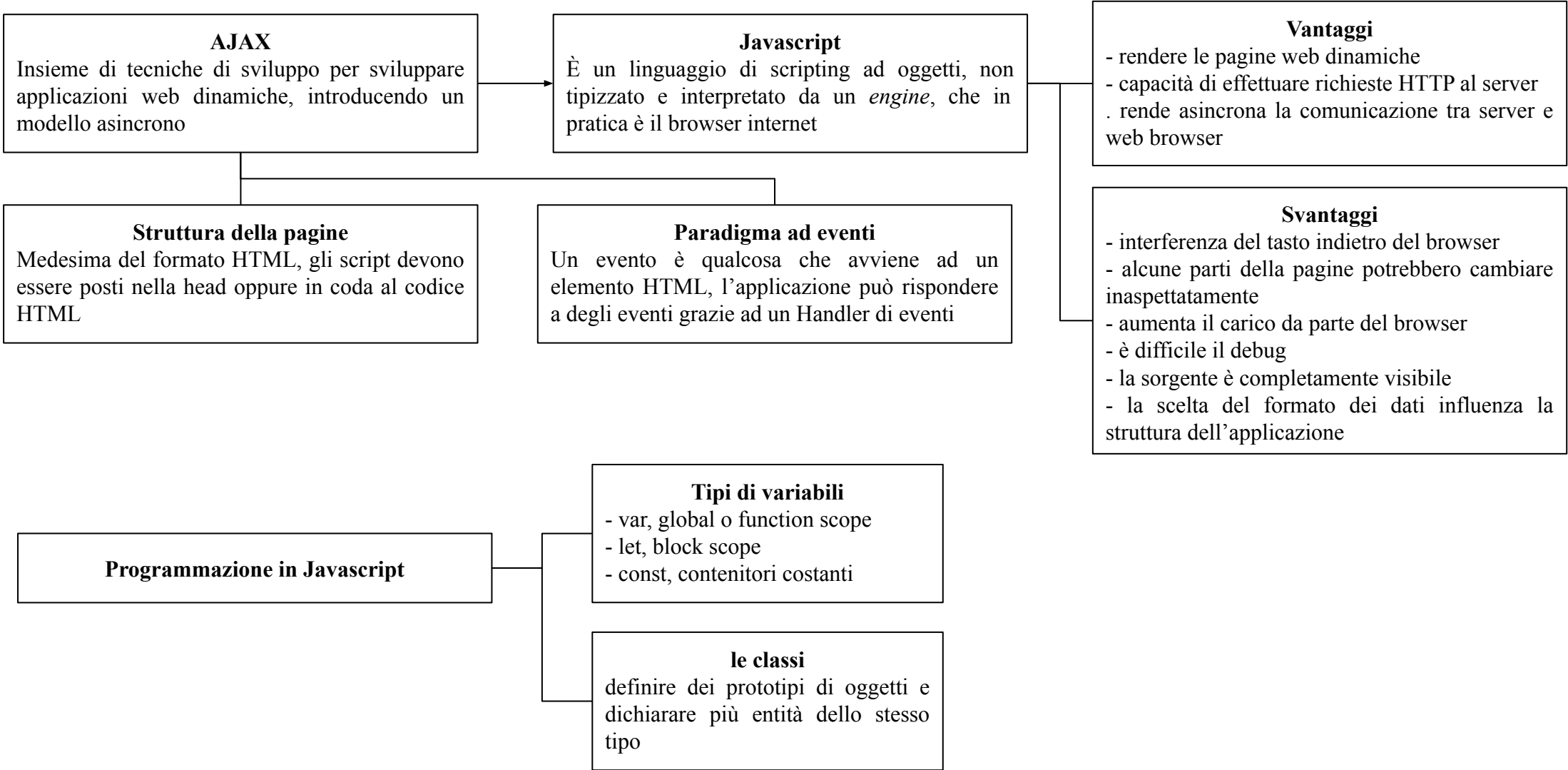
secondo parziale

LAUREA TRIENNALE IN SCIENZE INFORMATICHE

Andrea Broccoletti

Università degli studi di Milano Bicocca





XMLHttpRequest
Questo oggetto viene usato per scambiare dati con un server dietro le quinte, è possibile effettuare chiamate ad un server senza dover necessariamente ricaricare la pagina

Callback
È una funzione passata come parametro ad un'altra funzione.
È buona norma creare un oggetto XMLHttpRequest e richiamarlo per ogni task, se si hanno più task

Formato JSON
È un formato leggero per lo scambio di dati

Interoperabilità
C'è una corrispondenza univoca tra la codifica JSON e la codifica degli oggetti Javascript

onreadystatechange
È lanciato ogni volta che la readyState cambia, quest'ultima conserva lo stato di una richiesta

La richiesta è pronta quando si ha stato 4 e codice 200

Stati

- UNSENT
- OPENED
- HEADERS_RECEIVED
- LOADING
- DONE

ResponseText
Restituisce una stringa

ResponseXML
Ottiene un oggetto XML senza passare per il formato testuale

Node.js
Piattaforma realizzata con il motore Javascript che permette di realizzare applicazioni web veloci e scalabili, che usa un modello IO non bloccante e ad eventi e consente la realizzazione di applicazioni lato server scritte in Javascript

Framework express.js
Fornisce gli strumenti per realizzare un server che può ospitare sia risorse statiche, sia applicazioni che generano rappresentazioni dinamiche

Modello ad eventi
Il modello ad eventi è sviluppato su un singolo thread, è un loop che esegue e verifica periodicamente l'avvenire di un certo evento

Il *single event loop* è a livello logico, le operazioni sono eseguite da thread autonomi che permettono l'esecuzione parallela e concorrente

Basi

- si basa sull'esecuzione di operazioni stateless
- disaccoppia la programmazione dall'esecuzione favorendo la scalabilità

risposte

Dati semantici
Json può essere complicato ed esteso. Si aggiungono dei *metadati*, ovvero informazioni aggiuntive che qualificano il contenuto

Shared knowledge
Si definisce un vocabolario che descrive le entità rappresentate in documenti, dataset...

Ci sono delle repository collaborative che offrono un semplice schema per specifiche descrizioni di semantica, come schema.org

Json-LD
Offre una via semplice per aggiungere significato semantico, attraverso proprietà @
- @content
- @id
- @type

Caratteristiche
- indipendenza
- intercambiabilità
- scalabilità
- manipolabilità
- tutto può essere una risorsa

Linked data
Sono una collezione di best practice per pubblicare e connettere dati strutturati sul web

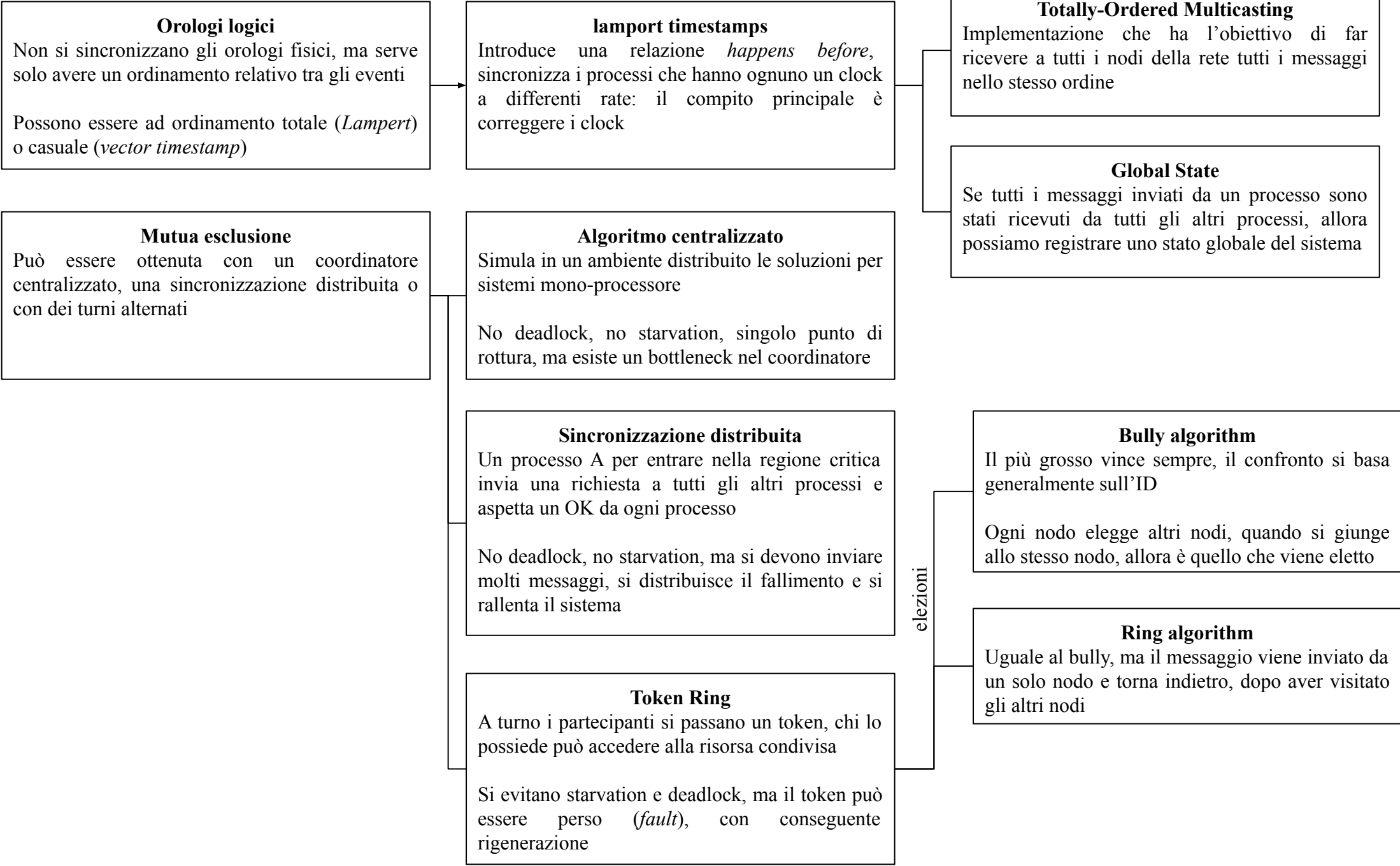
Fact
È una relazione tra un soggetto e un oggetto, o una proprietà e un'entità

Grafi
Si tende a rappresentare i fatti attraverso dei *knowledge graphs*, entità correlate e connesse da relazioni

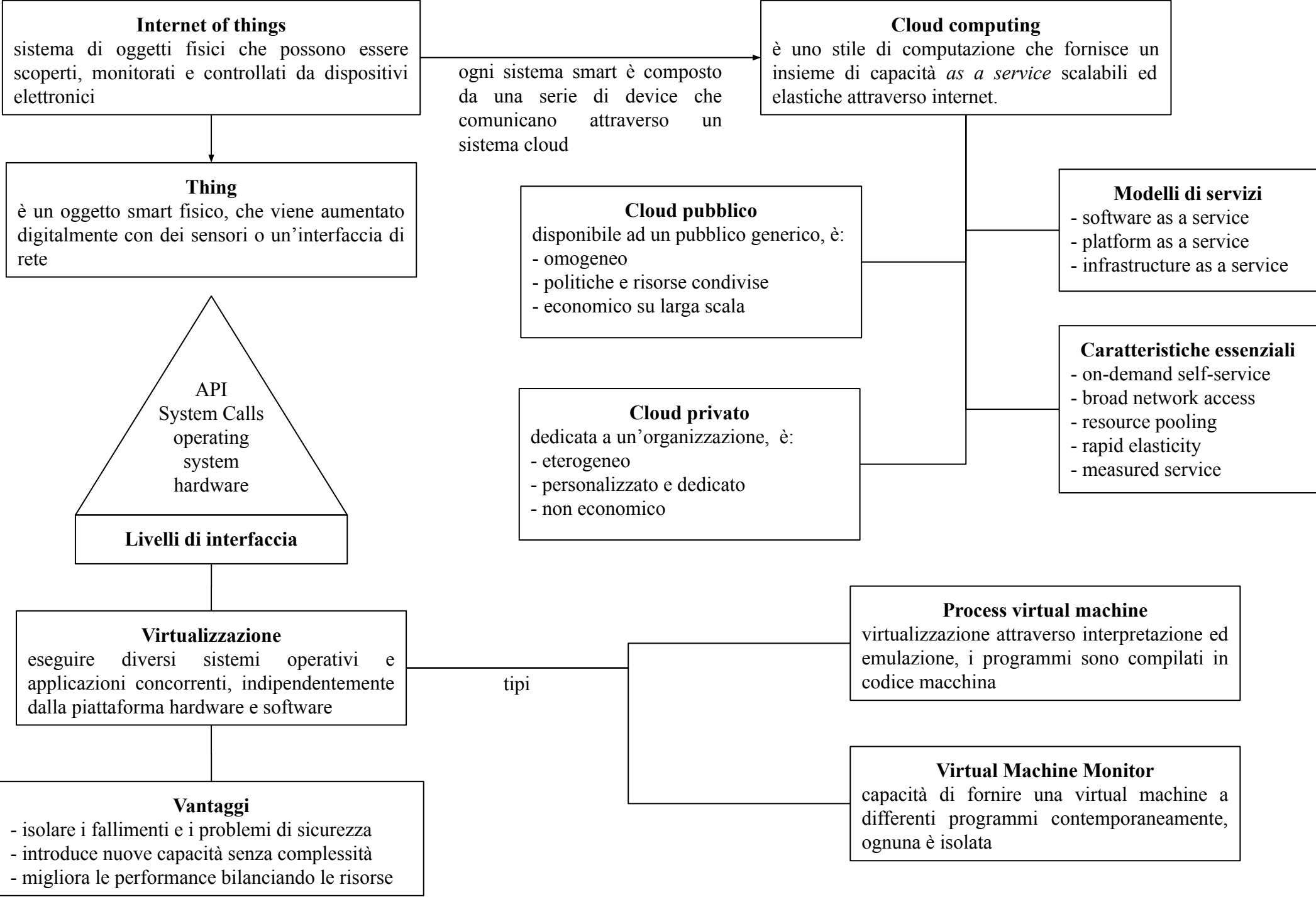
Sono efficienti da processare e offrono una più conveniente astrazione

RDF
Modello standard per rappresentare dati sul web, ovvero un linguaggio general-purpose per rappresentare fatti nel web
- triples
- URI
- statements

URI
Stringa di caratteri usati per identificare un nome o una risorsa:
- URL per identificatori che sono locatori
- URN, risorse indipendenti da locazioni e persistenti







Microservizi

Si basa sul *principio della singola responsabilità*, è un pattern semplice e leggero che minimizza le comunicazioni

Si suddivide l'applicazione in una collezione di piccoli servizi interconnessi, e la comunicazione è affidata ad un *API Gateway*

Benefici

- contrastano il problema della complessità
- sviluppo in team indipendenti
- ogni microservizio può essere distribuito indipendentemente dagli altri

Monolite

Insieme di logica di business, adapters e API che sono impacchettati e distribuiti come un monolite

Problemi

- difficili da mantenere ed evolvere
- problemi non isolati
- ha un costo complessivo maggiore

Regole di Martin Fowler

- la maggior parte dei sistemi software sono monoliti
- non confondere modularità di un monolite con dei microservizi
- non usare microservizi in un sistema non troppo complesso come monolite

Container

È una modalità standard per impacchettare un'applicazione e tutte le sue dipendenze

Unità leggere e non legate a un'infrastruttura specifica

Docker

Piattaforma aperta per costruire applicazioni distribuite

Image

Read-only snapshot di un container

Container

Unità standard in cui le applicazioni risiedono

Docker Hub

immagazzina, distribuisce e condivide immagini per container

Docker Engine

Un programma che crea ed esegue container

Costituito da dei file system impilati, alla cui base c'è un *boot*. Utilizza la *union mount* per permettere a diversi filesystem di essere montati insieme

Pattern copy on write

Appena un container viene avviato, il layer read-write è vuoto. Appena avviene un cambiamento, viene applicato il layer e utilizzato

Applicazione client-server con un server con un *processo demone* e una *REST API* che viene utilizzata attraverso dei *comandi CLI*.