WEB Platform - Layered System constraint

CS2B01 - Desarrollo Basado en Plataformas - Unidad 2

Dr. Jesus Bellido
jbellido@utec.edu.pe

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Logros

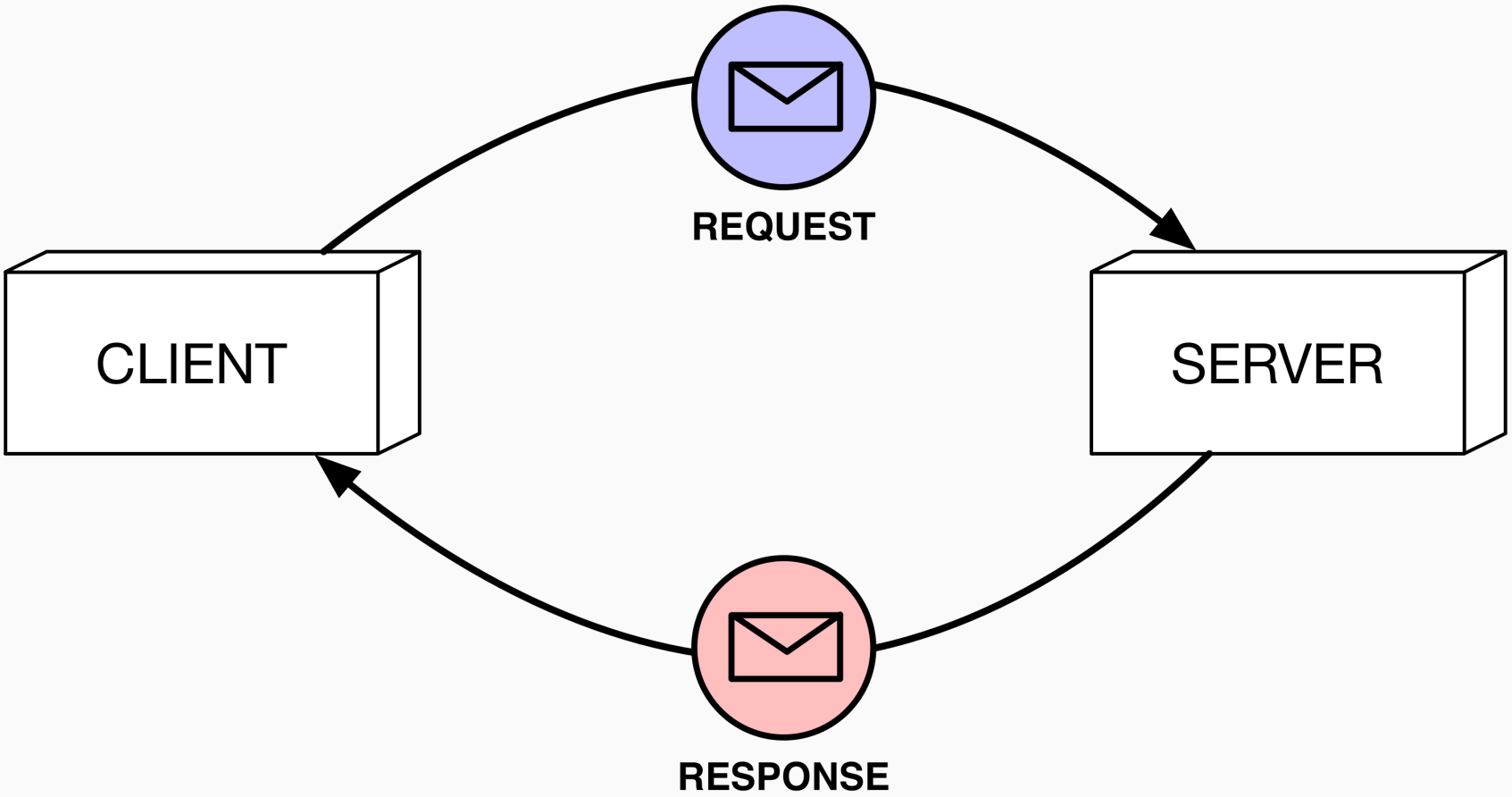
Logro de esta Sesión

Al finalizar esta unidad usted estará en la capacidad de:

- Diseñar e implementar una aplicación web sencilla
- Describir las restricciones que la web pone a los desarrolladores.
- Comparar y contrastar la programación web con la programación de propósito general
- Describir las diferencias entre software como un servicio y productos de software tradicionales.
- Discutir cómo los estándares de web impactan el desarrollo de software.
- Revisar una aplicación web existente con un estándar web actual.

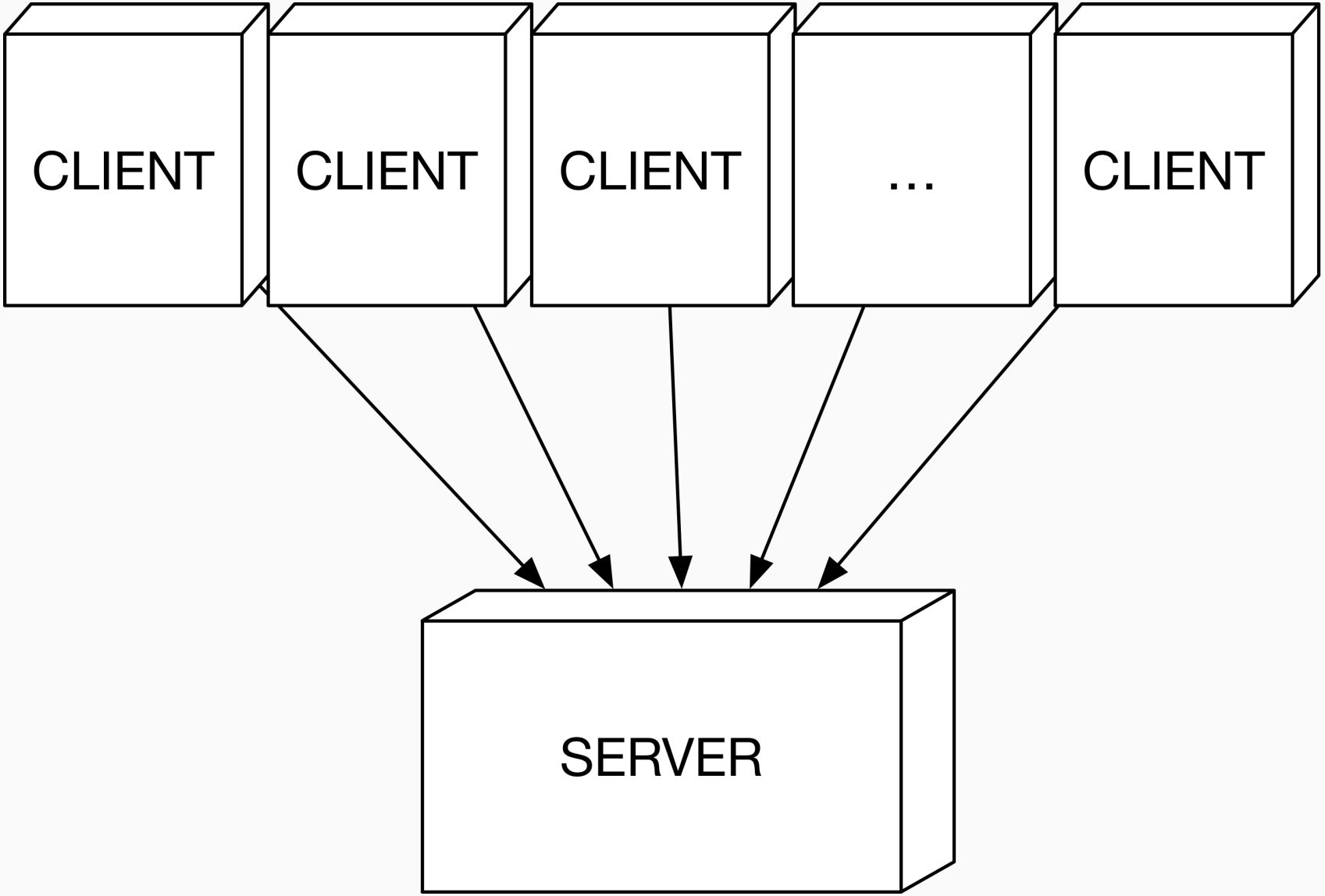
Introducción

Client-Server



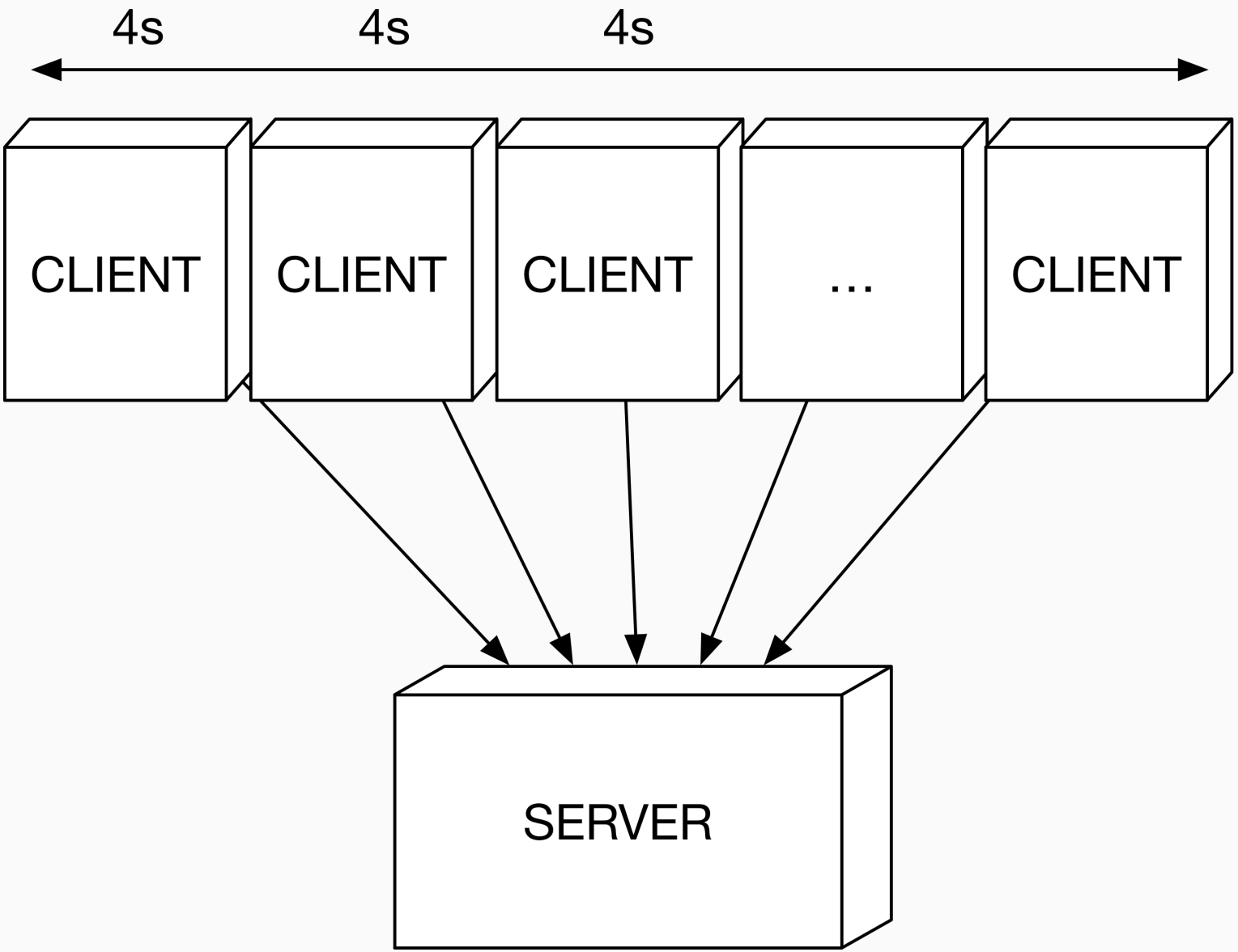
... pero ¿a cuántos clientes sirve un servidor?

How many clients can servers handle at a time?



¿todos los clientes envian sus solicitudes al mismo tiempo?

How many clients can servers handle at a time?

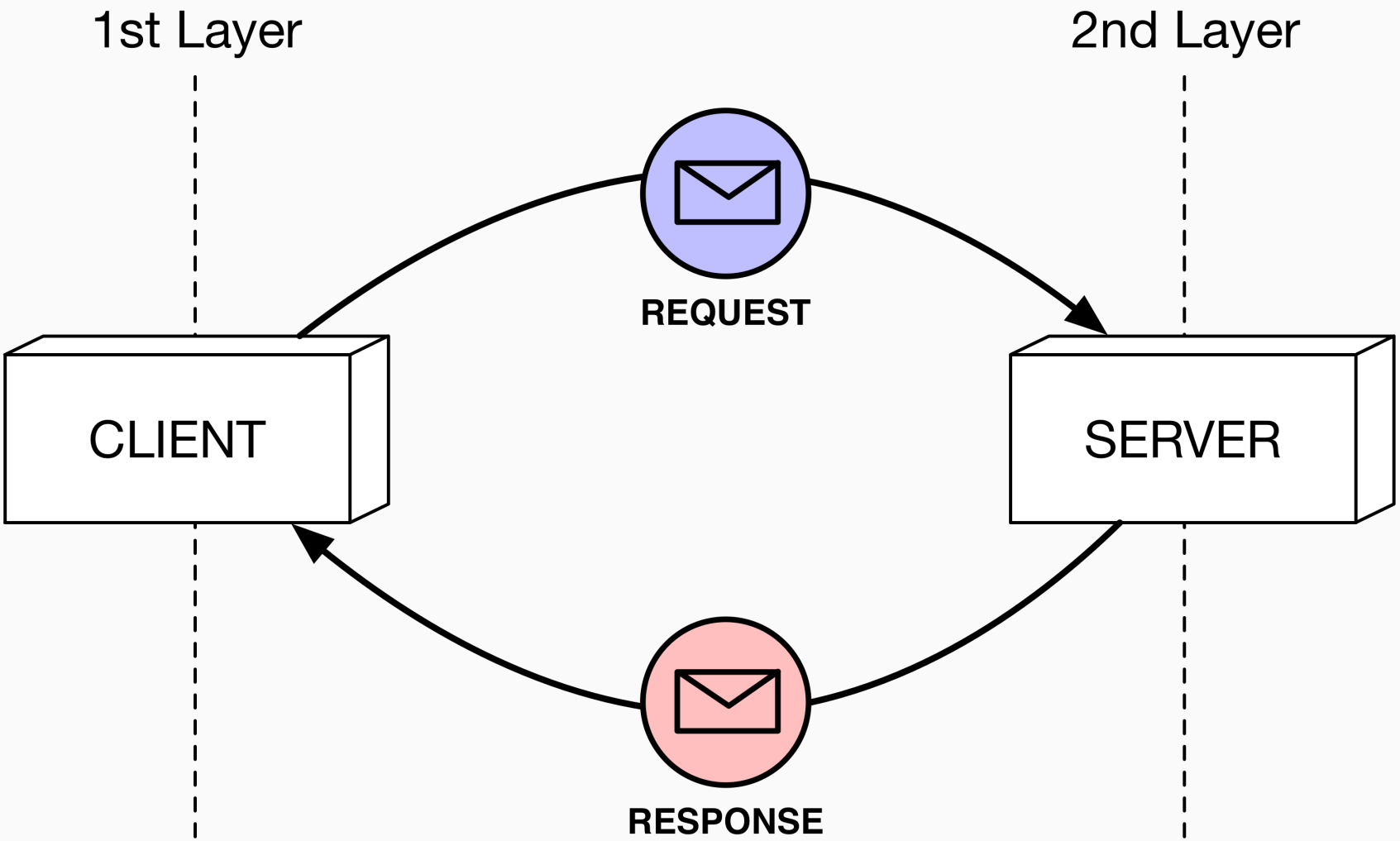


Group Work

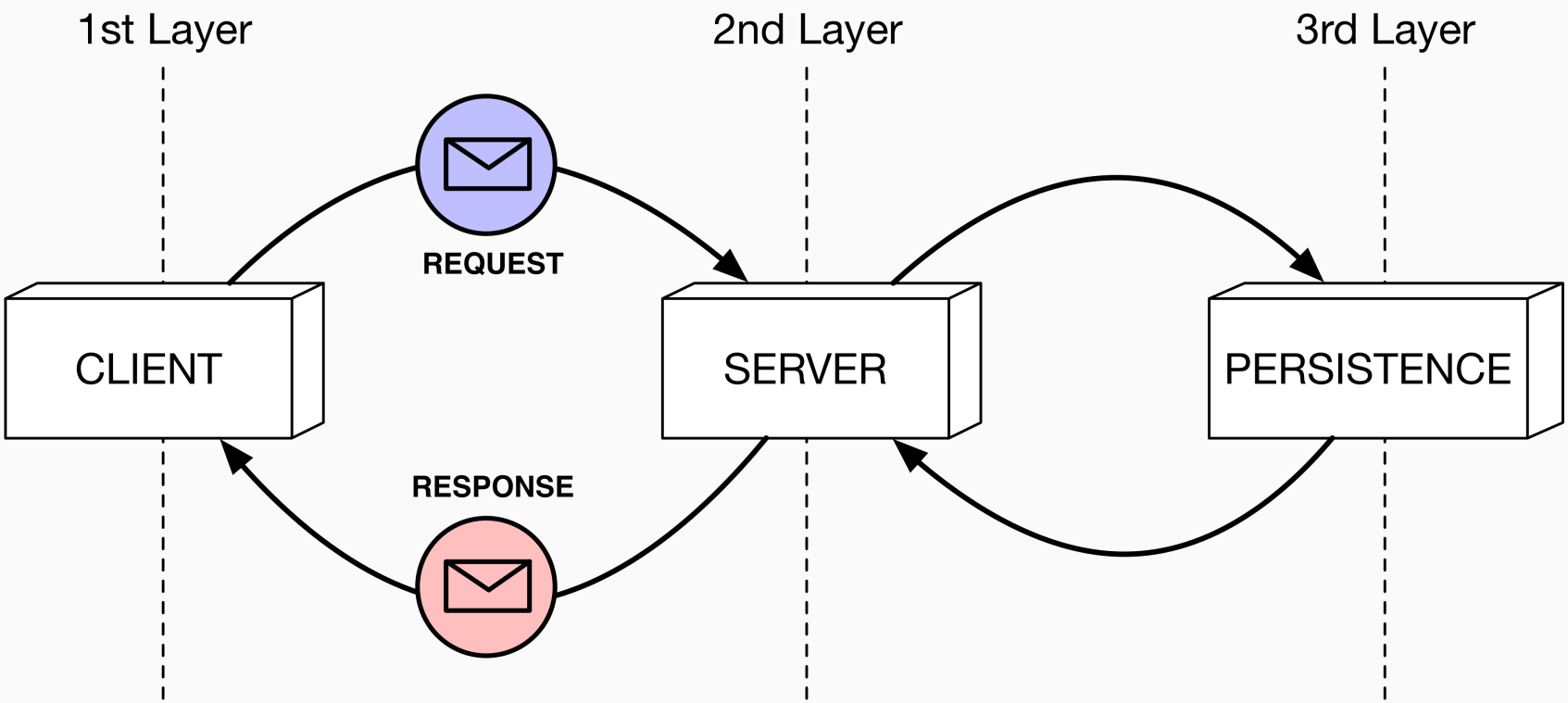
- El servidor se encarga de realizar cuatro tipo de operaciones: suma, resta, multiplicación y división.
- El cliente envía las solicitudes con problemas que sólo pueden contener un tipo de operación
- En grupos de 9 personas intenta resolver el problema de atender la demanda de 0.25 request/second.

Layered System

Client-Server: two layers



Client-Server: three layers



Persistence Layer 1

```
1  from sqlalchemy import Column, Integer, String
2  from sqlalchemy.ext.declarative import
       declarative_base
3
4  Base = declarative_base()
5  class User(Base):
6      __tablename__ = 'users'
7      id = Column(Integer, primary_key=True)
8      name = Column(String(50))
9      fullname = Column(String(50))
10     password = Column(String(12))
```

Persistence Layer 2

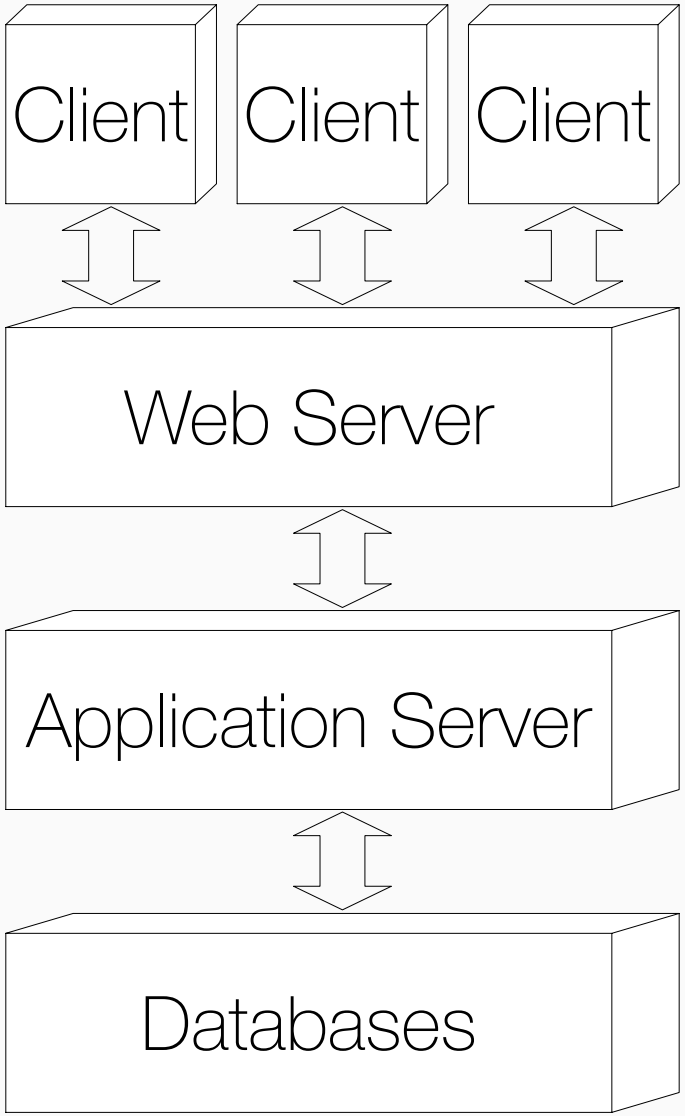
```
1  ...
2  from sqlalchemy import create_engine
3  from sqlalchemy.ext.declarative import
       declarative_base
4  from sqlalchemy.orm import sessionmaker
5
6  engine = create_engine('sqlite:///:memory:', echo=
       True)
7  Base.metadata.create_all(engine)
8
9  Session = sessionmaker(bind=engine)
10 session = Session()
```

Persistence Layer 3

```
...
user1 = User(id=1, name='ed', fullname='Ed Jones',
    password='hola123')
user2 = User(id=2, name='jb', fullname='Je Belli',
    password='bye123')
session.add(user1)
session.add(user2)
session.commit()
```

Ventajas y Desventajas

Cliente Servidor (N Capas)



Ventajas y Desventajas

Table 1: Revisión de los Atributos de Calidad

Atributo de Calidad	Ventajas
Disponibilidad	Réplica de servidores en cada capa, si uno falla, los otros responden. La calidad del servicio disminuye hasta restaurar el servidor fallado.
Manejo de Fallas	La mayoría de servidores web y de aplicación implementan una recuperación transparente (el cliente es redireccionado a un servidor replicado para responder la solicitud).
Flexibilidad	La modularidad facilita el cambio, debido a la encapsulación (presentación, negocios y datos). El impacto del cambio al interior de una capa debería ser mínimo en las otras.
Desempeño	Alto desempeño mediante replicación de software (threads concurrentes), velocidad de la conexión entre las capas y la cantidad de datos a ser transferidos. Es recomendable minimizar las llamadas entre las capas al procesar cada solicitud.
Escalabilidad	Como los servidores pueden replicarse, la arquitectura escala bien. En la práctica, la gestión de datos se convierte en el cuello de botella.

Implementación

Persistence Layer

Individual Work

- Implementa la capa de persistencia en tu aplicación de chat para guardar los usuarios registrados.
- Modifica tu algoritmo de validación teniendo en cuenta la nueva capa de persistencia.

Resumen

1. Logros
2. Introducción
3. Layered System
4. Ventajas y Desventajas
5. Implementación

Preguntas