

Unit 3 Client-server architectures

Unit Outcomes. Here you will learn

- how and why servers are usually layered in tiers
- how servers can be made more scalable

Further Reading: CDK2005 2.2

1 Client-server example

2 Multi-tiered architecture

Standard 3-tiered architecture

Almost 3-tiered system example

Evaluating tiered architectures

3 Improving server scalability

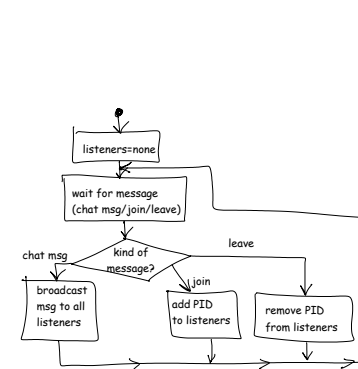
Proxies and caches

Replication

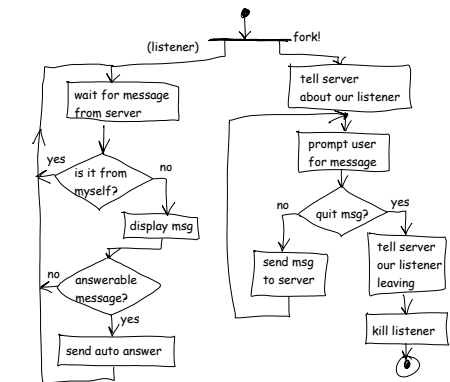
Mobile code

Client-server example

server activity:

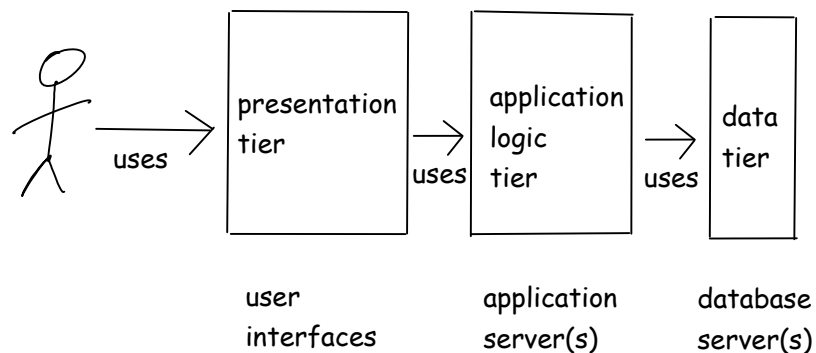


client activity:



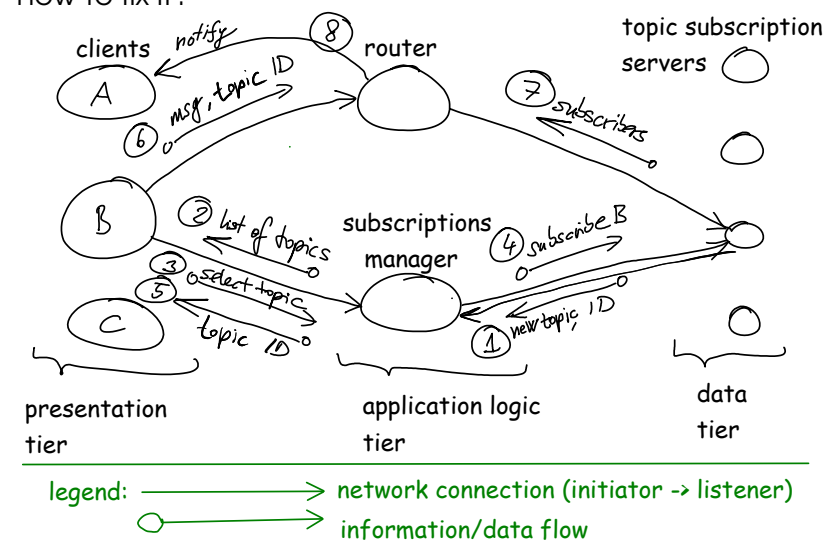
Multi-tiered architecture

Standard 3-tiered architecture



Almost 3-tiered system example

- why is the following not 100% 3-tiered?
- how to fix it?



Evaluating tiered architectures

- advantages/disadvantages of extra tiers

<i>factor</i>	<i>less tiers</i>	<i>more tiers</i>
---------------	-------------------	-------------------

network
load

development
time

scalability

maintenance

Improving server scalability Replication

- facilitated by a switching proxy
- when does it pay off to replicate? (eg in chat scenario)

<i>component</i>	<i>cache?</i>	<i>replicate?</i>
------------------	---------------	-------------------

router

subscription
manager

topic server

topic registry

Improving server scalability Proxies and caches

- proxy = fake server, giving all work to another one
- cache = like proxy but able to replay server interaction

- when is proxy useful? when cache? (eg in chat scenario)

Improving server scalability Mobile code

- eg Web server sends JavaScript to a browser
- how could it be used in the chat scenario?

Learning Outcomes

Learning Outcomes. You should now be able to

- describe the functionality of each tier in a standard 3-tiered system and how they interact
- describe the advantages/disadvantages of having more or less tiers in a client-server DS
- given an example tiered system, identify ways to improve its scalability, in particular which services can be cached, replicated and/or transferred as mobile code