Estilos y Patrones V

- IIC2173 - Arquitectura de sistemas de software
- Departamento de Ciencia de la Computación
- Escuela de Ingeniería – PUC
- Nicolás Acosta – Gerardo Crot

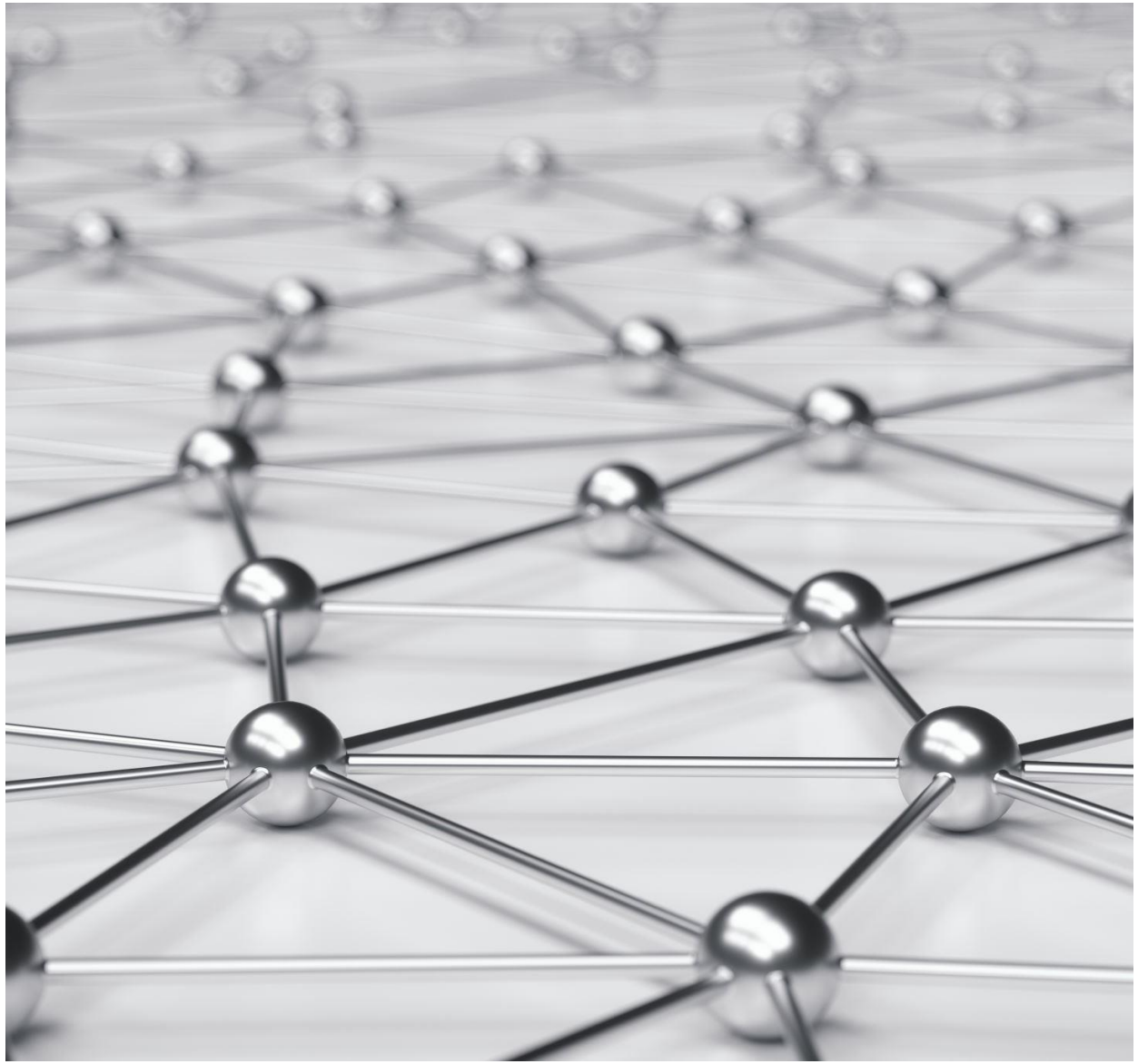


Estilos comunes

1.Influenciados por el lenguaje
    1.1 Main y sub-rutinas
    1.2 Orientado al objeto
2. En capas
    2.1 Máquinas virtuales
    2.2 Cliente - Servidor
3 Microkernel
4. Flujo de datos
    4.1 Por Lotes, secuencial
    4.2 Tubería y Filtro (Pipe & Filter)
5. Memoria compartida
    5.1 Pizarra
    5.2 Basado en reglas
6. Intérprete
    6.1 Intérprete
    6.2 Código móvil
7. Invocación implícita
    7.1 Basado en eventos
    7.2 Publisher – Subscriber
8. P2P (Peer to Peer)
9. Derivados
    C2, Corba

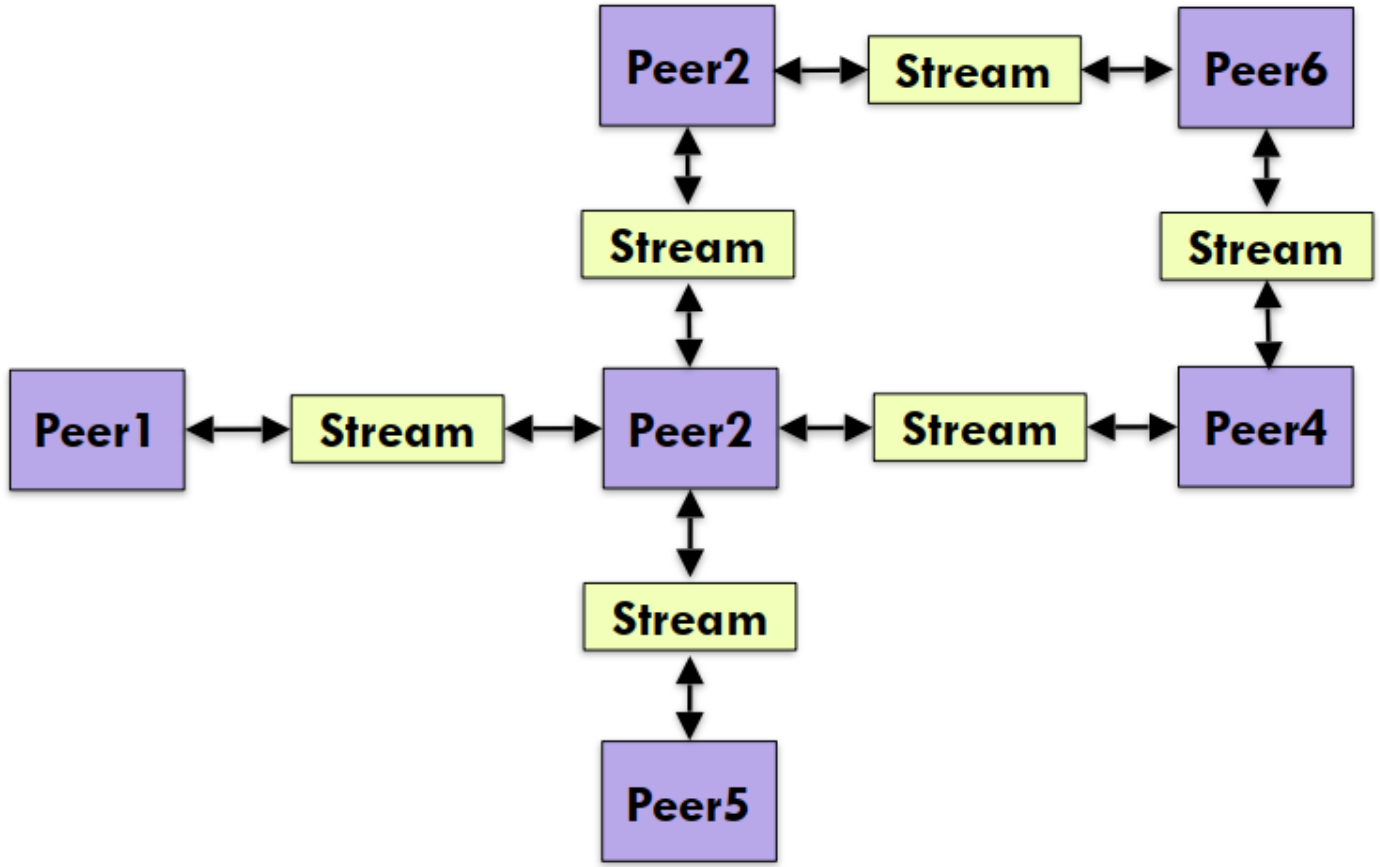
Peer-to-Peer (P2P)

- Estilo descentralizado, con flujos de control y recursos distribuidos entre pares
- Concepto de Peers: Máquinas idénticas que hacen tanto de cliente como servidor
- Sin necesidad de un componente central



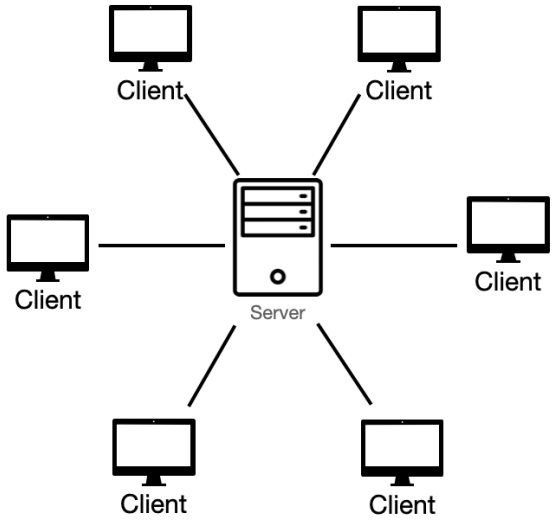
P2P: Características

- Componentes: Peers auto-organizados
  - Componentes independientes, con un estado propio y un hilo de ejecución propio
- Conectores
  - Protocolos de red
- Datos
  - Mensajes de red
  - Blobs

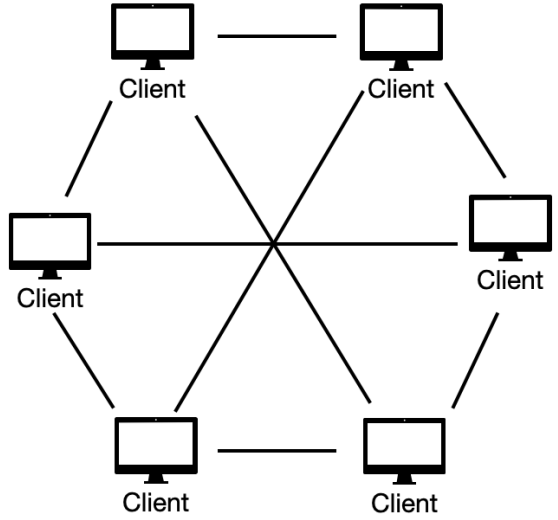


P2P vs Cliente-Servidor

- Topología
  - Conexiones de par a par: interfaces idénticas
  - Puede existir redundancia
  - No hay regulación en cantidad y duración
- Computación descentralizada
- Puede existir composición con estilos
- Múltiples formas de organizarse



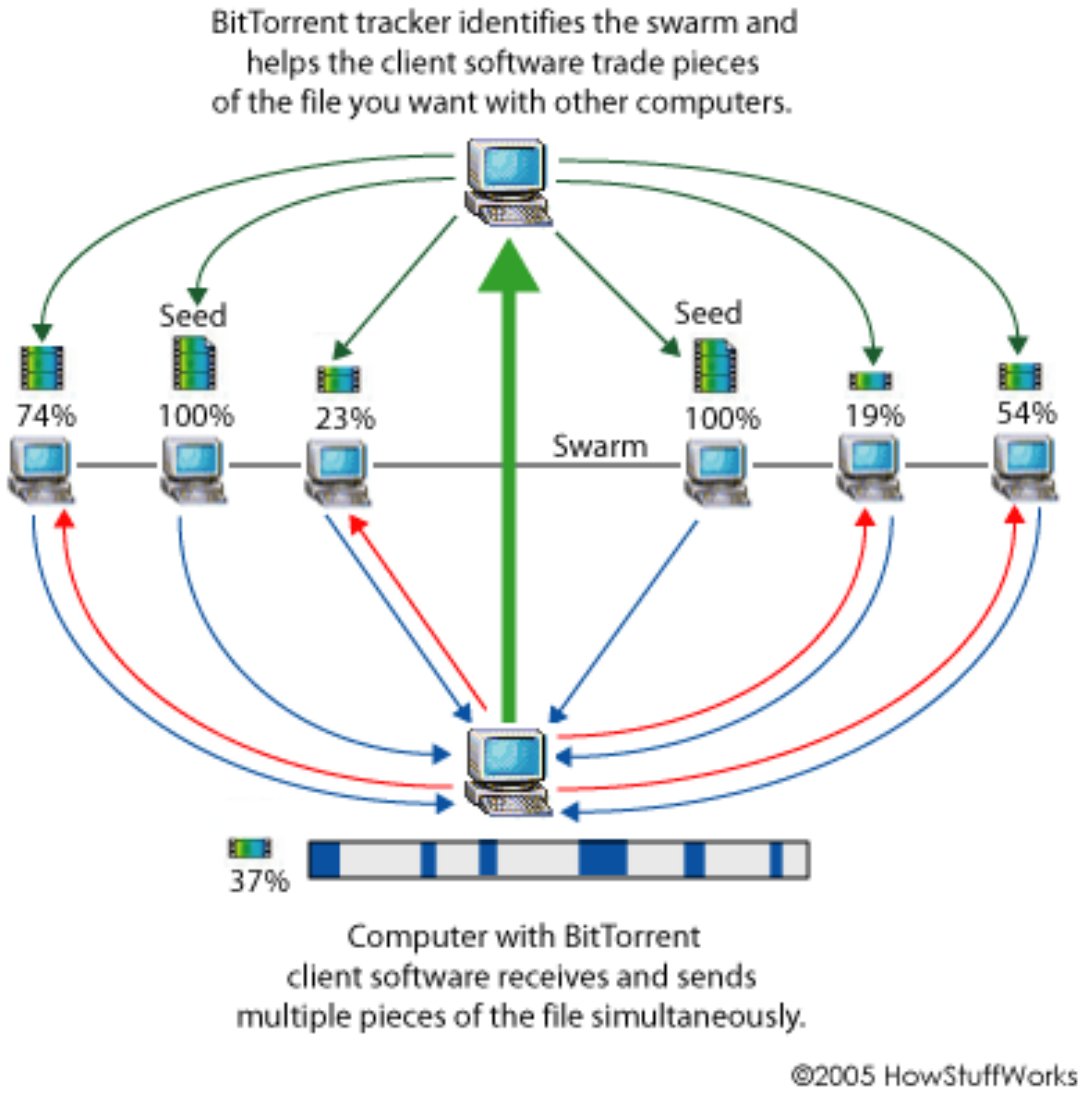
Client Server Architecture

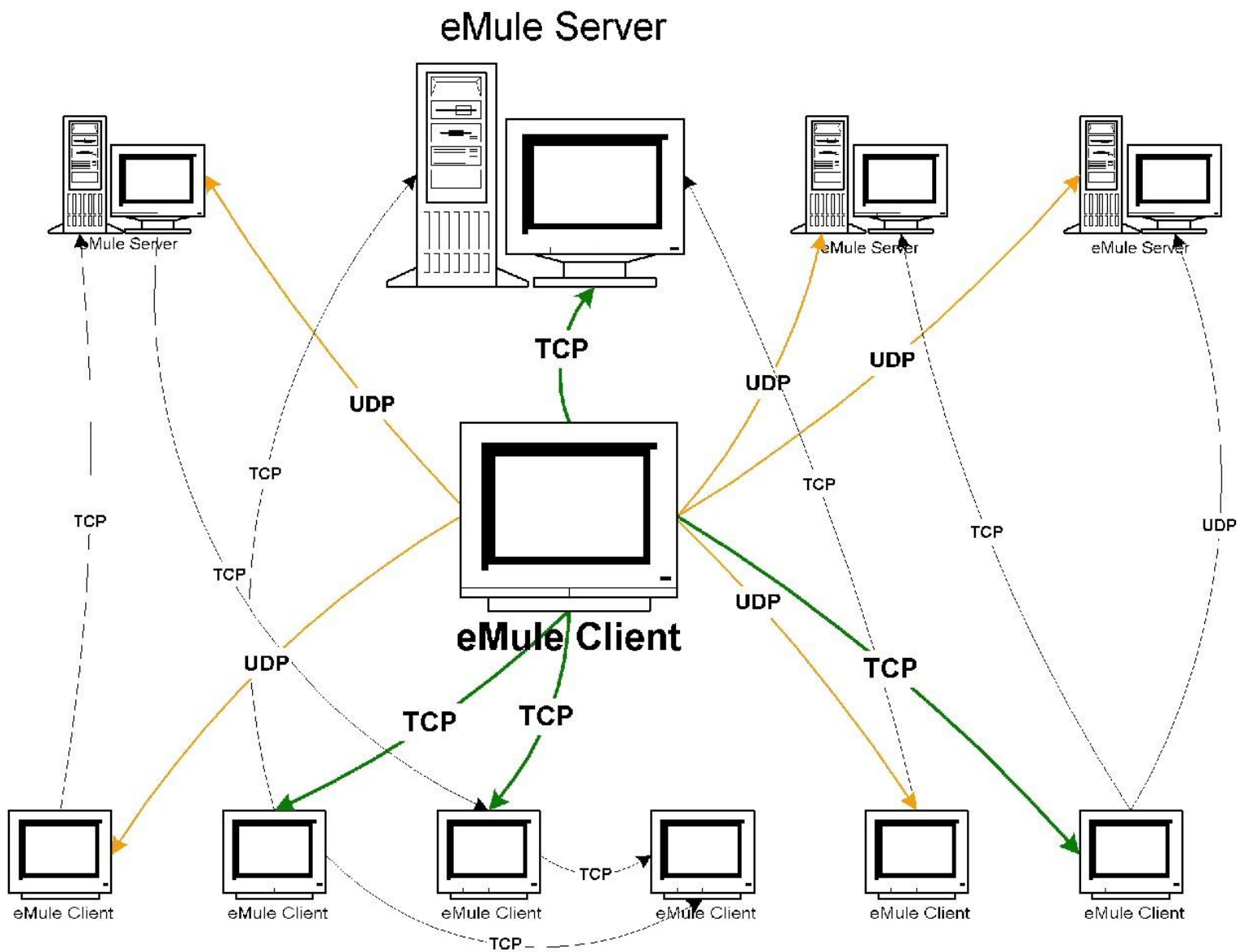


P2P Architecture

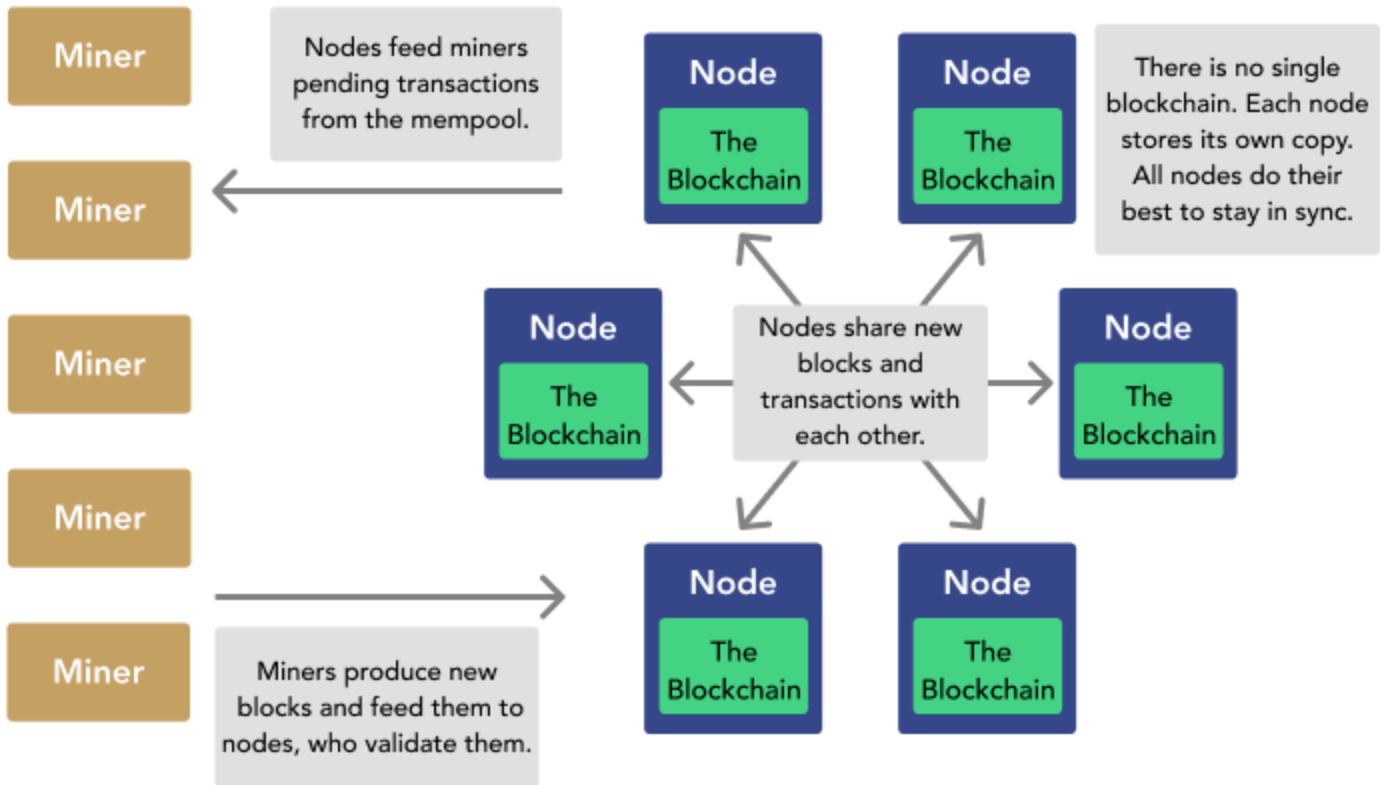
Ejemplos

- No estructuradas
  - BitTorrent en sus inicios
- Estructurados: Topología y organización predeterminada
  - Bitcoin
- Híbridos
  - Bittorrent + trackers + DHT
- Algunos casos iniciales
  - Spotify
  - Skype




eMule Server
eMule Server
eMule Server
eMule Server
TCP
UDP
UDP
UDP
TCP
TCP
TCP
TCP
TCP
UDP
UDP
eMule Client
TCP
TCP
TCP
UDP
TCP
TCP
eMule Client
eMule Client
eMule Client
eMule Client
eMule Client
eMule Client

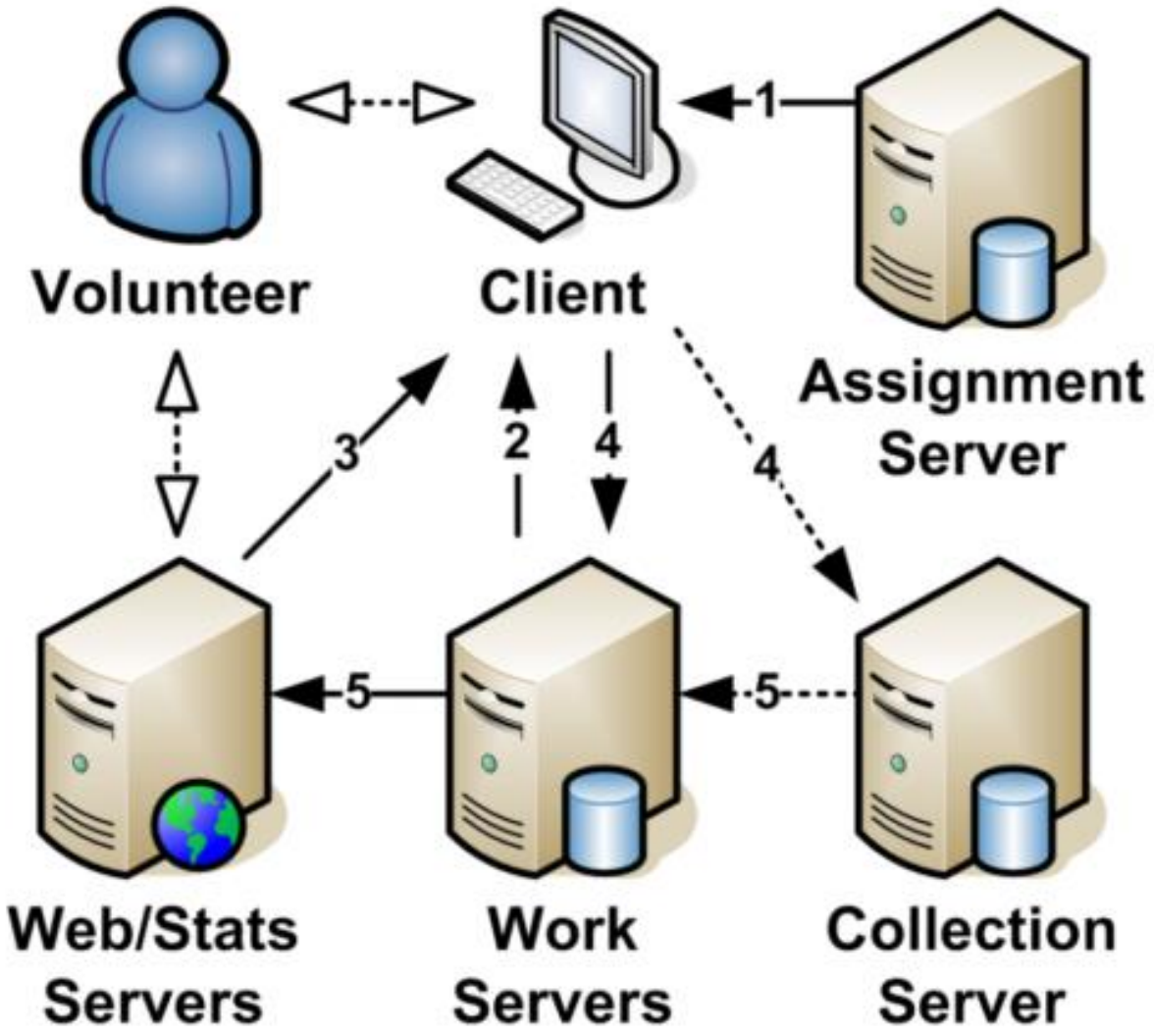
Bitcoin Network Interactions



https://river.com/learn/what-is-a-bitcoin-node/

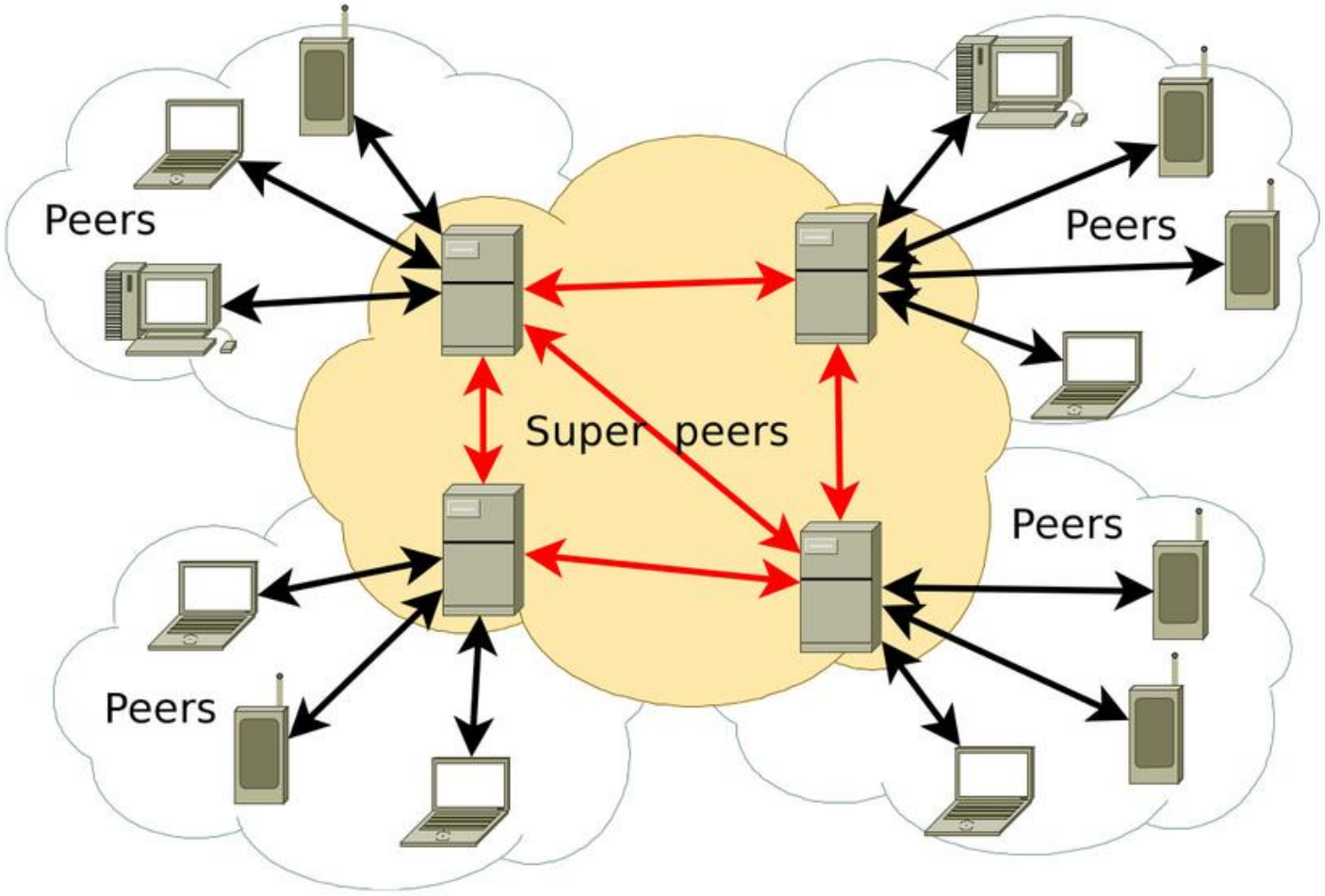
P2P: Pros

- Redundancia
  - Beneficia disponibilidad
  - Tolerancia a caída de nodos
  - Excepción: Consistencia
- Escalabilidad
  - Nodos pueden entrar con facilidad
  - DHT puede ayudar
- Performance
- En algunos casos: Transparencia y anonimidad



P2P: Cons

- Complejidad
  - Requieren protocolos anti caídas y de reinserción
- Seguridad
- Confiabilidad/Durabilidad
  - En escenarios de escasez (e.g. archivos distribuidos) pueden perderse trozos o archivos completos



¿Cómo elegir el estilo más apropiado?

- **¡Depende!**
- Todos los estilos tienen sus propios *trade-off*
- Pero existen algunos consejos generales



Algunos consejos

- Tengan en consideración y entiendan el estándar del momento en la industria
  - Usualmente se usan por varios motivos
  - Pero hay que saber si aplica a nuestro contexto (no debemos agregar *Blockchain* a todo solo porque se use)
- Tengan conocimiento de dominio en gran medida
  - Indicaciones de cómo se usará su aplicación
- Piensen en cómo se transformarán y enviarán los datos
  - Hay estilos que pueden servir más que otros



Algunos consejos

- No descarten la estructura de su organización
  - Quizás un estilo sea muy caro de mantener
  - ¿Podrán contratar más personal a futuro?

- Reconozcan situaciones similares entre sus sistemas y los estilos
  - Si mi sistema requiere capacidad de *customización* por usuario, me serviría implementar Microkernel
  - Mi sistema debe ser sumamente escalable, para evitar acoplamientos podría implementar un EDA



Estilos y Patrones

- Preguntas finales que se pueden hacer:
  - Hay veces en las que una quanta funciona, ¿estoy dispuesto a asumir eso?
  - ¿Dónde almacenaré mis datos? ¿Debo tener una única base de datos?
  - En los puntos clave de mi sistema, ¿cómo será la comunicación? ¿síncrona o asíncrona?
- No traten de abarcar todo, elijan sólo los principales
- Habrá más de una solución posible a una buena arquitectura, lo importante es que sepan justificar sus decisiones y convencer



Estilos y Patrones V

- IIC2173 - Arquitectura de sistemas de software
- Departamento de Ciencia de la Computación
- Escuela de Ingeniería – PUC
- Nicolás Acosta – Gerardo Crot

