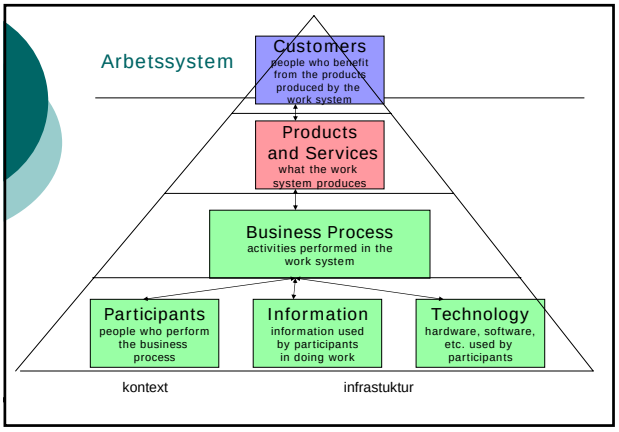
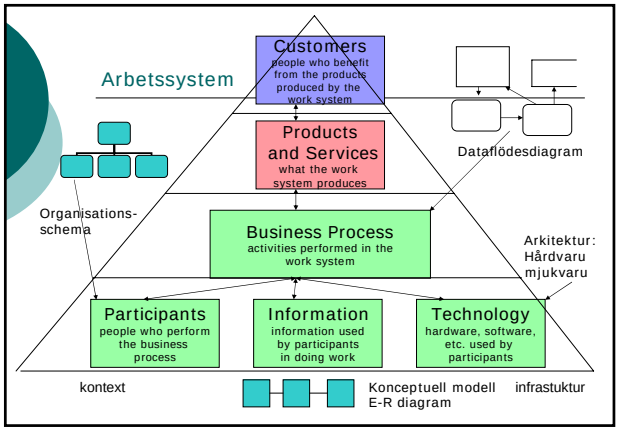
Systemarkitektur

Datortyper
Hårdvaruarkitektur
Mjukvaruarkitektur

Kapitel 8, 9, 10





Vilket beteende hos datorn tycker du är viktig när du använder den för ditt arbete?

Beteende variabler for IT

Performance Variables for IT

GROUP OF PERFORMANCE VARIABLES	TYPICAL ISSUE RAISED WHEN USING THIS TERM TO DESCRIBE A PARTICULAR TECHNOLOGY
Functional capabilities and limitations	**What types of processing is the technology supposed to perform and what capabilities does it have?**
• Capacity	How much information can it store or process?
• Speed	How fast can it process data or instructions?
• Price-performance	How many dollars does it cost per amount of information stored or for a given calculation speed?
• Reliability	How long will it likely continue operating without errors or unplanned interruptions?
• Operating conditions	How much space does it take up? How much does it weigh? What temperature does it require? How much electricity does it use?
Ease of use	**How easy is it to use this technology?**
• Quality of user interface	How intuitive and easy to learn is the method for instructing the technology to perform its task?
• Ease of becoming proficient	How much effort is required to become proficient in using the technology?
• Portability	How easy is it for the user to move the technology in the course of doing work?
Compatibility	**How easy is it to get this technology to work with other complementary technologies?**
• Conformance to standards	To what extent does the technology conform to accepted industry standards?
• Interoperability	To what extent does the technology use the same internal coding and external interfaces as other technology it must operate with or substitute for?
Maintainability	**How easy is it to keep the technology operating over time?**
• Modularity	Is it divided into modules that can be snapped together when building systems? Can these modules be replaced by equivalent modules if necessary?
• Scalability	Is it possible to significantly increase or decrease capacity without major disruptions?
• Flexibility	Is it possible to change important aspects of system operation without major disruptions?

Exempel på datorer, historiskt

Konrad's Zuse Z3 computer.
Germany, 1941

ENIAC
(**Electronic Numerical Integrator And Computer**)
United States, 1946

Besk
I november 1953 var BESK färdigbyggd
KTH lokaler på Drottninggatan 95A.

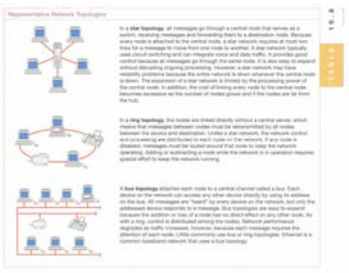
http://www.treinno.se/pers/okq/besk.htm

(Besk och Bark)

Dagens datorer

- Superdatorer
- Stordatorer (Mainframes)
- Minidatorer (Midrange computers)
- Workstations → Personal Computer (PC)
- PDAs (Personlig Digital Assistants, handdatorer)

Computer Networks (topologier)



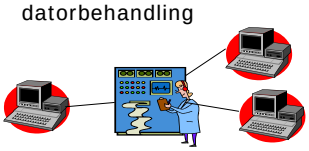
Stjärnnät

Ringnät

Bussnät

Nätverk i organisationer

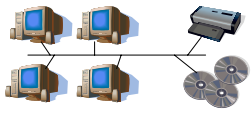
- Centraliserad databehandling (Centraliserad computing)
  - terminaler som är kopplade till en central dator som utför all datorbehandling



- Personlig databehandling (Personal computing)
  - Varje anställd har en egen dator som inte är förbundna med varandra i ett nätverk

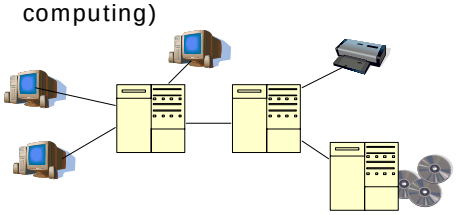
Nätverk i organisationer forts

- Distribuerad databehandling (distributed computing)
  - Arbetsstationer är ihopkopplade på ett sådant sätt att de kan dela på resurser t.ex. data



Nätverk i organisationer forts

- Databehandling i nätverk (Network computing)



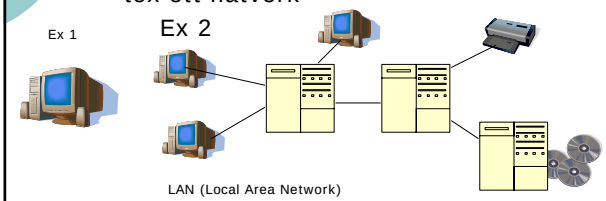
databas

Hårvaruarkitektur

Beskriver hur en organisations datorutrustning ev är ihoppkopplad i tex ett nätverk

Ex 1

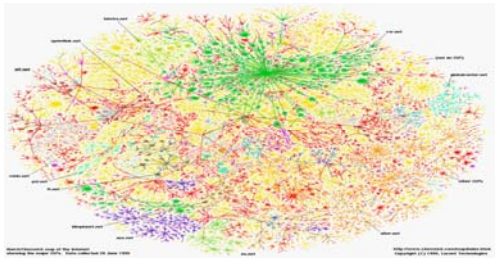
Ex 2



LAN (Local Area Network)

databas

Computer Networks (cont.)



WAN (Wide Area Network)

Funktioner inom IT (functions I IT)

System definition: Ett *informationssystem* är ett (arbets)system som samlar in, lagrar, bearbetar och presenterar information.

Six Functions of Information Technology

Function	Definition	Example of devices or technologies used to perform this function
Capture	Obtain a representation of information in a form permitting it to be transmitted or stored	Keyboard, bar code scanner, document scanner, optical character recognition, sound recorder, video camera, voice recognition software
Transmit	Move information from one place to another	Broadcast radio, broadcast television via regional transmitters, cable TV, satellite broadcasts, telephone networks, data transmission networks for moving business data, fiber optic cable, fax machine, electronic mail, voice mail, Internet

TABLE 1.5

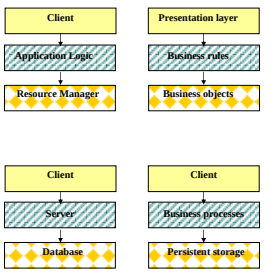
Functions of IT

Store	Move information to a specific place for later retrieval	Paper, computer tape, floppy disk, hard disk, optical disk, CD-ROM, flash memory
Retrieve	Find the specific information that is currently needed	Paper, computer tape, floppy disk, hard disk, optical disk, CD-ROM, flash memory
Manipulate	Create new information from existing information through summarizing, sorting, rearranging, reformatting, or other types of calculations	Computer (plus software)
Display	Show information to a person	Laser printer, computer screen

Architecture of an IS

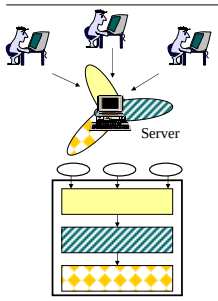
- Design → layers
- Implementation → tiers

Layers and tiers



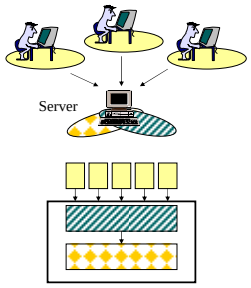
- Client is any user or program that wants to perform an operation over the system. Clients interact with the system through a presentation layer
- The application logic determines what the system actually does. It takes care of enforcing the business rules and establish the business processes. The application logic can take many forms: programs, constraints, business processes, etc.
- The resource manager deals with the organization (storage, indexing, and retrieval) of the data necessary to support the application logic. This is typically a database but it can also be a text retrieval system or any other data management system providing querying capabilities and persistence.

One-tier (Centralized)



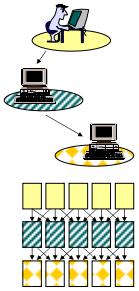
- The presentation layer, application logic and resource manager are built as a monolithic entity.
- Users/programs access the system through display terminals but what is displayed and how it appears is controlled by the server. (These are "dumb" terminals).
- This was the typical architecture of mainframes, offering several advantages:
  - no forced context switches in the control flow (everything happens within the system),
  - all is centralized, managing and controlling resources is easier,
  - the design can be highly optimized by blurring the separation between layers.

Two-tier (Client/server)



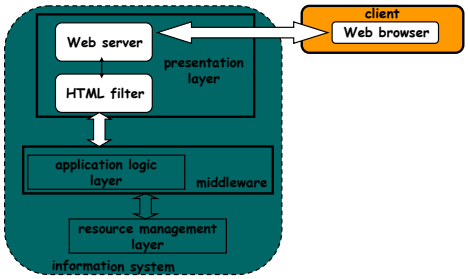
- As computers became more powerful, it was possible to move the presentation layer to the client. This has several advantages:
  - Clients are independent of each other: one could have several presentation layers depending on what each client wants to do.
  - One can take advantage of the computing power at the client machine to have more sophisticated presentation layers. This also saves computer resources at the server machine.
  - It introduces the concept of API (Application Program Interface). An interface to invoke the system from the outside. It also allows designers to think about federating the systems into a single system.
  - The resource manager only sees one client: the application logic. This greatly helps with performance since there are no client connections/sessions to maintain.

3-tier architecture (middleware)



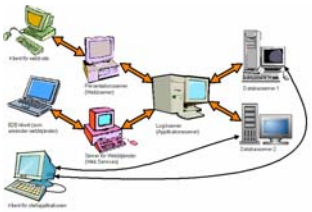
- In a 3 tier system, the three layers are fully separated.
- The layers are also typically distributed taking advantage of the complete modularity of the design (in two tier systems, the server is typically centralized)
- A middleware based system is a 3 tier architecture. This is a bit oversimplified but conceptually correct since the underlying systems can be treated as black boxes. In fact, 3 tier makes only sense in the context of middleware systems (otherwise the client has the same problems as in a 2 tier system).

N-tier architecture (Web)



GK:ITO ITea Lab

- Multi-tiered enterprise system



http://itea.dsv.su.se/tutorial/

GK:ITO ITea Lab

- Use ITea as a customer → Website
- Use ITea as a business customer → B2B
- Use ITea as a manager → Chef
- Explore the ITea architecture → Arkitektur
- Explore the ITea databases → Databaser
- Understand the system calls inside the different parts of the ITea system
- Look at the ITea documentation → Dokumentation
- Take the test! → Formulär

GK:ITO ITea Lab

Start here:
http://itea.dsv.su.se/tutorial/

Work individually or in pairs, but do the final assignment individually!

Questions?

Ask: rafa@dsv.su.se

References (1/3)

- [ALDER] "Information Systems" (4th Edition), Steven Alter. Prentice Hall. 2004. http://www.prenhall.com/alter/
- [ALONSO] "Web Services: Concepts, Architectures and Applications". Gustavo Alonso et al. Springer-Verlag. 2004. http://www.inf.ethz.ch/personal/alonso/teaching.html
- "It's the economy, stupid!" http://en.wikipedia.org/wiki/It%27s_the_economy%2C_stupid
- The Network Really Is the Computer http://www.oreillynet.com/pub/a/251

References (2/3)

- Imperative programming: http://en.wikipedia.org/wiki/Imperative_programming
- Declarative programming: http://en.wikipedia.org/wiki/Declarative_programming