

# Att välja och vidmakthålla arkitekturer

## Tekniska perspektiv

SESAM Höstseminarium 2001-10-24

*Jaan Haabma*

*Basesoft Open Systems AB*

*jaan@basesoft.se*

*08-13 17 20*

# Disposition

- Förutsättningar - arkitektur ?
- Avvägningar vid utformning av arkitekturer
  - samverkansarkitektur - tekniska perspektiv
  - infrastrukturbaserad utveckling, "ramverk"
- Typarkitekturer
  - "tillämpningsarkitekturer", "mönster"
  - "middleware"
- Arkitekturdriven utveckling - livscyklar
- Arkitekturdriven utveckling
  - Exempel olika perspektiv/vyer

# System och arkitektur

- Varje "system" har en arkitektur (implicit, explicit)
- Vad är "system" (i detta sammanhang)
  - *System - individuell tillämpning, programvaru-system, subsystem, system av system, produkt, produktlinje, produktfamilj, företagssystem, verksamhetssystem , ...*
- "System" befolkar en "miljö" (eng "environment"). Miljön (omgivningen) påverkar systemet. Miljön kan innehålla andra (samverkande) system. Miljön anger därmed gränserna (eng. "scope") till det aktuella "systemet".
- Ett "system" har ett antal intressenter ... (som ställer krav på egenskaper ...)
- Ett "system" har en eller flera uppgifter (eng. "mission") i sin miljö/omgivning

# Arkitekturbeskrivningar - perspektiv och vyer

- Arkitekturen behöver kunna beskrivas i ett antal arkitekturbeskrivningar ...
- Arkitekturbeskrivningar kan organiseras i ett antal vyer ("views") för att adressera olika parterers intressen ("concerns") (intressenter, krav, egenskaper, användning), dvs beskriva systemets arkitektur ur olika perspektiv ("viewpoints")
- En arkitekturvy kan bestå av ett antal modeller för arkitektur
- => *IEEE Recommended Practice for Architectural Descriptions of Software-Intensive Systems (IEEE Std 1471-2000)*

# Vad är "arkitektur" (term)

- Ingen enhetlig definition - jämför din respektive din samtalsparts uppfattning!
- IEEE STD 610.12 - "The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time"
- IEEE Std 1471-2000 - "The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution"
- Arkitektur ?
  - Stil, Principer, Mönster, ...
  - eller Struktur, Definitioner, Indelningar ...

# Definitioner - IEEE Std 1471-2000

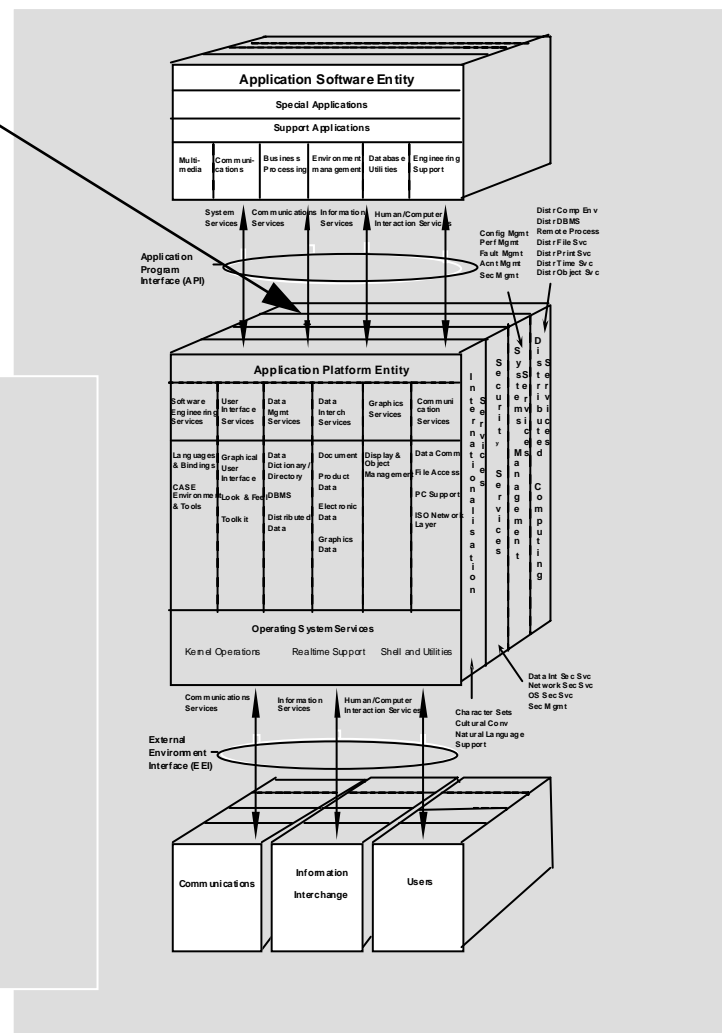
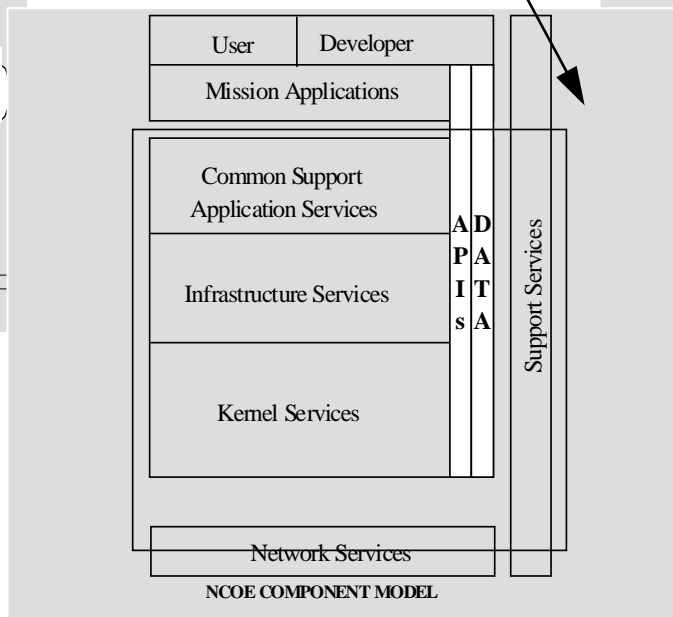
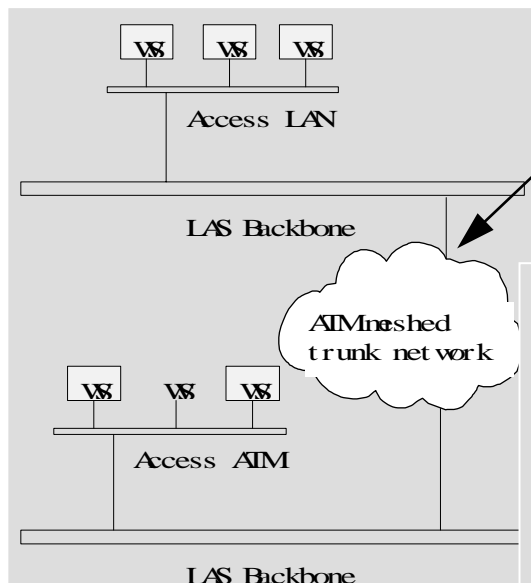
- 3.1 acquirer: An organization that procures a system, software product, or software service from a supplier. (The acquirer could be a buyer, customer, owner, user, or purchaser.)
- 3.2 architect: The person, team, or organization responsible for systems architecture.
- 3.3 architecting: The activities of defining, documenting, maintaining, improving, and certifying proper implementation of an architecture.
- 3.4 architectural description (AD): A collection of products to document an architecture.

# Definitioner - IEEE Std 1471-2000

- 3.5 architecture: The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
- 3.6 life cycle model: A framework containing the processes, activities, and tasks involved in the development, operation, and maintenance of a software product, which spans the life of the system from the definition of its requirements to the termination of its use.
- 3.7 system: A collection of components organized to accomplish a specific function or set of functions.
- 3.8 system stakeholder: An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system.
- 3.9 view: A representation of a whole system from the perspective of a related set of concerns.

# Olika modeller för arkitektur (struktur)

(examples)





# Disposition

- Förutsättningar - arkitektur ?
- **Avvägningar vid utformning av arkitekturer**
  - **samverkansarkitektur - tekniska perspektiv**
  - **infrastrukturbaserad utveckling, "ramverk"**
- Typarkitekturer
  - "tillämpningsarkitekturer", "mönster"
  - "middleware"
- Arkitekturdriven utveckling - livscyklar
- Arkitekturdriven utveckling
  - Exempel olika perspektiv/vyer

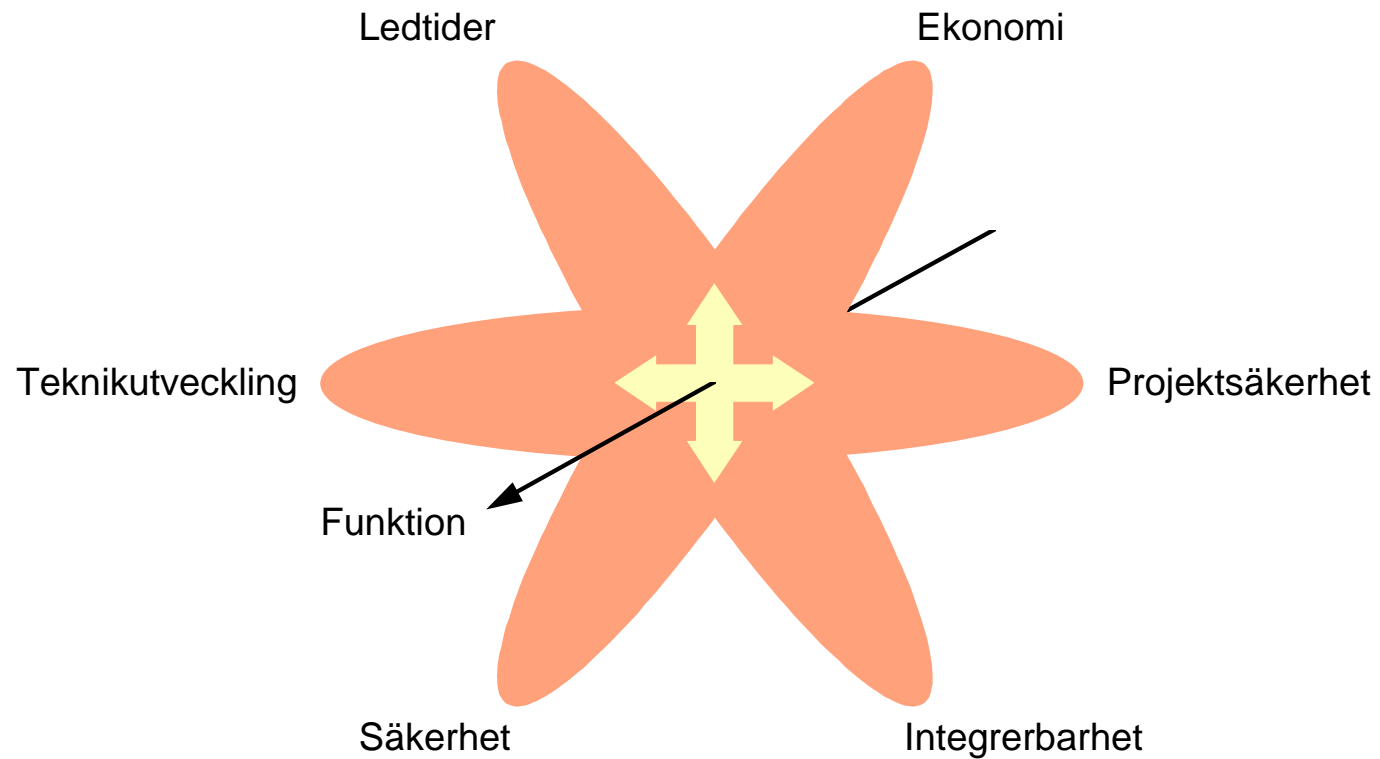
# Avvägningar

- Avgränsning - Arkitekturer för samverkande system - "samverkansarkitektur" :
- "Systemarkitektur som främjar samverkan mellan IT-lösningar och som därför överbryggar geografisk och teknisk heterogenitet"
- Hierarkier av "Arkitekturer" eller hellre strukturer, ex samverkande ...
  - komponenter
  - tillämpningar ("applications")
  - system
  - system av system

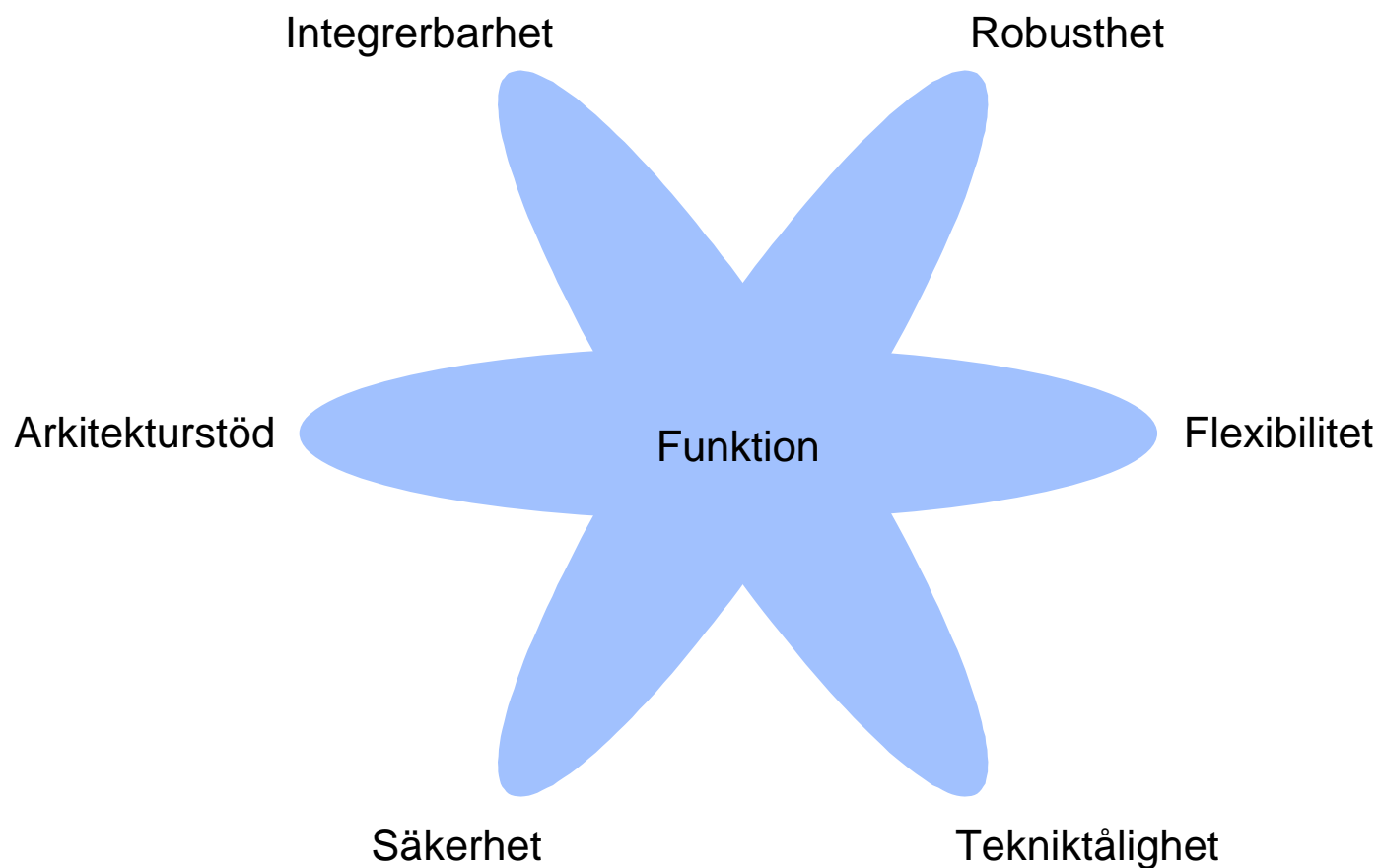
# Affärsidé / Affärskrav

- Vad är drivkraften bakom en samverkansarkitektur ...
- Dvs affärskrav som leder till en samverkansarkitektur, exempel
  - Organisatoriska krav - strukturförändringar
  - Verksamhetskrav - ändrad verksamhet, ny uppgifter, nya hot
  - Ledtid - "time to market" (=> återanvändning?)
  - Ny funktion - Krav på "kunden i centrum", krav på nytt ledningsstöd, ...
  - Nya egenskaper - prestanda, säkerhet, ...
  - Ekonomi - "protected investments", rationell utveckling, små/stora serier, ...

# "Business" - avvägningar



# Systemegenskaper - kvalite'er



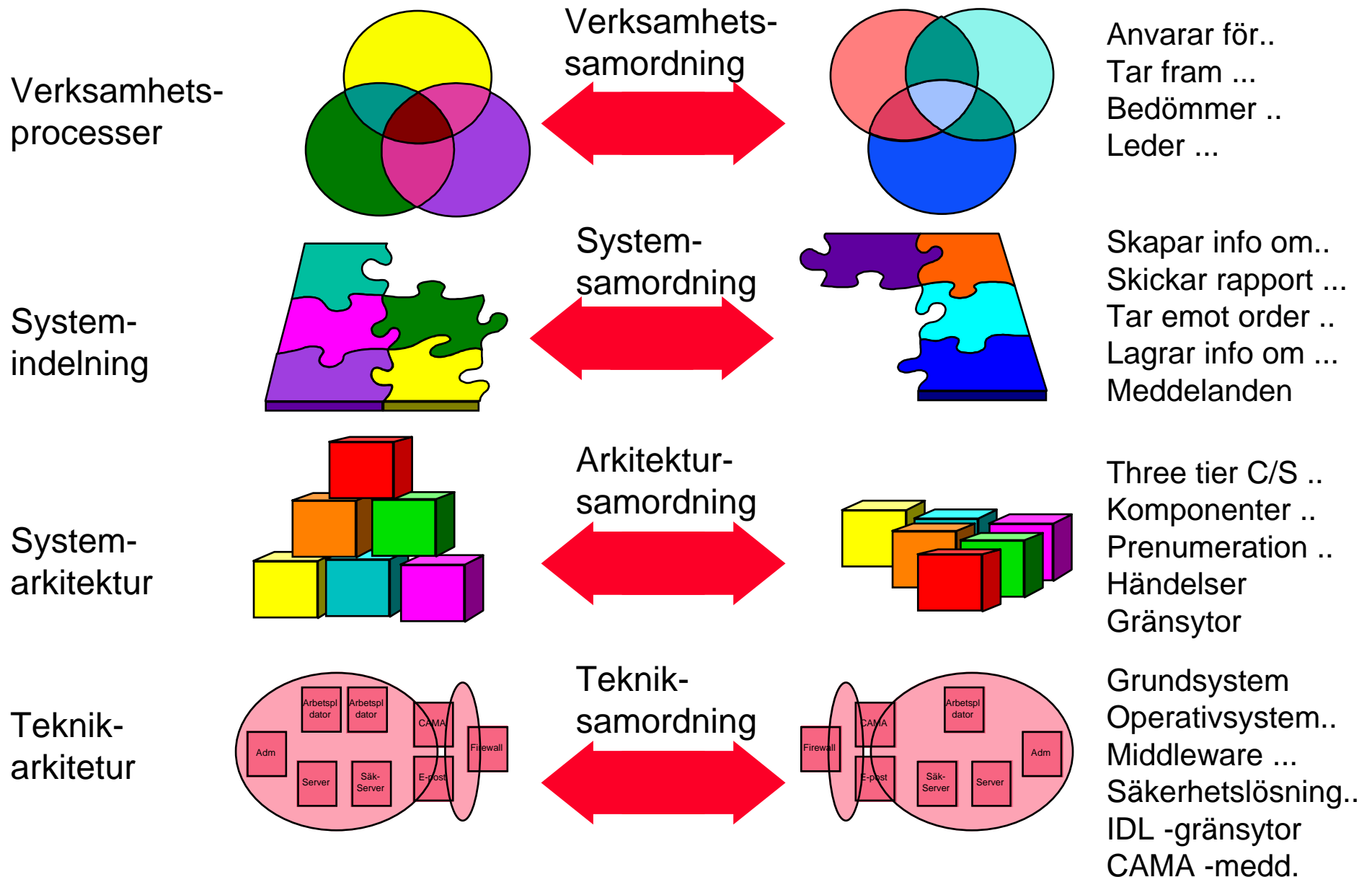
# Strategi/Vision för Teknisk arkitektur

- Oftast implicita (?) krav!
  - Exempel - Time-to-market vs ekonomi (affärskrav)
  - Exempel - Integritet (riktighet) vs COTS; sekretess vs tillgänglighet (systemegenskaper)
- Strategi för informationshantering
  - Exempel - flexibilitet, datadrivet, ...
- Strategi för konstruktion (tillämpningsarkitektur)
  - Komponentbaserad, objektorienterat, "software-factory", återanvändning, stateful/ stateless, "olika stora objekt", prenumeration vs pollande, distr objekt
- Strategi - Infrastruktur för tillämpningar *[behöver ett helt eget pass ...]*
  - Teknologi: DCE, DCOM, CORBA, Java/RMI, EJB, ...
  - Strategi: Teknikoberoende (tekniktålighet) vs optimerat tekniknyttjande; produkter vs arkitektur i produkt *[".NET och transaktionshantering"]*
  - *Obs:* Infrastrukturen måste stödja flera olika middleware-produkter!
- Strategi - Teknik
  - Standarder, Teknik, Protokoll, Leverantörer, ...

# Samverkansarkitektur ?

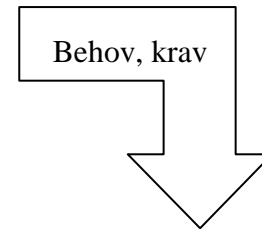
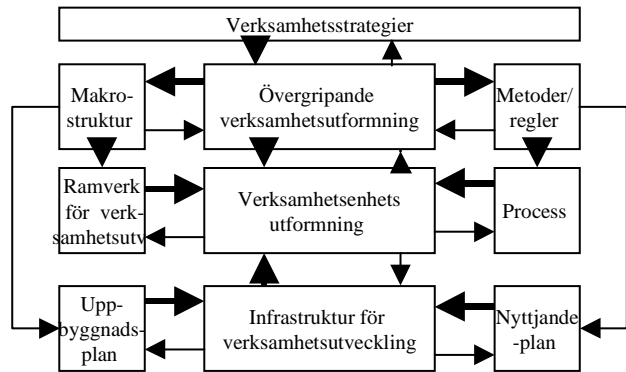
- Hantera/behov av samordning på olika nivåer av verksamhet/organisation
- Skilj på: perspektiv på, vyer av, skikt av, principer för ...
  - Verksamhet - varför, process, ...
  - Information - vad, ...
  - Teknik - hur, ...
- Olika modeller för arkitektur (struktur)
- Business Component Factory [*bl a kurslitteratur i DFS IT-arkitekt kurs*]

# Problem/Erf - Samordning på olika nivåer



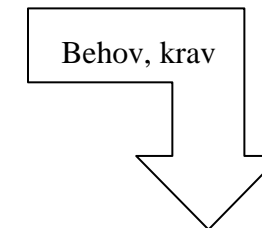
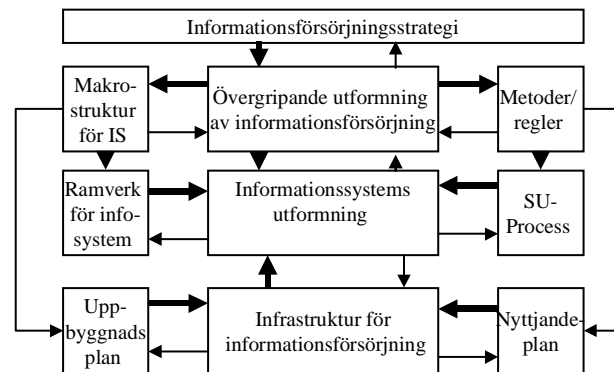
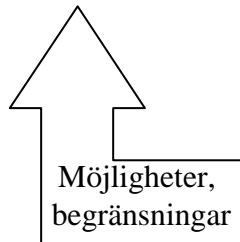


# Perspektiv - verksamhet, information, teknik

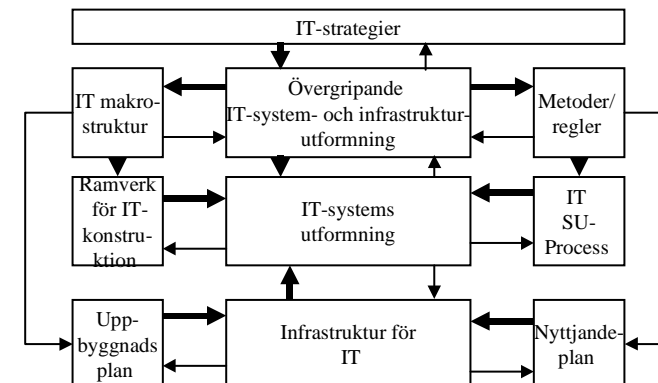
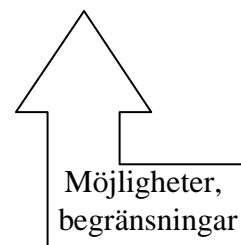


Information

Verksamhet

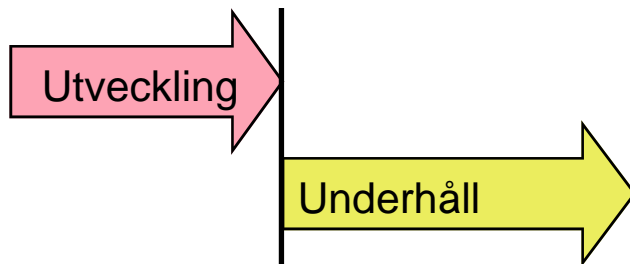
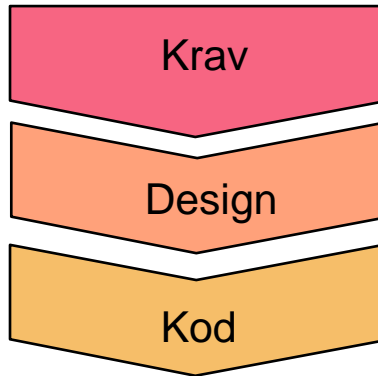


Teknik



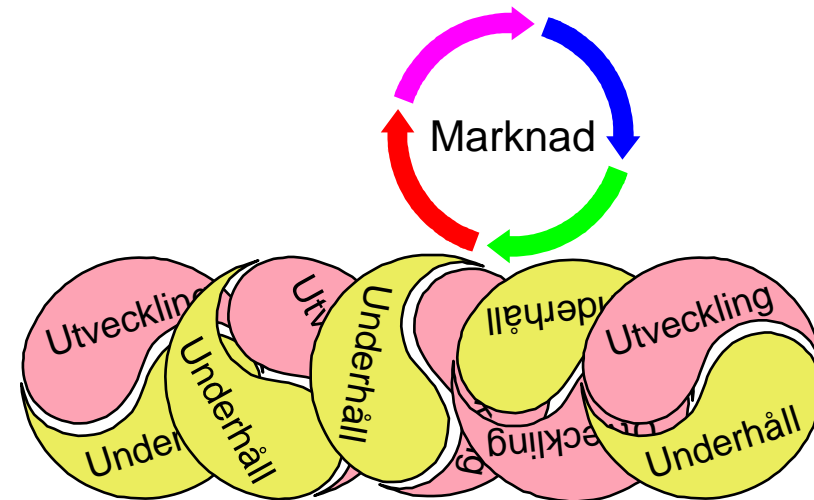
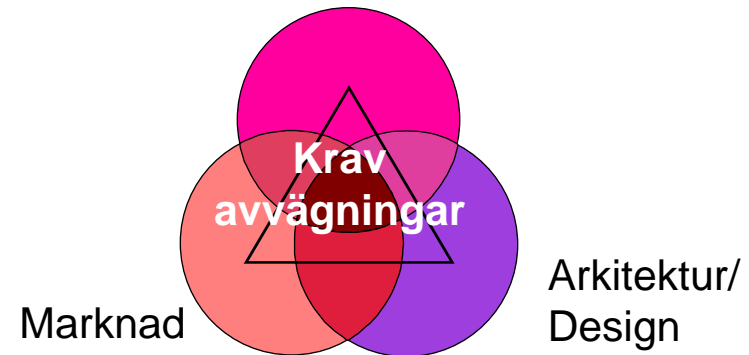
# Utvecklingsmiljö och verktyg

Traditionellt



med COTS

System context



# Disposition

- Förutsättningar - arkitektur ?
- Avvägningar vid utformning av arkitekturer
  - samverkansarkitektur - tekniska perspektiv
  - infrastrukturbaserad utveckling, "ramverk"
- **Typarkitekturer**
  - **"tillämpningsarkitekturer", "mönster"**
  - **"middleware"**
- Arkitekturdriven utveckling - livscyklar
- Arkitekturdriven utveckling
  - Exempel olika perspektiv/vyer

# *Exempel - komponentbaserd utveckling*

## *Business Component Factory ("BCF")*

### *- Fem dimensioner*

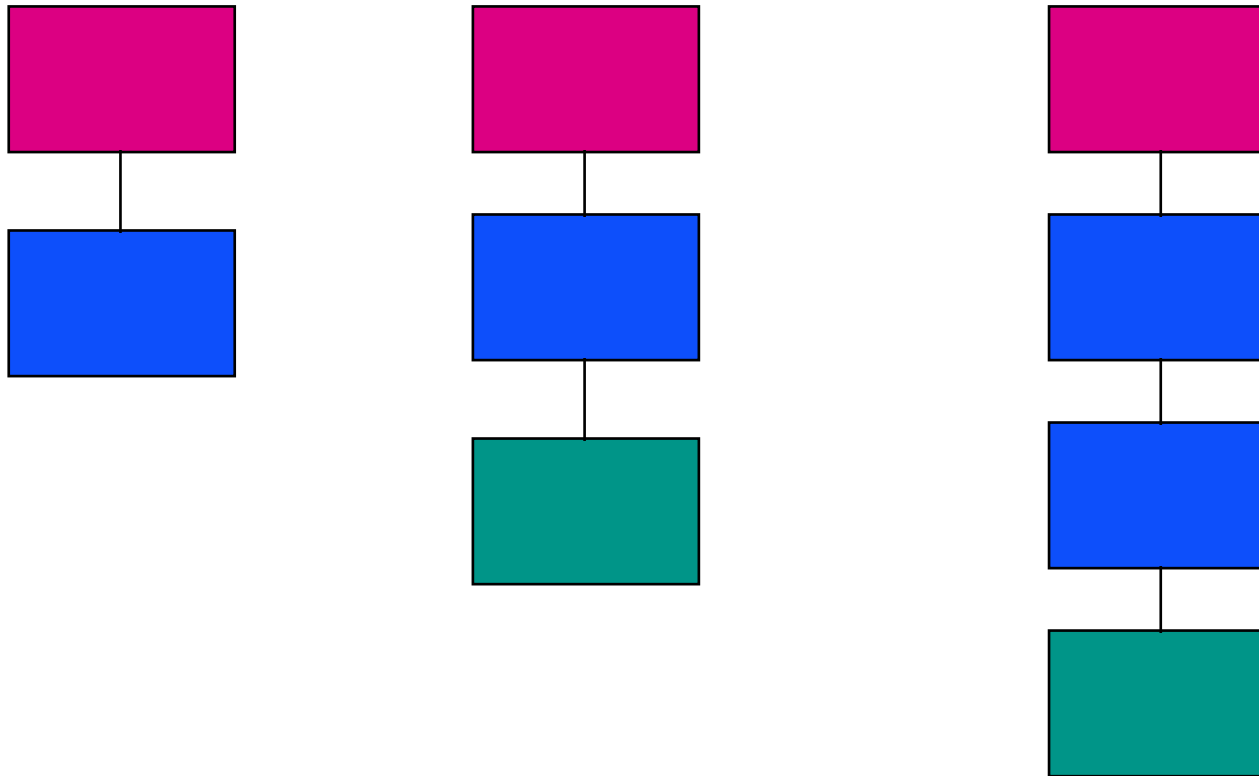
- D1 - Storlek/granularitet på komponent
  - (Klass i programspråk), komponent, affärskomponent ("business component"), system av affärskomponenter, federation av systemkomponenter
- D2 - **Arkitektur**
  - **Teknisk-** (exekveringsmiljö), **Tillämpnings-** (principer, mönster), **Projektlednings-**, **Funktions-** (spec och impl verksamhetsfunktion)
- D3 - Utvecklingsprocess
- D4 - "Distribution Tiers"
  - User, Workspace, Enterprise, Resource
- D5 - Funktionsnivåer ("Functional Layers")

# *Teknisk arkitektur - vyer enl "BCF"*

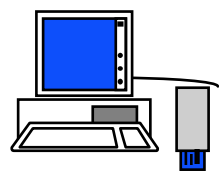
- Tillämpningsarkitektur (konstruktion)
  - Designmönster - Client(server (2-tier), Internet, Mainframe/ Host-based/ Terminal-based, N-tier, Office Automation, Communications (E-mail, E-Calendar), ...
  - Egenskaper - Tjänster och krav (ex. TPS, lokalitet hos användare), Obligatoriska/rekommenderade produkter
- Teknisk arkitektur - dela upp i T1, T2 ?
- T1 - Infrastruktur för tillämpningar/komponenter
  - Middleware (RPC, meddelande/kö), DBMS's, WorkFlow, Säkerhet (behörighet, spårbarhet), Utv. verktyg, ...
- T2 - Teknik
  - Maskinvara, OS, Säkerhet (identifiering), Nätverk (katalog, distr filsystem), Protokoll, ...

# Tillämpningsarkitekturer - patterns

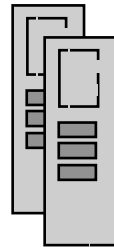
- Patterns - Client(server (2-tier), Internet, Mainframe/ Host-based/ Terminal-based, N-tier, Office Automation, Communications (E-mail, E-Calendar), Agent based ...



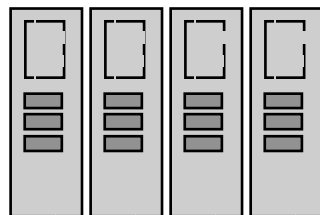
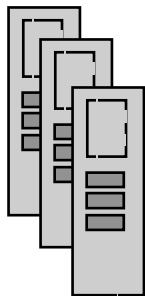
# Tillämpningsarkitekturer - "Distribution Tier" (enl "BCF")



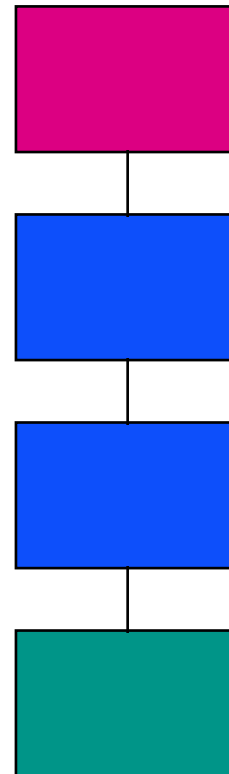
*Presentations  
logik (P)*



*Affärsslogik (F)*



*Data (D)*



User DC (UDC)

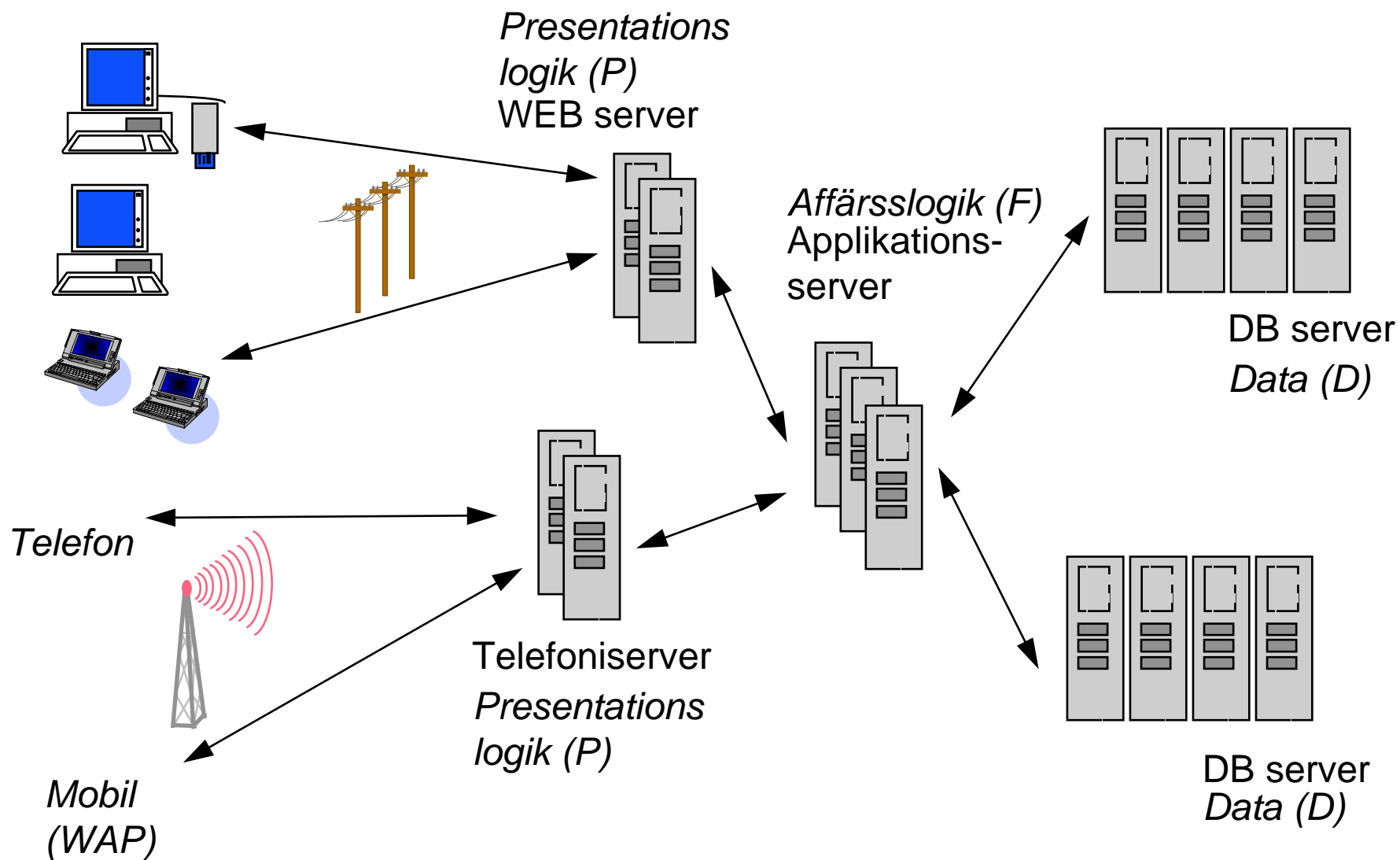
Workspace DC (WDC)

Enterprise DC (EDC)

Resource DC (RDC)

DC = (Distribution Component)  
- med varför begränsat till det ?

# "E-lager" samverkansarkitekturer



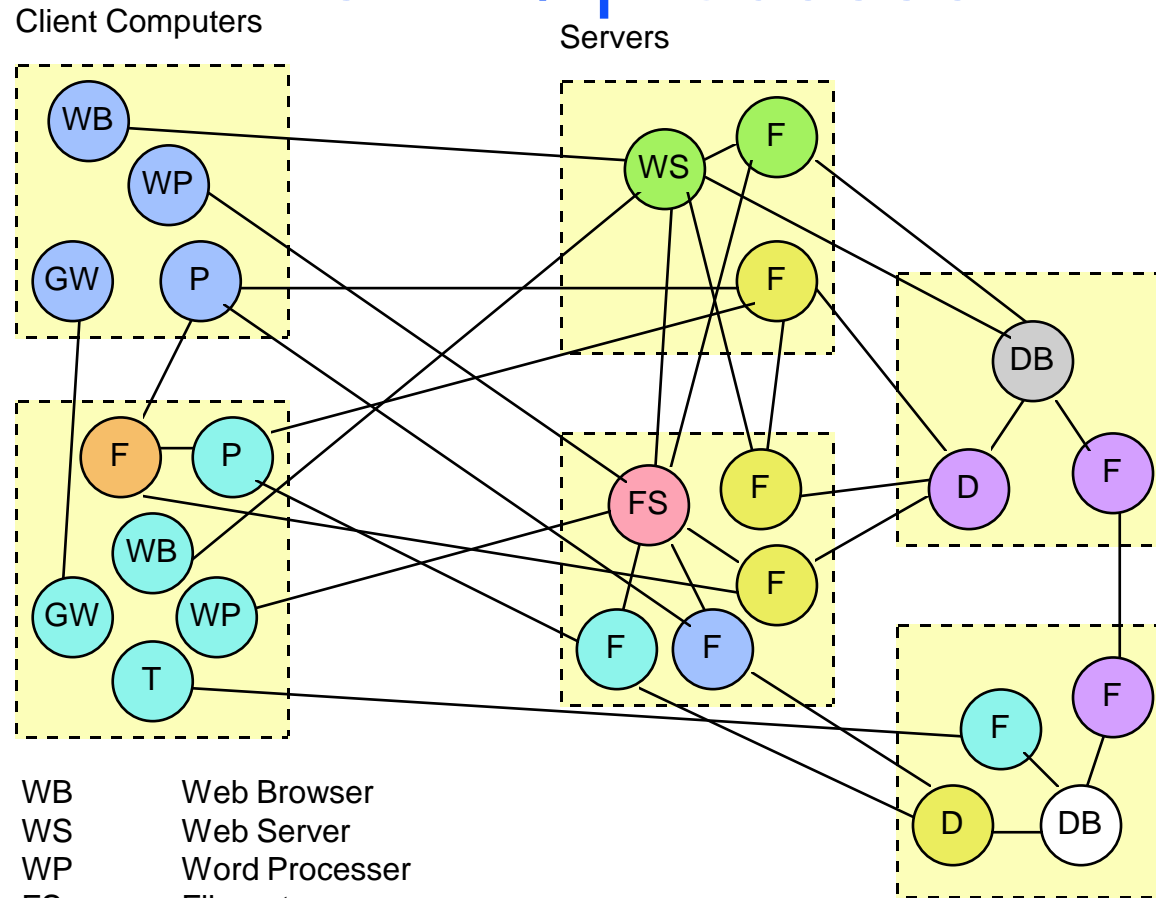


# Exempel - Middleware

- Middleware ?
  - Programvara mellan tillämpningar och operativsystem
  - Stöder komponentsamverkan (någon slags "RPC")
    - Arkitektur - komponenter, gränssytor, olika mönster för "RPC" - request/reply, publish/subscribe, ...
  - Tillhandahåller olika uppsättningar tjänster för drift (run-time) och utveckling (adk)
- Exempel - CORBA, COM/DCOM, DCE, Java/RMI, EJB
  - RPC/IDL, objektmodell/komponentmodell, produkter vs standard
  - Interoperabilitet -
    - mellan produkter inom samma "arkitektur" (ex olika Corba produkter)
    - mellan olika "arkitekturer" (ex Corba och EJB)
- Generella tjänster
  - Distribuerade filsystem, distr databaser (ODBMS), katalog/namntjänster, systemadministration, säkerhet

# Exempel - teknisk samverkan

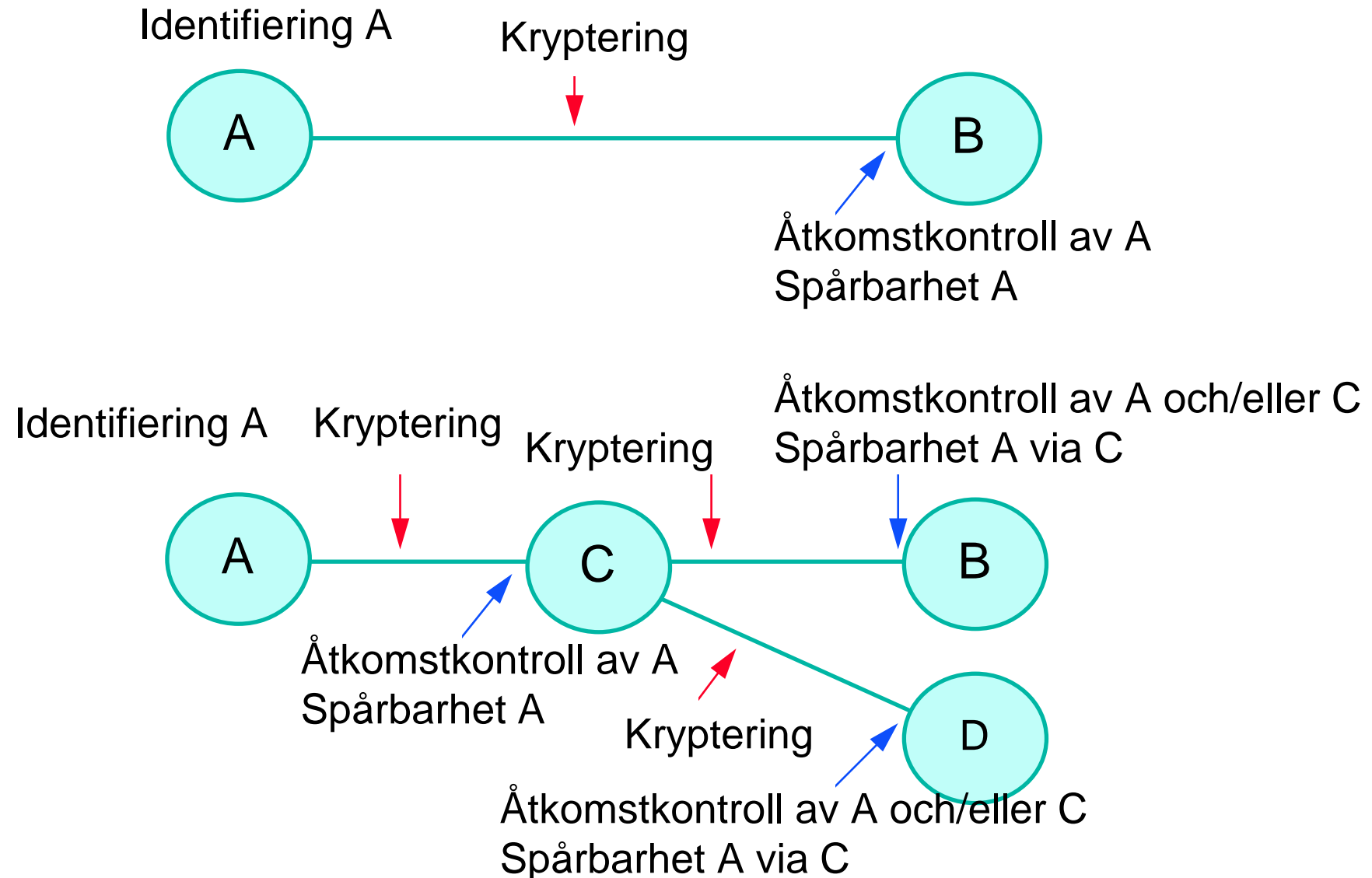
## - skikt / processer



- WB Web Browser
- WS Web Server
- WP Word Processor
- FS Filesystem
- P Presentation comp.
- F Function comp.
- D Data handling comp
- DB Database process
- GW GroupWare (C-C comm)
- T Terminal Emulator

The colors indicate different "users" (or authenticated principals) in a network based system. The circles are operating system processes that act on behalf of the "user"

# Exempel - Säkerhet krav/funktioner



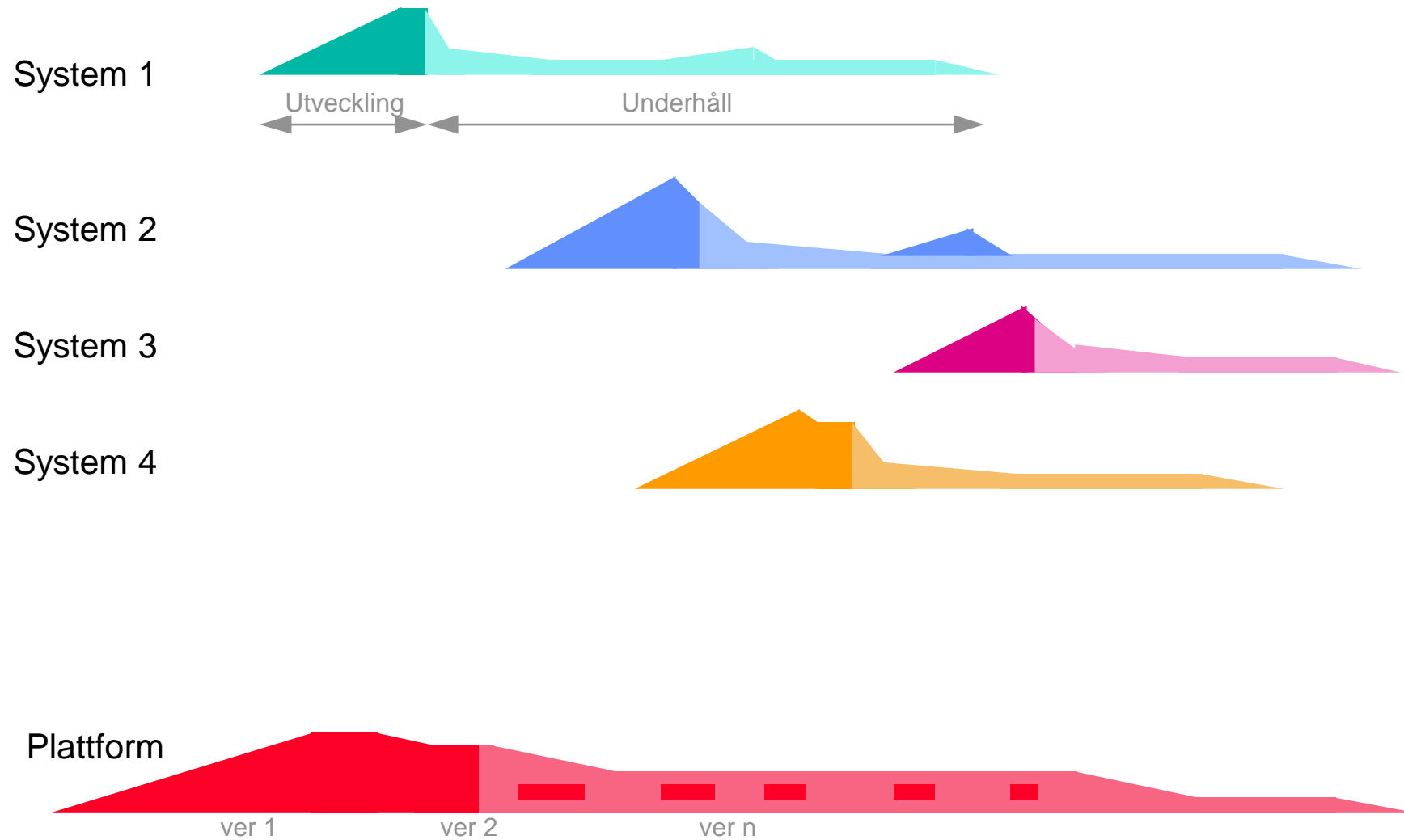
# Disposition

- Förutsättningar - arkitektur ?
- Avvägningar vid utformning av arkitekturer
  - samverkansarkitektur - tekniska perspektiv
  - infrastrukturbaserad utveckling, "ramverk"
- Typarkitekturer
  - "tillämpningsarkitekturer", "mönster"
  - "middleware"
- **Arkitekturdriven utveckling - livscyklar**
- Arkitekturdriven utveckling
  - Exempel olika perspektiv/vyer

# Arkitekturdriven utveckling - livscyklar

- "Arkitektur" har längre livslängd än "system"
- "System" som realiserar arkitektur för "system av system", som har och kommer ha olika livslängder och teknik - "teknisk infrastruktur"
- "Arkitektur" för teknisk infrastruktur ("plattform")
- => Infrastrukturbaserad utveckling

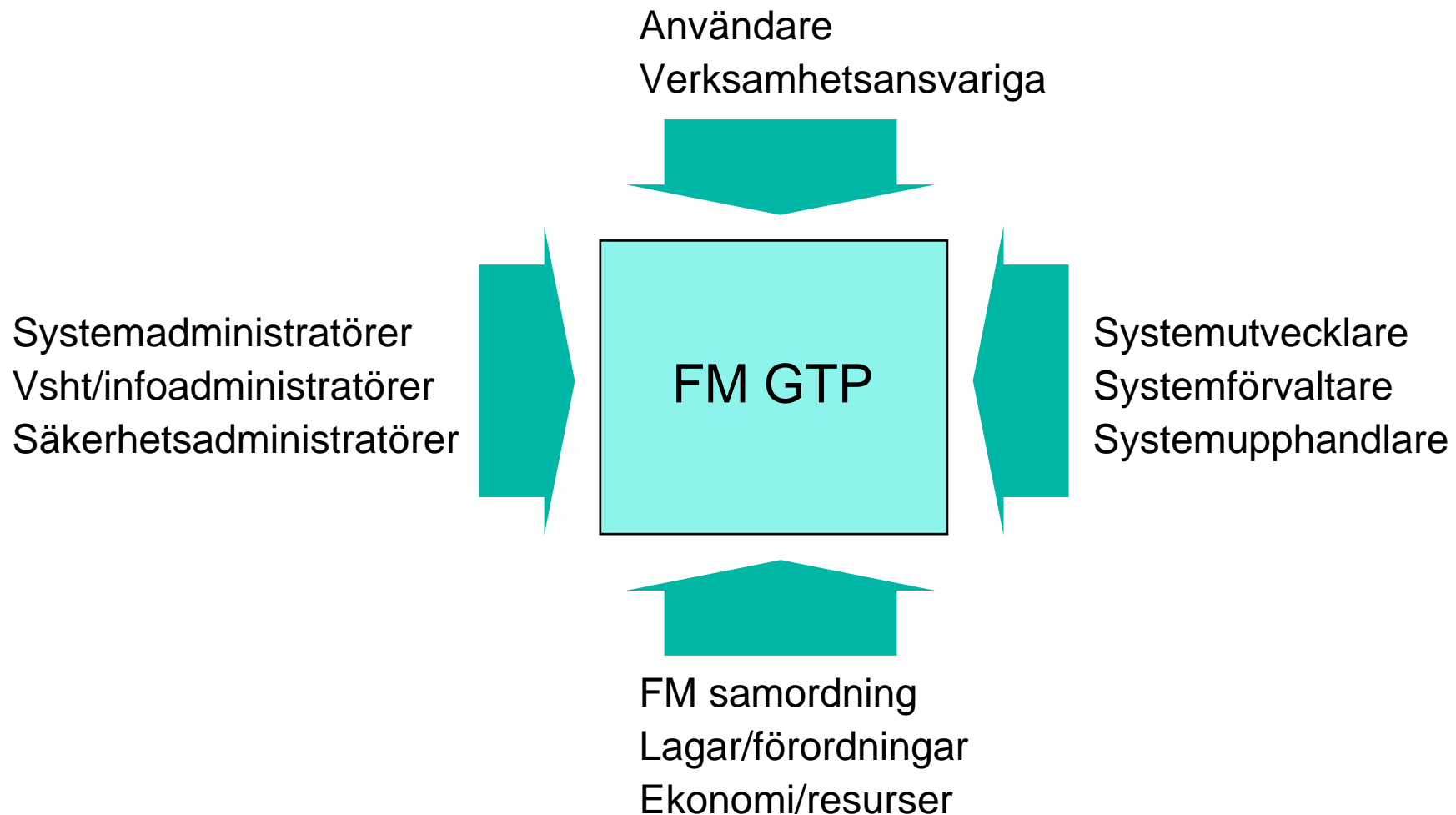
# Livscyklar



# Disposition

- Förutsättningar - arkitektur ?
- Avvägningar vid utformning av arkitekturer
  - samverkansarkitektur - tekniska perspektiv
  - infrastrukturbaserad utveckling, "ramverk"
- Typarkitekturer
  - "tillämpningsarkitekturer", "mönster"
  - "middleware"
- Arkitekturdriven utveckling - livscyklar
- **Arkitekturdriven utveckling - exempel**
  - **FM GTP - Generell Teknisk Plattform**
  - **Exempel olika perspektiv/vyer**

# Intressenter ("Stakeholders")





# Krav/Egenskapskrav ("Concerns") från olika intressenter - på Teknisk Plattform / Teknisk infrastruktur

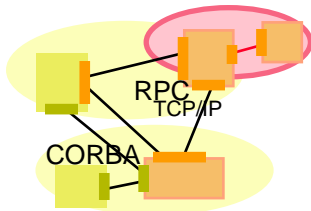
- Kunna användas i olika sammanhang ...
- Tillhandahålla funktioner för IT-integritet ("säkerhet" - hantering öppen respektive hemlig information ...)
- Stödja olika integritetsmekanismer ber. på krav
- Stödja användning av omodif. COTS och GOTS
- Stödja utveckling av informations- och ledningsystem med olika (tekniska) tillämpningsarkitekturer
- . . .

# GTP verksamhetsstöd

## Verksamhetsfunktion ...



- Kontorsautomation (Office etc)
  - producera, lagra dokument; samverka med filer
- Moderna ledningssystem (IS FV, MSS, LedsystT etc)
  - komponentsamverkan, någon slags funktionsanrop över nätet (s.k. RPC)
  - systemsamverkan



- Användarsamverkan genom Mail, Web (Intranät, Extranät) etc
  - skicka mail till person/roll; söka/uppdatera gemensam information
- Äldre informations- (resurslednings-) och ledningssystem genom kapslingsfunktion och terminalemulering
- Driftstöd - Systemadministration, logghantering, konfigurationshantering, övervakning



## Realiseras som funktion i GTP ...

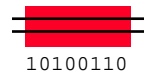
- Säker, robust fildelning (SFS)
  - hanterar alla tillämpningar som arbetar med filer
- Säker komponentsamverkan (LCom, TeS)
  - hanterar alla tillämpningar som byggs med samverkande, distribuerade komponenter (RPC, Corba etc.)



- Standardprogramvara (Mail, Web); Kapsling av standardprotokoll för Mail, Web etc (ComE). Alla (standard) TCP/IP baserade tillämpningar/protokoll kan hanteras.

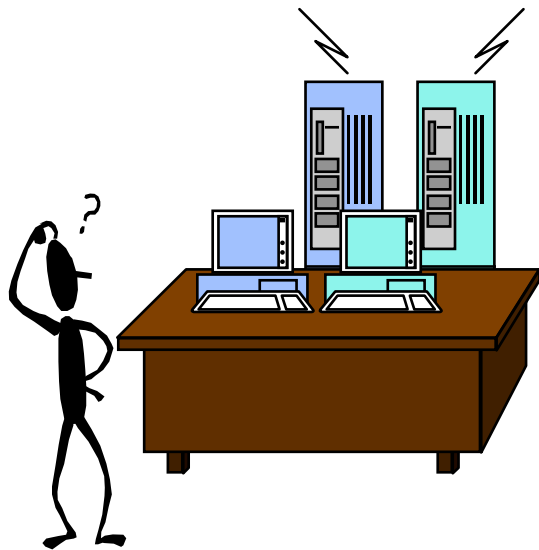
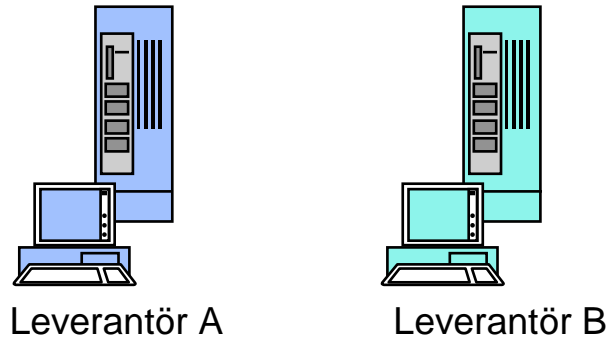


- Äldre system nås genom standard terminalemulering eller kan kapslas med kapslingsfunktioner (ComE, GSS, TeS etc.)
- Gemensamma driftstödsverktyg

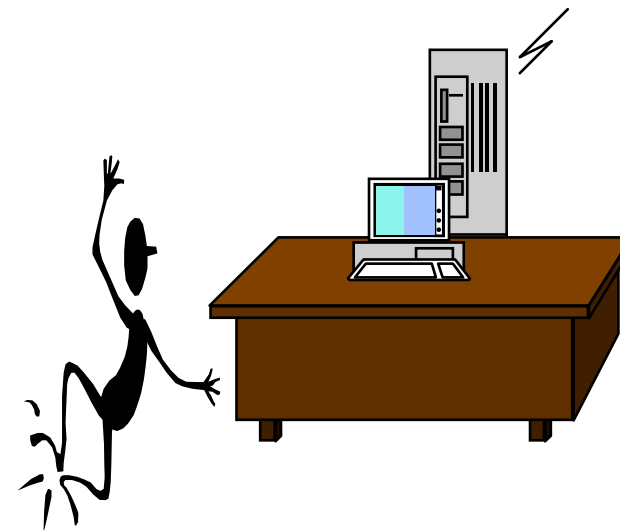
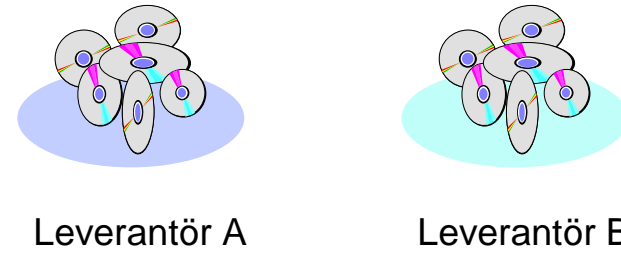


# Gemensam plattform

Utan ...

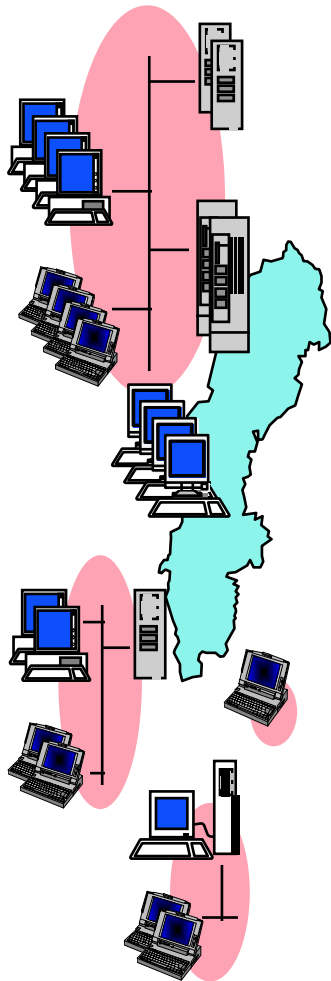


Med ...

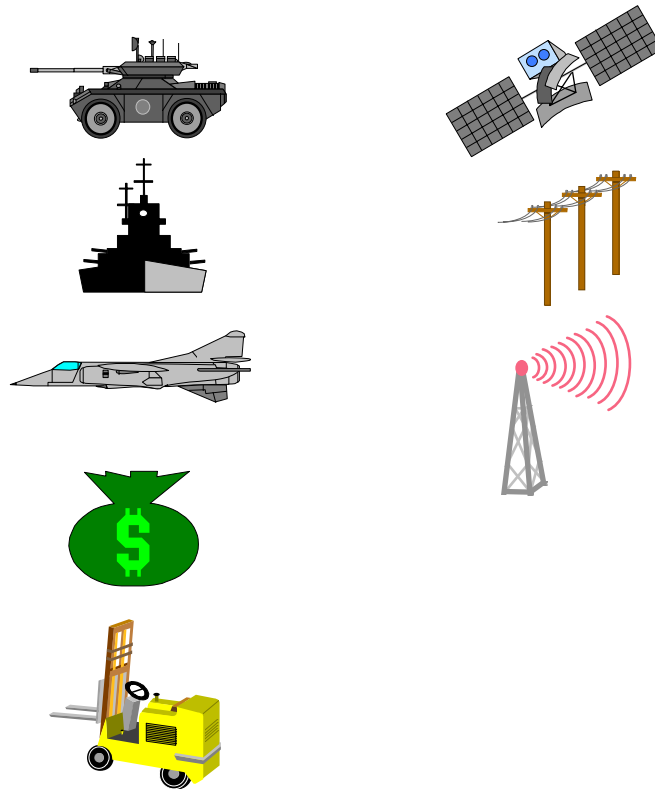


# GTP - Flexibilitet - Olika användning

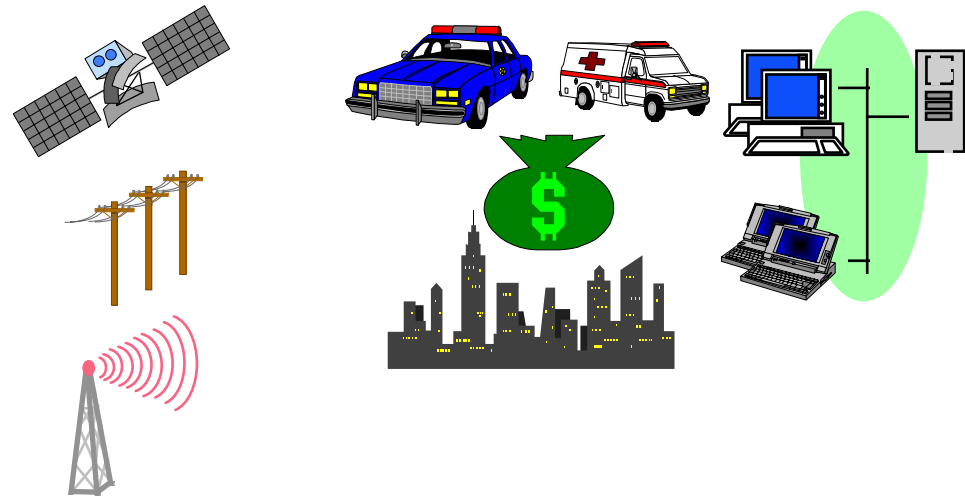
Skalbar



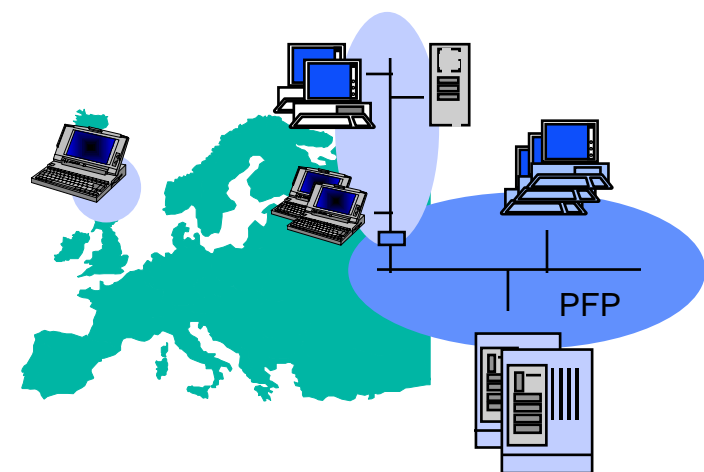
Olika verksamheter



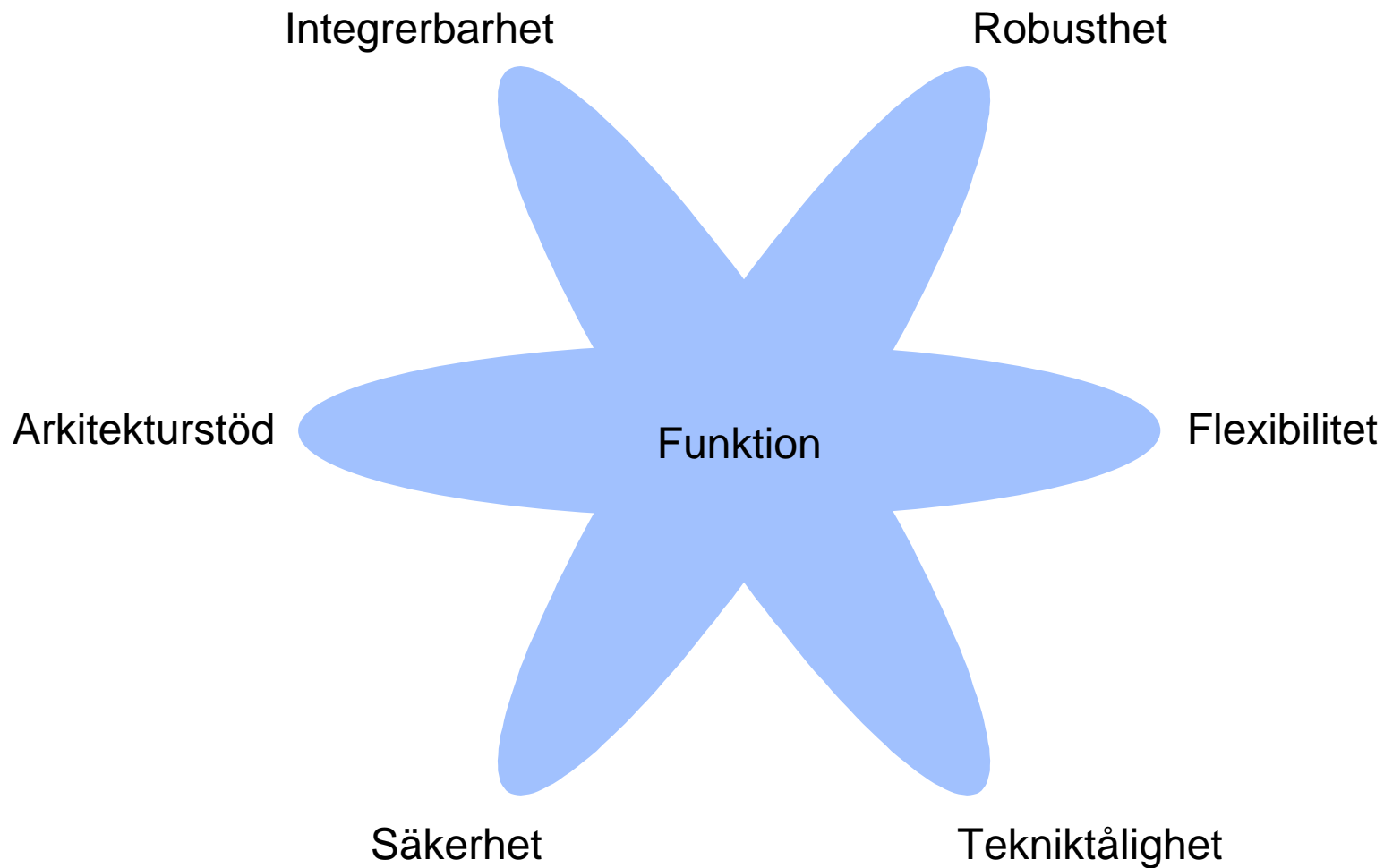
Kommunikation Civil version



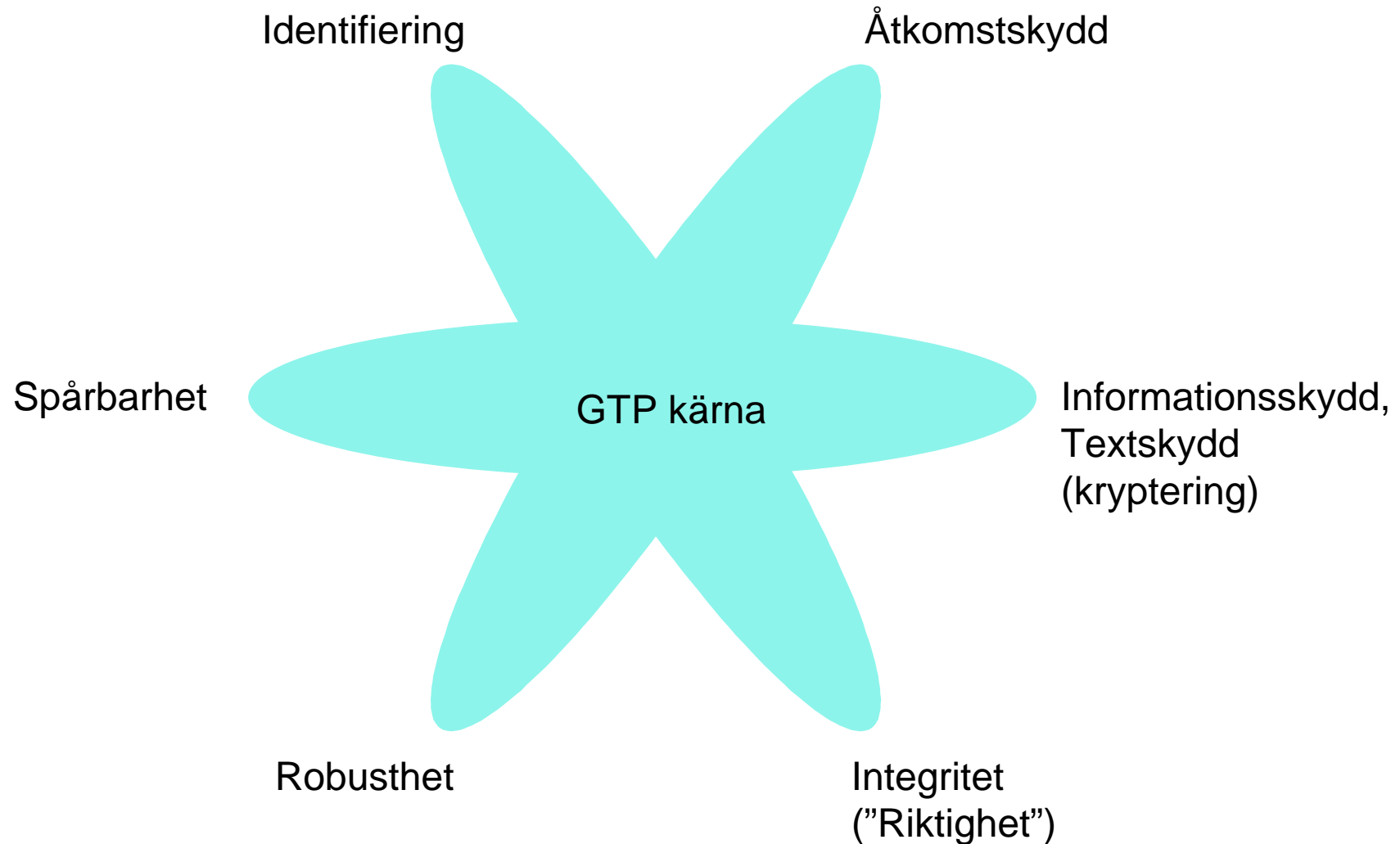
Internationell samverkan



# Tillhandahålla stöd för Systemegenskaper

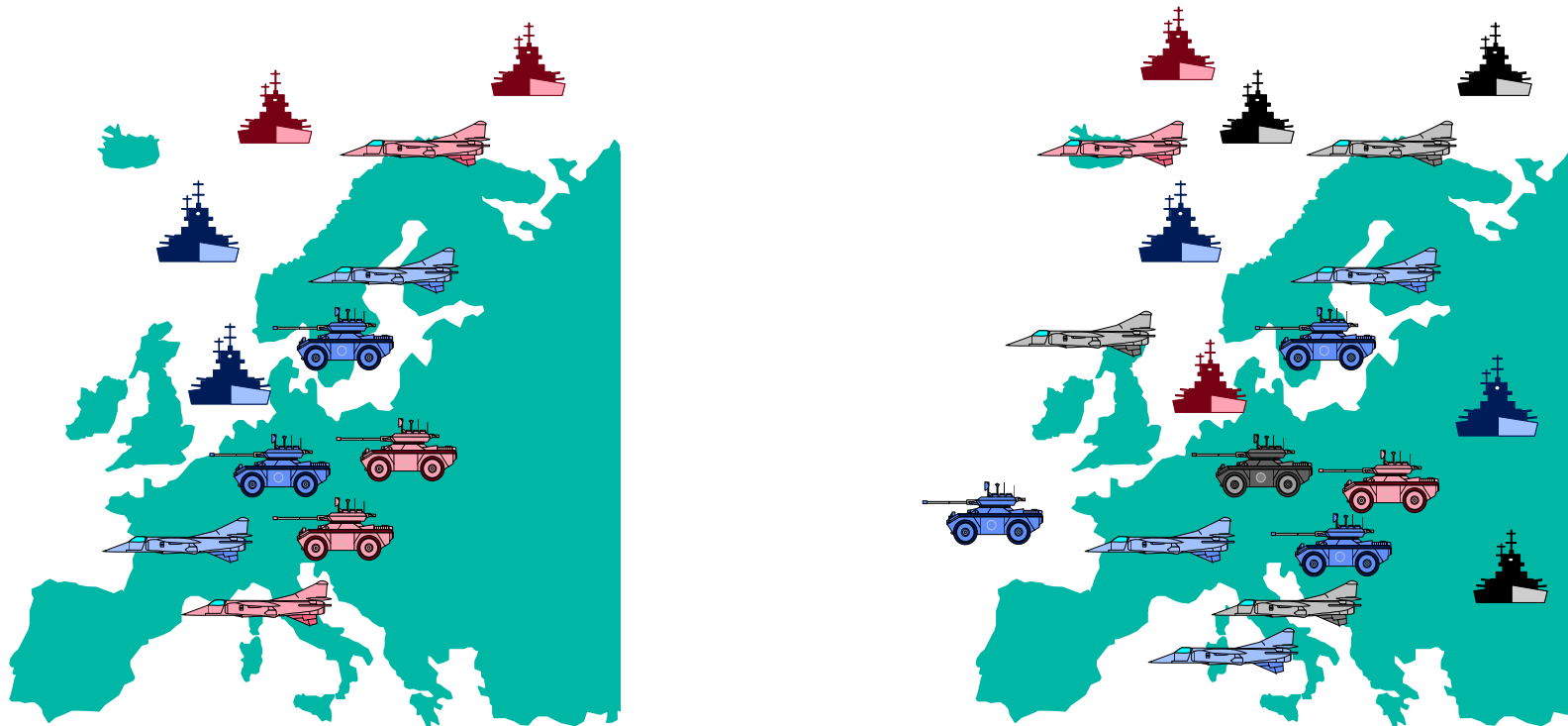


# Säkerhetshöjande funktioner



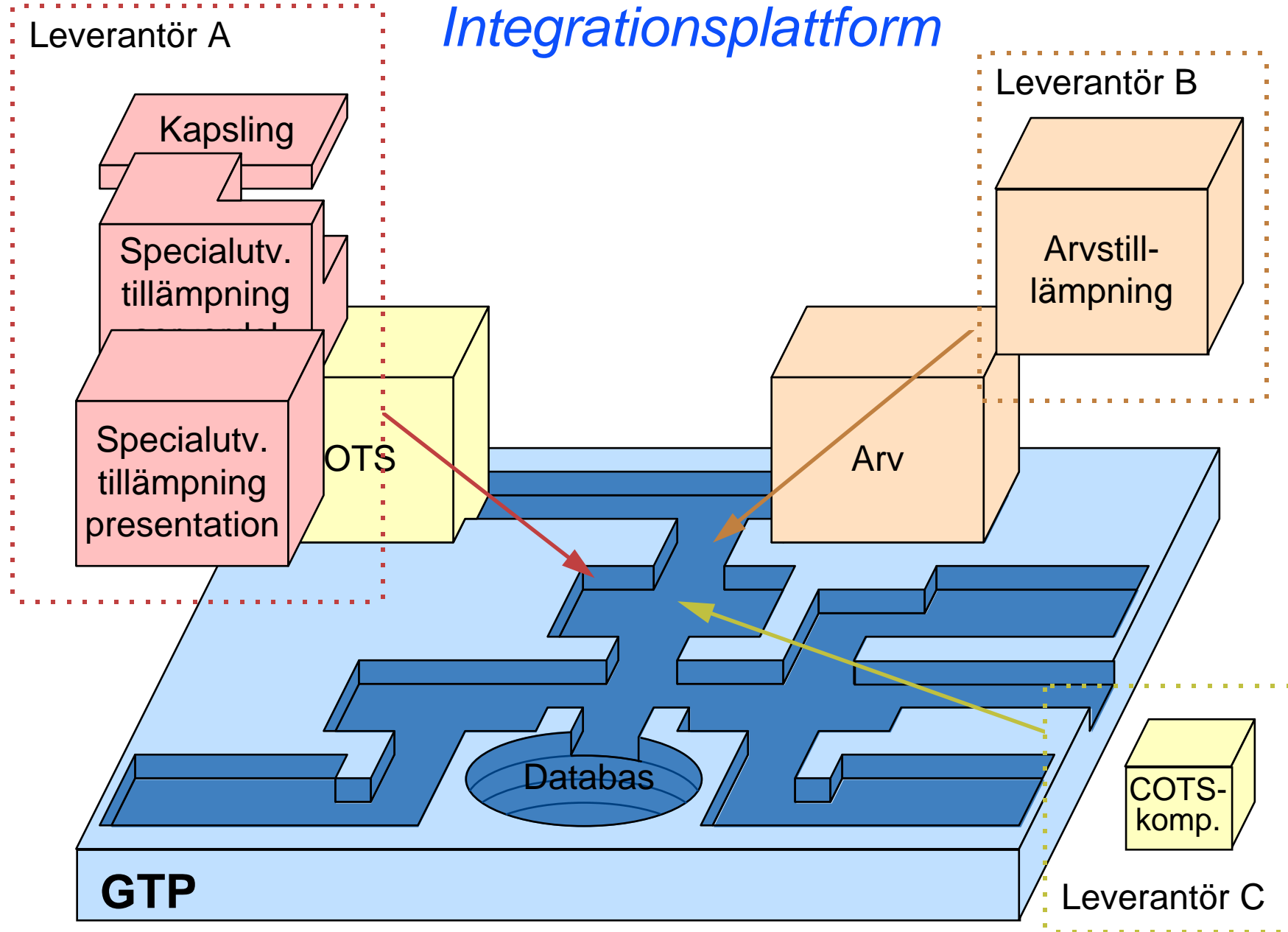
# *"Riktighet"? Situation Awareness*

If this is the real-world situation... ...you don't want the systems to show this



The use and dependence of IT is accelerating  
=> New possibilities but also new risks

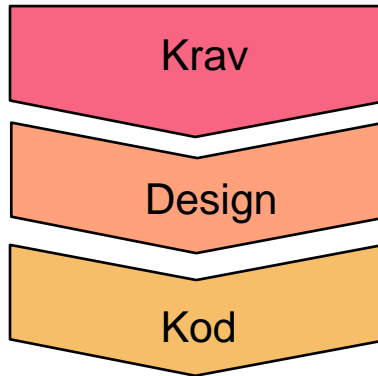
# Integrationsplattform



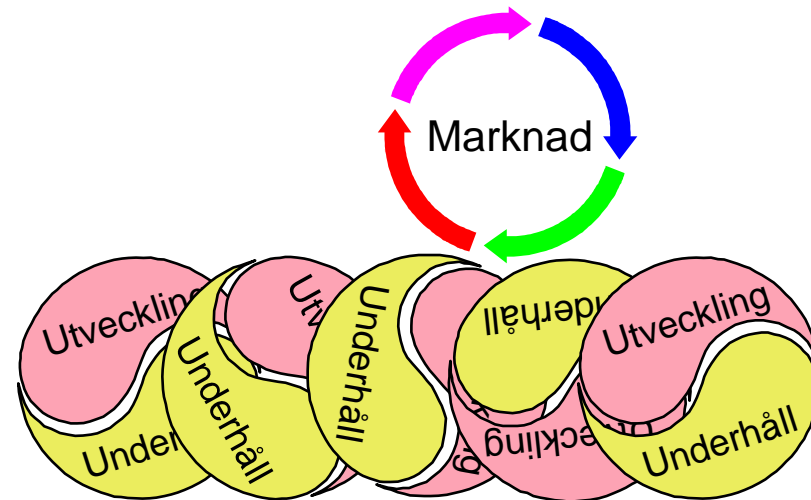
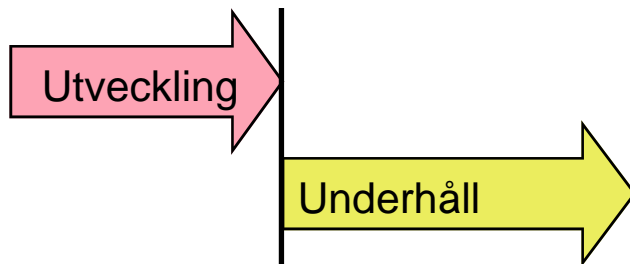
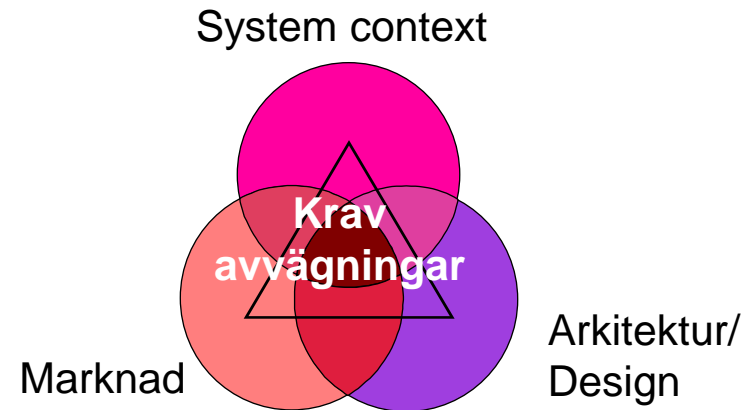


# Med COTS för COTS

Traditionellt



med COTS



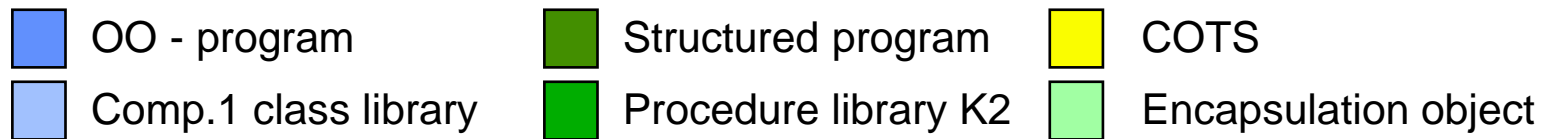
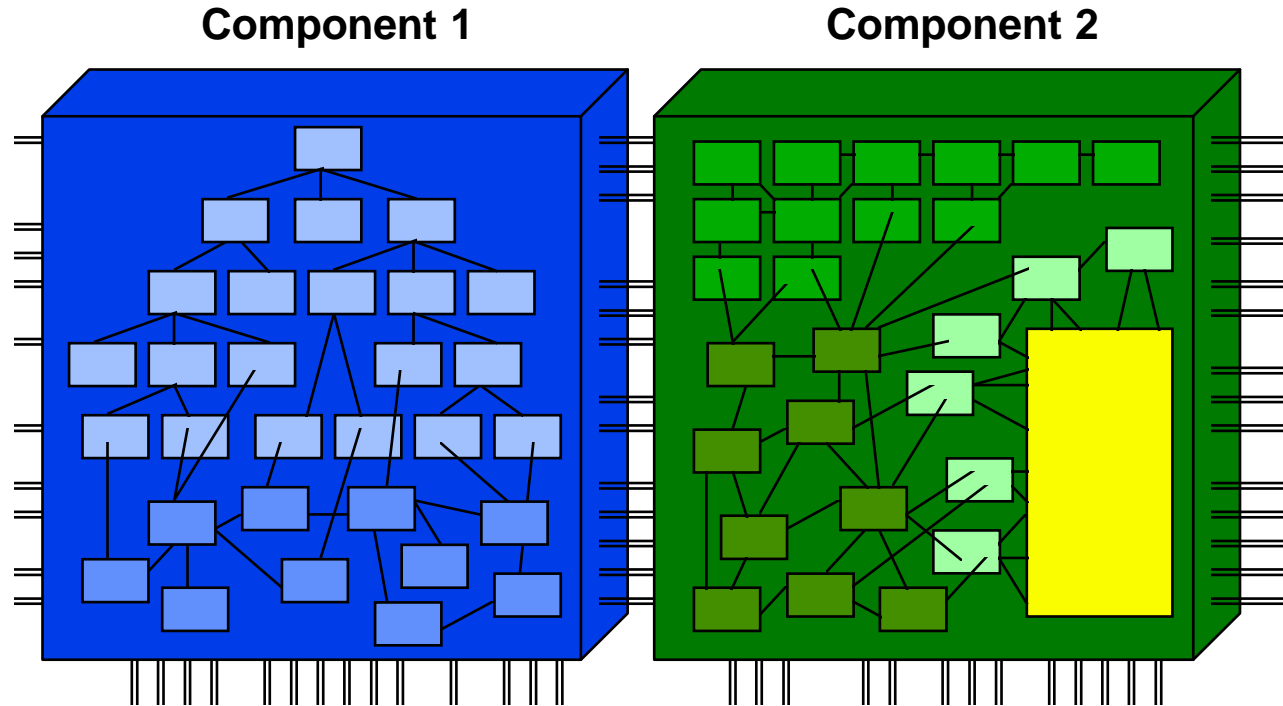
# GTP Teknisk Arkitektur - Strategi

- GS/GTP stöder olika tekniska tillämpningsarkitekturer
- GTP självt är komponent- och nätverksbaserat, körs i / stödjer heterogena systemmiljöer (NT, olika UNIX), middleware som systembus
- Tjänster i middleware (och infrastruktur) är *kapslade* (e.g. i TeS komponenter)
- Säkerhet och robusthet är implementerade i TeS, genom nyttjande av underliggande funktionalitet i middleware och OS
- Användningen av GTP funktioner för systemegenskaper och säkerhet är automatiserad för COTS respektive special- ("egen"-) utveckling
  - För COTS - automatiskt i fil-baserade, protokoll-baserade eller via standard API-er (GSS API)
  - Vid utveckling - "Säkerhets pre-processor" (TeS SecP) - utgår från komponenters gränssytor beskrivna i IDL
- Enhetliga verktyg för administration och konfiguration via TeS komponenter - SysA, SecAdm, RTCS, ...

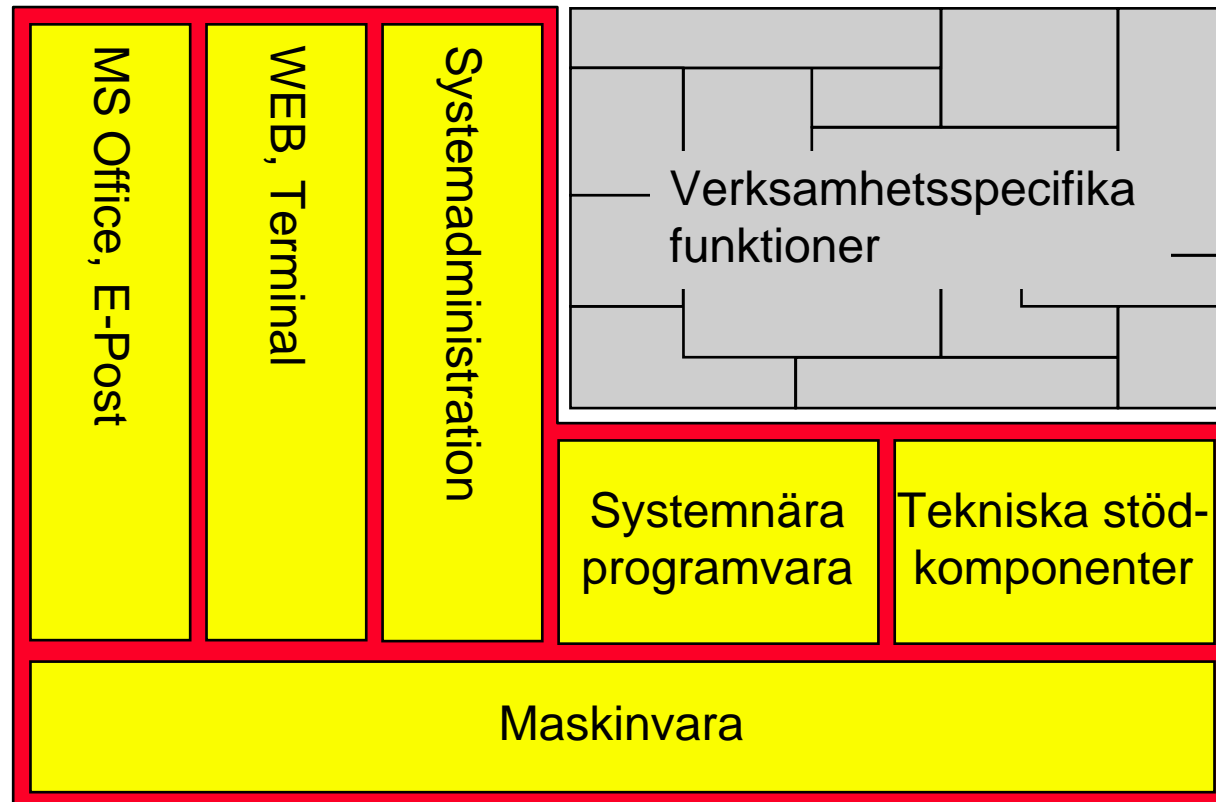
# Arkitektur - vy komponentsamverkan

- Objektorientering - grovgranulära objekt
- Komponenter
  - Flerskikts client/server, olika samverkanssätt
  - Middleware som integrationsplattform/systembuss
- System
- Konstruktionsprinciper
  - Komponenter
  - Komponentersamverkan
  - System
  - Systemsamverkan
  - ...

# Components

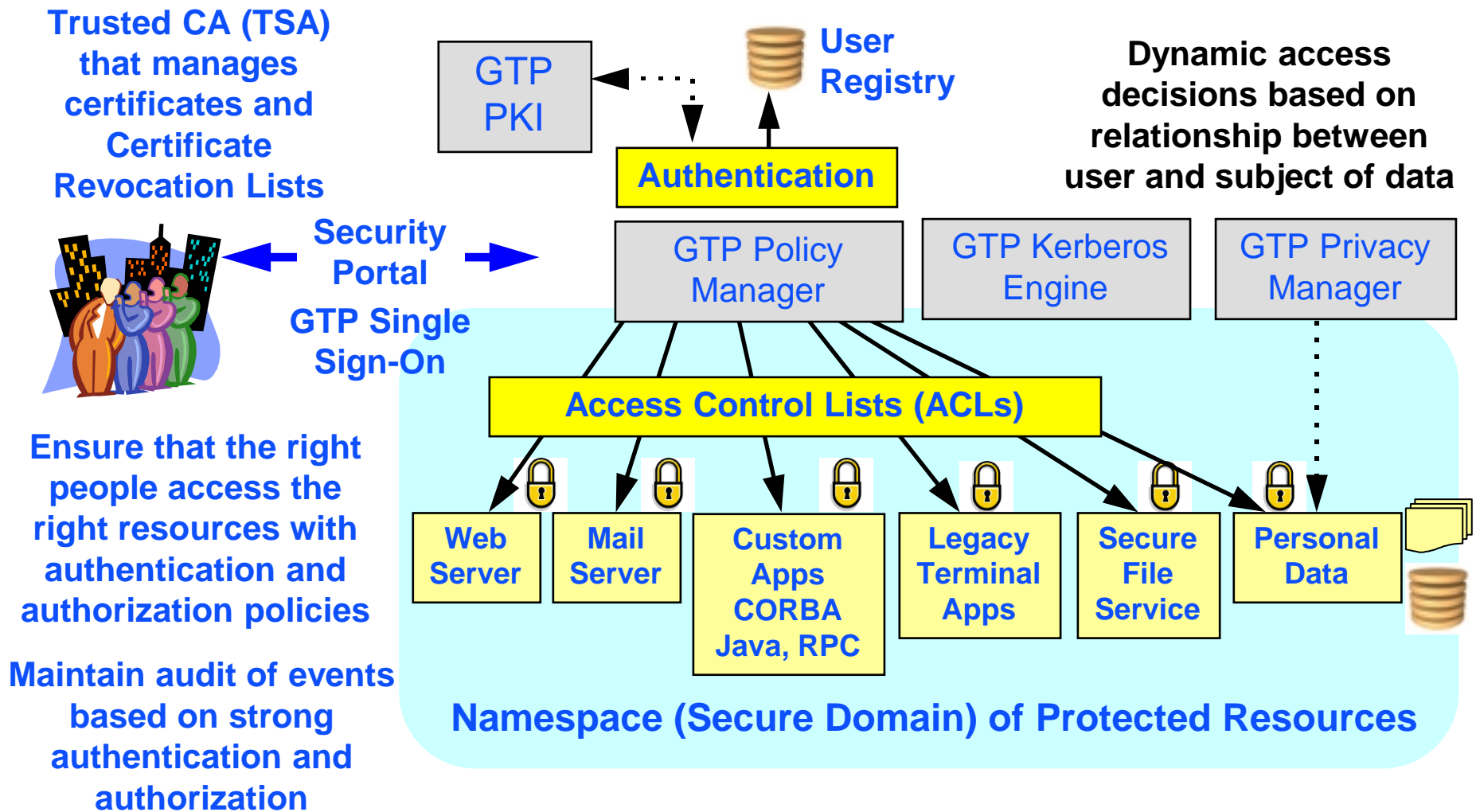


# GTP - Produktgruppsindelning



- Funktionalitet indelat i logiska komponenter som kan realiseras på olika sätt (med olika produkter) över tiden / parallelt

# GTP Network Centric Security

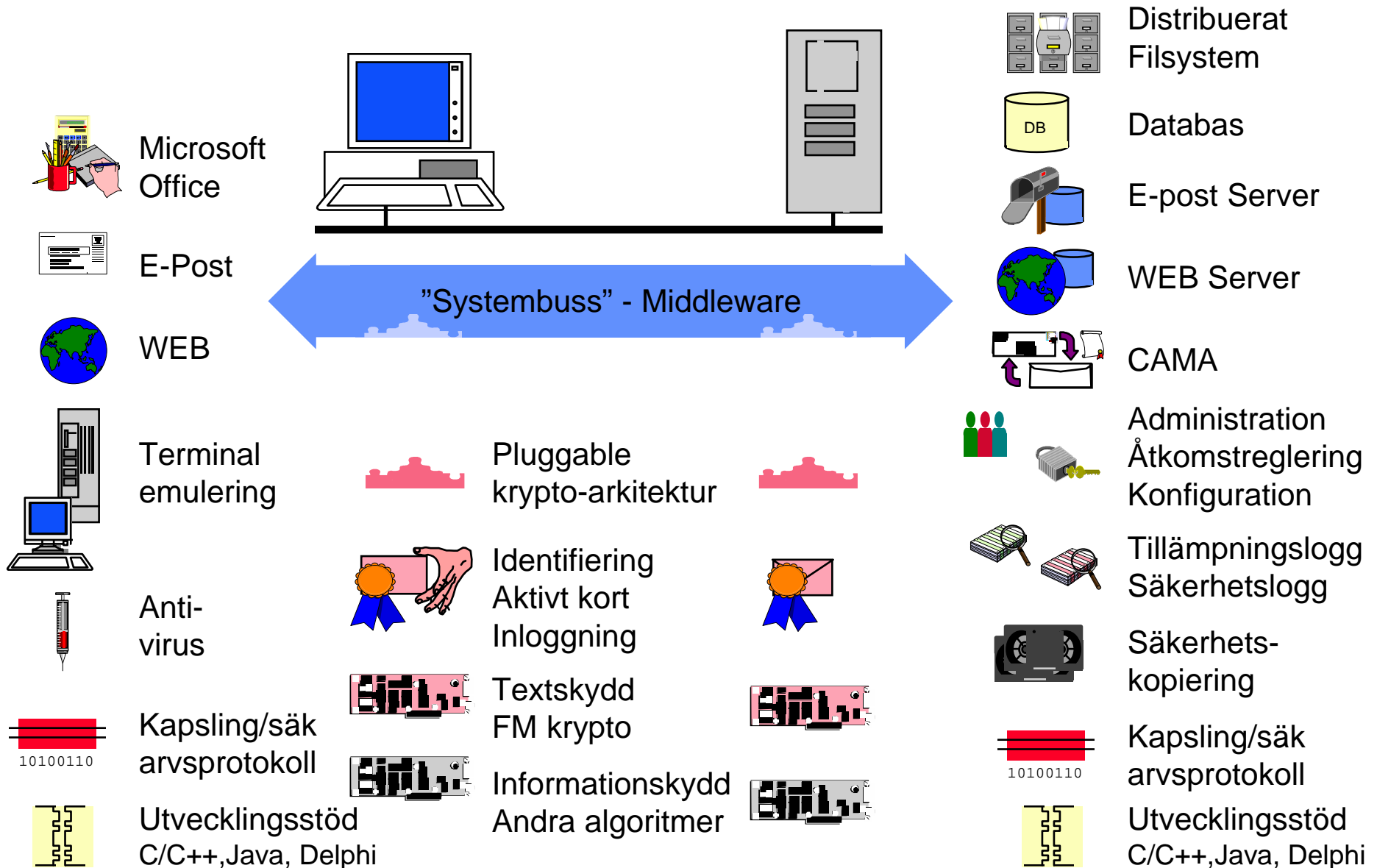


Ensure that the right people access the right resources with authentication and authorization policies

Maintain audit of events based on strong authentication and authorization

Implement permissions policies and personal data access controls, to enable the trust and security needed for mission critical applications

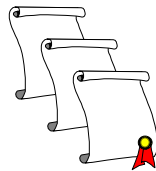
# GS/GTP funktioner



# Focus on "What" not "How"

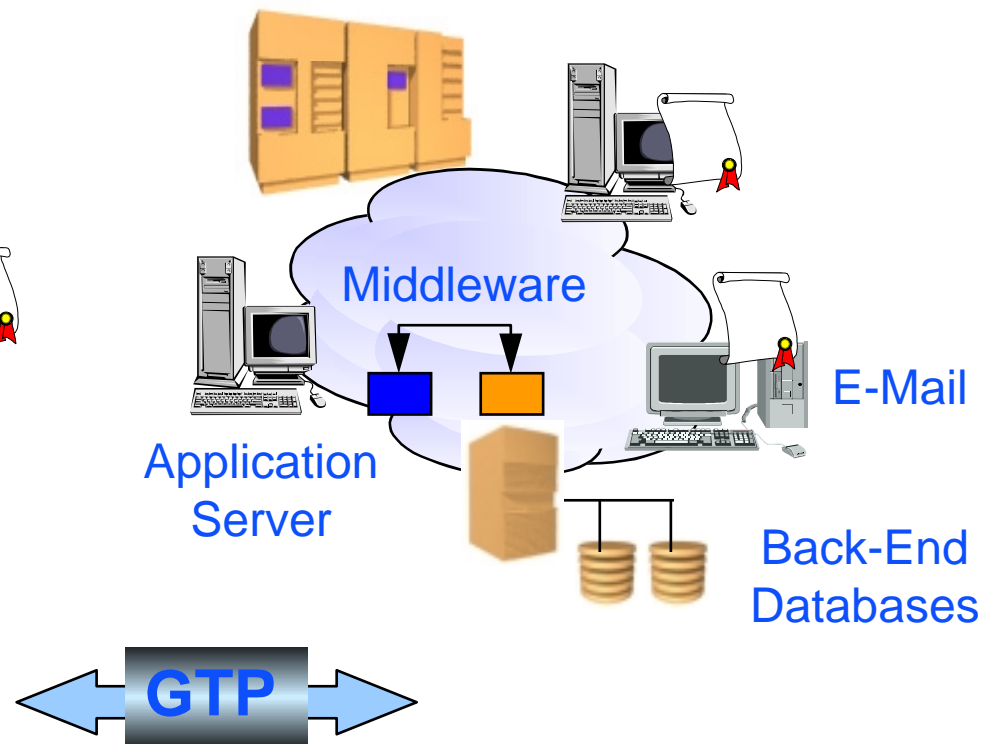
**GTP enables administrators to focus on what they want to accomplish**

**1) Define GTP Profiles that define the "What".**



**2) Apply the right policies to the right systems, applications or functions. Inspect and try out "What if" scenarios.**


**3) GTP handles the details in "How" tasks are executed across heterogeneous platforms.**





# SecL - GTP Single SignOn


**Inloggning GTP**




Inloggnings information

Behörig personal skall identifiera sig för systemet.  
Obehörigt utnyttjande beivras.  
Loggning av aktiviteter i systemet sker.

Sätt i aktivt kort i kortläsare eller  
tryck Ctrl + Alt + Del för att logga in.

 **FÖRSVARSMAKTEN**


**Uppdatera rättigheter**



Rättigheterna för nuvarande inloggning upphör snart, ange din användarpin (PIN ANV) för att förnya dina rättigheter.

OK

**Välj användare för inloggning.**



Flera konton finns kopplade till denna användare, välj önskat konto.

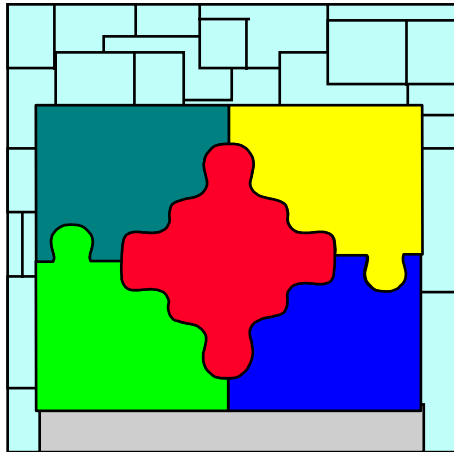
NT Användarnamn:

DCE Användarnamn:

OK Cancel

# GTP Example Configurations

## SD Command & Control System



Specialpurpose C2I components



End-user general purpose applications, OA



Software for systems administration



System software (OS, DBMS etc.)



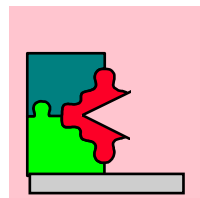
Technical Support Products, (sec./robustn./flex.)



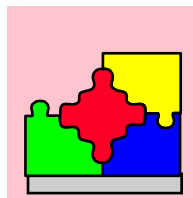
Development tools



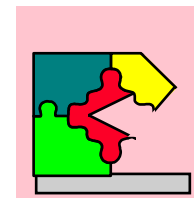
Hardware



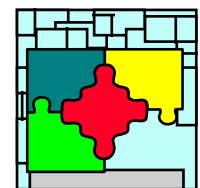
Configuration for Staff work, OA



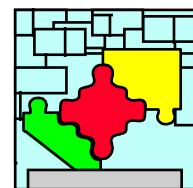
Configuration for dev. of C2I-sys.



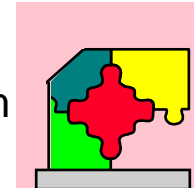
Basic config. for systems with lower req. on sec./robustn.



Configuration for Tactical/Op. C2-system with OA

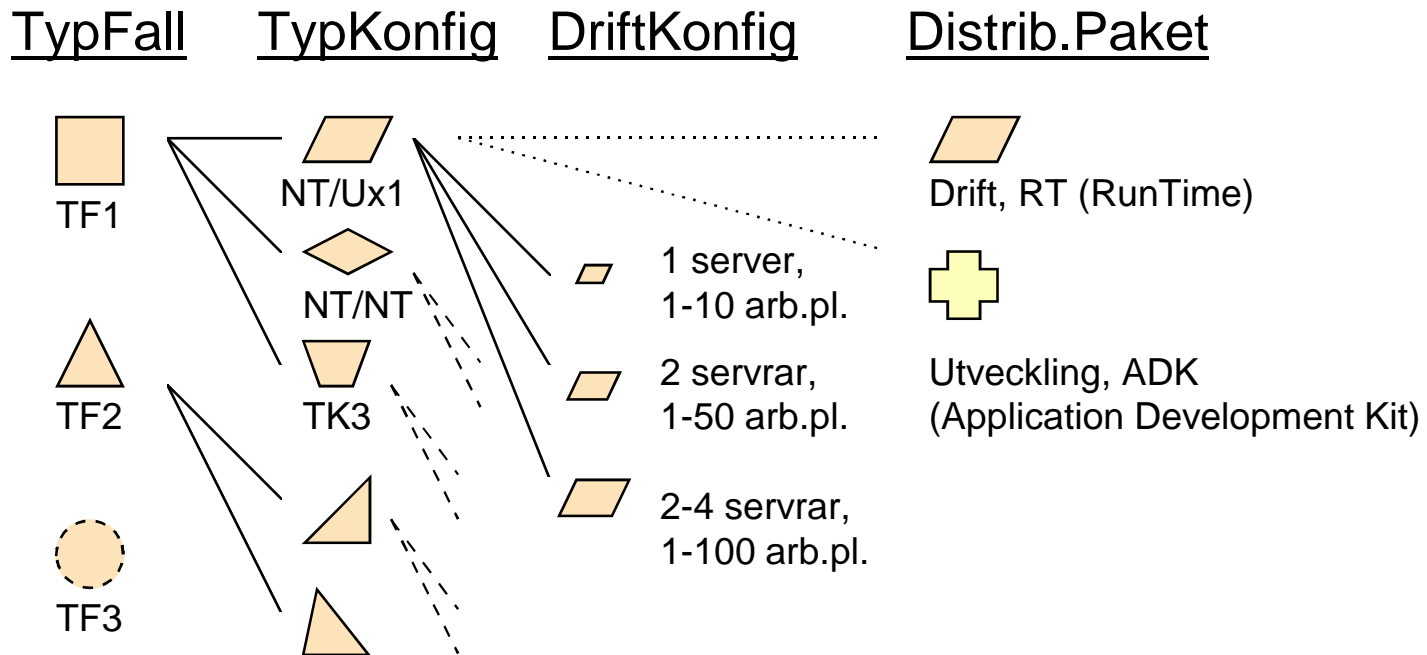


Configuration for Tactical C2-system without OA



Basic config. for systems with higher req. on sec./robustn.

# Driftkonfigurationer

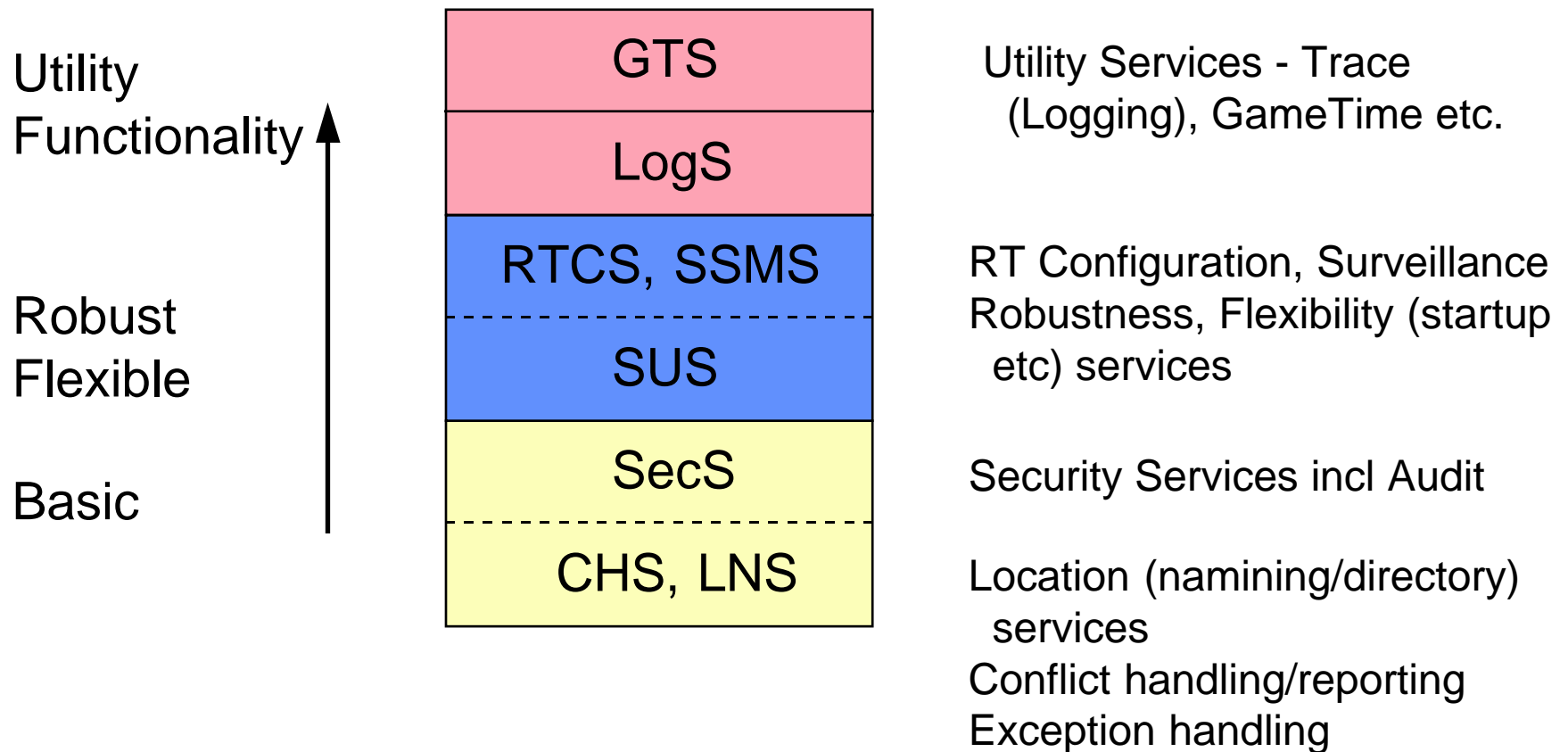


# Arkitektur - vy komponentsamverkan

- Realisering med teknisk arkitektur - middleware som systembuss
- Komponentssamverkan - grundläggande behov
  - via gränssytor, kunna "förstå" - konflikter, hitta tjänster, inbyggd säkerhet, ...
  - kompletterande tjänster ber. på användning
- Middleware - strategi, arkitektur, produkter
  - identifiera och abstrahera grundläggande tjänster ...
  - . . . felåterhämtning, lokalisering tjänster, säkerhet, övervakning/styrning tjänster, etc
  - realisera med COTS, kapslingar och tillägg där så nödvändigt

# StepWise use of TeS

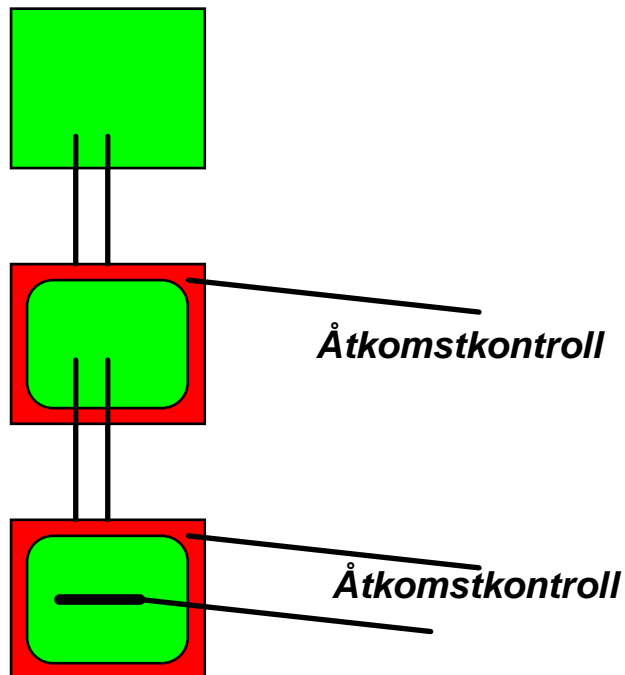
*TeS role in Architecture? To encapsulate and abstract services for networked based systems - to achieve ease-of-use, flexibility, technology idependence.*



# Säkerhet i "middleware"

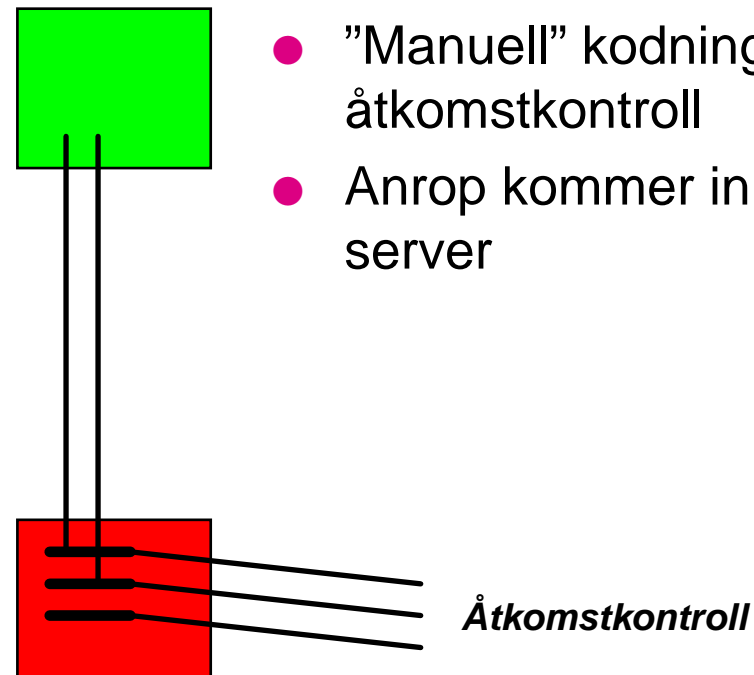
"Nu ..."

- Assurans och automatik
- Skaltänkande, skydd i gränssytor



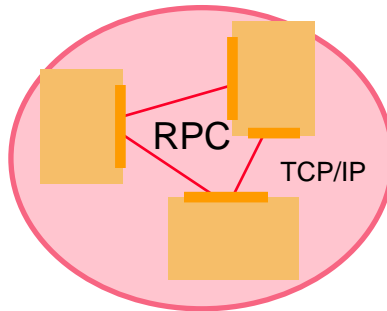
"Förr ..."

- "Manuell" kodning av åtkomstkontroll
- Anrop kommer in i server



# Plattform av komponenter och för komponenter

## Förr (och nu)

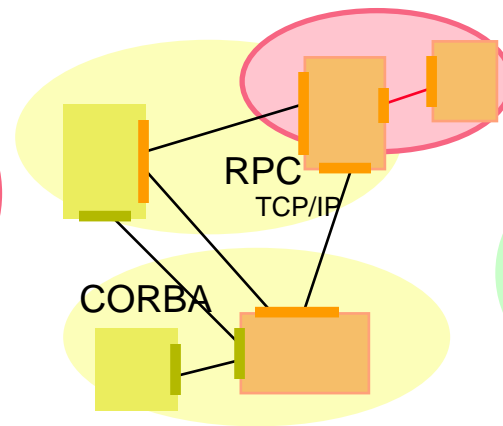


### Samverkan inom ett fysiskt skyddat LAN.

Komponentsamverkan med RPC via väldefinierade gränssytor (IDL).

COTS samverkar med egna protokoll baserat på TCP/IP (och ofta RPC).

## Nu (och senare)

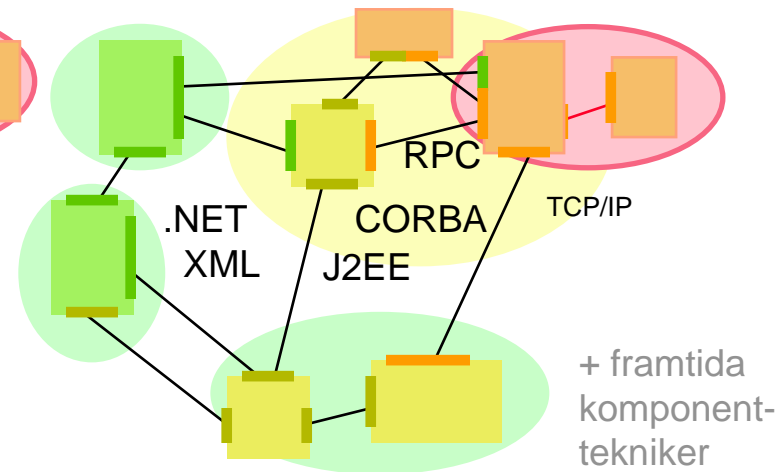


### Samverkan även mellan LAN och utanför skyddade LAN.

Komponentsamverkan med CORBA IIOP och/eller RPC via väldefinierade gränssytor (IDL).

COTS samverkar med egna protokoll baserat på TCP/IP (och ofta RPC).

## Framöver (och senare)

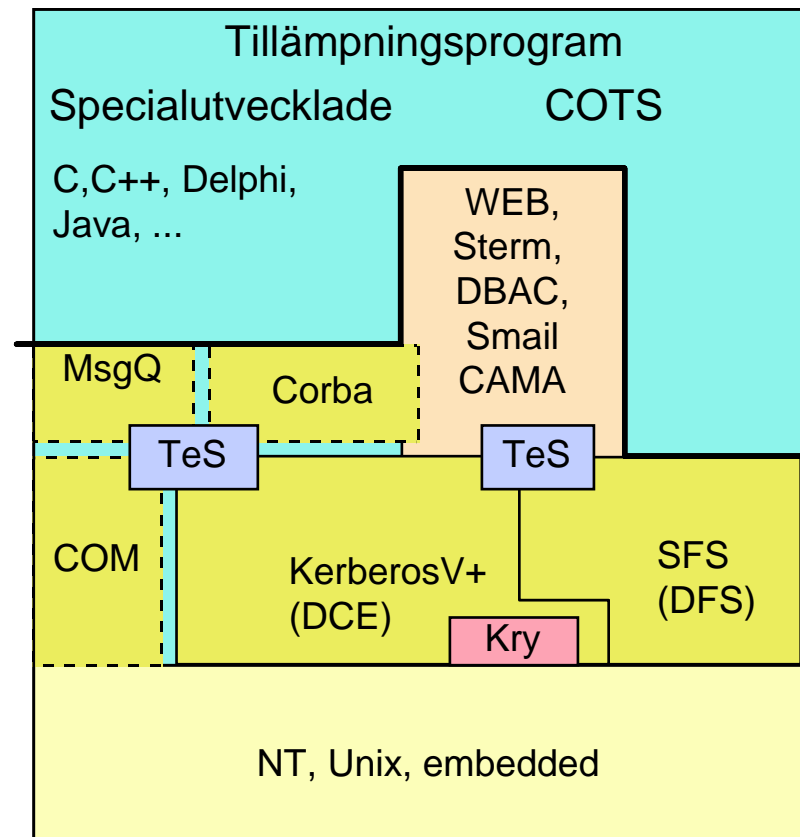


### Samverkan även mellan fristående komponenter i nät.

Komponentsamverkan med XML och/eller annat gränssytebeskrivningspråk samt CORBA IIOP och/eller RPC.

COTS samverkar med egna protokoll baserat på TCP/IP.

# Integration andra middleware/RPC - Corba, Java, ... - lagermodell





# Summering

- Förutsättningar - arkitektur ?
- Avvägningar vid utformning av arkitekturer
  - samverkansarkitektur - tekniska perspektiv
  - infrastrukturbaserad utveckling, "ramverk"
- Typarkitekturer
  - "tillämpningsarkitekturer", "mönster"
  - "middleware"
- Arkitekturdriven utveckling - livscyklar
- Arkitekturdriven utveckling - exempel
  - FM GTP - Generell Teknisk Plattform
  - Generell teknisk arkitektur, logiska komponenter, olika realisiering (parallelt) över tiden ...  
*infrastrukturbaserad utveckling*
  - Exempel olika perspektiv/vyer