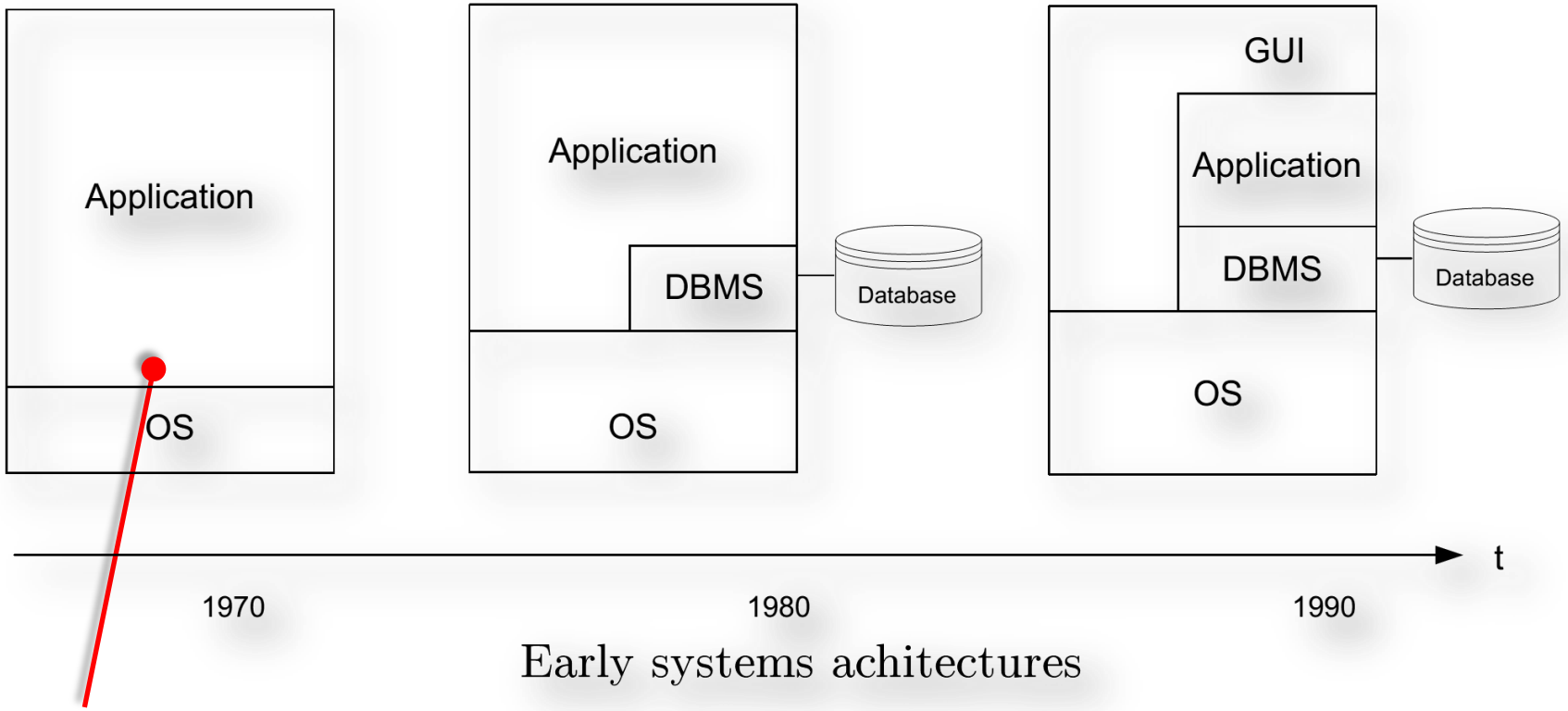
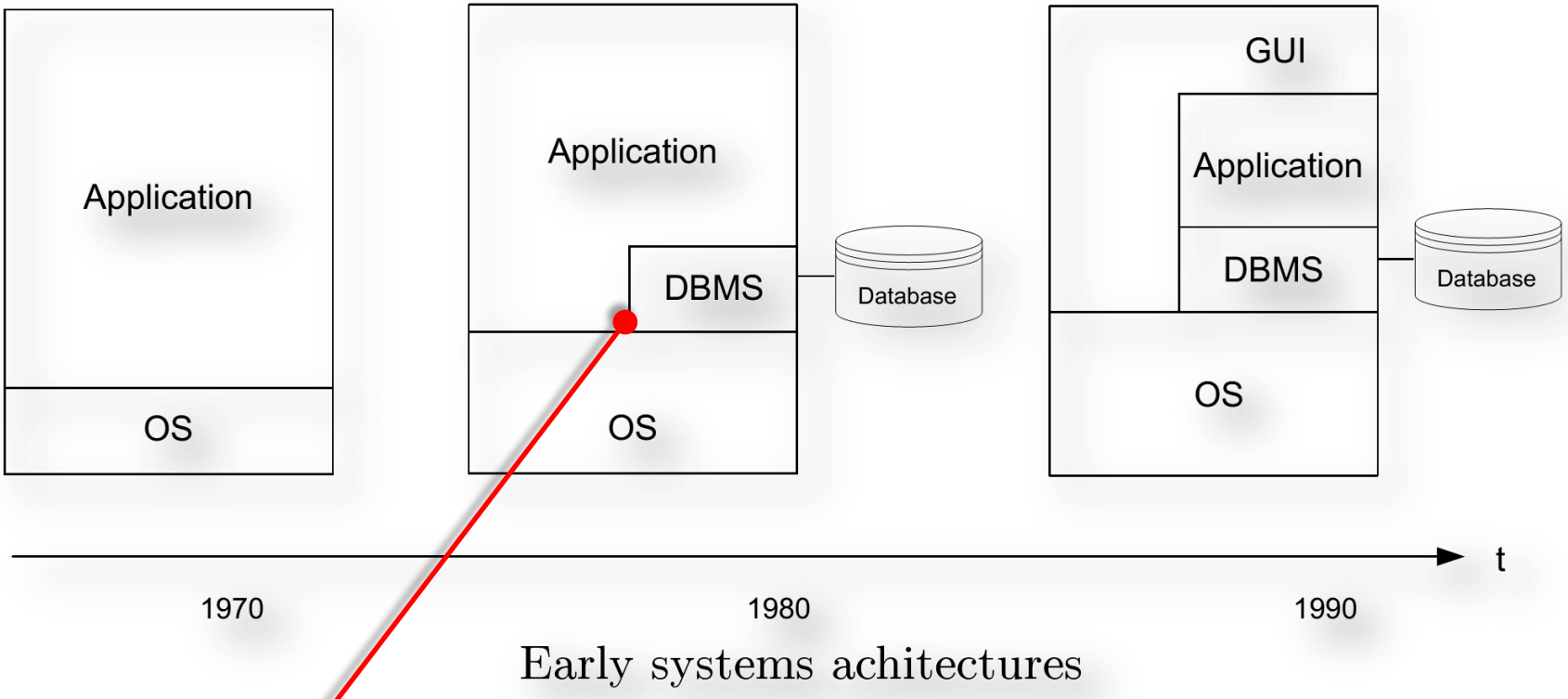


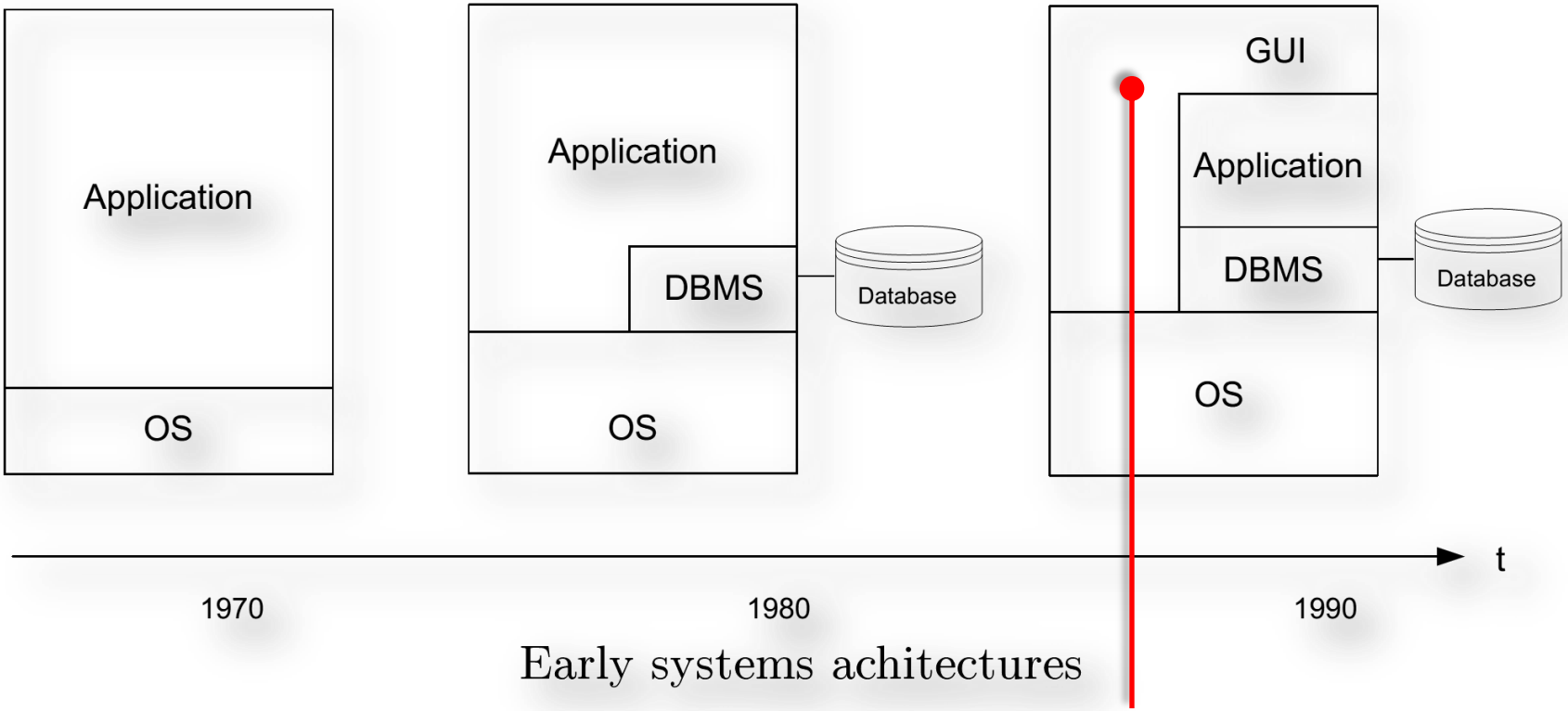
Early systems achitectures

- ✓ 1st era: (early years of computing): applications developed over Operating System (OS) subroutines, coding basic functionality such as access to hardware (storage and memory management). Application porting to a new computer results in almost complete redevelopment.
- ✓ OSs start to provide programming interfaces. Applications can be implemented more efficiently, and a change in the hardware does not impact their lifecycle.

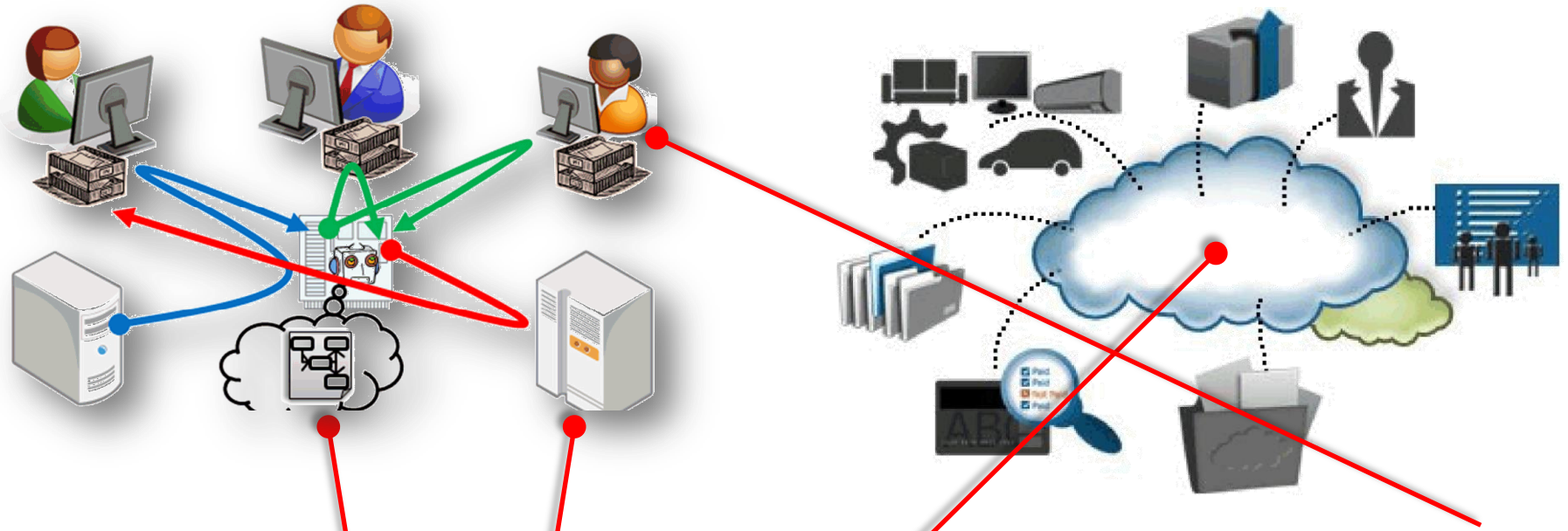


Early systems achitectures

- ✓ 2nd era: to reduce the effort on storage and retrieval of data, and data inconsistency issues due to multiple applications access. With Data Base Management Systems (DBMS) the physical structure of data can be modified without affecting applications.
- ✓ Declarative query languages (SQL), transaction, recovery and security capabilities are managed at the DBMS level. Moreover, new OS functionalities are offered to applications.



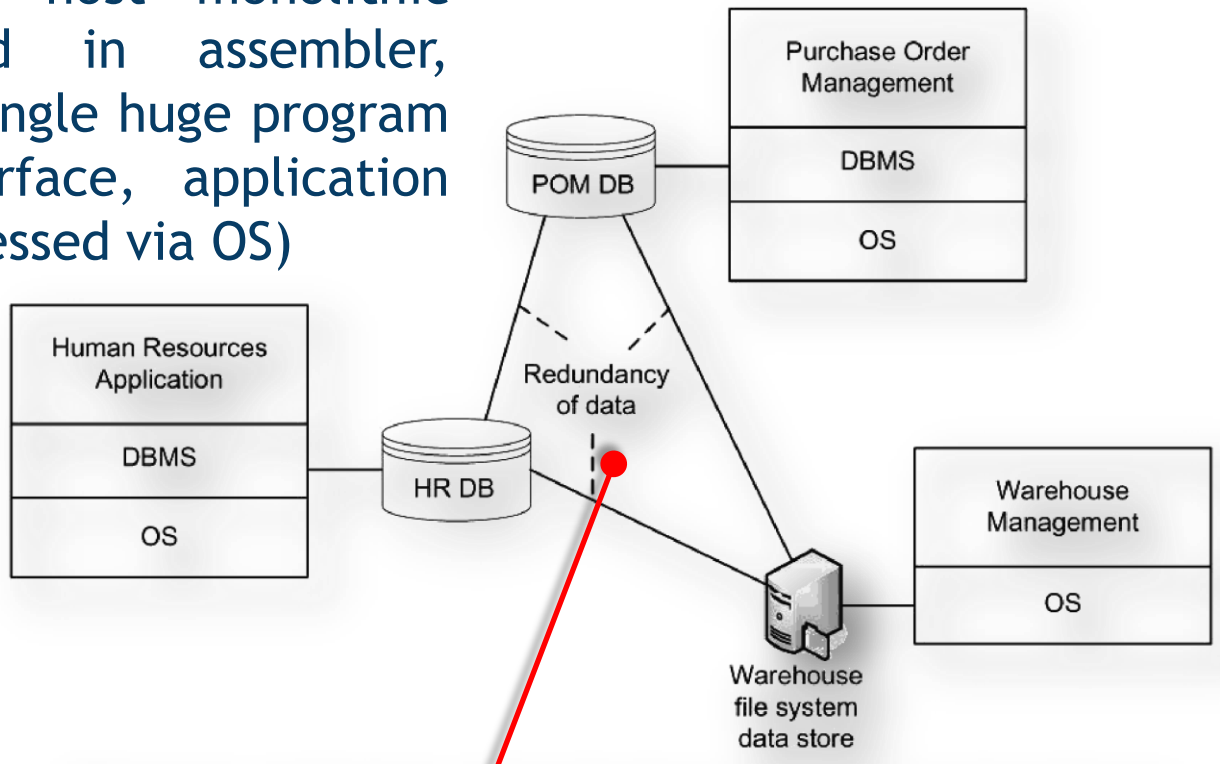
- ✓ 3rd era: to ease human interaction with applications, to avoid extensive user training and specialized employees. Graphical User Interfaces (GUI) separates the interaction with the application from its underlying business logic.
- ✓ Usability is a user-centered paradigm providing applications more efficient to use, easier to learn, and more satisfying to use.



- ✓ A computer Information System (IS) is a system, composed of people and computers systems, that processes or interprets information.
- ✓ An Enterprise Information System (EIS) is an IS which supports enterprise business processes.
- ✓ A business process is a collection of related, structured activities/tasks producing a specific service/product for a particular kind of customer (customer-centric perspective).

- 1st era: mainframes host monolithic applications, developed in assembler, managing all tasks in a single huge program with textual user interface, application logic, and data files (accessed via OS)

- With the advent of DBMS and GUI, lowering cost of computers, it is typical for an enterprise to have different applications: for HR, PO, and for Production Planning; each with its own DBMS.



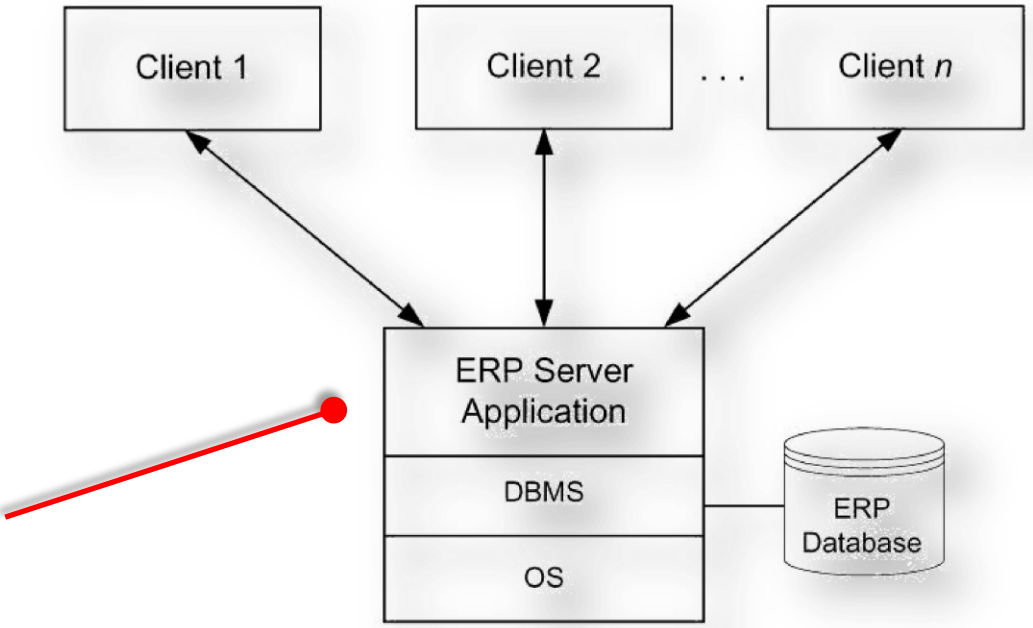
Enterprise applications with redundant data and data dependencies

- In large enterprises with different departments, different application systems are used for the same issue.

- Dependencies between data stored in multiple systems are represented by identifiers (e.g. contract id, employee id). However, changes (e.g. a customer address) are hard to propagate without inconsistency.

- In this context the first Enterprise Resource Planning systems (**ERP**) are developed: to host disparate enterprise applications over an centralized DBMS.

- An ERP is accessed by client applications, which access an application server issuing requests to a DBMS.



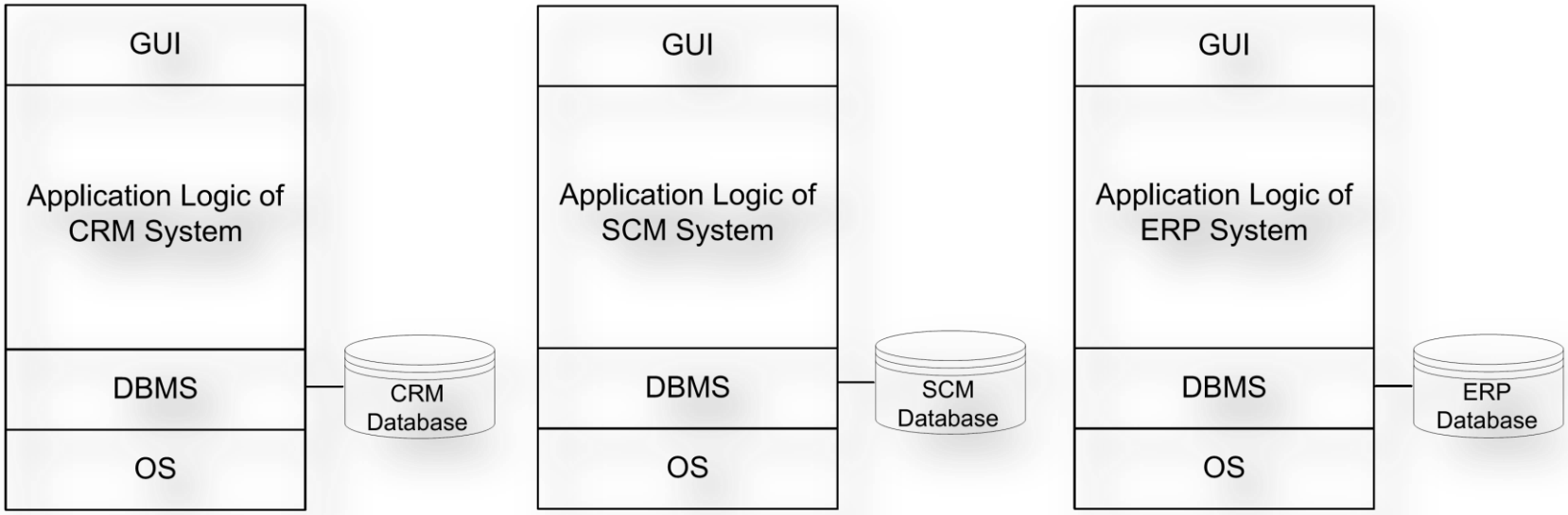
Two-tier client-server architecture

- With the growth of enterprises and new market requirements, driven by new customer needs around the year 2000, new software systems enter in the market:

- Supply Chain Management (**SCM**) systems, Customer Relationship Management (**CRM**) systems, with the purpose of supporting the planning, operation, and control of supply chains, including inventory management, warehouse management, management of suppliers (and distributors) relationships (**SRM**), and demand planning.

- New types of ISs enter the market, often developed by different vendors, hosting their own DBMS. System architects face again the problem of heterogeneous enterprise applications: for instance, call centers are not able to know the complete status of the customer.

- This unsatisfactory situation is called "siloed applications": while application systems can be physically connected by a local network, they are not logically integrated; manual integration made by the user consumes considerable resources and is error-prone.



Siloed enterprise applications

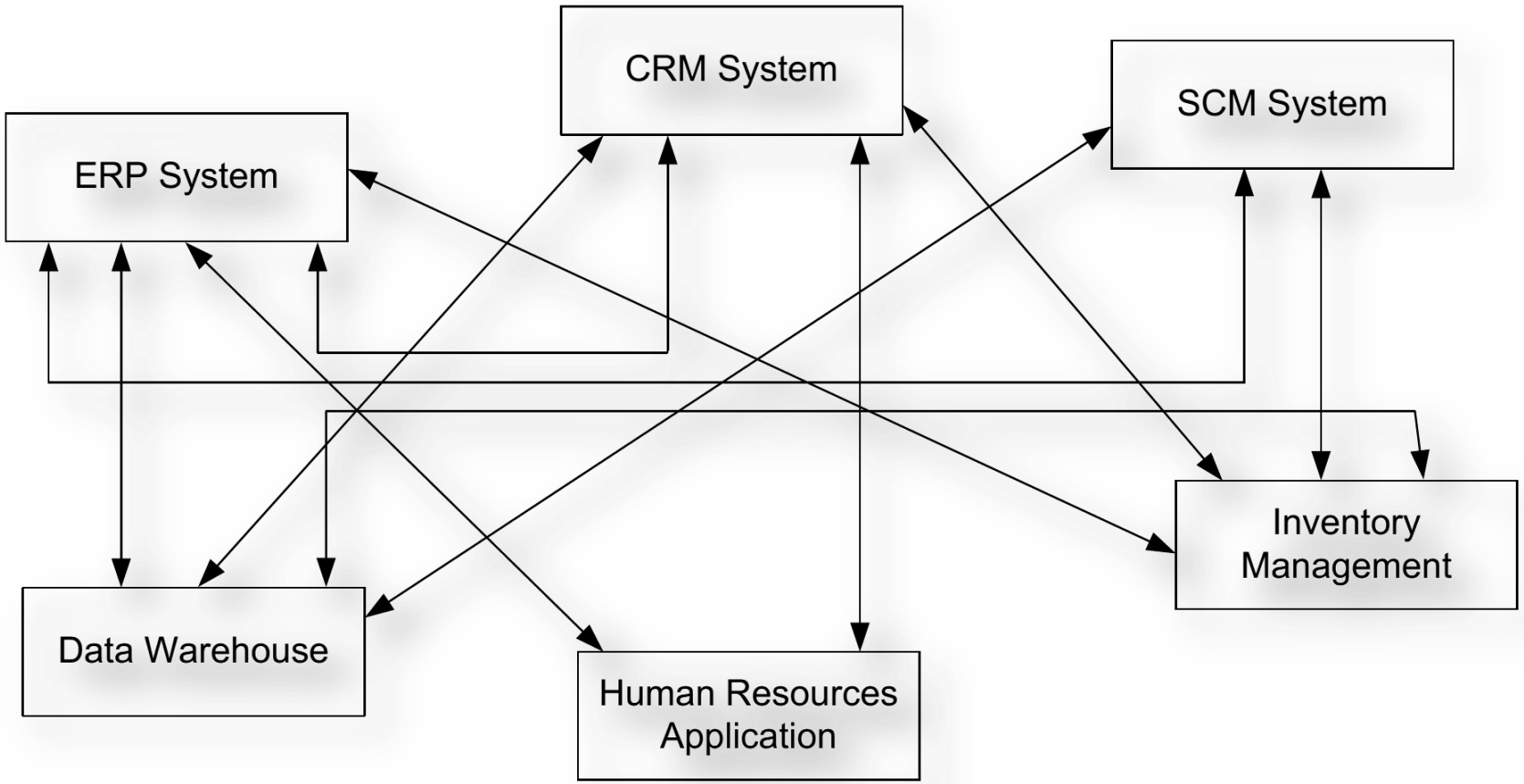
- Unfortunately, due to the large complexity of the systems at hand, the same approach used with ERPs, i.e., to re-implement systems functionality in an integrated way, is not feasible in the new context.

- This leads to new middleware systems: **EAI** (Enterprise Application Integration) systems; in EAI, a system performs certain steps and transfers control to another system, which takes results and continues operation.

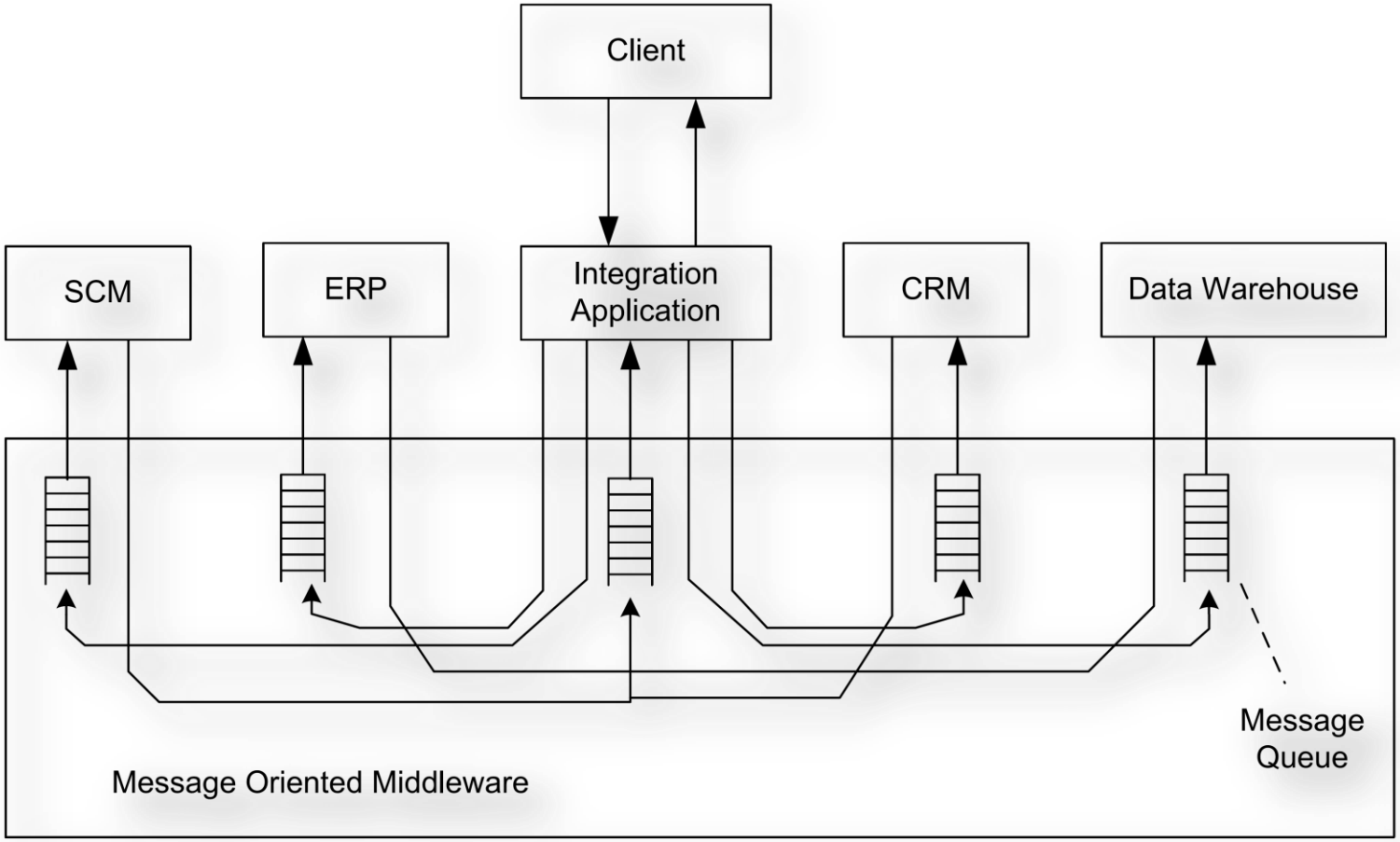
- EAI technology can be used to cope with syntactic and semantic differences between data (data integration): e.g. the customer address is represented in one system by the attribute "CAddr" and in the other system by the attribute "StreetAdrC"; e.g. in one system the attribute "Price" includes value-added tax, in the other system it is excluded.

- In enterprise computing changes are abundant, and a system architecture should support changes in an efficient and effective manner.

- Early inventions in the EAI architecture are: (i) message oriented middleware; (i) application adapters; (iii) message broker with declarative rules.

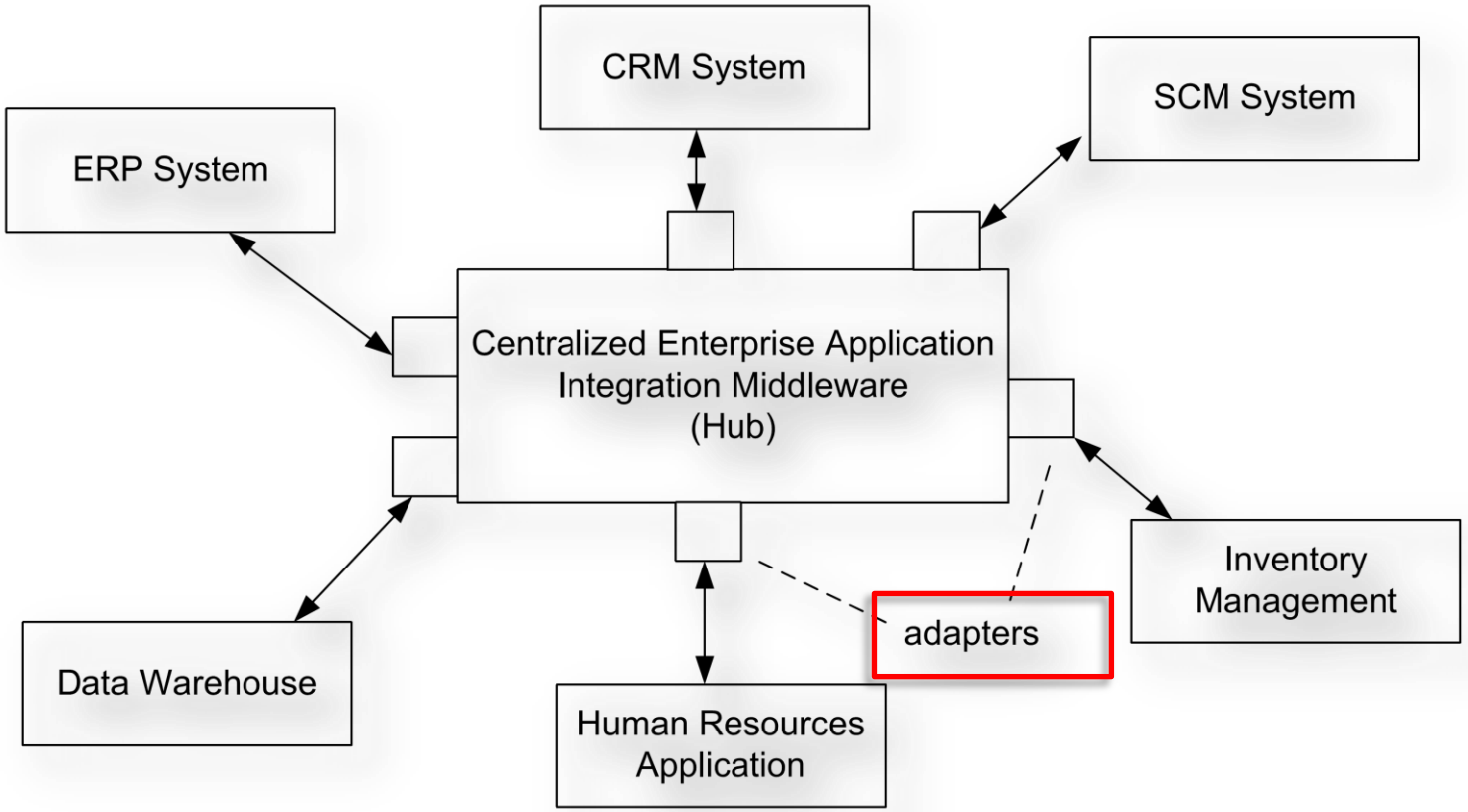


Early enterprise application integration: hard-wiring of application systems results in $N \times N$ problem



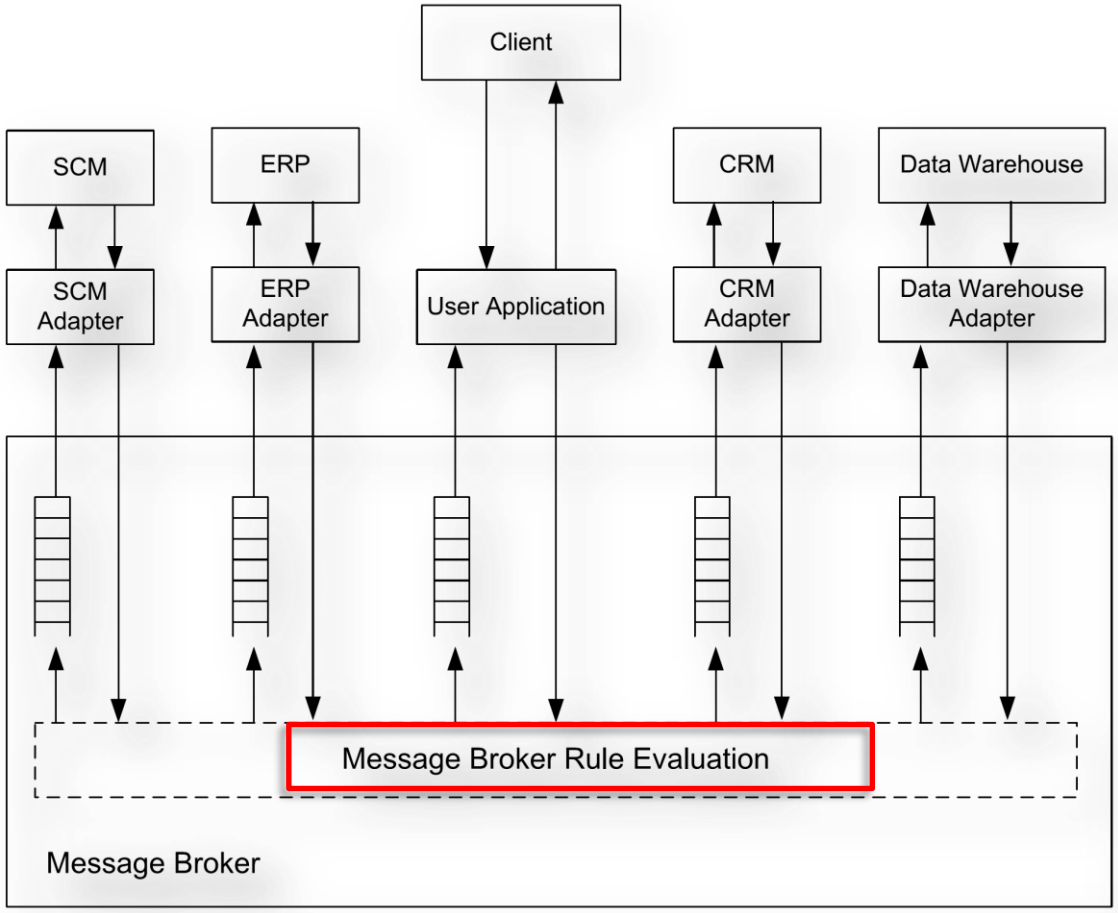
Message-oriented middleware for reliable communication between applications. Senders of messages encode receivers, and process logic is encoded in applications

- Guarantee of message delivery. But the problem of point-to-point communication still exists (response to change is not improved).



Hub-and-spoke enterprise application integration architecture

- The sender of a message does not encode the receiver, since the message structure and content is used to automatically detect the receiver or receivers of a message (content-based routing). No *N x N* connections. Each application requires the development of a dedicated adapter.
- Message brokers are used to define rules for communication between applications, in a declarative way, in the central hub. Applications can link to message brokers via publish/subscribe mechanism: applications can subscribe to certain types of messages and can publish messages.



Message broker with declarative rules that de-couples senders from receivers and eases response to change

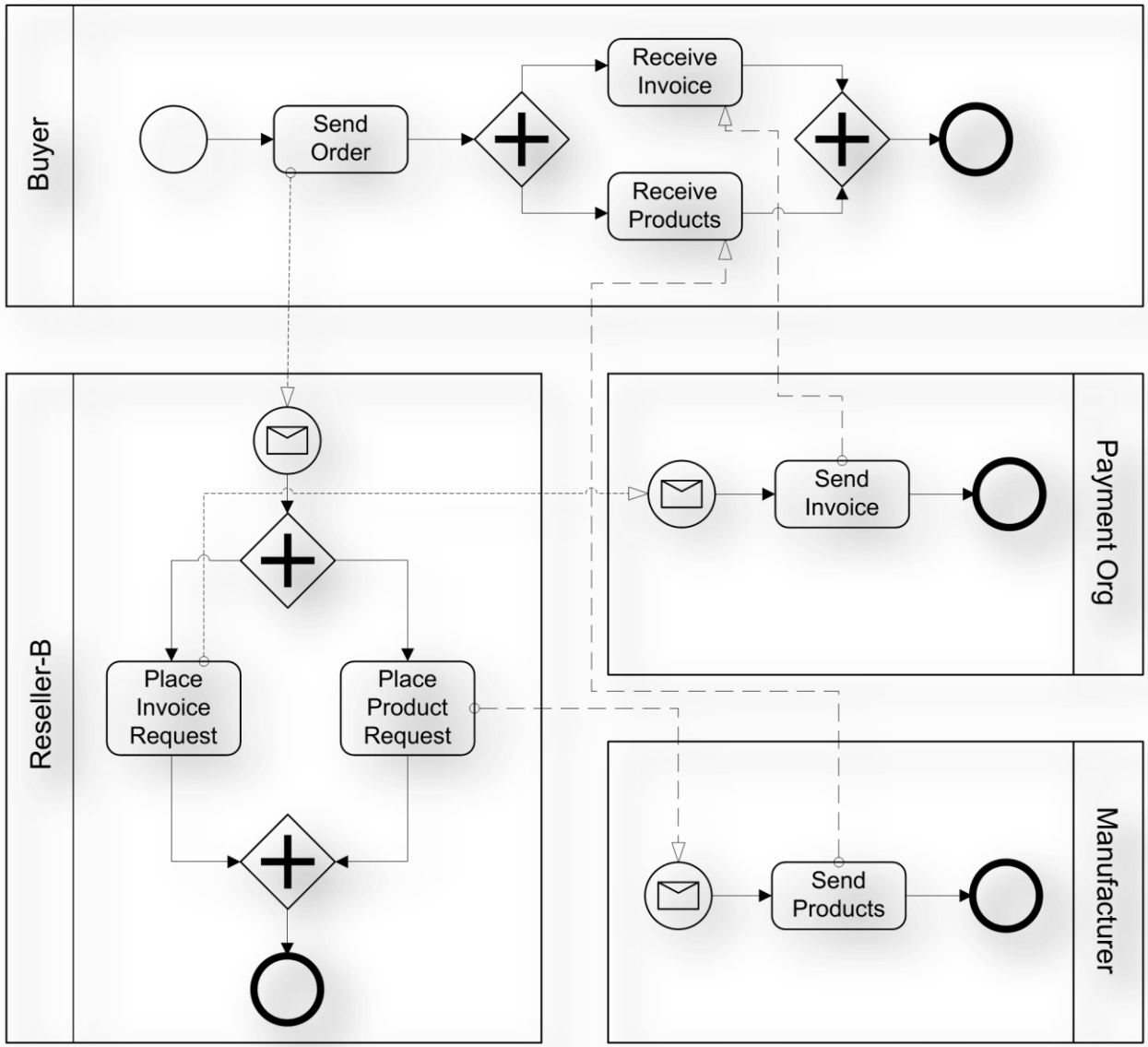
- This approach requires a global data model hosted by the message broker that all applications have agreed upon. This requires programming and low-level configuration of adapters.
- Data integration is typically performed using data mapping tools allowing the mapping of data structures of the application to data structures of the message broker.

- In typical enterprise application integration scenarios, the functionality of the integrated applications is organized by **a sequence or partial order of steps, realizing a process**. This process consists of activities that are executed, under business constraints, to achieve an overall business goal.

- While in enterprise application integration discussed so far these process structures are buried in rules hosted by the message broker, an explicit representation of processes is more appropriate. **Workflow management** is the fundamental invention in the evolution of information systems.

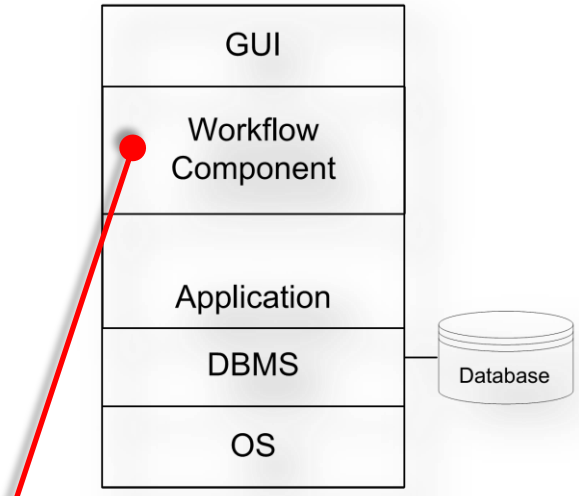
- In parallel, another factor emerges from business administration rather than from software technology: **process orientation (PO)**

- PO is based on a critical analysis of **Taylorism** (small-grained activities conducted by highly specialized personnel), which was good until 80s, when products were typically assembled in a few steps of a simple nature ($\Rightarrow$ transfer of work between companies does not introduce delays, no information on previous steps is required)
- In modern business organizations, that mainly process information, the steps during a business process are often related to each other, context information on the complete case is required during the process, and the transfer of work between companies causes a major problem.



Example of business-to-business collaboration through interacting business processes

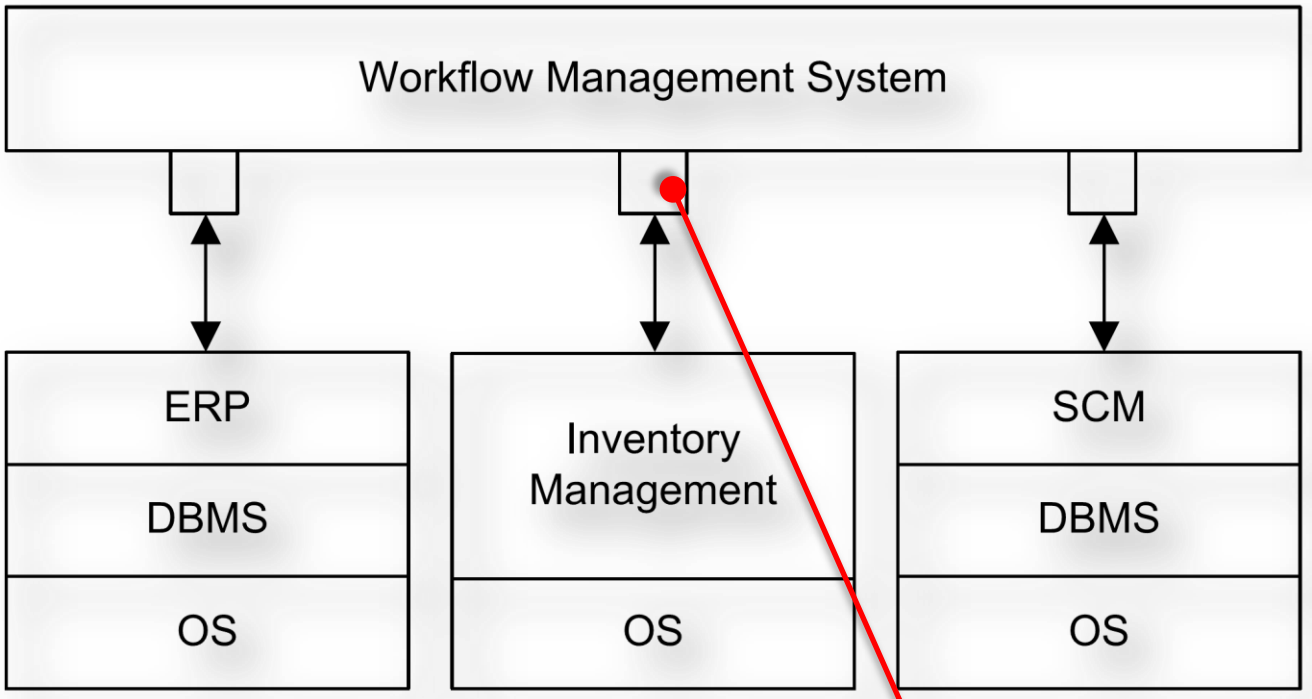
- The important achievement of workflow management is the explicit representation of process structures in process models and the controlled enactment of business processes according to these models.
- The **model-driven approach** facilitates a high degree of flexibility, because process models can be adapted to fulfill new requirements, and the modified process models can immediately be used to enact business processes.
- A **workflow management system (WfMS)** is a software system that defines, creates, and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants, and, where required, invoke the use of IT tools and applications.



Single-application workflow systems achitecture

- Today, most enterprise application systems, such as ERP, embed a workflow engine (called workflow component) to facilitate the flexible customization of business processes within these systems.

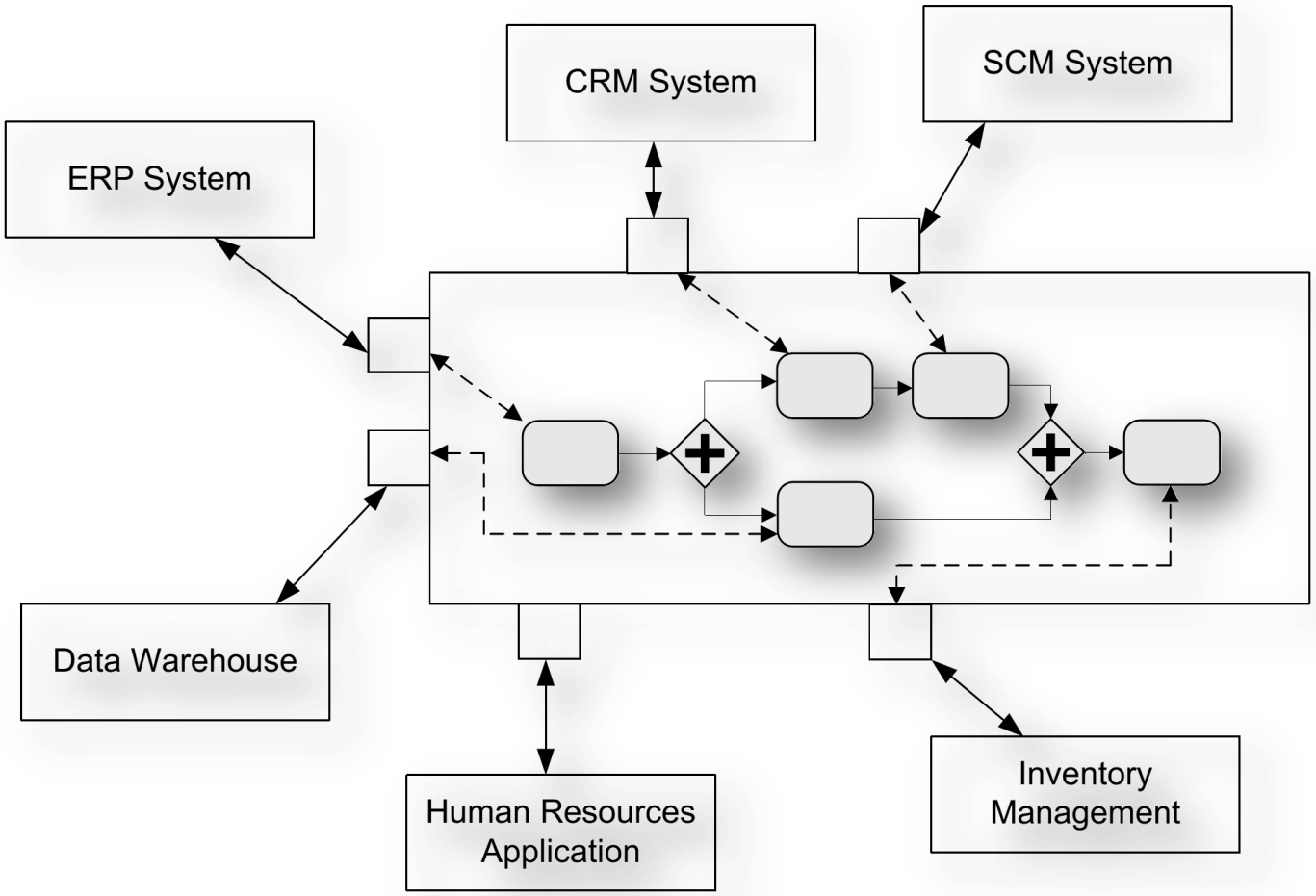
• In the case of multiple-application workflows, a dedicated workflow management system makes sure that the application systems are invoked as specified in the process model.



Multiple-application workflow systems architecture

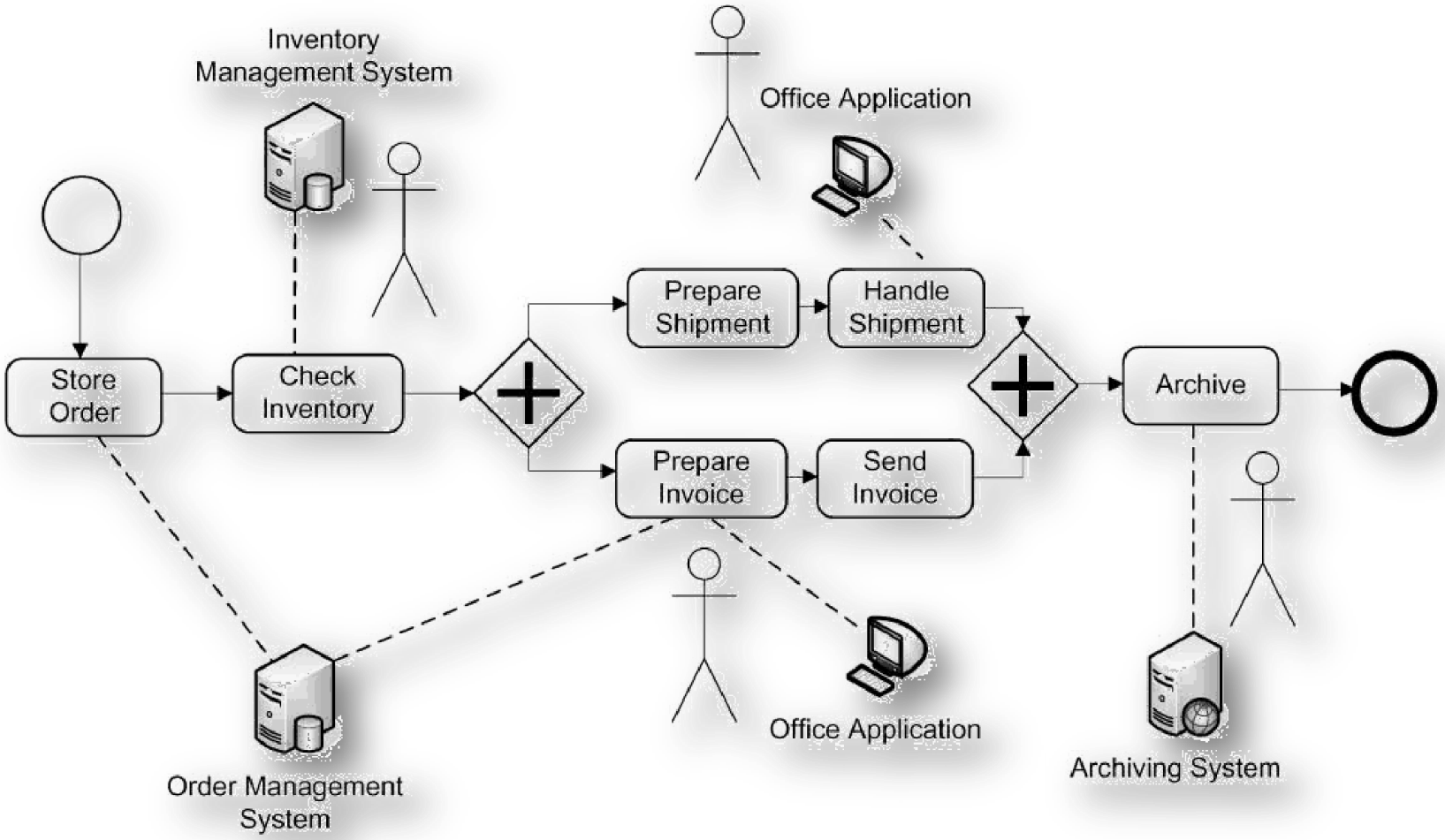
• In addition, data transfer between application systems is also handled by the workflow management system, by using adapters.

- 1st type of Wf: Systems Workflow, which consists of activities implemented entirely by software systems without any user involvement



System workflow integration scenario; a process model defines if and when enterprise applications are invoked

- 2nd type of Wf: human interaction workflow, in which humans are actively involved and interact with information systems.

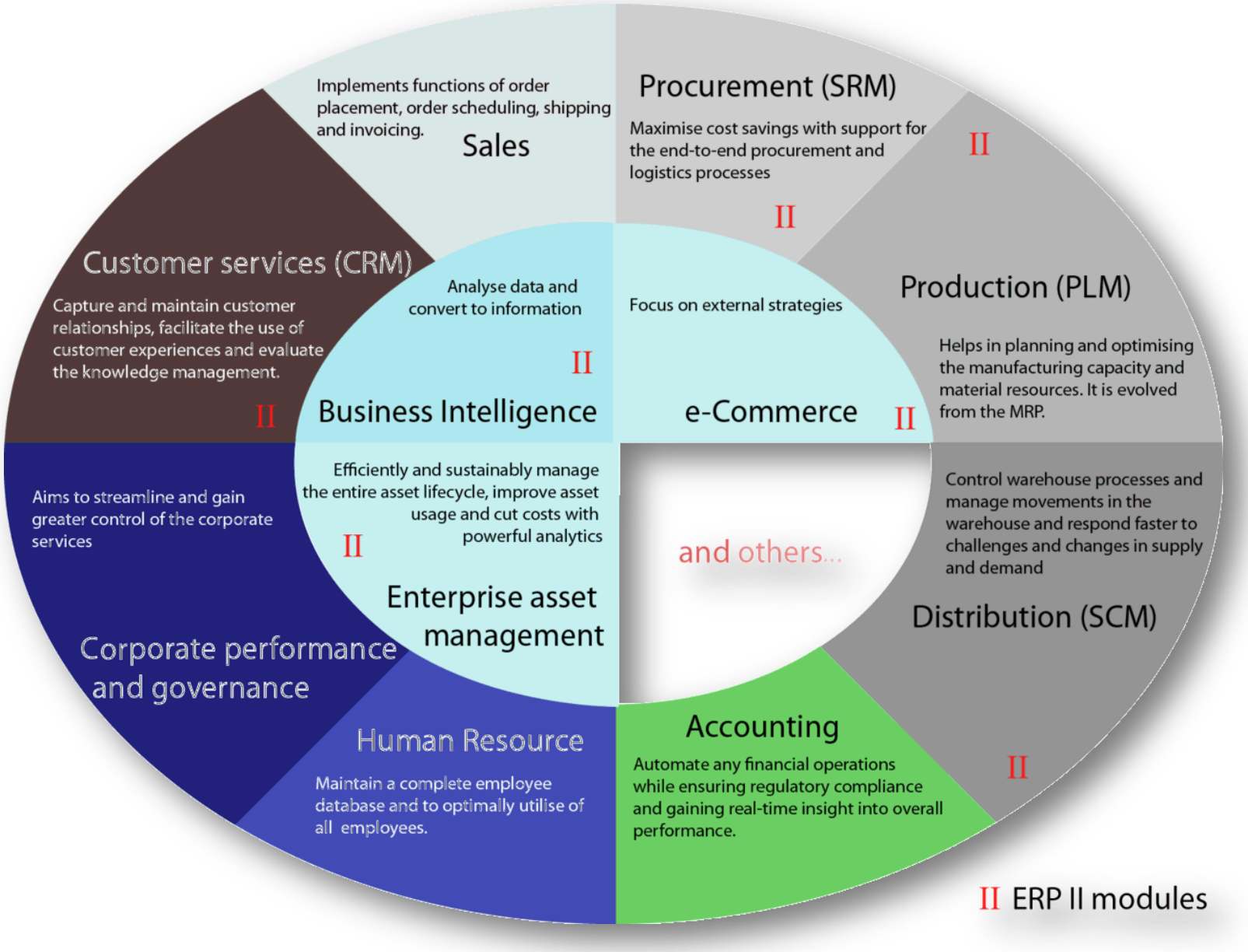


Sample human interaction workflow

- Not all environments ask for a WfMS. In cases where no changes to the process structure are envisioned, a coding of the process flow can be an attractive and adequate choice: e.g. store procedures in database administration, print workflow in publishing environments.

- Business processes are also realized in online shops, such as train reservation systems, where steps of an interaction process are graphically represented to guide the user interaction. Since this type of interaction process can be realized using Web page design, a WfMS is not required.

- ERPs realize literally thousands of BPs, which can be customized to fit particular needs. In most cases, the BPs are realized within the system, without integration issues. If the predefined BPs cannot be tailored to fits the needs, integrated process modeling can be used for new processes.

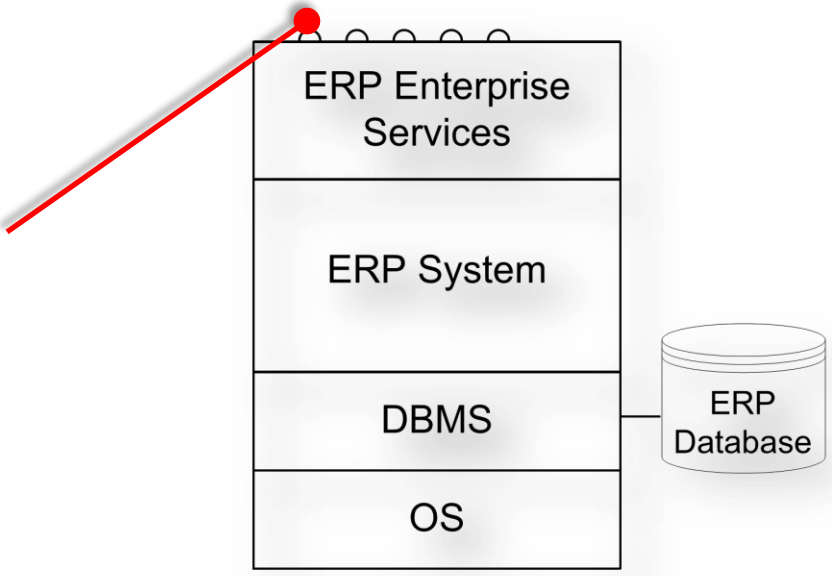


- Main modules in a second generation ERP

- One of the major trend both in business engineering and software technology is **Service-Oriented Architecture (SOA)** implemented by **Web Services (WS)**.

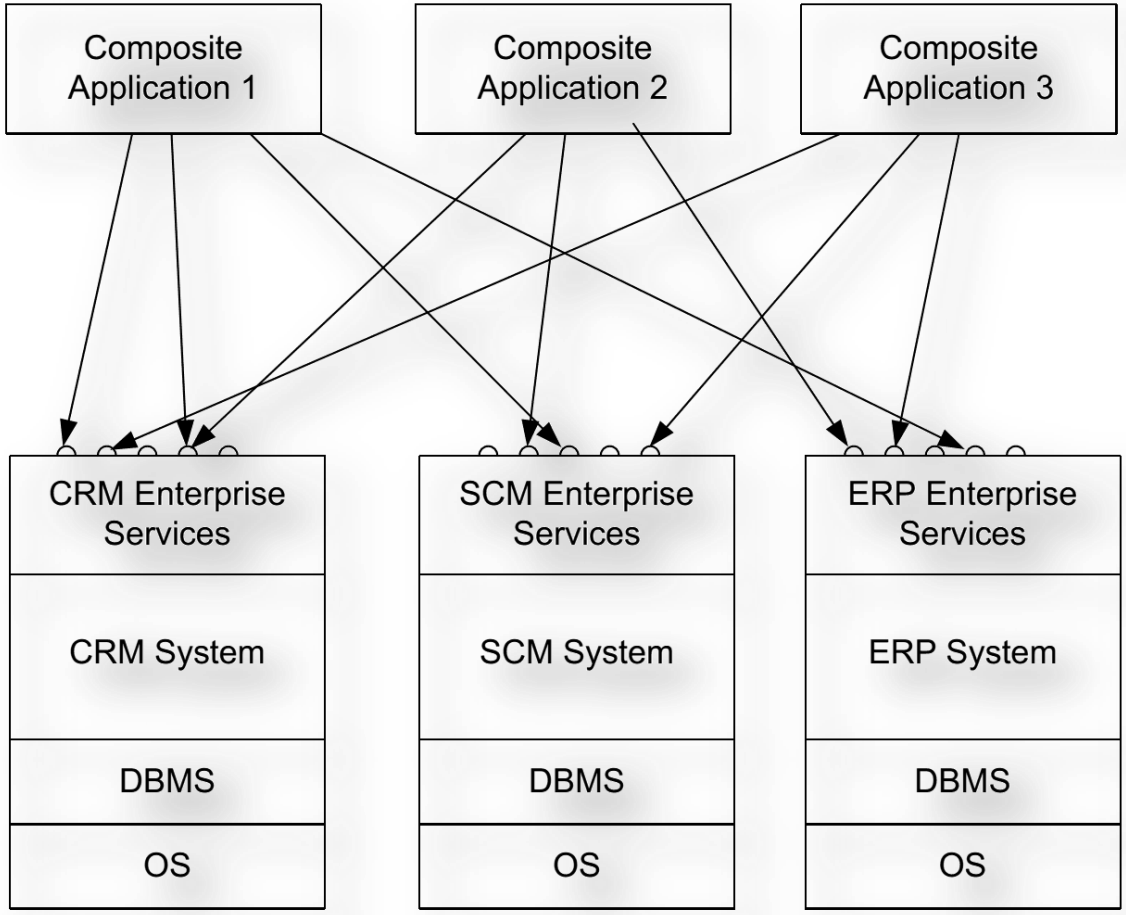
- A WS is a software whose operations are provided, in a platform independent format (XML), by a host to any another host of the WWW. In a WS, web technology such as the HTTP protocol, originally designed for human-to-machine communication, is utilized for machine-to-machine communication, to invoke software operations and transfer machine readable data (XML).

- The functionalities of an enterprise application system can be provided through services (depicted by semicircles in figure) via XML-based standardized interfaces.
Thus, complex applications can be dynamically built on top of existing functionalities.



Service-enabled application system

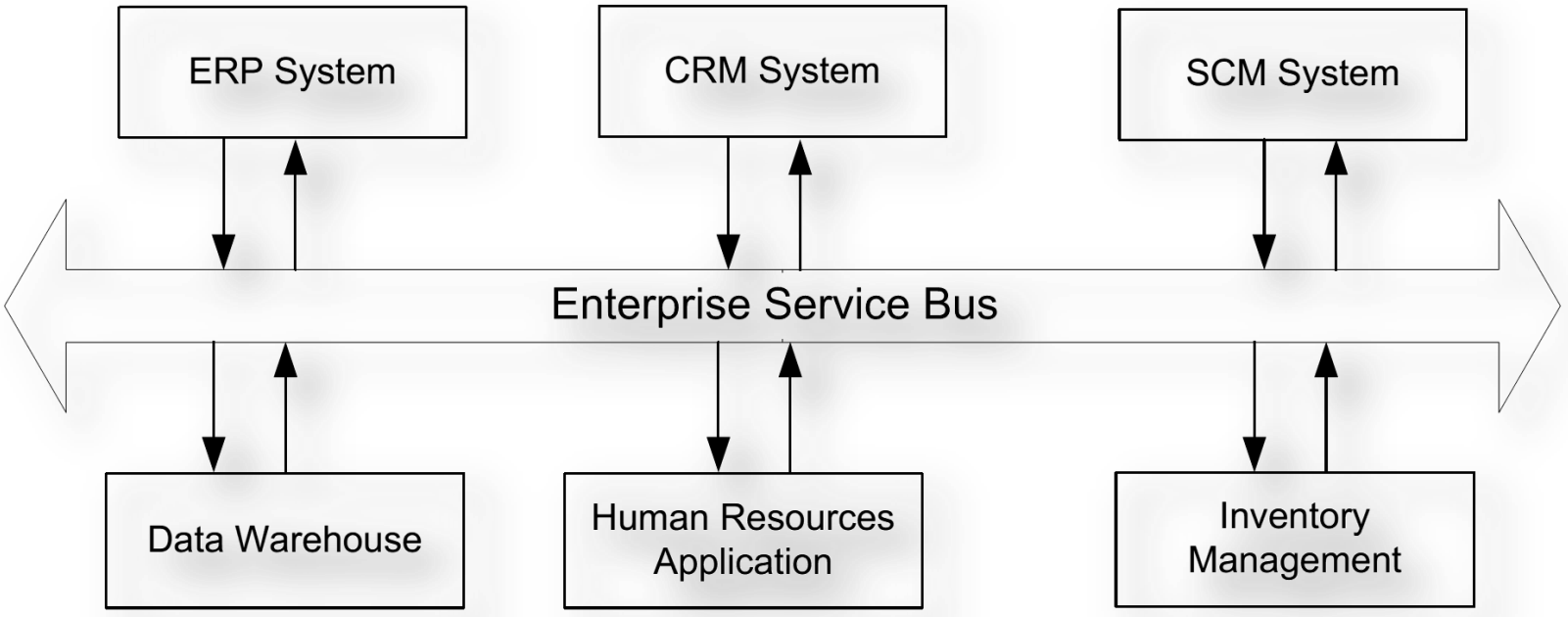
- **Composite applications** invoke enterprise services that provide the functionality of the underlying back-end systems. User interaction is realized by dedicated graphical user interfaces that sit on top of composite applications.



Enterprise systems expose functionality through enterprise services

- Modern EAI middleware provides Web Services interfaces to the enterprise applications.

- The term **Enterprise Service Bus (ESB)** means that each enterprise application is attached to the bus, which acts as an application independent integration middleware.

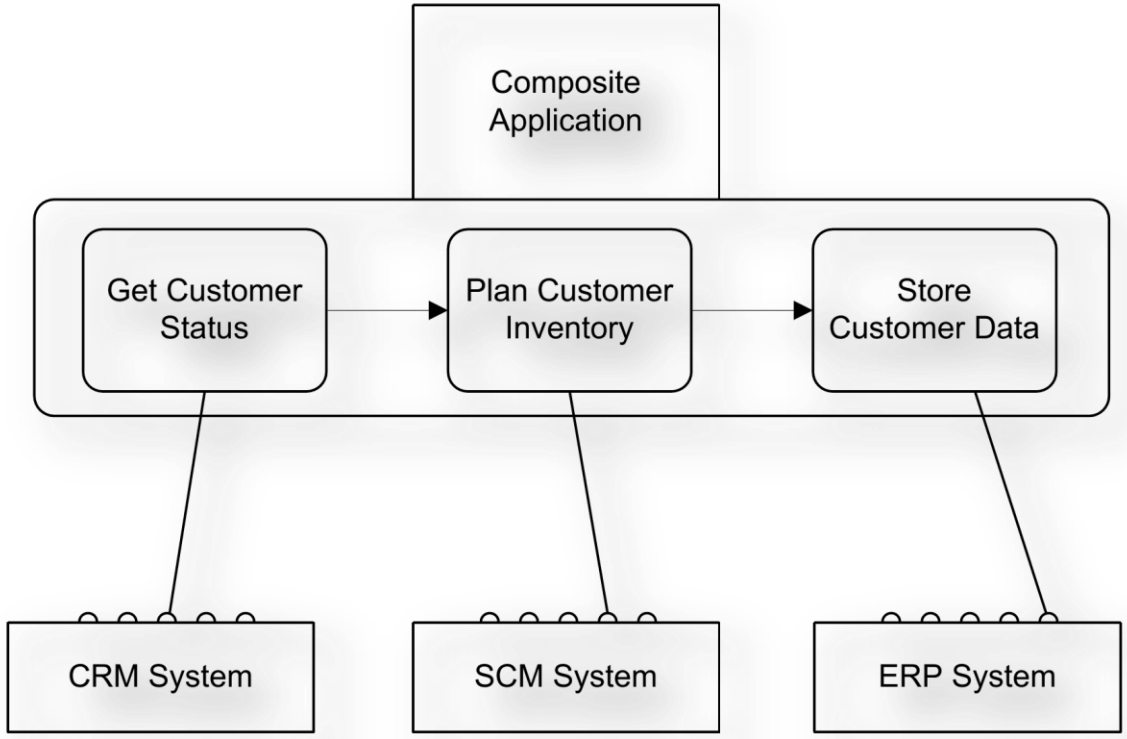


Enterprise service bus

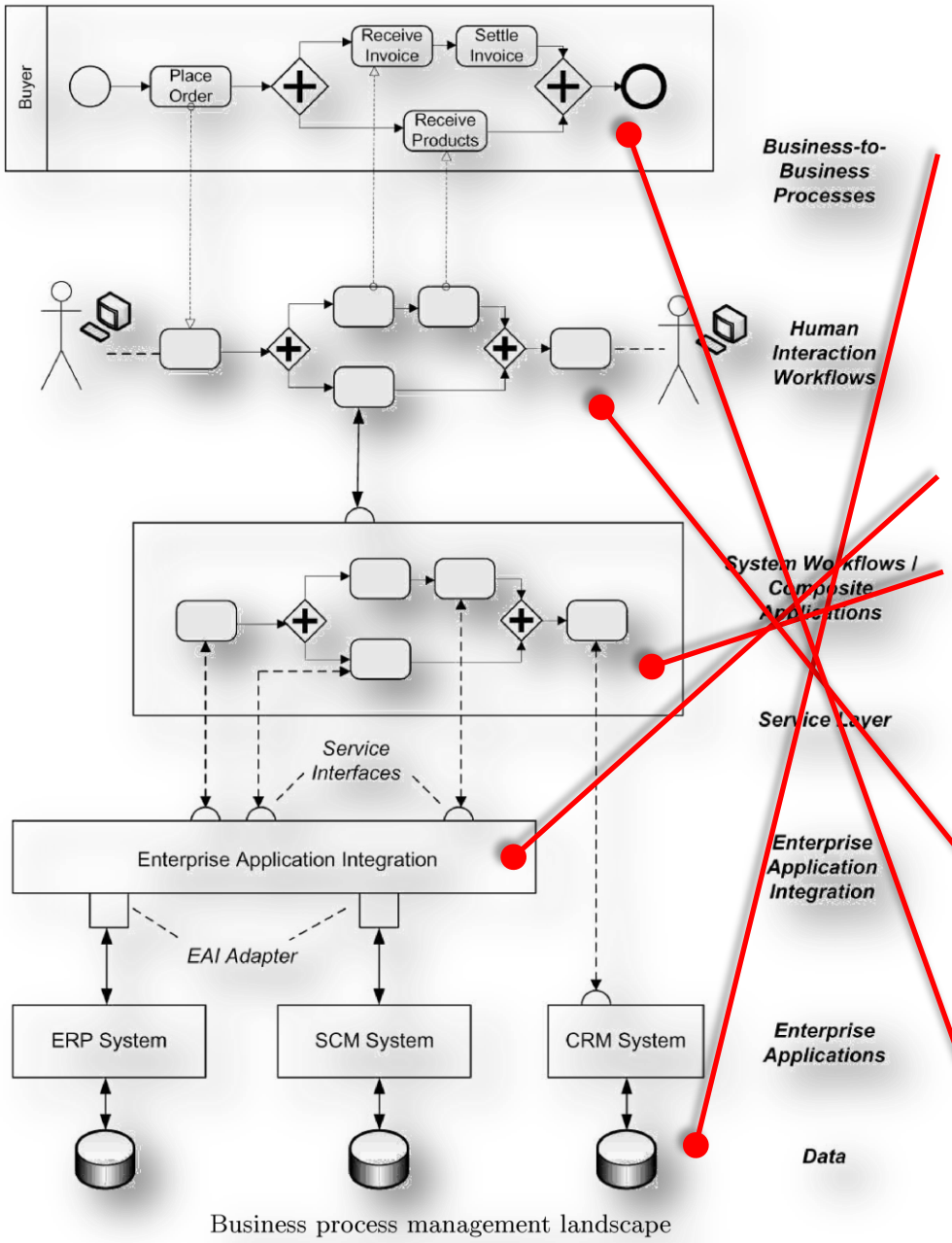
- The structure of composite applications can in many cases be expressed as a business process.

- The activities of these processes are implemented by invoking enterprise services. Additional execution constraints like conditional execution can be represented by business process models

- Enterprise services can also be used to realize business interactions of multiple enterprises (multiple pools).



Using service composition to realize composite applications



Business process management landscape

- The business process management (BPM) architecture is shown. At the lowest level, heterogeneous applications, such as ERP and CRM, but also tailor-made applications.
- Integration issues are covered by an EAI middleware, via adapters for heterogeneous applications.
- The functionalities of enterprise applications are provided through services to the system workflow (**service tasks**)
- The activities of human interaction workflows can be then associated **(user tasks)**
- Finally, activities in human interaction workflows can also be part of a business-to-business process interaction.