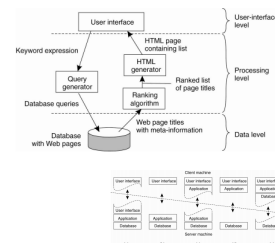
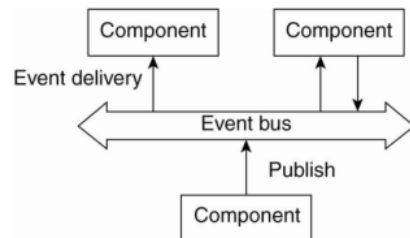
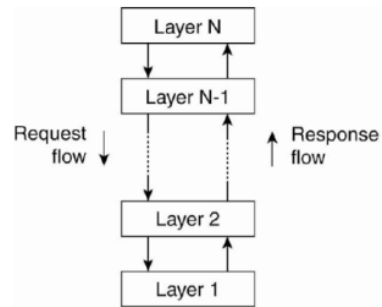


2. Architectures

Ideen bag organiseringen af et DS kaldes en arkitektur.

Software arkitektur (forsøger at give transparens) – logisk (sw). Opbygget af komponenter.

1. **Lagdelt arkitektur:** Lagene kan kalde nedad og modtage svar opad.
2. **Objektbaseret arkitektur:** Hvert objekt er en komponent, og de er forbundet via RPC
3. **Data-centrerede arkitekturer:** Man snakker igennem et fælles repository.
4. **Hændelsesbaserede arkitekturer:** Man sender hændelser rundt i systemet, publish/subscribe – dette giver en løs kobling (reference afkobling).
5. **Delt data-space:** Skyen. Kombi af data-centreret og event-baseret.



System arkitektur (realiseringen af en software arkitektur) – fysisk (hw)

- **Centraliseret:** Klient-server - requests/replies. Realiseringer af den lagdelte arkitektur. Vertikal opdeling.
 - **App layering:** Generelt eksempel på opdeling af opgaver. [TRANSPARENT]
 - **Multitieret:** Måder at realisere App layering på. [TRANSPARENT]
- **Decentraliserede: P2P:** Klient er også server. Horizontal opdeling. Symmetrisk kommunikation → overlay netværk
 - **Overlay:** Logisk netværk ovenpå det fysiske. Intern kommunikation er lettere. Muligheder for algoritmer indenfor netværket. Tager ikke hensyn til maskinernes fysiske placering, medfører:
 - Link stress:** Hvor ofte en multicast pakke krydser det samme link.
 - Stretch:** Forholdet mellem forsinkelsen i overlay netværk mod det underliggende.
 - Tree cost:** Samlede omkostning ved træet. Ved at minimere det, optimeres træet.
 - Proces = knude. Kommunikation gennem direkte forbindelser knuder imellem.
 - **Struktureret:** Fx Chord-netværket, hvor både data-enheder og knuder tilknyttes en identifikator fra et identifikator-space. Data-enhed med identifikator key, knyttes til den knude med id>key. [TRANSPARENT]
 - **Ustruktureret:** Her laves overlaynetværket af randomiserede algoritmer. Problemet er at floodet netværket request for at finde en node (brug superpeers).

