



An Architectural Pattern for Designing Intelligent Enterprise Systems

Eugenio Zimeo[⌘], Gianfranco Oliva[§], Fabio Baldi[§], Alfonso Caracciolo[§]

[⌘]Department of Engineering – University of Sannio

[§]Poste Italiane – TI-RS Centro di Ricerca e Sviluppo - Napoli

The context

- Shift from enterprise to intelligent enterprise systems
 - Drivers: web-based collaboration, social mining, data semantics
 - Target functions: context-awareness and personalization
- **Challenge**: *providing enterprise systems with intelligent behaviors still ensuring typical non-functional features of these systems, such as flexibility, efficiency and scalability*

What are enterprise systems?

- They are **large-scale** application software that support business processes, information flows, reporting and data analytics in complex organizations
 - are designed to manage large volumes of critical data and to provide high level of transaction performance
 - can provide B2C or B2B services outside the organization
- **E-Commerce** is a well-known class of enterprise systems
 - Intelligent e-Commerce software systems represent today the main trend

Expected impacts on architecture design

- Exploiting architectural patterns is a recognized approach for aiding software architects and developers to handle the complexity of large-scale software systems
- Enterprise architectural patterns need to be revised to take into account the desired ability of intelligent systems to “generate” knowledge from multiple external sources
 - sources could be tightly related to the e-Commerce application, different enterprise subsystems that support cross-business features, or external to the organization boundary
 - sources can change over time or can be enriched with additional ones when new needs emerge

Background and related work

- Several patterns have been proposed in the literature to design enterprise systems
 - IBM has proposed a good catalogue that includes: *Self-Service*, *Collaboration*, *Information Aggregation*, and *Extended Enterprise*
 - *Model View Controller* represents the most known pattern to implement the Self-Service model in the web context
 - *Multi-tier architectures* are often exploited to ensure scalability, security and flexibility of enterprise systems
 - *Service Oriented Architectures* propose the adoption of *Enterprise Service Buses* to loosely couple enterprise sub-systems
 - *MAPE-K* is a recently proposed pattern to design autonomic systems
- Many scientific papers on intelligent enterprise systems mainly focus on specific architectures for OLAP (see the paper)
- Context-aware recommendation and personalization in e-Commerce introduce a new class of enterprise systems that need specific architectural studies to provide architects and developers with new reference models

The Inference Pattern (1/4)

- Problem

- We want to design a software system able to perform processing on its own data with the aim of making it explicit the implicit knowledge that may be useful to users and to the system itself

- Context

- Interactive applications generate data from users interactions. These data can be processed together with the data coming from other sources to generate new information that can be useful to understand users' behaviors and preferences. This improves the effectiveness of interactions for achieving the business objectives

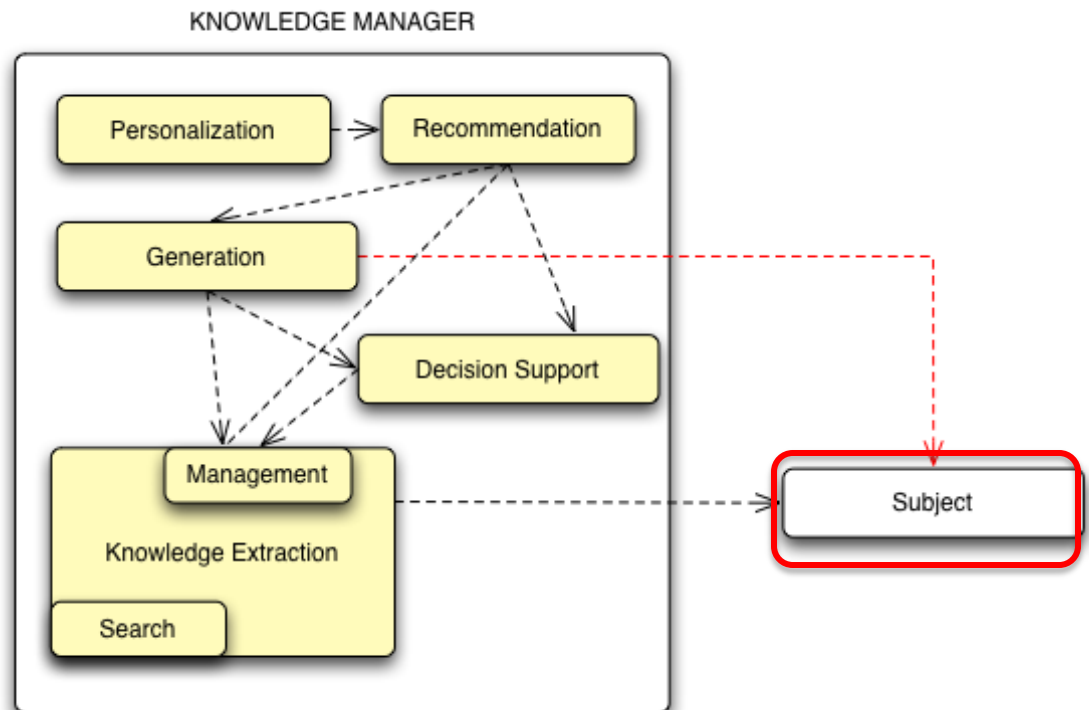
The Inference Pattern (2/4)

- Solution
 - The system is decomposed in two logically separated subsystems:
 - the one in charge of answering to user requests and the one responsible of collecting data and preprocessing them for context-aware recommendation
 - The logical separation promotes:
 - Extensibility of the knowledge base
 - Continuous and concurrent processing to extract knowledge with respect to the transactional activity generated by the interactive subsystem
 - Easy and optimized deployment of the implemented software system

The Inference Pattern (3/4)

- Participants

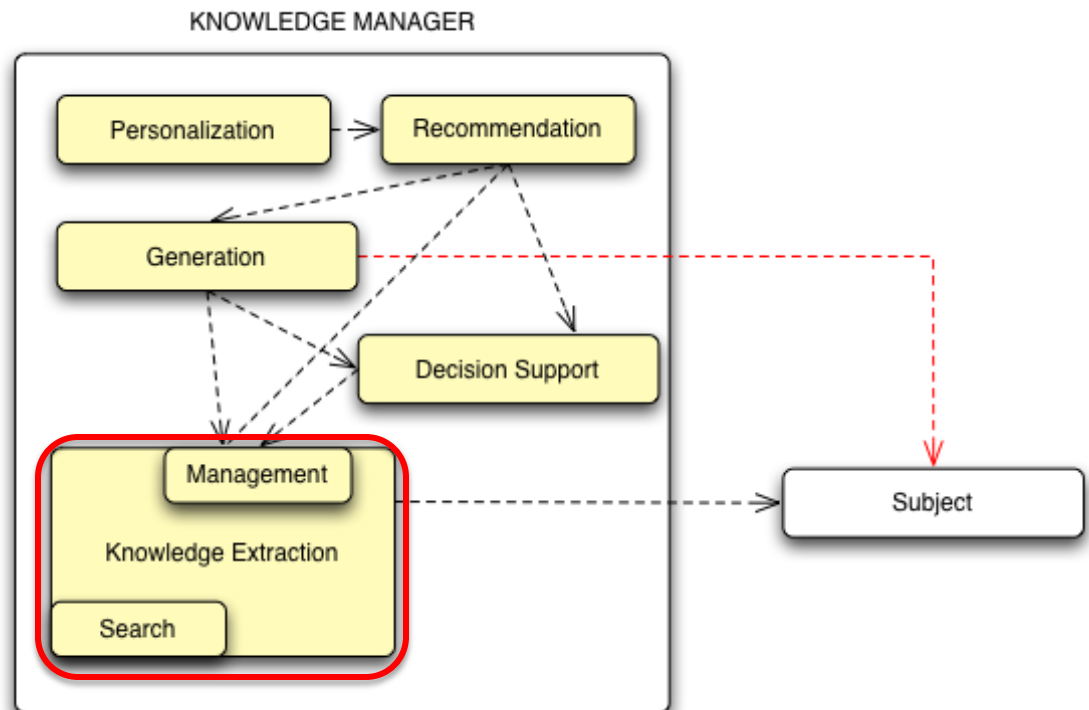
Subject: is the observed subsystem from which implicit knowledge is extracted.



The Inference Pattern (3/4)

- Participants

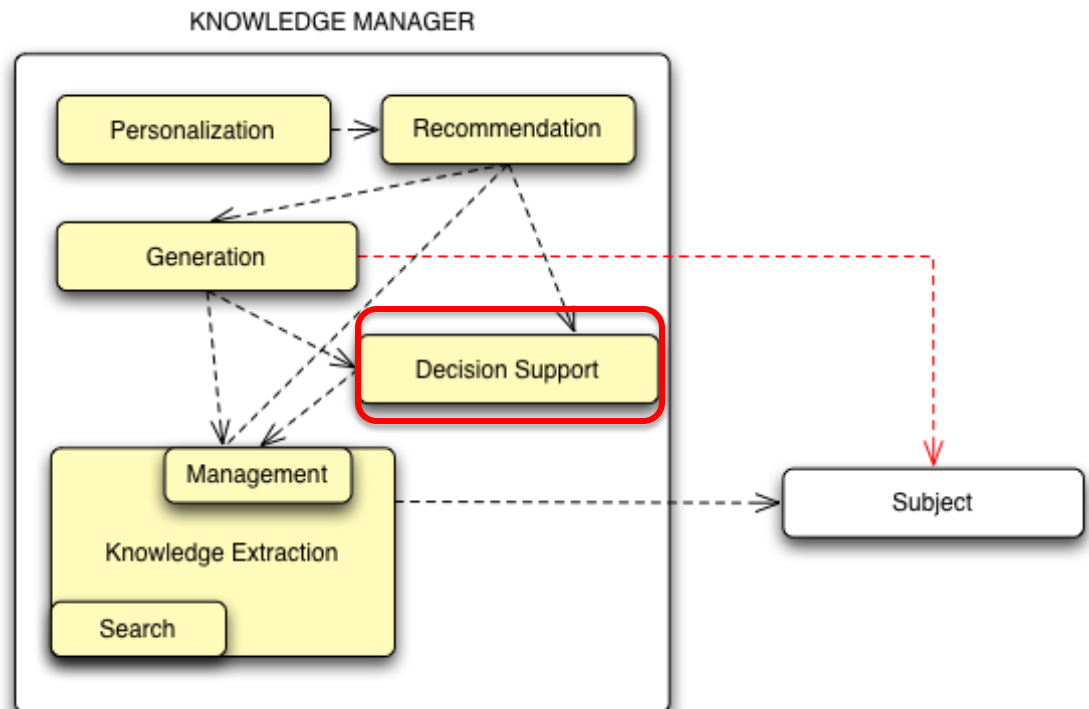
Knowledge extraction: collects the data produced by the interactions between users and the subject and extracts the implicit knowledge.



The Inference Pattern (3/4)

- Participants

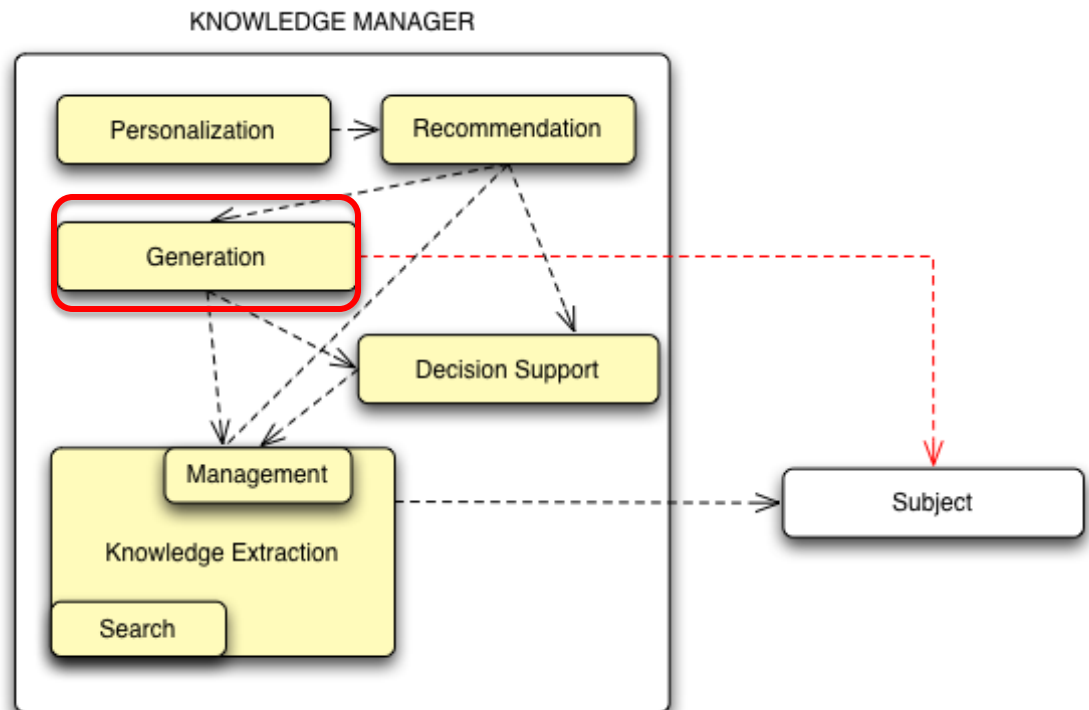
Decision Support: it provides the user with a new knowledge extracted by the KE subsystem to support proper decisions



The Inference Pattern (3/4)

- Participants

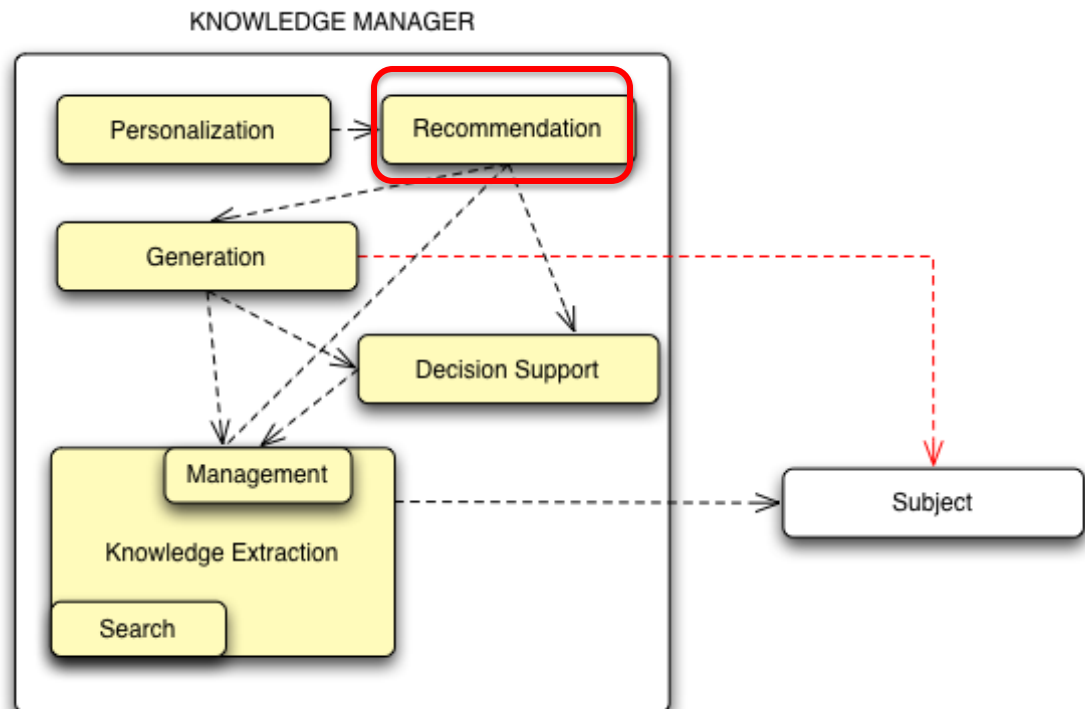
Generation: generates new knowledge for the subject, by using the implicit knowledge extracted by the Knowledge Extraction (KE) subsystems.



The Inference Pattern (3/4)

- Participants

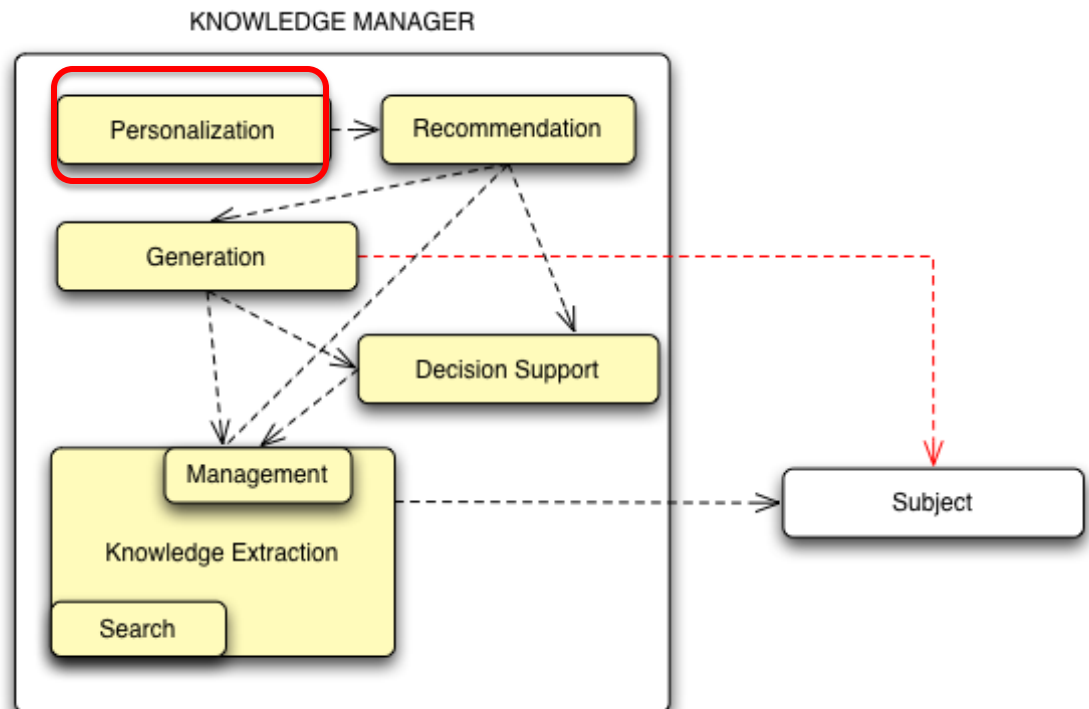
Recommendation: suggests information to users. It may use the Generation subsystem to ask for the generation of aggregated data or the Decision Support subsystem to recommend a choice to a user.



The Inference Pattern (3/4)

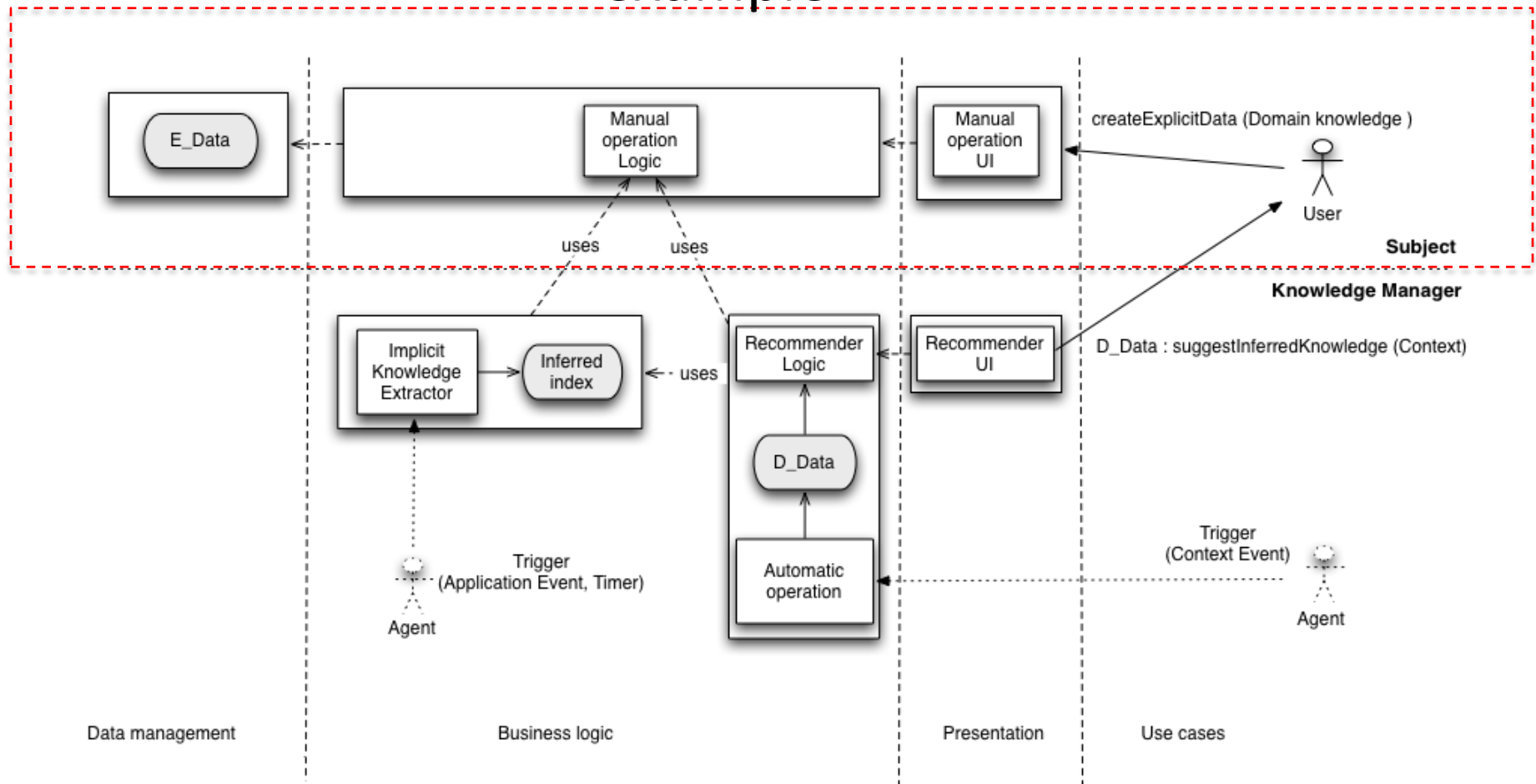
- Participants

Personalization: it personalizes the suggestions with respect to the specific user (or context in general). It is a sort of recommender supporting personalization.



The Inference Pattern (4/4)

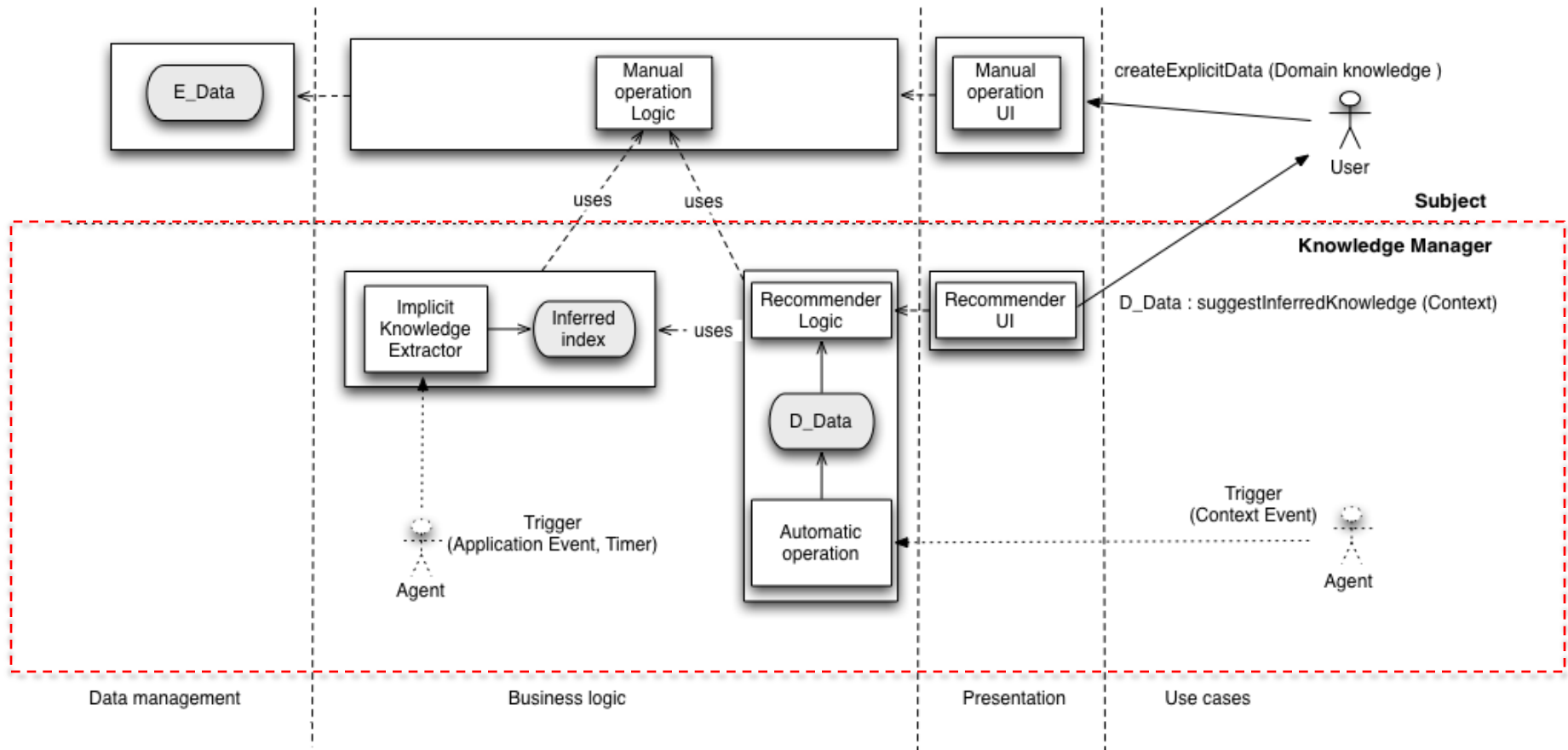
example



Application of the inference pattern to a multi-tier architecture

The Inference Pattern (4/4)

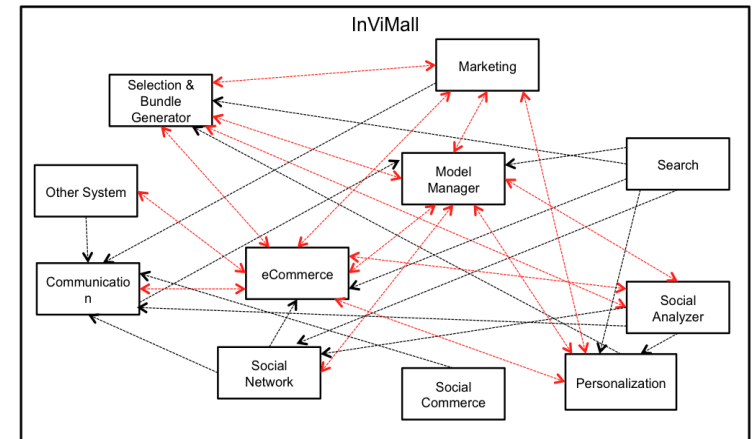
example



Application of the inference pattern to a multi-tier architecture

Case study (1/2)

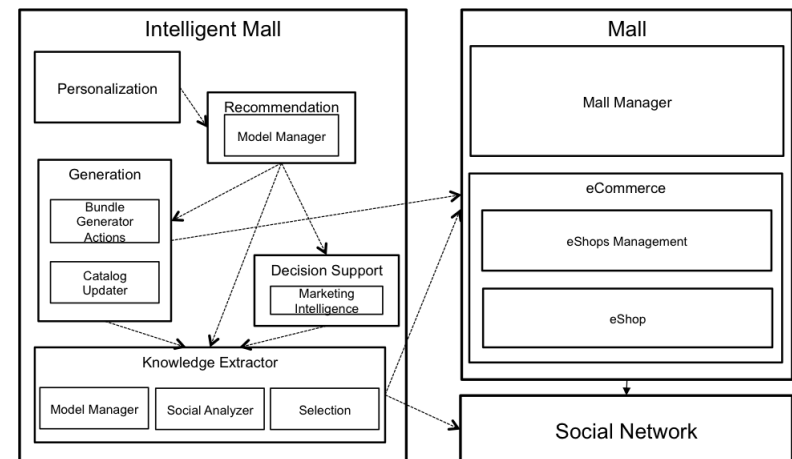
- The Inference Pattern has been exploited to design a scalable e-Commerce system, called InViMall (Intelligent Virtual Mall), at Poste Italiane
 - InViMall is an electronic mall, based on e-community concepts
 - It exposes several innovative features through a multichannel interface:
 - personalized suggestions
 - automatic generation of personalized product bundles
 - advanced marketing intelligence
 - faceted navigation



Analysis based on
Dependency Structure
Matrix

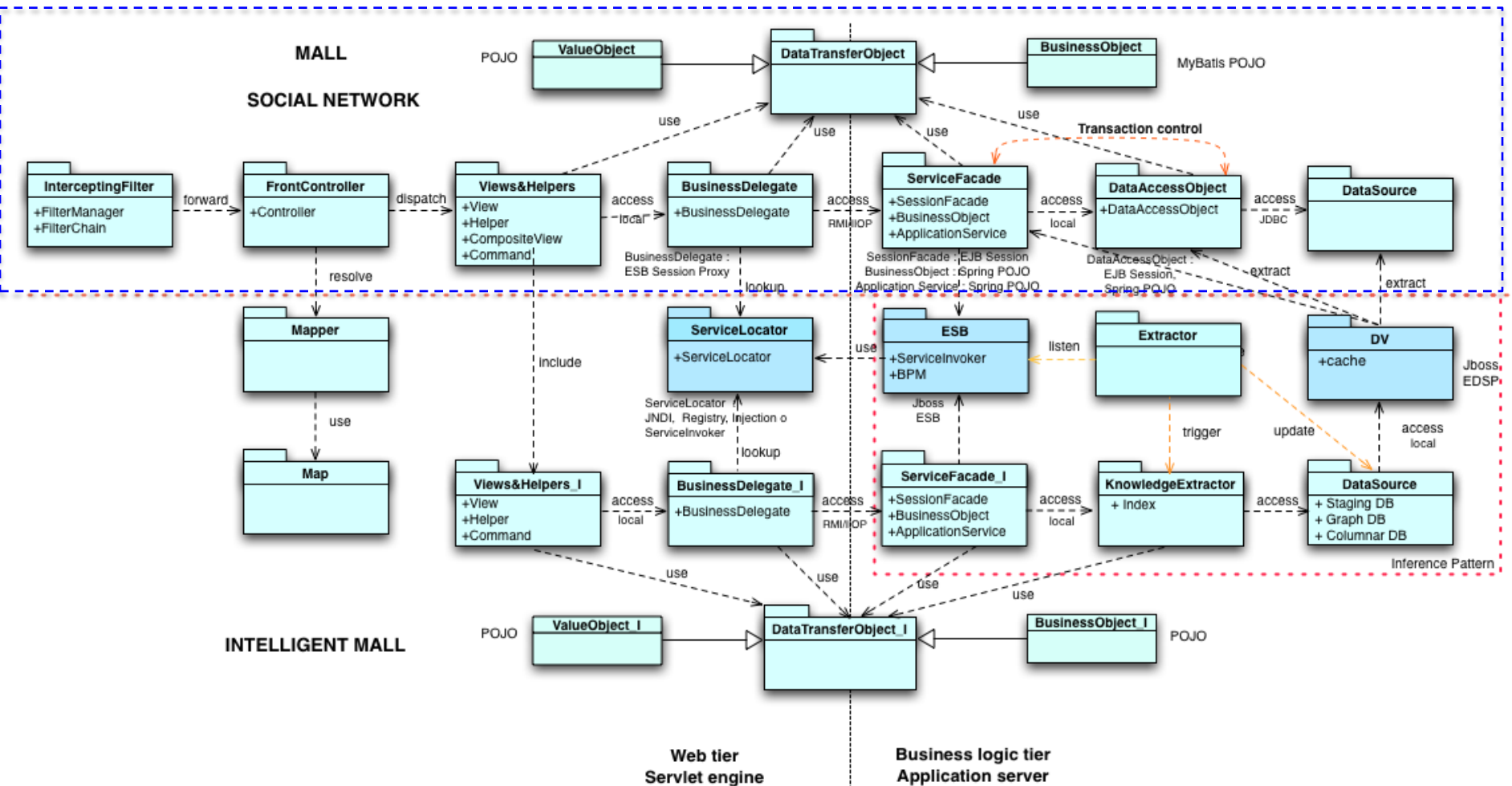


Attribute Driven Design



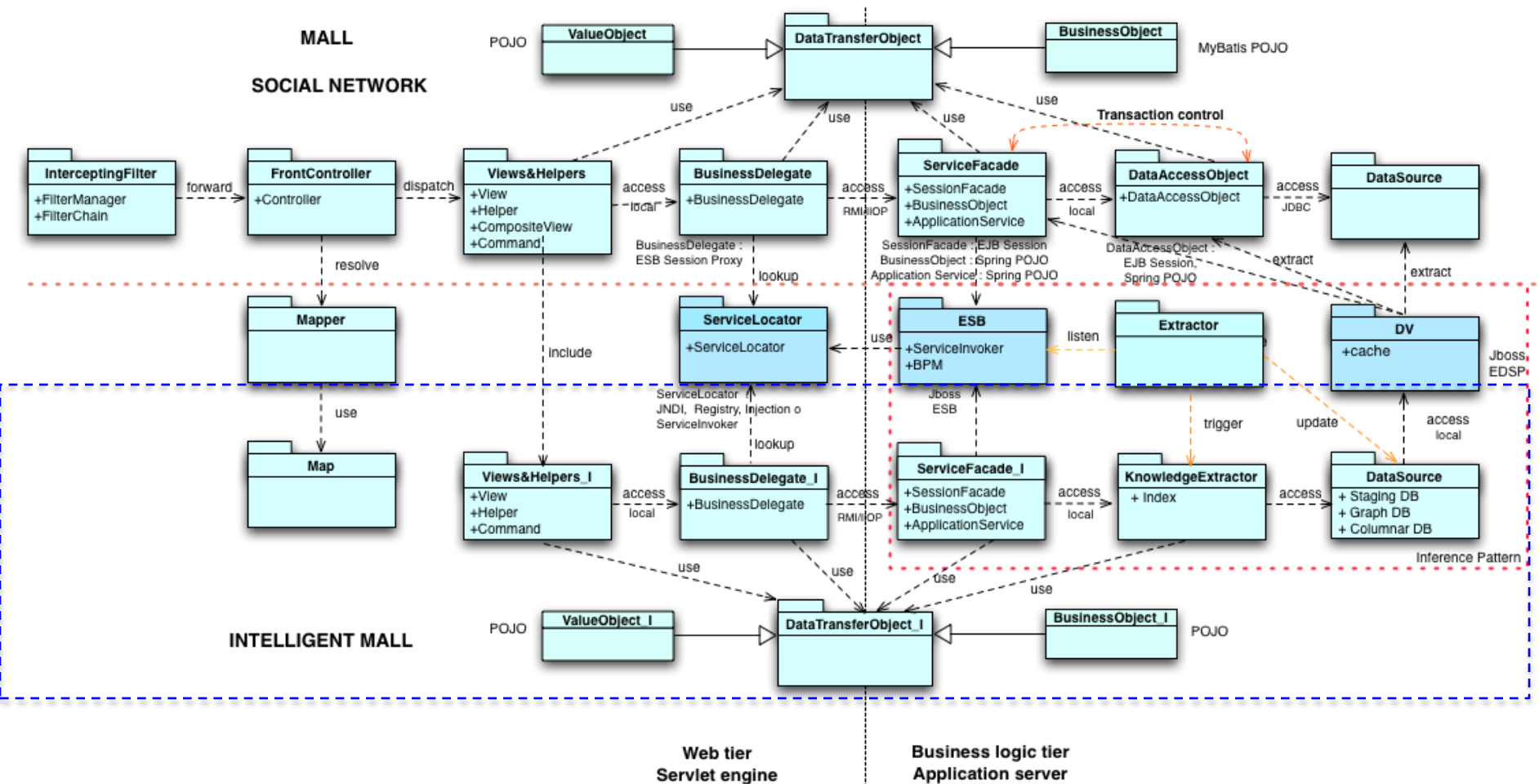
Case study (2/2)

Applying the Inference Pattern to the **InViMall** system (integration view)

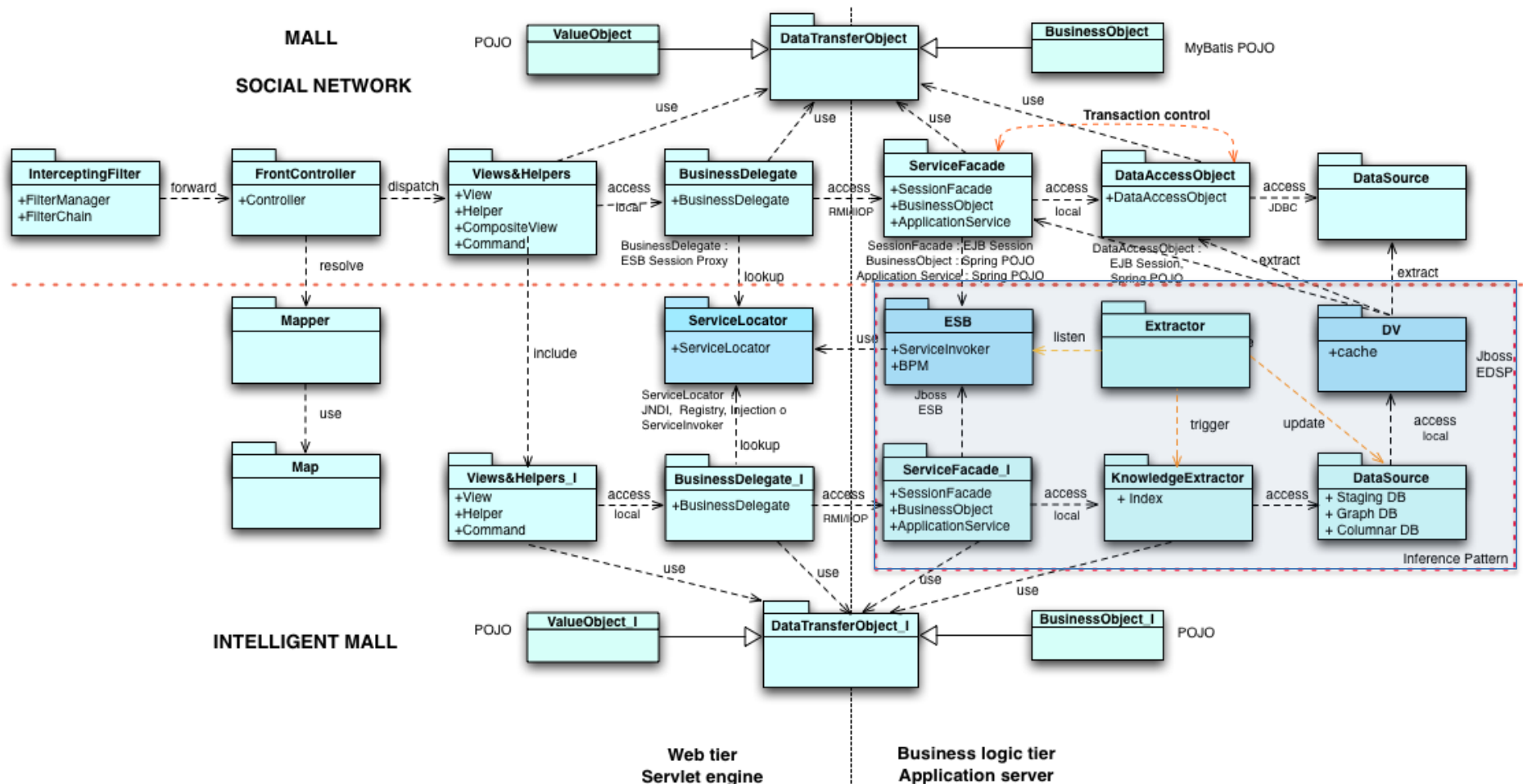


Case study (2/2)

Applying the Inference Pattern to the **InViMall** system (integration view)



Applying the Inference Pattern to the **InViMall** system (integration view)



Conclusion and future work

- The paper presented an architectural pattern that helps software architects to design intelligent enterprise systems that combine interactive features with batch processing to infer knowledge from data
- The pattern was applied with success to design a complex application that implements an electronic mall with a clear logical separation between the transactional and intelligent subsystems
- The pattern enables the integration of both real-time, event-driven interactions, and data accesses for large volumes of data
 - In the future, more sophisticated integration middleware platforms that transparently combine the two benefits above will be investigated

Thank you for your attention

eugenio.zimeo@unisannio.it