

# WSOS (ex TSDW): **W**eb & **S**ervice **O**riented **S**ystems (6 CFU) con laboratorio (3 CFU, prof. S. Nicotra)

*Giuseppe Pappalardo*

- **Sistema Distribuito**: sistema software i cui componenti interagiscono attraverso la **rete**
- **Architettura** tipica di un SD: **client-server**
- Il corso **WSOS** presenta le principali **tecnologie** disponibili per l'**interazione** e l'**interconnessione** tra i componenti di un SD
- La principale di queste è il **Web** (protocollo HTTP), costruito su **Internet** (TCP) e le **socket** di rete

# Tecnologie per i Sistemi Distribuiti

- **Socket** (API di "astrazione", per accesso *programmer-friendly* a TCP)
  - un componente invia un messaggio di richiesta, un altro risponde
    - in C/Unix (ambiente "nativo" delle socket) e in Java (nel corso di Laboratorio)
- **RPC** - chiamata di funzione/metodo remoti, un componente invoca un metodo, un altro componente, **remoto**, lo esegue
  - RPC per C/Unix, RMI per Java
- **Thread** – per applicazioni distribuite ad **alta disponibilità**
  - consentono a più componenti di eseguire **attività parallele**, indipendenti o interagenti
    - Thread Java (nel corso di Lab.)

# WSOS: architetture per il Web

Architettura **3-tier** per interazione via **HTTP/Web**

- Tier 1: il **cliente**, un **browser** Web (con Javascript)
  - ▮ principalmente nel corso di Web Programming
- Tier 2: il **server** (Web server: Apache, Nginx, MS IIS) e/o **Application Engine** (PHP, Tomcat (Java), etc.) che esegue la *business logic* (in Java (corso Lab), PHP, Python ?...)
- Tier 3: un **database** (es. mySql/mariaDB) con i dati

Allineandoci allo stato dell'arte, potremo implementare (v. corso TAP) i tier 2 e 3 anche come **container** Docker (efficienza e indipendenza da piattaforma)

# WSOS: tecnologie per il Web

- **PHP** per il Tier 2 – *business logic*
- **Servlet / Web App** (Java, corso Lab.) per Tier 2 – *business logic*
- Pattern di fruizione *programmatica* di **servizi remoti** (p. es. servizi Google, Amazon, ChatGPT, etc.) da codice:
  - interazione **REST** via HTTP tra cliente e servizio; p.es. l'host <http://biblioteca.dmi> offre il servizio, sul port 80 e:
    - il messaggio HTTP GET /books elenca i libri
    - il messaggio GET /books/23 mostra il libro 23 dell'elenco
    - il messaggio DELETE /books/23 elimina il libro 23
    - POST /books inserisce un nuovo libro, i cui dati fanno parte del messaggio POST, spesso in formato *JSON*
  - **JSON**: linguaggio Javascript di codifica per scambio dati

# WSOS: tecnologie per il Web - MVC

- Pattern **MVC** (**M**odel-**V**iew-**C**ontroller) per app Web
  - **M**: interazione con DB, **V**: interazione con il cliente, **C**: business logic (implementa le funzionalità dell'app)
  - MVC si sposa bene a REST
  - MVC con PHP/**Laravel**
  - M\_C con Java **Spring Boot** (corso Lab); per la V si ricorre al framework **ThymeLeaf**
  - MVC con Python/**Django** ?
- App Web MVC (p.es. Laravel) e cliente "**thin**" (browser e HTML, no JavaScript)
- Alternativa (cliente "**rich**")
  - API Web (es. app Spring Boot) implementa **M**odel/**C**ontroller
  - **V**iew assicurata da cliente "**rich**" (browser con Javascript)

# Modalità d'esame

## Tematiche d'esame

1. Socket (C o Java, a scelta dello studente)
2. Thread (Java)
3. PHP
4. Java Servlet
5. Laravel
6. Spring Boot

## Modalità (appelli): prove di laboratorio. Quali?

- Selezione random 1-6
- Selezione tipica: (1 oppure 2) + (3,6 oppure 4,5)
- Altra selezione tipica: 3,6 oppure 4,5

# Prove in itinere

## **Prove di laboratorio o esoneri, in itinere:**

1. Socket (C o Java, a scelta dello studente)
2. Thread (Java)
3. PHP
4. Java Servlet
5. Laravel
6. Spring Boot

**Quali sostenere:** ad arbitrio dello studente (modularità totale)

**Validità:** l'anno accademico (...solare!)

**Vincoli:** nessuno, un esonero conseguito in itinere non si perde anche se non si supera il successivo esame

## **Esame:**

- breve colloquio per chi ha superato 1-6
- per gli altri prova di laboratorio sui temi non esonerati in itinere

# Piattaforma per studio: quale?

**Linux!!!** (anche perché è quella di ExamBox). Oppure...

**Unix**, p.es. OsX!! Oppure...

**Windows...** Come?

- WSL (Windows Subsystem for Linux, macchina virtuale HyperV con Linux installato, p.es. Ubuntu): a volte si presenta qualche problematica di configurazione, oppure...
- XAMPP (<https://www.apachefriends.org/>): Apache, Mysql, PHP, PERL; qualche difficoltà per Laravel, oppure...
- Herd (per Laravel, ha PHP ma serve qualche spiegazione in più per usarlo, non ha mysql), oppure...
- VM (VirtualBox?) Linux (occorre hw performante), oppure...
- container Docker (per mysql, PHP...)