DR TEKSTAB

System Integration Architecture.

How to manage it in an IT-based Broadcast environment

NMC Seminar 16th of June 2004

Scope for the presentation:

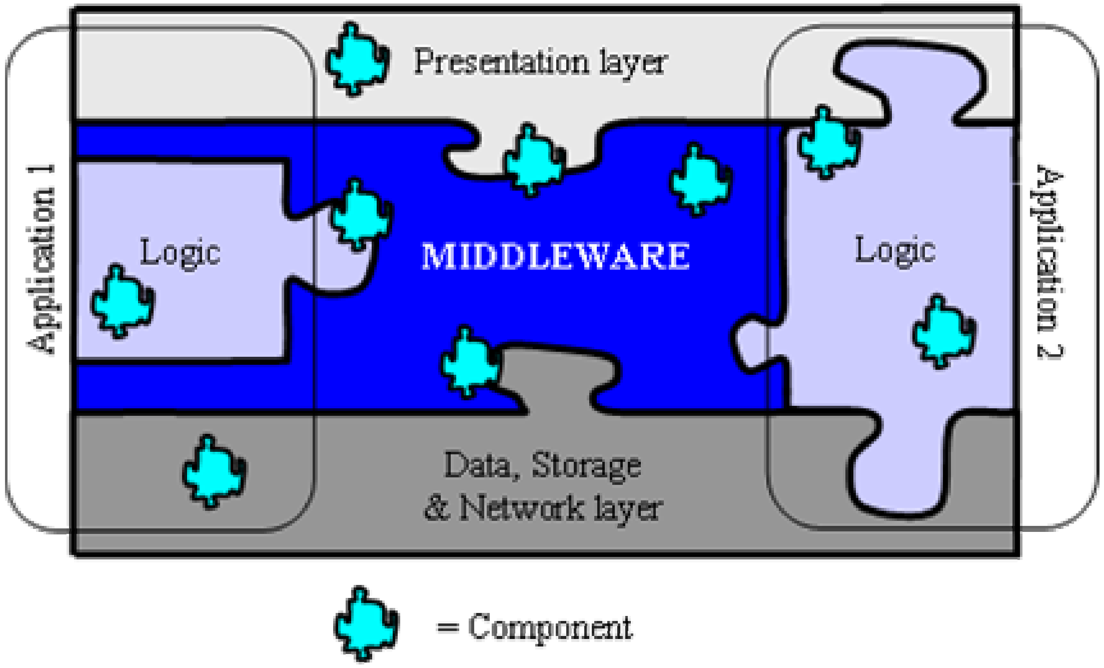
- The overall look at System integration in DR – under the pressure of moving to a new Broadcast house, where production will be ‘tapeless’
- Our experiences as a broadcaster with System integration and Middleware
- The intro told under - and enriched by - the growing consensus in the p-mdp working group about terms and arguments

The landscape

- We used to work with isolated systems, interfaced by people - co-existence.
- The main task of system integration today is turning systems co-existing into systems co-operating. To collect systems into Super systems when appropriate and efficient to optimize our workflow.

The landscape

- To make this work, we need some glue to connect it all together.



Why - the challenges

- We work increasingly across formerly isolated business systems.
- The products (e.g. programmes) we create are more diverse and spread over many media platforms.
- Workflows are changing rapidly as well
- We need optimisation of our processes and their supporting technology to make the needed improvements in outputs or costs

Why - the challenges

- Operations become much more spread out in our physical environment.
- The technology used is also much more diverse and distributed.
- Systems are getting more and more split into components
- Isolated systems are replaced by modular combinations of functions and data.

Why - the challenges

- The amount of integrations are exploding
- Then systems can't be managed as isolated islands with few (but important) connections anymore
- We are forced away from the system perspective with a few genius developers around each system to make things happen and fix the bugs from their (and the systems) point of view alone.

Why - the implications

- Increasingly central structure, control and change-management are needed
- We must decide: What we do and what we don’t – which tool we use and which ‘endless opportunities’ we leave behind
- We must ‘own’ our metadata - and data models, - and make our:

System Integration Architecture

Why – the expected benefits

Flexibility: Fast changing work-processes and work-flows need flexible access to functions and data.

Speed: Where functions and data have to be used across systems, middleware speeds development and changes (in the long term - not necessarily in the short term).

Why – the expected benefits

Cost: The costs of building and maintaining unique one-to-one integrations between systems are rapidly growing.

A hub-and-spoke approach is more efficient as you only have to change the connection from the primary system to the hub.

Why – the expected benefits

New products and services: Easy and secure access to information across systems opens up opportunities for new products. Especially on our new delivery platforms.

Prevent lock-in by vendors: You can replace subsystems from vendor A by a new system from vendor B, without having to reconfigure other systems.

Why – the expected benefits

Standardization: As methods and interfaces are re-used fewer components and skills are needed.

Overview: As a common way of achieving integration is established, better overview is gained (processes and systems).

Data integrity: Much more data are used across systems and work processes. Better data integrity can be obtained.

DR- System Integration project

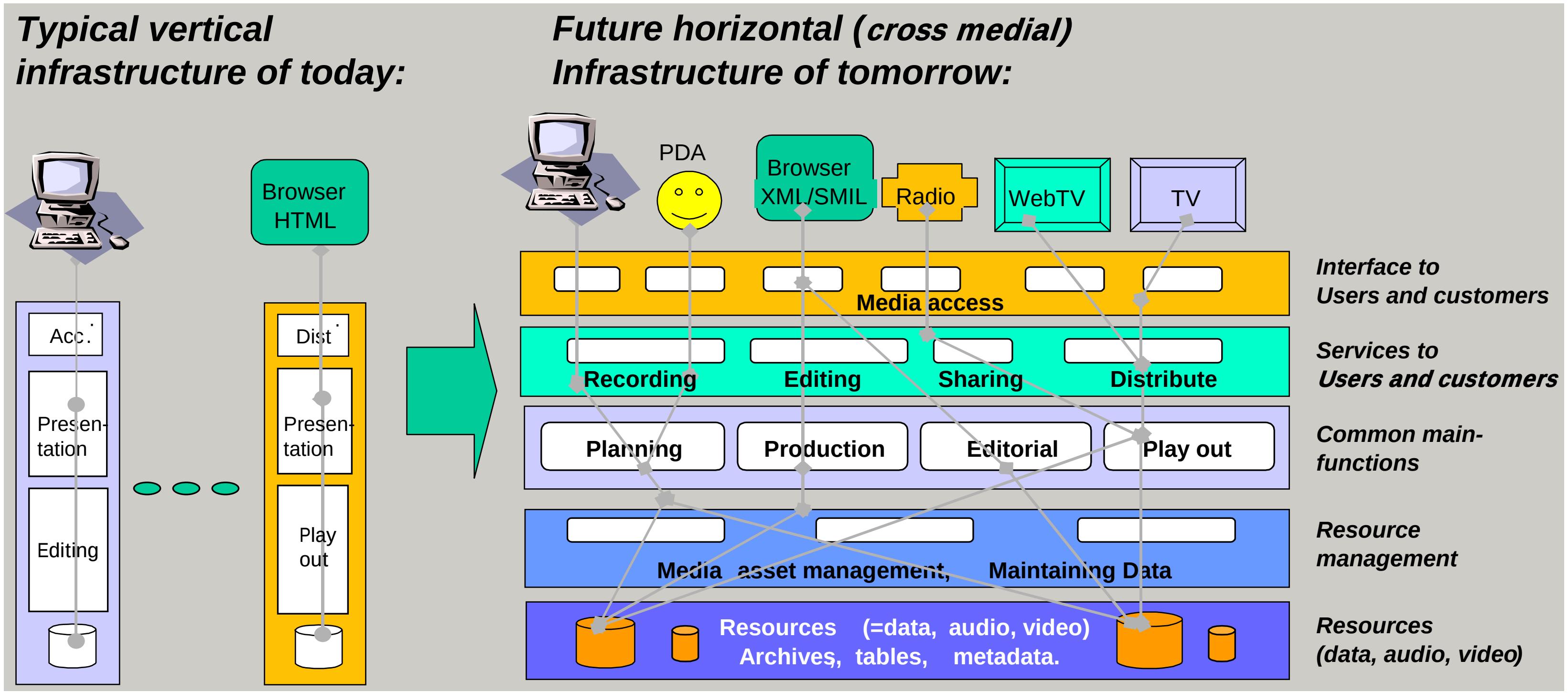
- Strategic analysis in 1999 regarding DR's future residence: DR-byen.
- One of the major conclusions regarding technology:

  'Communication between layers and modules shall be handled by middleware'

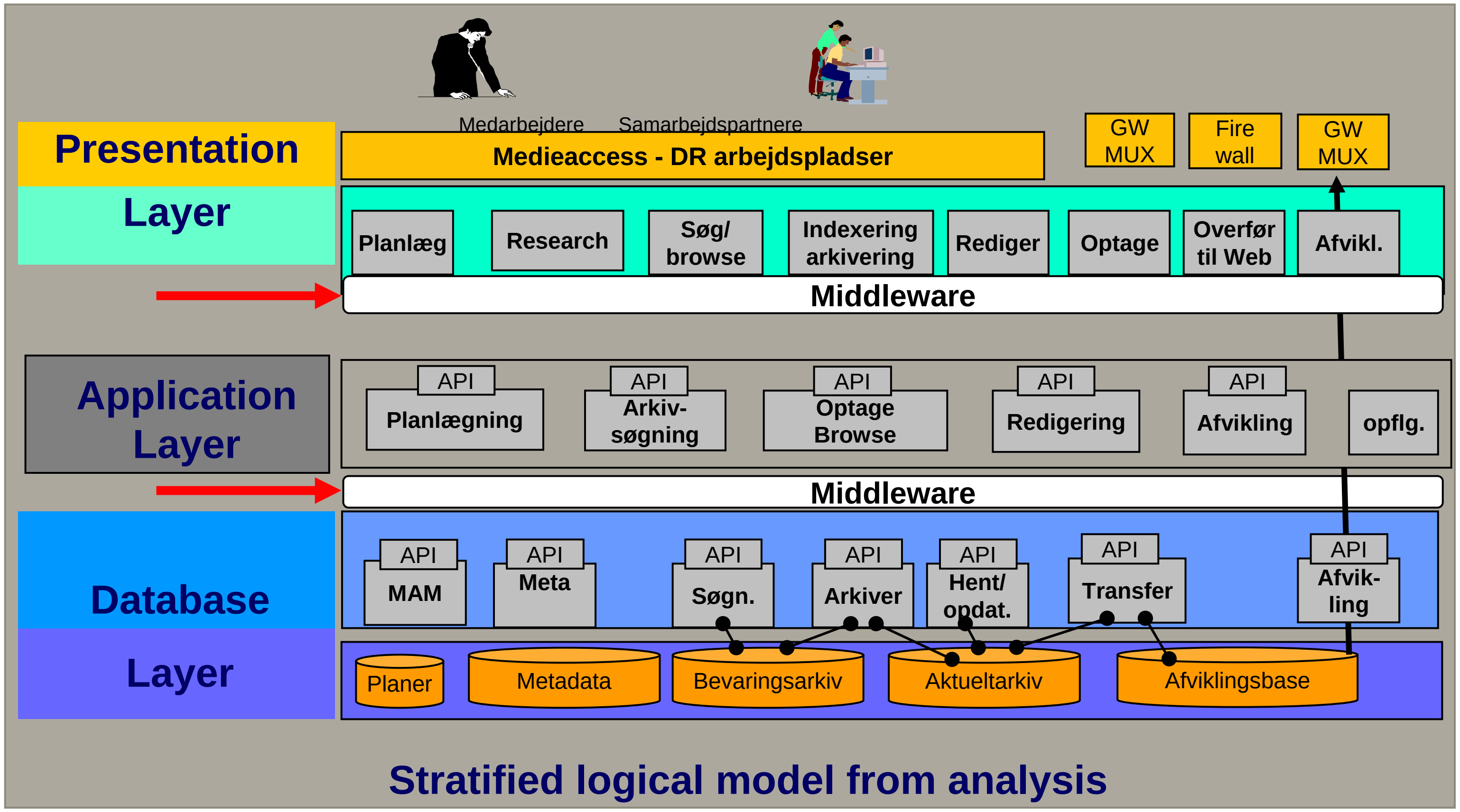
- System integration project formed 2001

DR- The expected change -

from vertical to horizontal IT-architecture


Typical vertical infrastructure of today:
Future horizontal (cross medial) Infrastructure of tomorrow:
Browser HTML
Acc.
Presen-tation
Editing
Dist
Presen-tation
Play out
PDA
Browser XML/SMIL
Radio
WebTV
TV
Media access
Recording
Editing
Sharing
Distribute
Planning
Production
Editorial
Play out
Media asset management,
Maintaining Data
Resources (=data, audio, video) Archives, tables, metadata.
Interface to Users and customers
Services to Users and customers
Common main-functions
Resource management
Resources (data, audio, video)

DR- The expectation of 'Middleware'



Stratified logical model from analysis

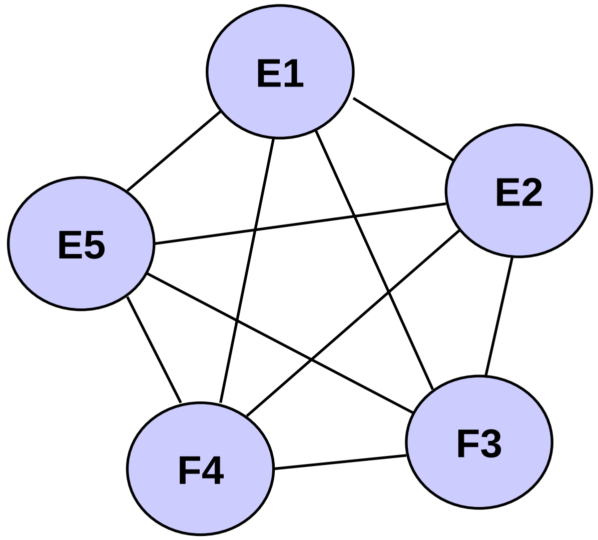
DR- System Integration project

- +60 relevant integrations localized
- 16 most important chosen and ‘prove of concept’ carried out
- +80 requirements for the platform
- 8 standard requirements to all systems
- Set of DR standards and policies for system integration developed

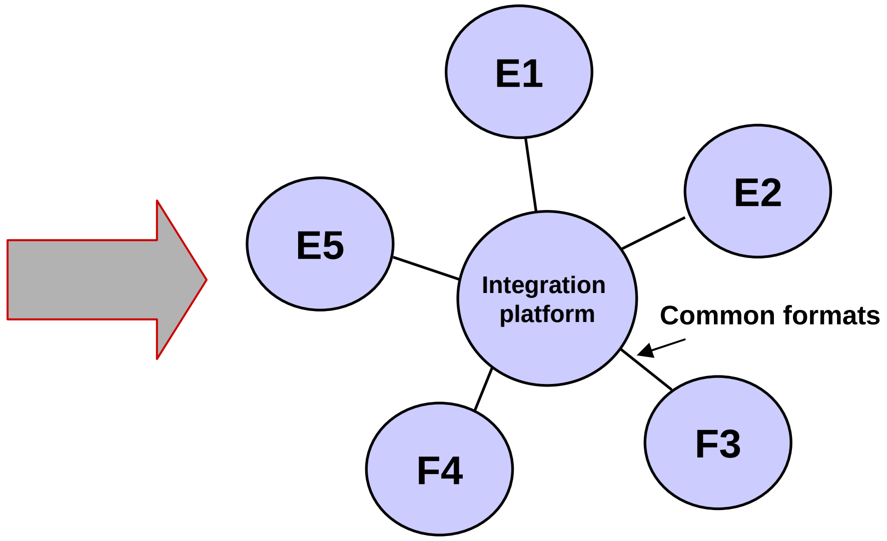
DR- System Integration project

- Integration platform acquired after market survey
- Basic setup and test done
- In use in daily operation
- Re-use of connections in progress (more and more systems gets the benefits)
- Integration standards and the estimation of the integration effort are more widely included in all major projects by now

DR- Moving to 'Hub and spoke'



N x (N-1)/2
convertings/agreements



N convertings/agreements

Complex, cloudy
And expensive to maintain – but cheap to do with few Systems

Greater effort to start with, but easier to maintain, scalable, cheap to connect to

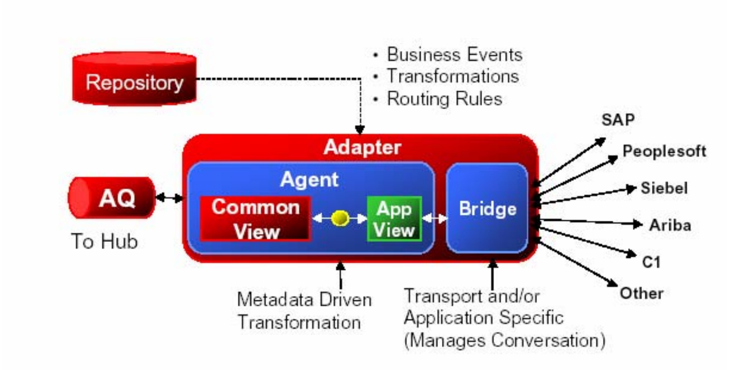
DR- 'Hub' in different ways:

- 'Hub' as a broker

  MOM

  Message oriented Middleware

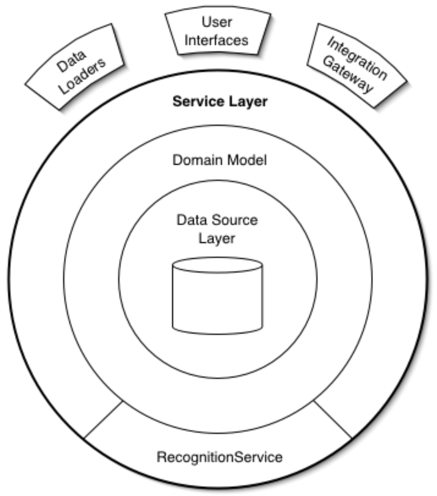
  Metadata-based transforming of data.
  Transport-technically protocol translation



- 'Hub' as a well-described and re-useable way of making complex services

  SOA

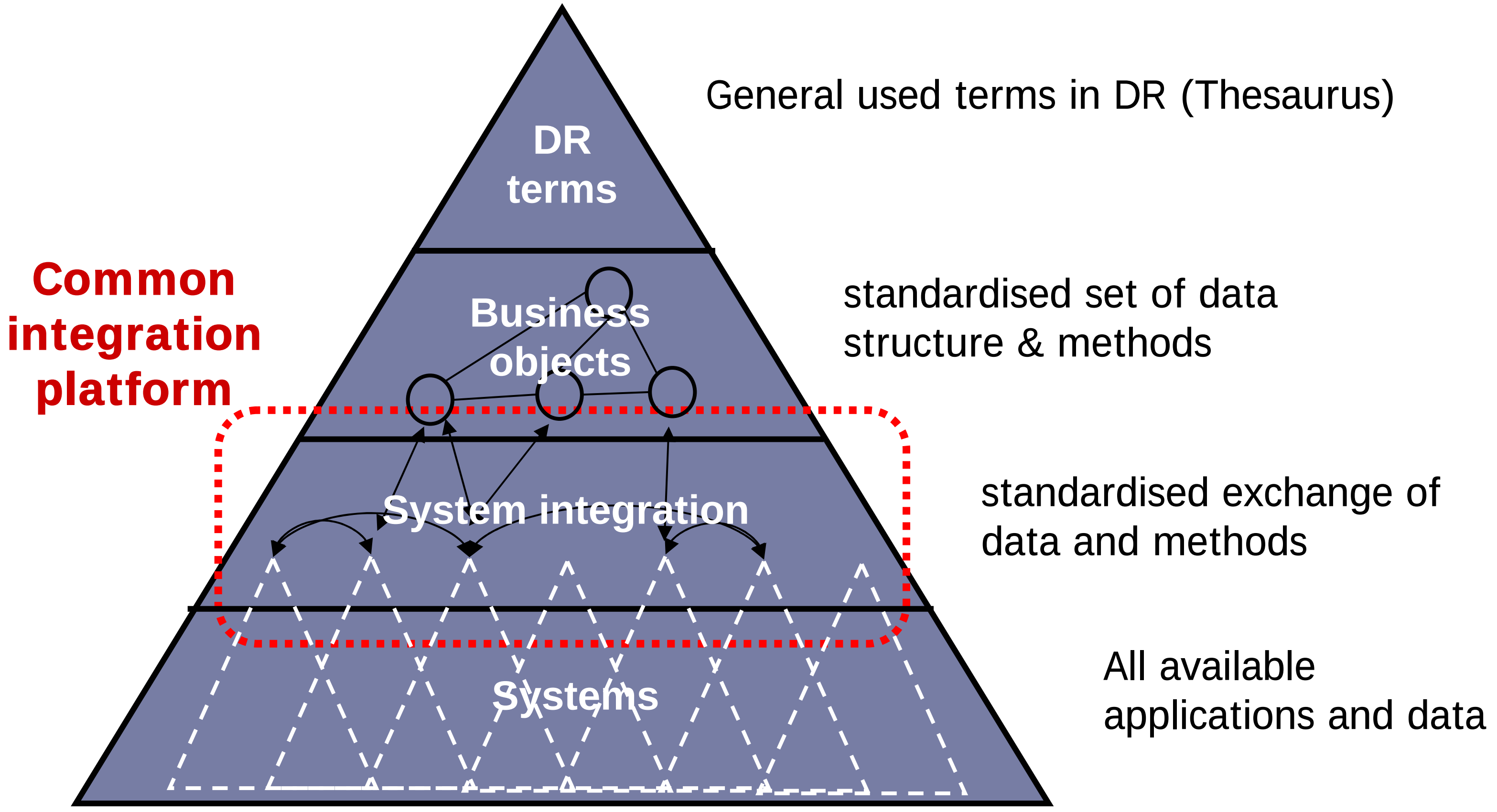
  Service oriented Architecture



DR- ‘Hub’ in different ways:

- ‘Hub’ as a central way of keeping the overview
- ‘Hub’ as a central way of manage and control the development process
- ‘Hub’ as a central way of manage and control the ongoing changes

DR- ‘Middleware’ is not enough:


Common integration platform
DR terms
Business objects
System integration
Systems
General used terms in DR (Thesaurus)
standardised set of data structure & methods
standardised exchange of data and methods
All available applications and data

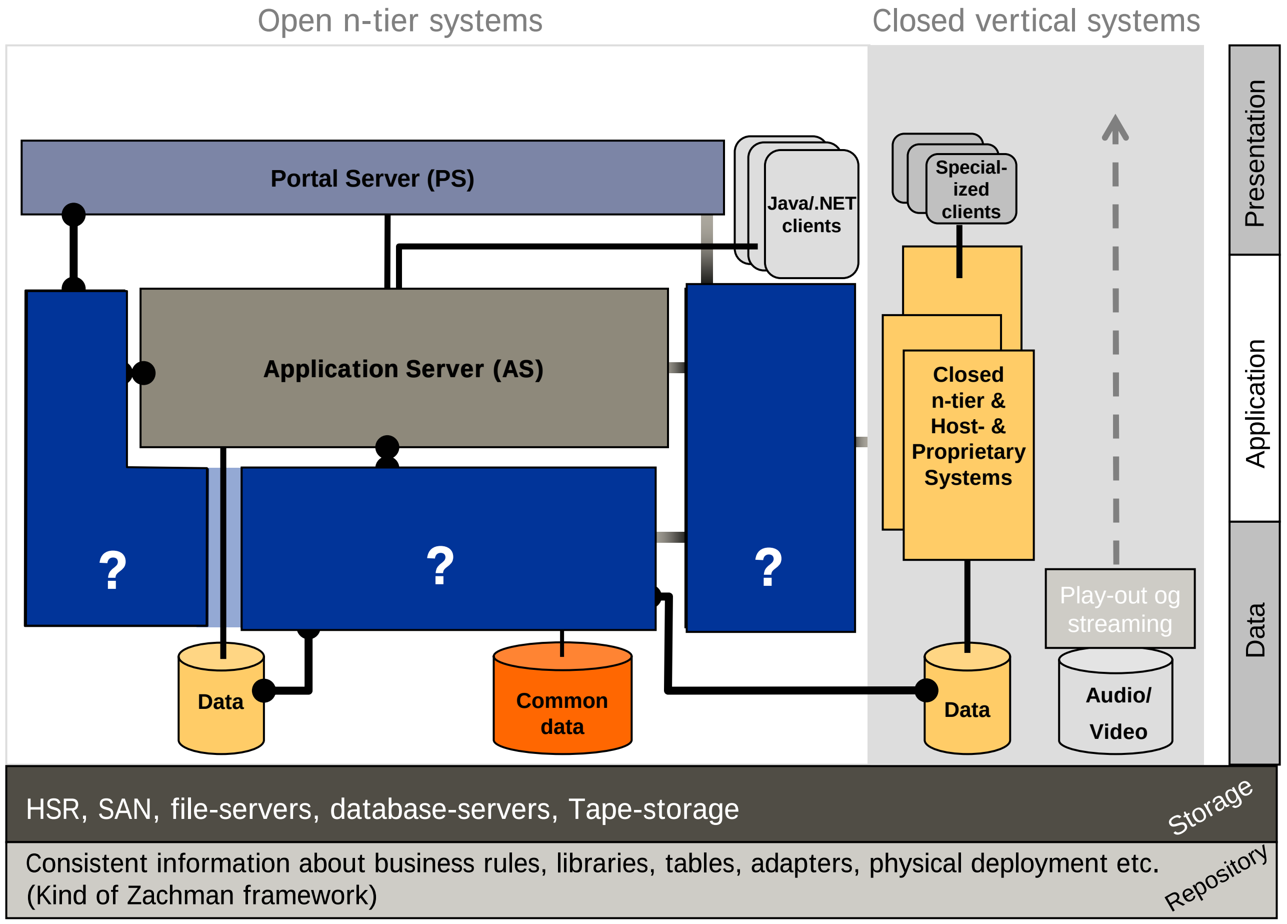
DR- Four major types of integration

- Message based integration
  (copying data between systems)
  - The same data is needed in different systems and an automated exchange mechanism can provide copies of data in these systems. It can be set up dependable or event-based. Functionality is made by integration broker.
- Data-carrying integration
  (Different systems access to the same data)
  - Data is only located in one copy in one place. All systems are working real time on the same instance of data. Integration is made by agreement between the systems about the data model they use, and about a common logic for reading and updating data. This is made by a data-carrying integration platform, primarily made by an application server and a database.

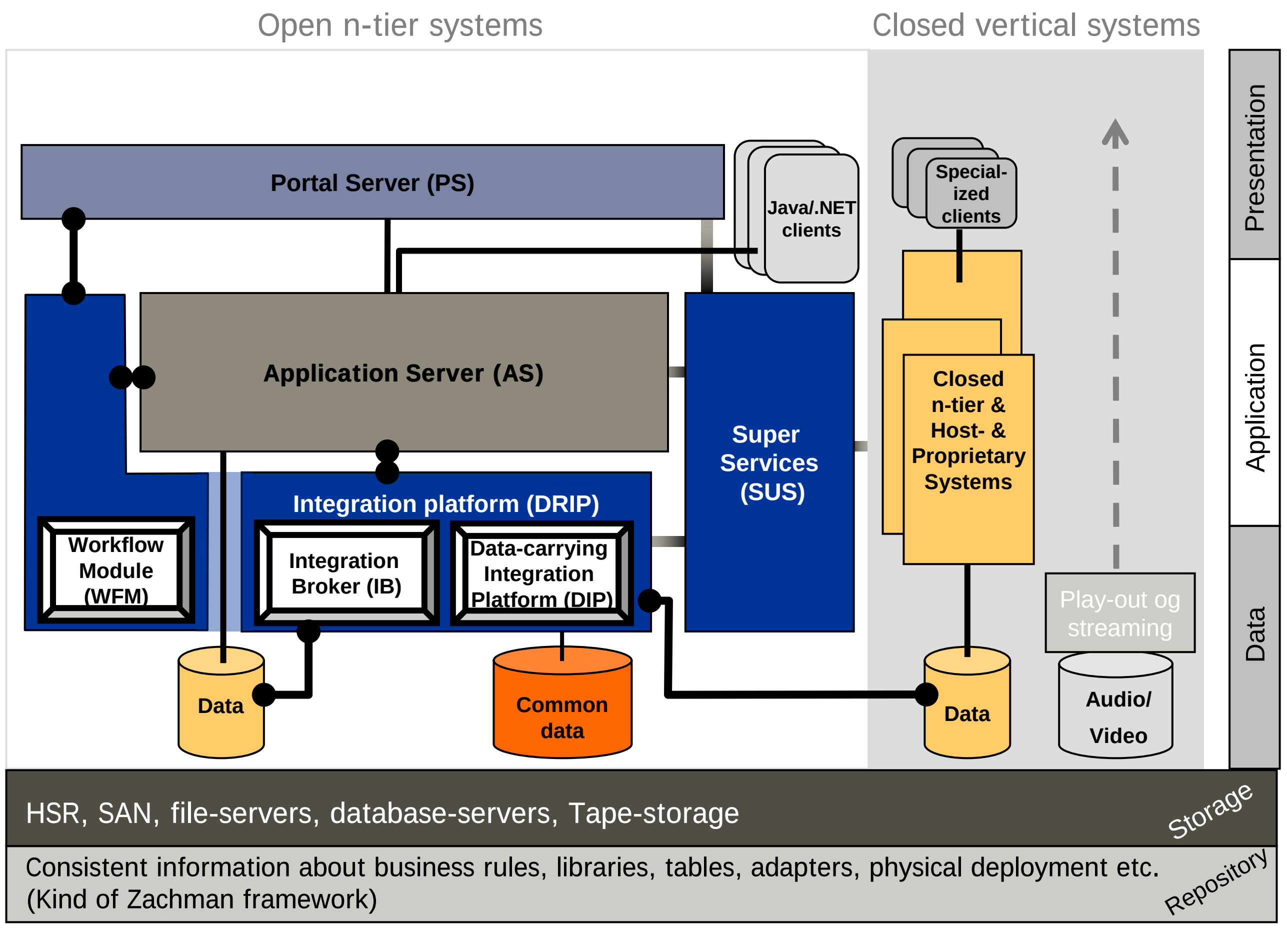
DR- Four major types of integration

- Portal integration
  (Wrapping of user interfaces to the systems)
  - Used where users require a more integrated user dialog or user interface than provided by the separate applications. Where individual customizing of the user interface is required, or where certain functions should be automated, or where single sign-on is needed a integration of the user interface is needed. Web-based user interfaces provide integration by a portal.
- Workflow
  (Mechanism's to coordinate events in systems)
  - Automating or semi-automating workflows, use of functions or data integration. To support tightly connected or integrated business processes it should be possible in advance to define which actions to take place, when certain data are exchanged or updated. Work-flow mechanisms can be found in each of the three former described components.

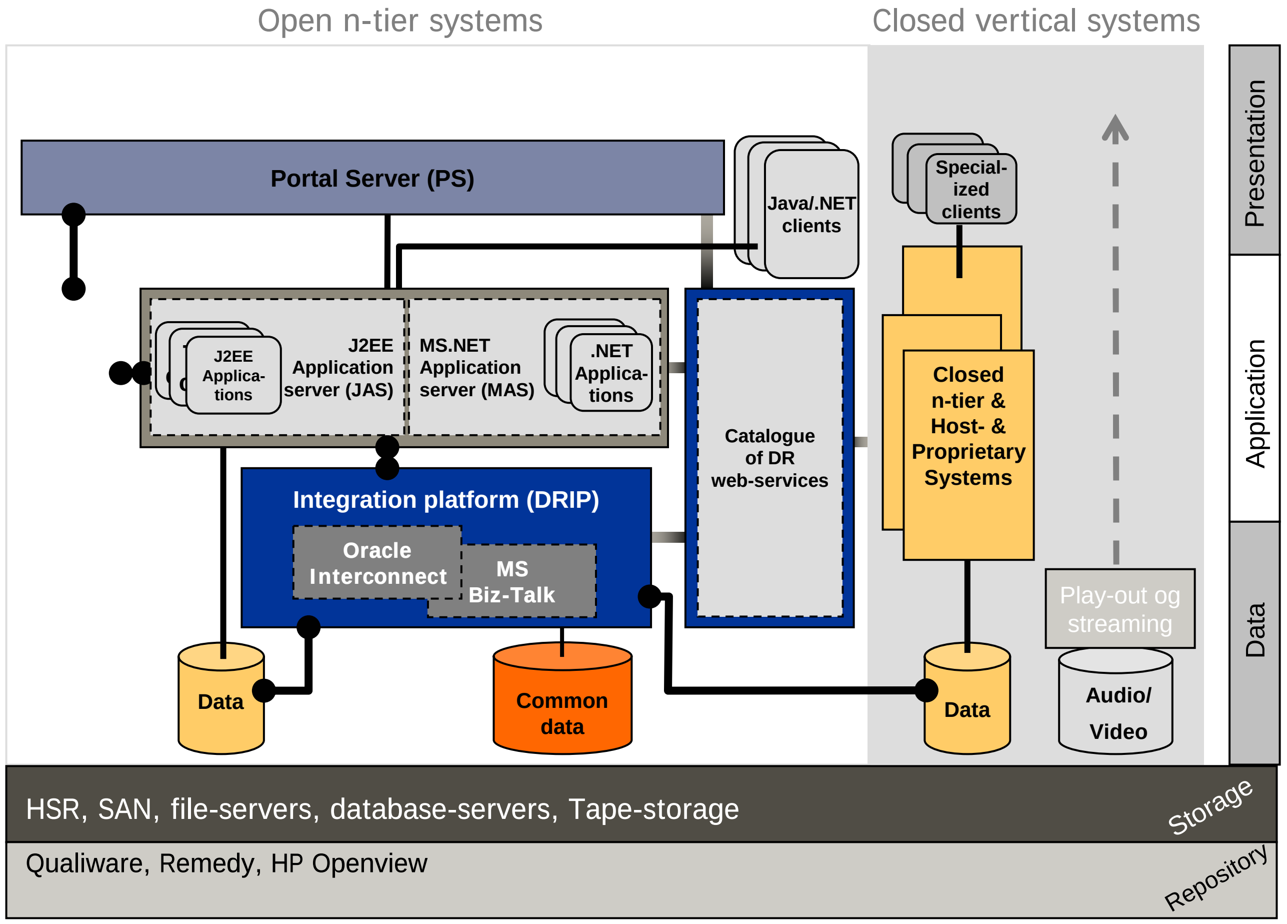
DR- The main tools in our world


Open n-tier systems
Closed vertical systems
Portal Server (PS)
Java/.NET clients
Special-ized clients
Application Server (AS)
?
?
?
Closed n-tier & Host- & Proprietary Systems
Play-out og streaming
Data
Common data
Data
Audio/ Video
Presentation
Application
Data
HSR, SAN, file-servers, database-servers, Tape-storage
Storage
Consistent information about business rules, libraries, tables, adapters, physical deployment etc. (Kind of Zachman framework)
Repository

DR- The main tools in our world


Open n-tier systems
Closed vertical systems
Portal Server (PS)
Java/.NET clients
Special-ized clients
Application Server (AS)
Super Services (SUS)
Closed n-tier & Host- & Proprietary Systems
Integration platform (DRIP)
Workflow Module (WFM)
Integration Broker (IB)
Data-carrying Integration Platform (DIP)
Data
Common data
Data
Play-out og streaming
Audio/ Video
Presentation
Application
Data
HSR, SAN, file-servers, database-servers, Tape-storage
Storage
Consistent information about business rules, libraries, tables, adapters, physical deployment etc. (Kind of Zachman framework)
Repository

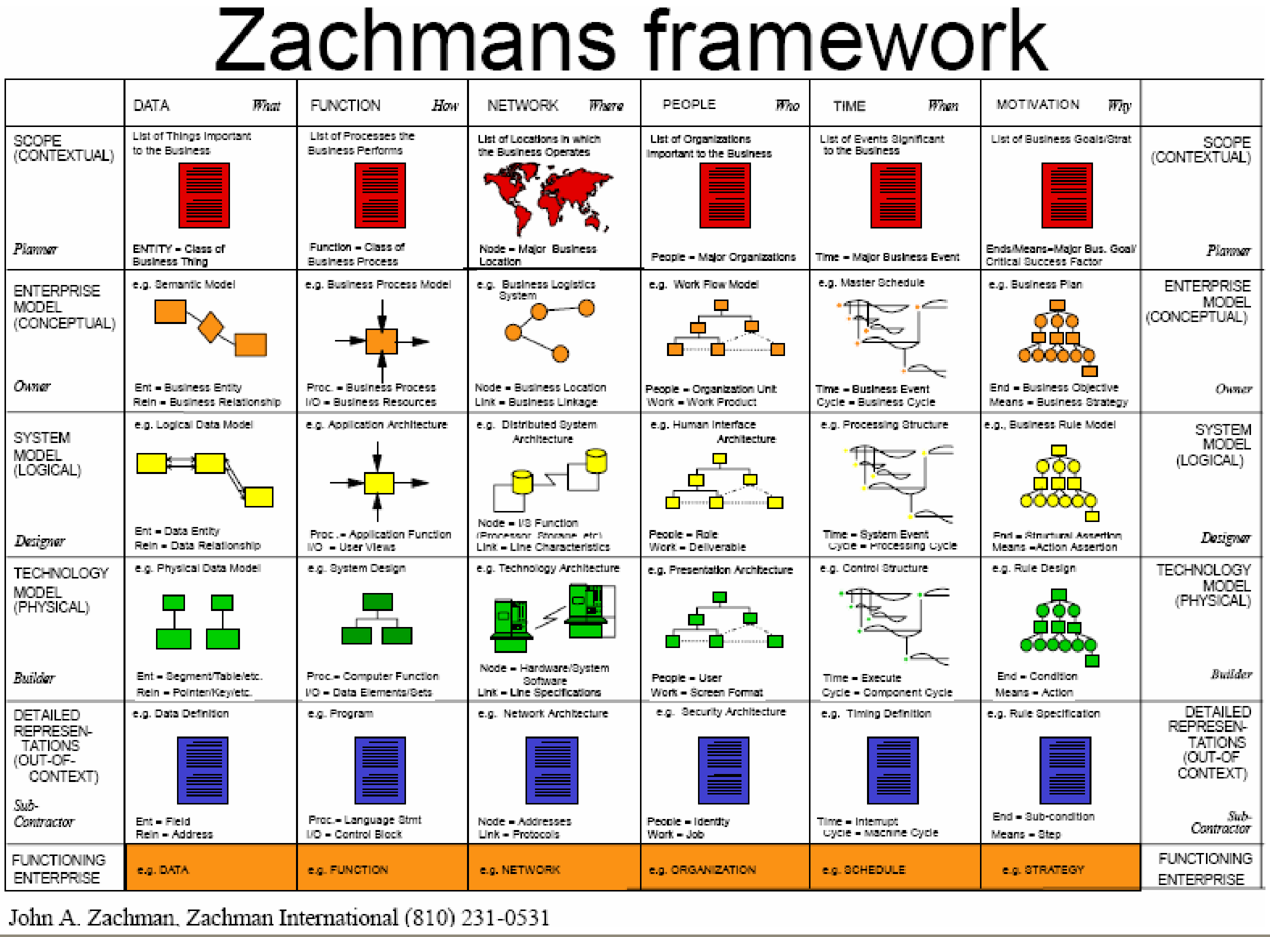
DR- The first tools in our world



DR- Our skills and roles

- System ownership
- Process ownership starting up
- More IT (system) architects
- Metadata manager
- Additional focus in Investment plans and IT-governance
- We are looking at new management roles around Archives - given their different role

DR- sorting out the dimensions



Zachmans framework

	DATA *What*	FUNCTION *How*	NETWORK *Where*	PEOPLE *Who*	TIME *When*	MOTIVATION *Why*	
SCOPE (CONTEXTUAL) *Planner*	List of Things Important to the Business ENTITY = Class of Business Thing	List of Processes the Business Performs Function = Class of Business Process	List of Locations in which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organizations	List of Events Significant to the Business Time = Major Business Event	List of Business Goals/Strat Ends/Means=Major Bus. Goal/ Critical Success Factor	SCOPE (CONTEXTUAL) *Planner*
ENTERPRISE MODEL (CONCEPTUAL) *Owner*	e.g. Semantic Model Ent = Business Entity Reln = Business Relationship	e.g. Business Process Model Proc. = Business Process I/O = Business Resources	e.g. Business Logistics System Node = Business Location Link = Business Linkage	e.g. Work Flow Model People = Organization Unit Work = Work Product	e.g. Master Schedule Time = Business Event Cycle = Business Cycle	e.g. Business Plan End = Business Objective Means = Business Strategy	ENTERPRISE MODEL (CONCEPTUAL) *Owner*
SYSTEM MODEL (LOGICAL) *Designer*	e.g. Logical Data Model Ent = Data Entity Reln = Data Relationship	e.g. Application Architecture Proc .= Application Function I/O = User Views	e.g. Distributed System Architecture Node = I/S Function (Processor, Storage, etc) Link = Line Characteristics	e.g. Human Interface Architecture People = Role Work = Deliverable	e.g. Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means =Action Assertion	SYSTEM MODEL (LOGICAL) *Designer*
TECHNOLOGY MODEL (PHYSICAL) *Builder*	e.g. Physical Data Model Ent = Segment/Table/etc. Reln = Pointer/Key/etc.	e.g. System Design Proc.= Computer Function I/O = Data Elements/Sets	e.g. Technology Architecture Node = Hardware/System Software Link = Line Specifications	e.g. Presentation Architecture People = User Work = Screen Format	e.g. Control Structure Time = Execute Cycle = Component Cycle	e.g. Rule Design End = Condition Means = Action	TECHNOLOGY MODEL (PHYSICAL) *Builder*
DETAILED REPRESEN- TATIONS (OUT-OF- CONTEXT) *Sub- Contractor*	e.g. Data Definition Ent = Field Reln = Address	e.g. Program Proc.= Language Stmt I/O = Control Block	e.g. Network Architecture Node = Addresses Link = Protocols	e.g. Security Architecture People = Identity Work = Job	e.g. Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g. Rule Specification End = Sub-condition Means = Step	DETAILED REPRESEN- TATIONS (OUT-OF CONTEXT) *Sub- Contractor*
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANIZATION	e.g. SCHEDULE	e.g. STRATEGY	FUNCTIONING ENTERPRISE

John A. Zachman, Zachman International (810) 231-0531

DR- have we reached Nirvana yet?

Done

- Integration platform acquired and basic setup and test done
- In use in daily operation (broadcast as well as ERP systems)
- Re-use of connections in progress (more and more systems gets benefits)
- Integrations are more widely standard in requirements and estimations of projects. (News & TV production, NCS, Play out, etc.)
- DR Metadata model launched and implemented

In progress

- Common description of all major processes and workflows
- Repository and change management system under implementation
- Project and process owners respect the decisions here and now, and are looking more at the long term benefits
- Investment planning cope with compensation to first projects in line
- Middle layer management understand and respect the need for transition to a new development paradigm

To come

- Internalize the goals, standards and methods in each developer
- Firm change managements routines for developing and maintaining integrations

DR- have our suppliers ?

Done

- Respect the need for establish some kind of neutral area at the borders of their systems
  - Here exchange of data in our way and format takes place
  - Some minor functionality (components) are integrated in their applications
  - Acting as a broker-to-broker bridge between complex collections of applications

In progress

- Implementing DR-metadata model
- Gradually realizing the need for standardization of integration in the broadcast domain
- Implementing standardized integrations from traditionally ERP- over planning- to production- or play out systems

To come

- Integrating functionality as much as pure data
- Fully interchangeable components collected as a system (may the best component win)

Conclusion: This is the way to go

- We definitely know, design and describe our operations and our technology better by now
- We got far better system integration – but still have a long way to go
- We succeed in laying a hard pressure on the vendors to accept, meet and solve our need for widely implemented integration
- This will definitely take time in our organization: 4-6 years from start in 2001 – and irreversible