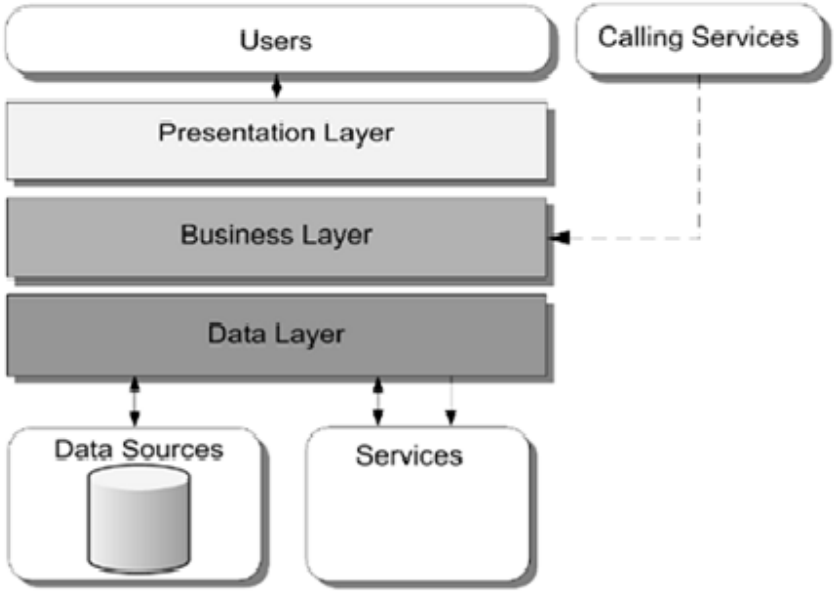
Distributed architectures

A very brief overview

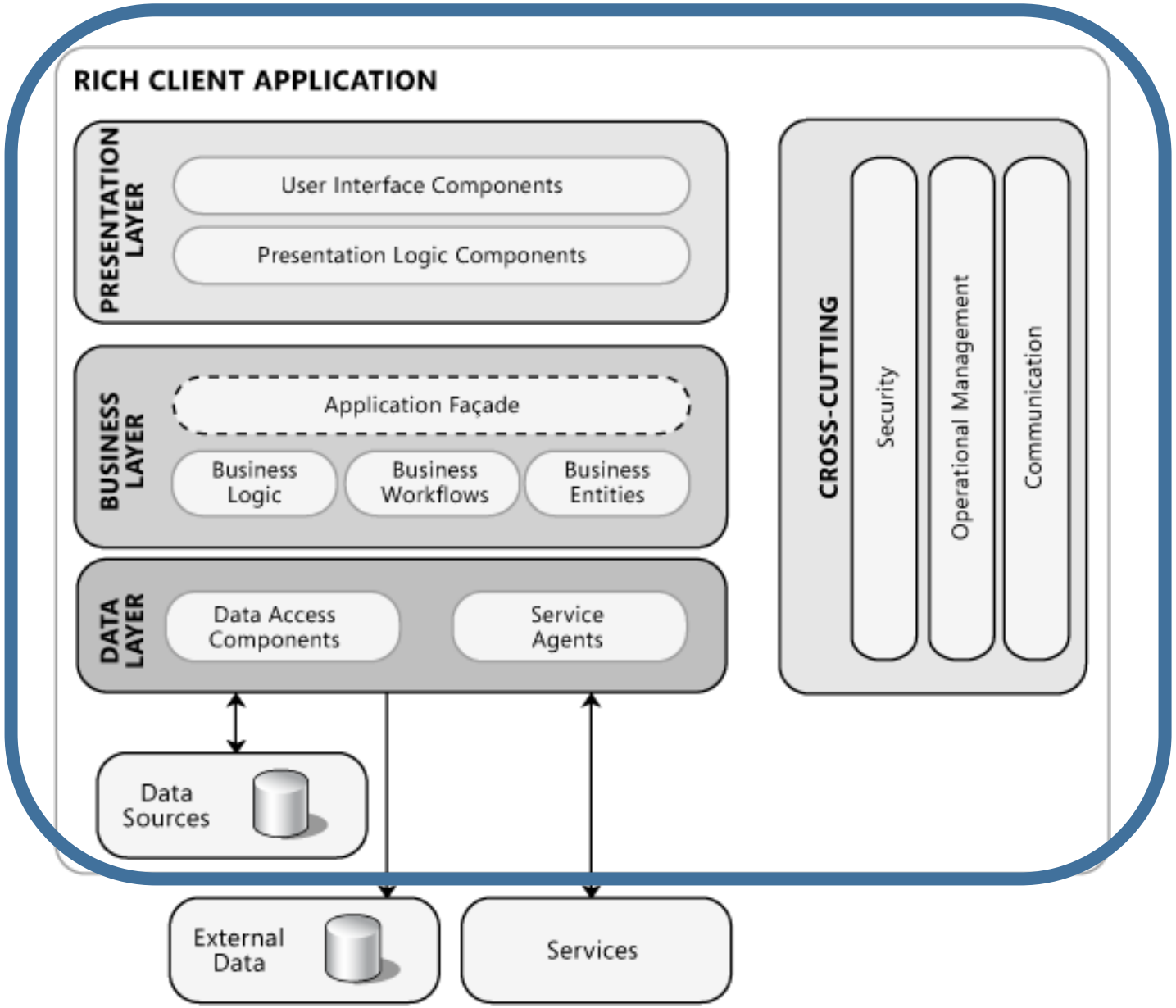
Layers / Tiers

- An application is often divided into layers.
- The layers might be on different hosts connected by a network
  - The layers are often called Tiers, then
  - Example
    - 2sem: Application –REST - Database
    - Browser – Web/application server (middle tier) - Database



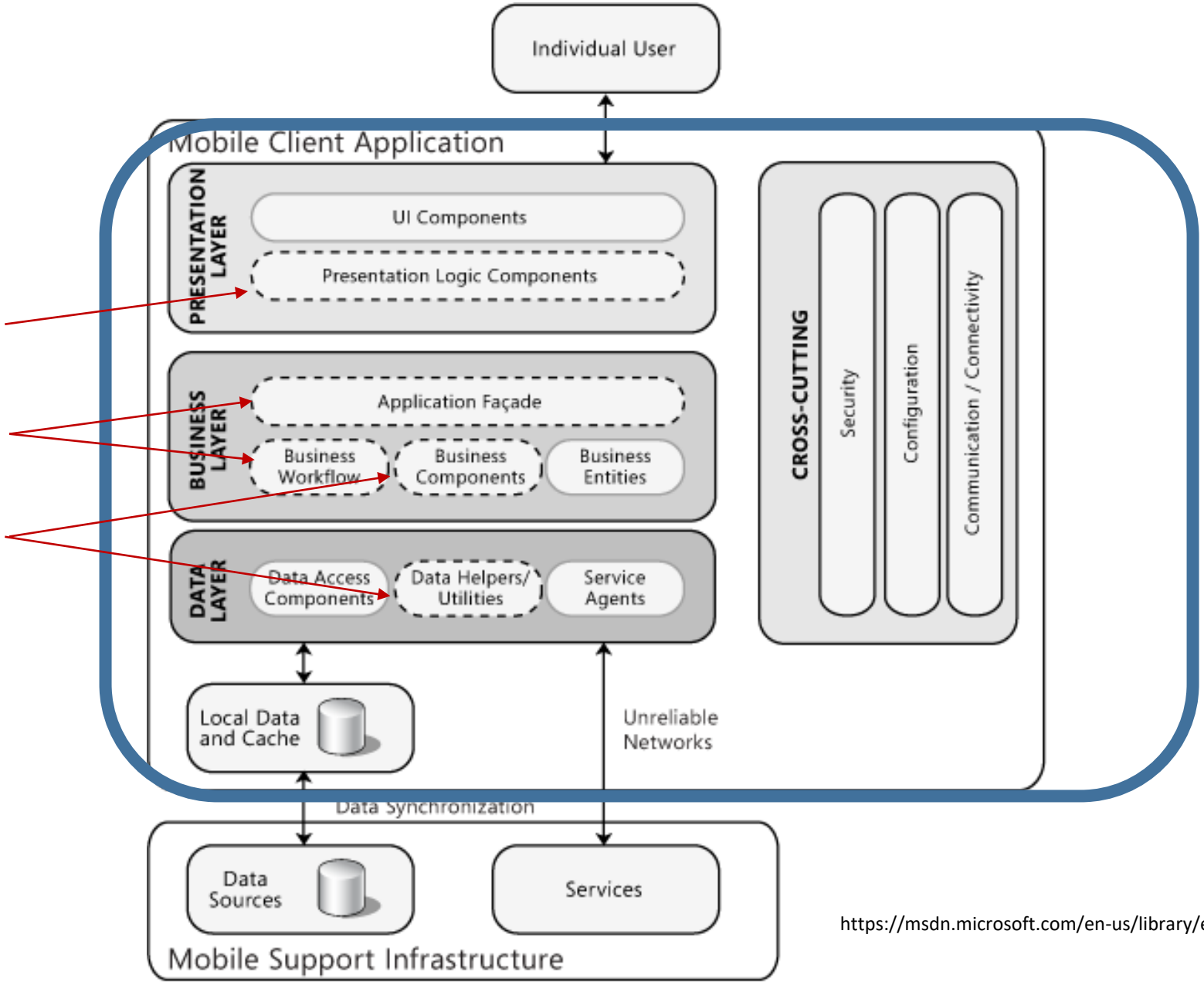
http://www.guidanceshare.com/wiki/Application_Architecture_Guide_-_Chapter_9_-_Layers_and_Tiers

Example Rich/Fat/Thick applications



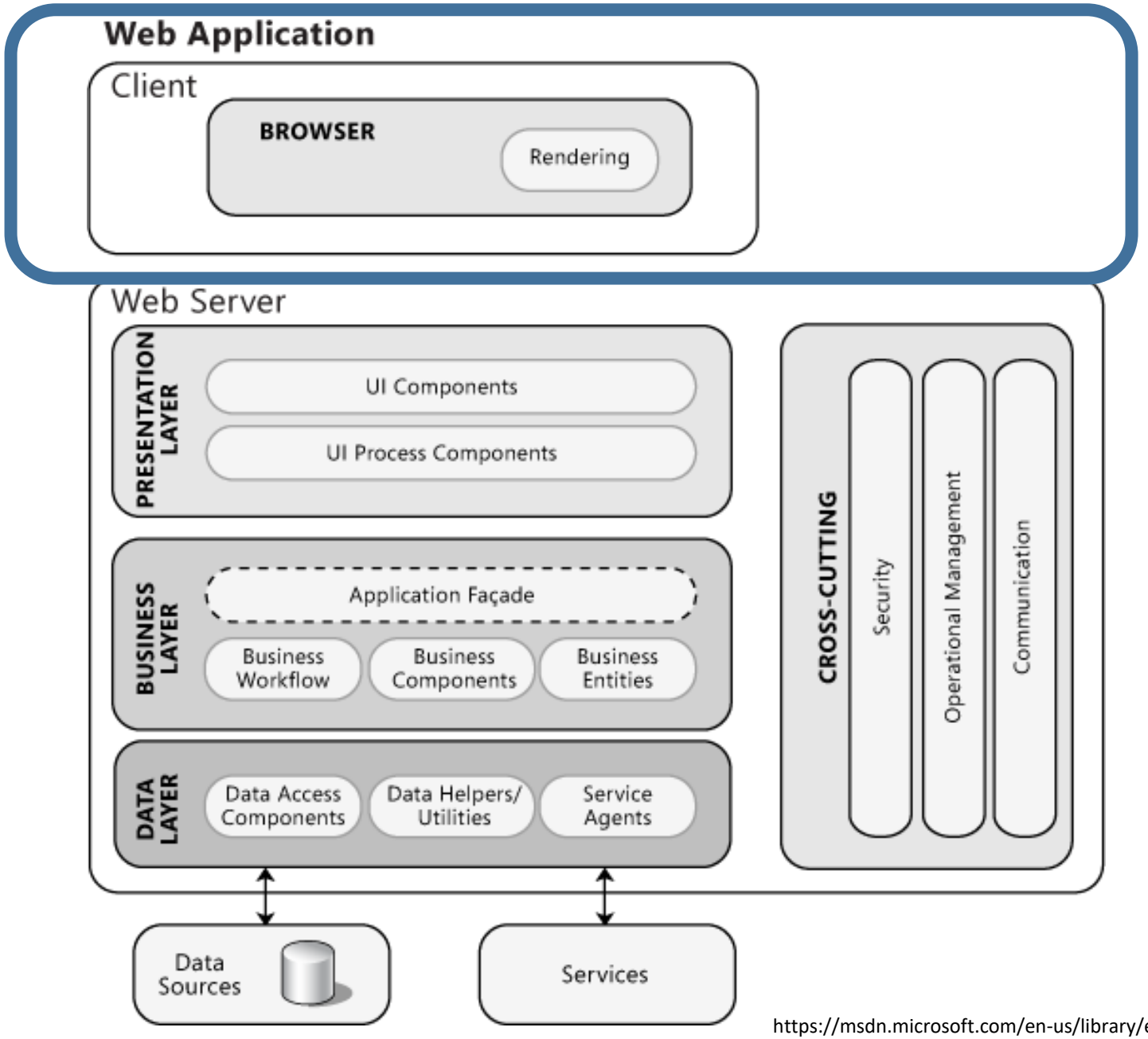
https://msdn.microsoft.com/en-us/library/ee658104.aspx

Example Mobile applications



https://msdn.microsoft.com/en-us/library/ee658104.aspx

Example Web /Thin applications



https://msdn.microsoft.com/en-us/library/ee658104.aspx

Three types of distribution

- Client servers
- Peer to peer
- Pipelines

Client-Server

Server

- Always on
- Welknown address (host + port)
- TCP
  - Waits for incoming client connections
- UDP
  - Waits for incomming client request

Client

- Takes the initiative
- TCP
  - Connects to the server
- UDP
  - Send request (Datagram) to receiver / server

SOAP a twist of Client-Server

Provider

- Always on
- Welknown address (host + port)
- Provide the service by the WSDL-file
- Wait for a method call (from consumer)
- All calls are stateless

Consumer

- Implements the wsdl-file
- Take the initiative – by making the method call

REST another twist of Client-Server

Restfull-service

- Always on
- Welknown address (host + port)
- Provide the service by the URL
- Wait for a call to a specific URL
- All calls are stateless

Client

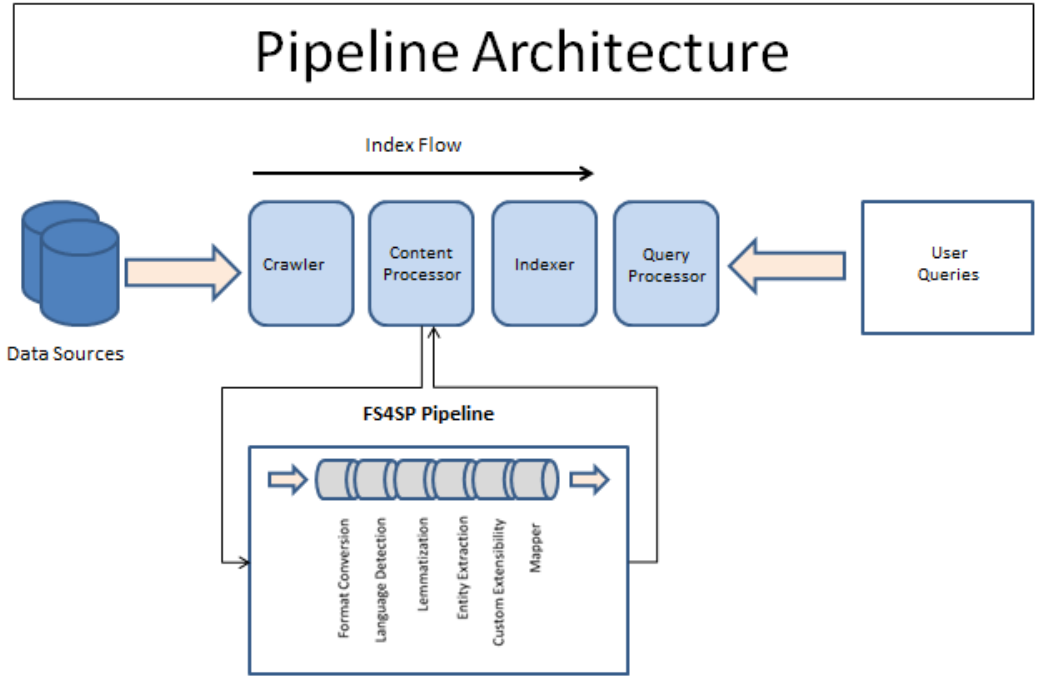
- Take the initiative
- make the call by using the url
- The call uses the HTTP methods GET, PUT, POST, DELETE

Peer-to-Peer

- Peers are equal
- Peers may come and go
- **Each peer play the role of both a client and a server.**
  - The client part connects to another peer's server part and asks for service
- General problem
  - How do peers find each other?
  - Solution 1: Mixed architecture
    - Registry / Index / Repository has information on location of peers
      - A server ! E.g. skype
  - Solution 2: Pure distributed system
    - No registry … (CN: Distributed Hash tables) e.g. bittorrent

Pipeline: Pipes and filters

- A filter read input from a pipe, process the input, and writes it to another source (pipe)
  - Data flows from filter to filter
    - Never backwards
- Like an assembly line in a factory
- Filters may be on different hosts.
  - Pipes are network connections
  - Dragging or pushing data through the filters



http://social.technet.microsoft.com/wiki/contents/articles/7244.sharepoint-2010-customize-fast-search-to-crawl-related-data-lookup-columns.aspx