
Conception of Information Systems

Lecture 1: Basics

8 March 2005

<http://lsirwww.epfl.ch/courses/cis/2005ss/>

Information System: Definition

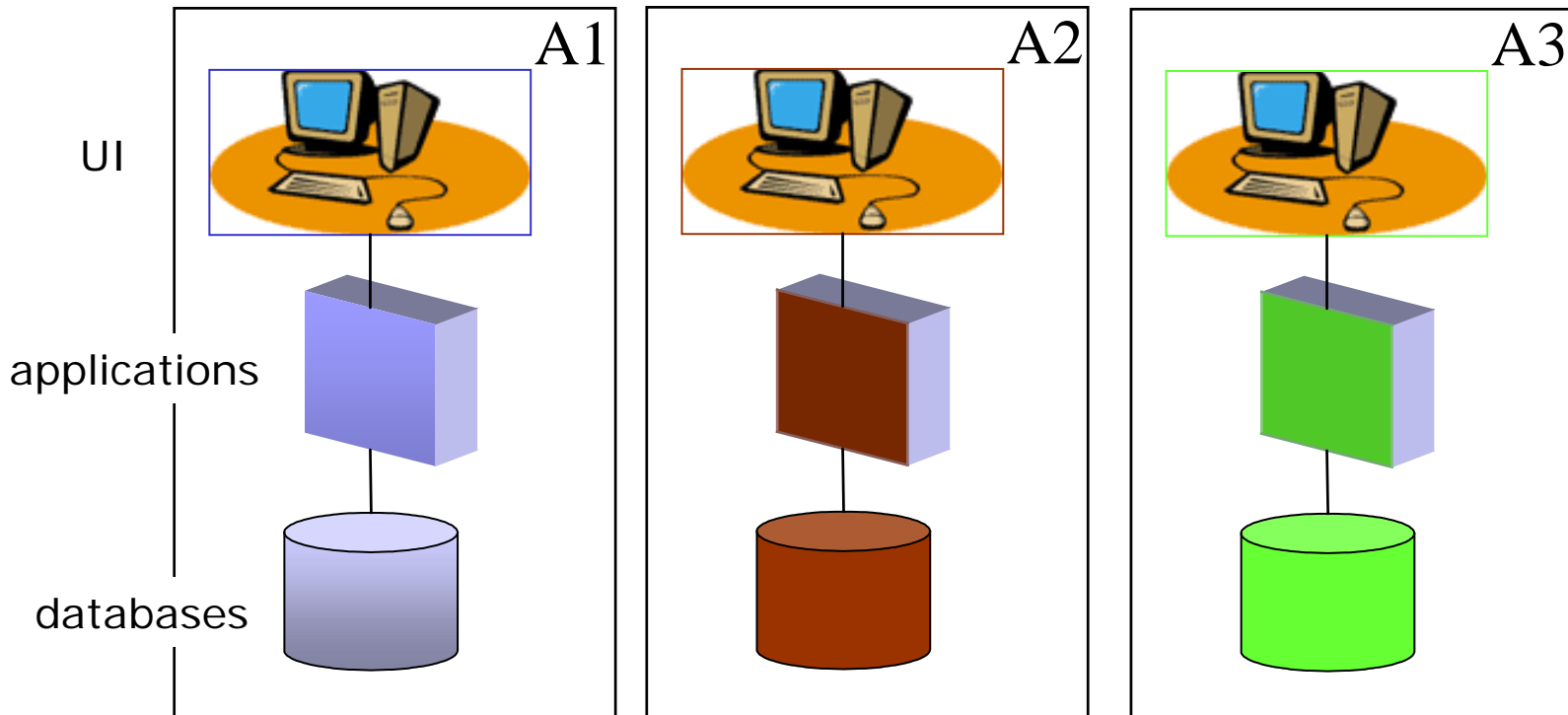
- Webopedia:

An *information system* is a system that collects and stores data

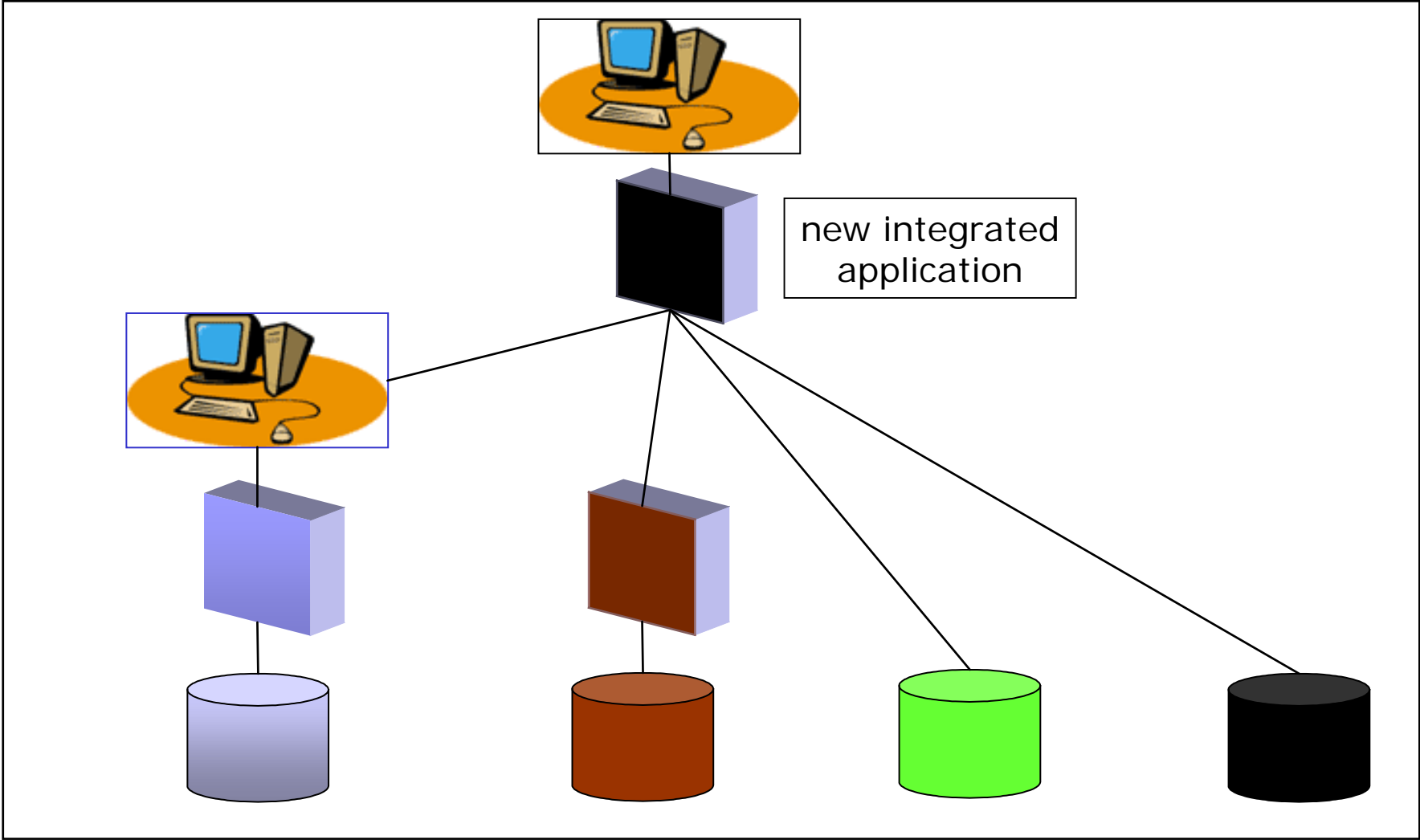
A *system* is a group of interdependent items that interact regularly to perform a task

Information Systems: Integration

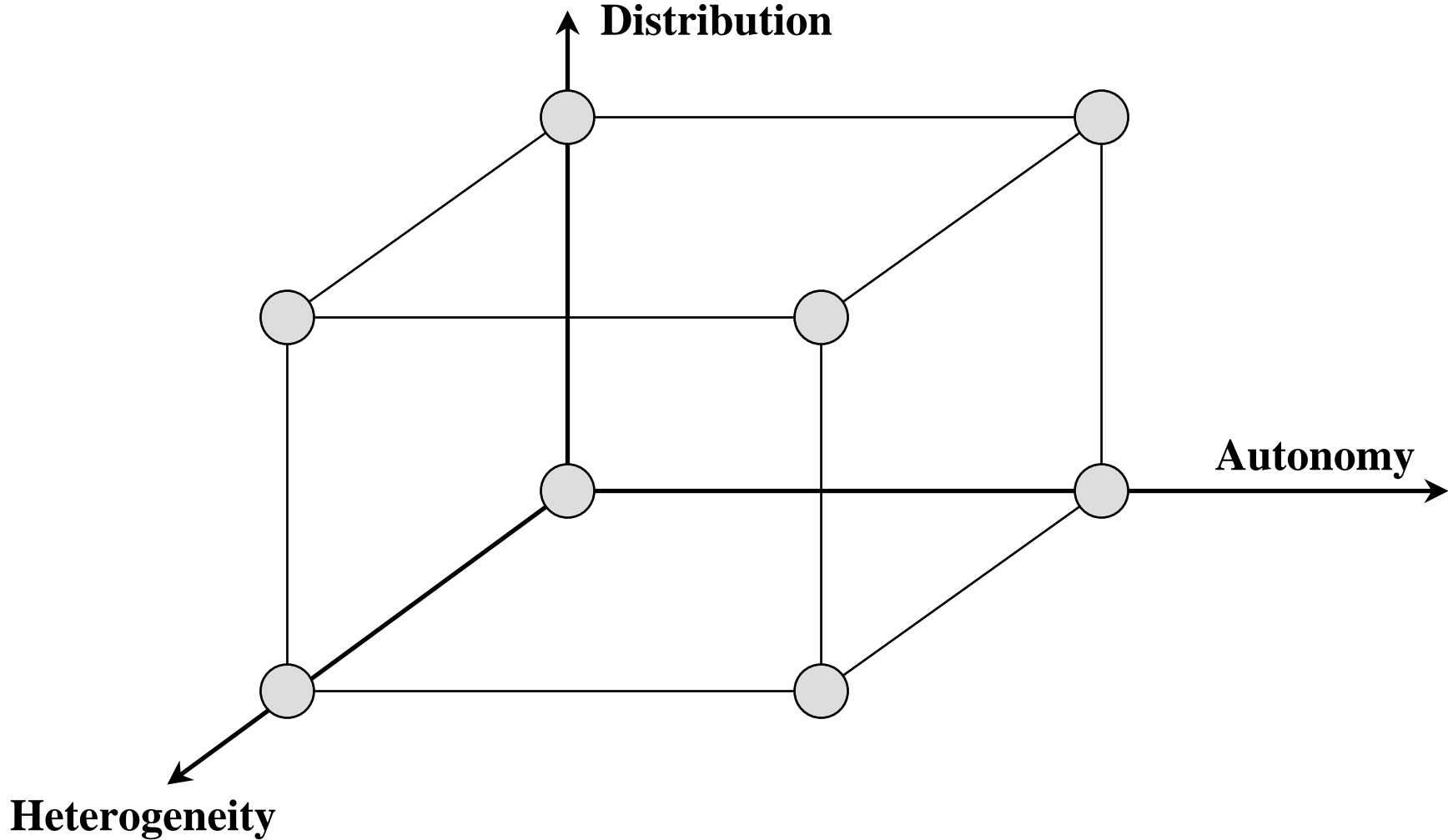
- 30% development of new systems
- 70% integration of existing systems into new applications
- Example: 3 applications implementing a three-tier architecture:



Integrating Legacy Applications



Challenges in Information Systems Integration



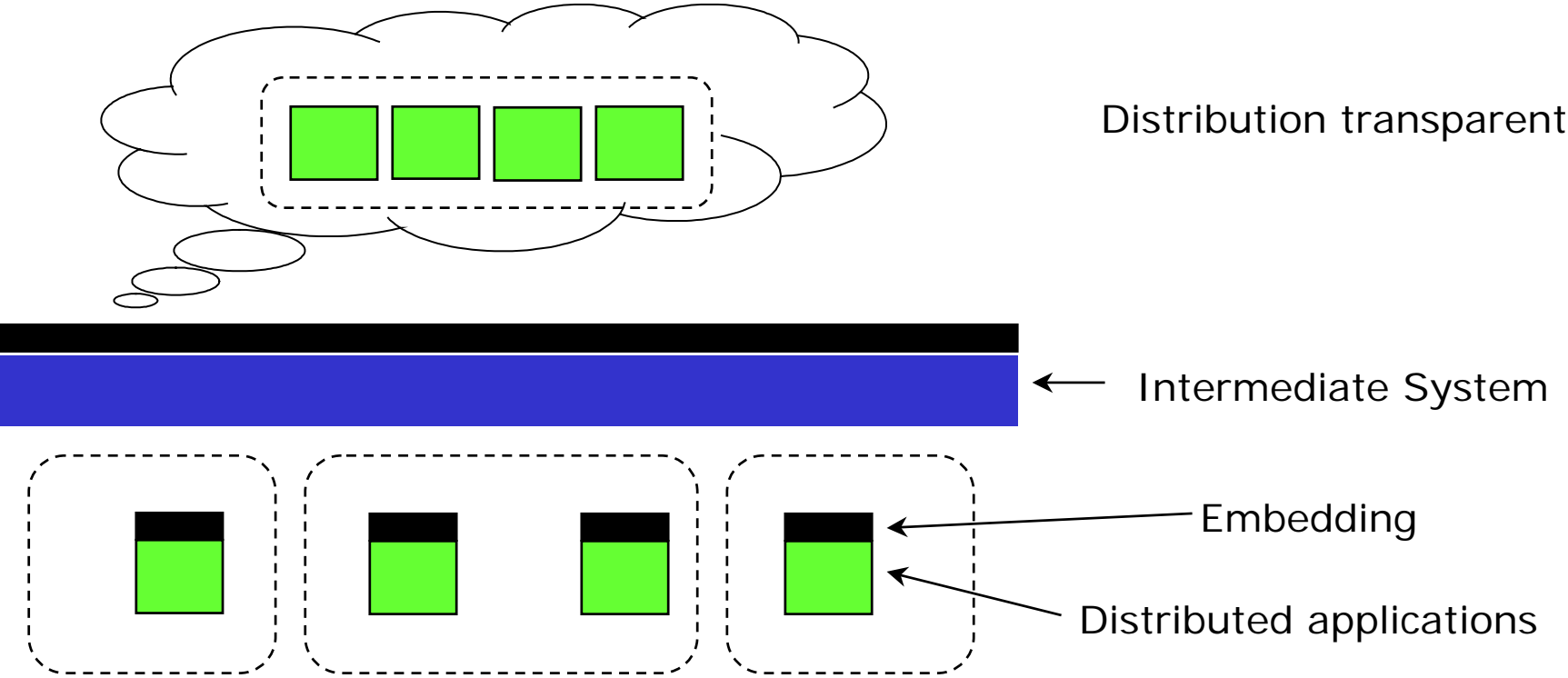
Distribution

- Communication
 - Applications running on different nodes
 - Locating the applications
 - Establishing and handling connections
 - Invoking operations
 - Asynchronous vs. synchronous
- Distributed data
 - Fragmentation and replication of data
 - Consistency
- Distributed operations
 - Distributed transactions
 - Coordination
- Performance
 - Replication and caching
 - Network latency
 - Data vs. operation shipping



Dealing with Distribution

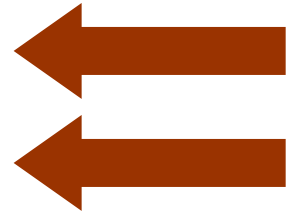
- Making distribution transparent: Intermediate systems support an abstraction layer such that the developer does not bother about details of distribution



Heterogeneity

- Syntactic heterogeneity

- Interfaces
 - Protocols and languages: http, CORBA, SQL, ODBC,
- Data models
 - Relational, XML, object-oriented, ...
- Coupling
 - Stateless- or stateful connections
 - Security and admission procedures



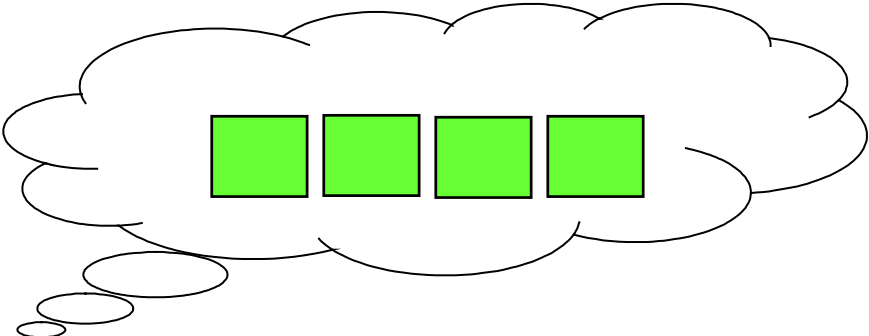
- Semantic heterogeneity

- The semantics of the models do not coincide
- The semantics coincides but the structural representation differs
- Applies both to data and execution models (e.g. transaction model)
- The data representation (schema) is the same but the extensions differ



Dealing with Heterogeneity

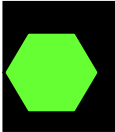
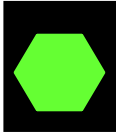
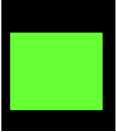
- Making heterogeneity transparent: Intermediate systems provide an abstraction layer such that the developer does not see differences in the underlying models



Heterogeneity transparent



← Intermediate System



Embedding

Heterogeneous applications

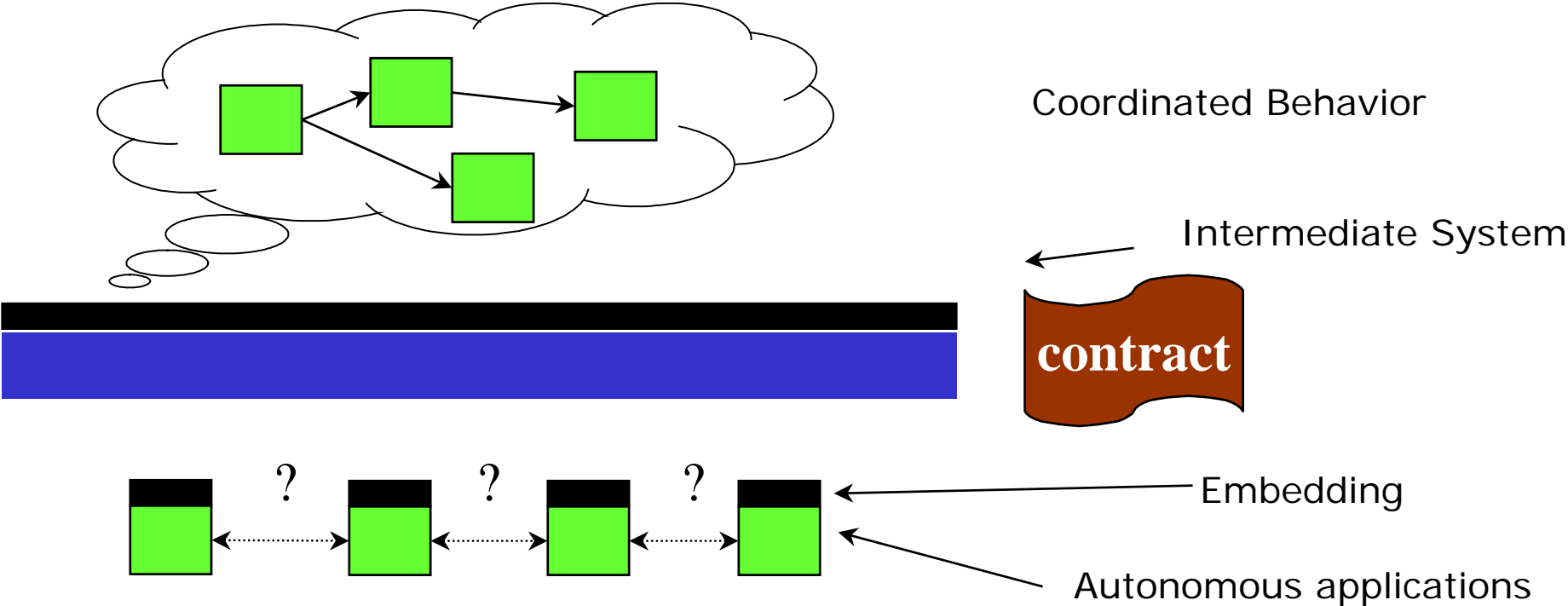
Autonomy

- Autonomy requirement
 - The local operation of a system and system consistency is not affected by its participation in a global system
- Semiautonomous systems
 - Give up some of their autonomy to participate in a global system (contract)
- Design autonomy
 - Choice of data and process model
 - may lead to ***heterogeneity***
- Communication autonomy
 - Choice when and to what extent provide access to its resources
- Execution autonomy
 - Choice of how to respond to requests for operation execution



Dealing with Autonomy

- Establish contracts among autonomous applications such that they can work together in useful ways



Solutions

To successfully integrate information systems, we need:

- Global abstract models in which the integrated systems are represented
 - Data and process models
 - Interfaces offered
- System framework that implements the integration
 - Middleware
- Mechanisms for embedding the existing systems
 - Mappings for data and operations
 - Interfaces to comply with
- Usage methodologies
 - How to use integration technology

Solutions typically address multiple aspects (distribution, heterogeneity, autonomy) at the same time

Types of Middleware

- Data-oriented
 - Federated databases
- Document-oriented
 - XML middleware
 - Web servers
- Transaction-oriented
 - Transaction monitors
 - Message queuing systems
- Object-oriented
 - Distributed object architectures
 - Component technologies
- Process-oriented
 - Workflow management systems