

# Design Patterns

MSc Computer Science

---

Produced  
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics  
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



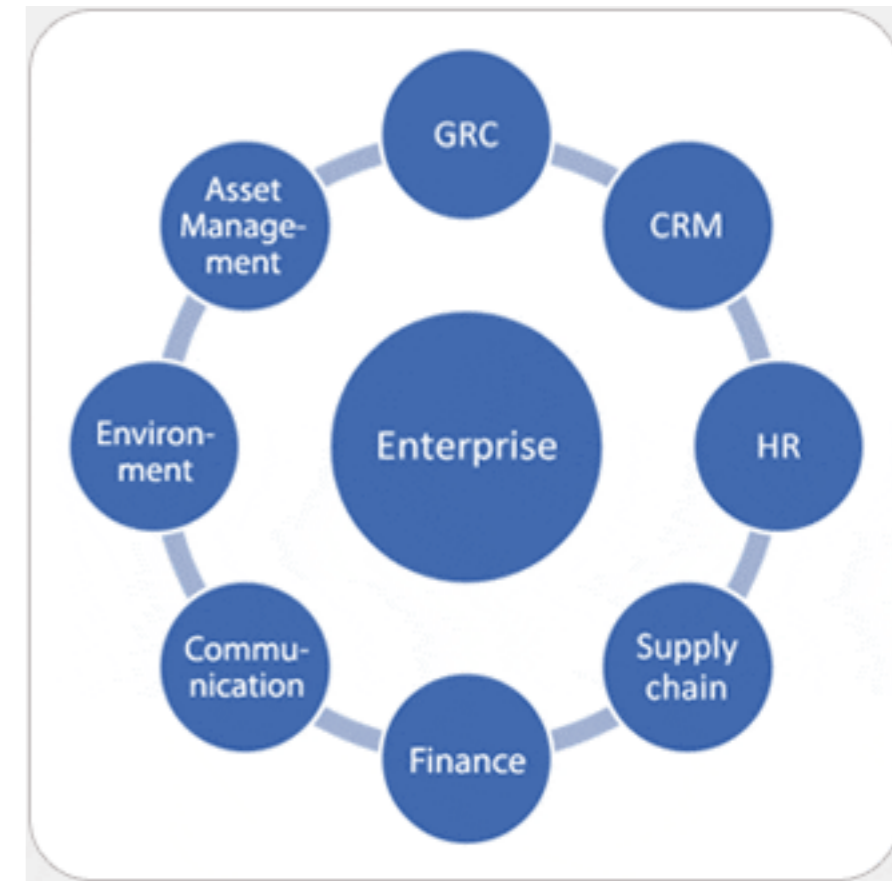
# Characteristics of Enterprise Systems



# Characteristics of Enterprise Applications

---

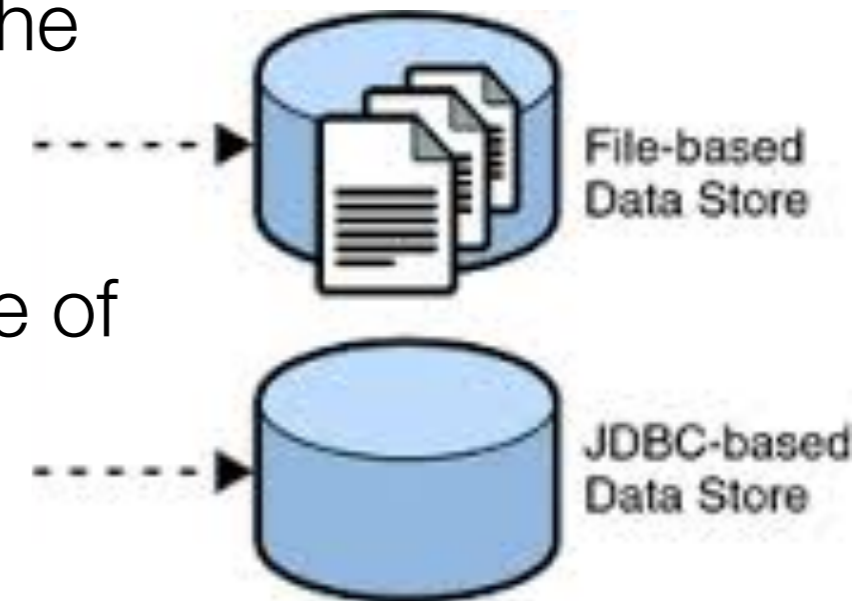
- persistent data
  - a lot of data
  - access data concurrently
  - a lot of user interface screens
- integrate with other enterprise applications
  - conceptual dissonance
  - complex business logic
  - evolution of requirements



# Persistent Data

---

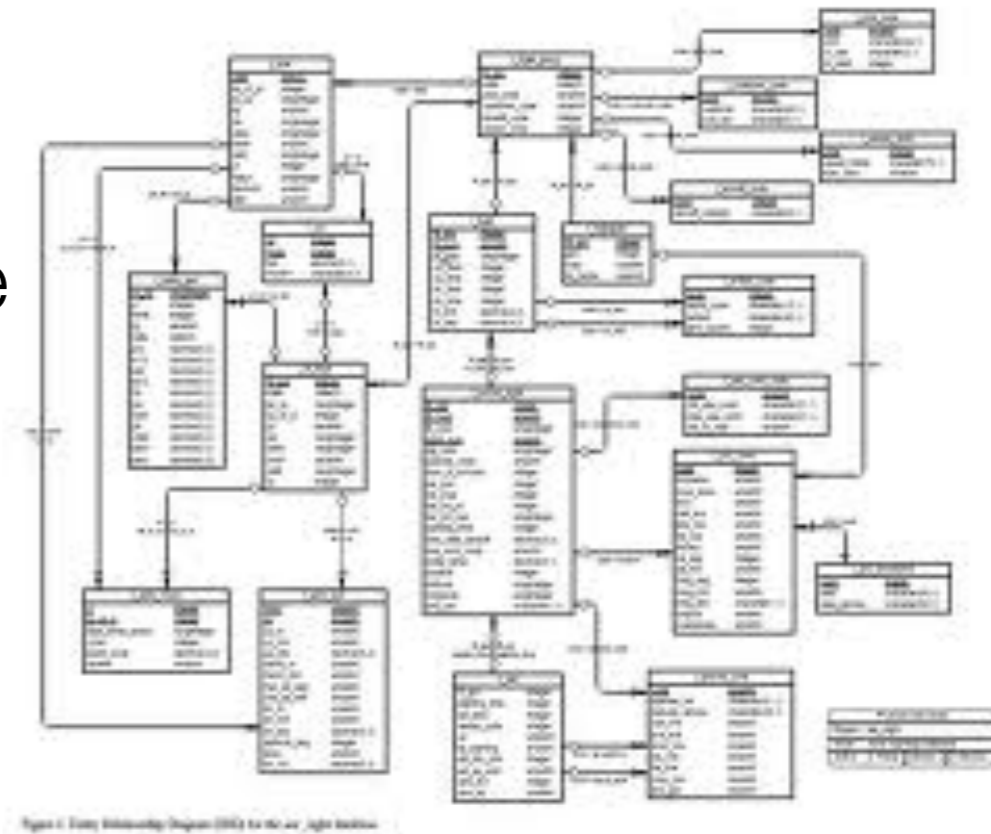
- Needs to be around for months and usually years...
- .. during which there will be many changes in the applications that consume/produce it.
- During lifetime may be changes to the structure of the data ...
- ... requiring 'evolution' of data based on older schemes
- May even outlive hardware, operating systems and compilers that created it.



# A Lot of Data

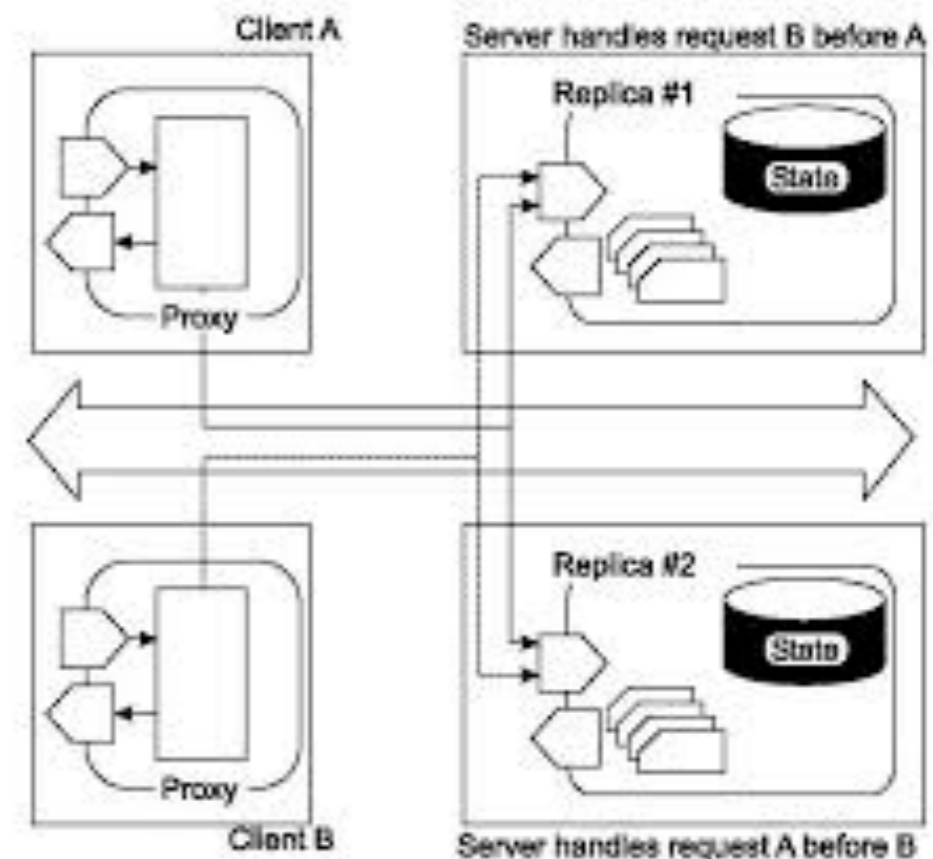
---

- A moderate system will have over several GB of data organized in tens of millions of records...
- ...replacing older systems used indexed file structures such as IBM's VSAM and ISAM.
- Use mostly relational databases...
- ... although NoSQL beginning to be an option.
- The design and feeding of these databases has turned into a sub-profession of its own.



# Data Accessed Concurrently

- For many systems this may be less than a hundred people...
- ... however, for Web-based systems may be thousands or hundreds of thousands.
- Major issues in ensuring that all users can access the system properly....
- ... with transaction managers required to ensure two users don't access the same data at the same time in a way that causes errors.



# Many UI Screens

---

- Potentially hundreds of distinct screens...
- ...with users have highly varying technical expertise.
- Information has to be presented in many different ways for different purposes....
- ... and may have a elements of batch processing, which presents its own challenges for the UI

# Integrate with Other Enterprise Applications

---

- Usually need to integrate with other enterprise applications scattered around the enterprise...
- ... and these may be built at different times with different technologies, and with different collaboration mechanisms (COBOL data files, legacy messaging systems etc.)
- Occasionally the enterprise will attempt to integrate its different systems using a common communication technology...
- ...but often not completed, leaving several different unified integration schemes in place at once.



# Conceptual Dissonance

---

- Differences in business process and conceptual dissonance with the data.
  - One division of the company may think a customer is someone with whom it has a current agreement;
  - another division also counts those that had a contract but don't any longer
- As a result, data has to be constantly read, adapted, and written in all sorts of different syntactic and semantic formats.

# Complex Business Logic

---

- ‘there are few things that are less logical than business logic’
- contrast with systems software - highly logical and scientific.
- Business systems - strange conditions that often interact with each others in surprising ways.
- Many one-off special cases is what leads to the complex business "illogic" that makes business software so difficult.

# Enterprise Actors

---

- B
- C
- E
- G
- C

# Enterprise Actors

---

- Business
- Customer
- Employee
- Government
- Citizen

# Examples

---

- B2C - e.g. online retailer
- B2B - e.g. processing of leasing agreements
- B2E - e.g. expense-tracking system for a small company
- C2C - e.g. classified adds
- G2G, G2E, G2B, B2G G2C, C2G?

# B2C - Online Retailers

---

- Handle a very high volume of users - solution needs to be efficient in terms of resources used but also scalable load can be increased by adding more hardware.
- The domain logic for such an application may be straightforward: order capturing, simple pricing and shipping calculations, shipment notification.
- To be widely available - Web presentation that can be used with the widest possible range of browsers.
- Data source includes a database for holding orders + communication with an inventory system for availability and delivery information.

# B2B: Processing of leasing agreements

---

- Perhaps simpler than B2C retailer's because there are many fewer users —no more than a hundred or so at one time.
- Business logic more complex - calculating monthly bills on a lease, handling events (early returns and late payments), and validating data.
- Many intricate tasks, which may be in the form of little variations over deals done in the past.
- Requires more complex UI - with users requiring a more sophisticated presentation than conventional HTML pages.
- A more complex user interaction also leads to more complicated transaction behavior: Booking a lease may take an hour or two, during which time the user is in a logical transaction

# B2E - Expense Tracking System

---

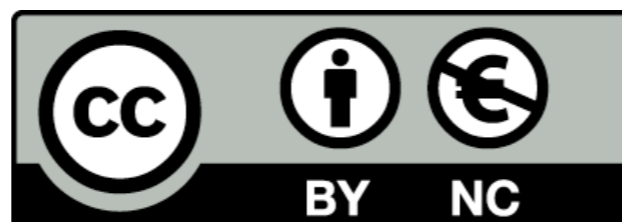
- Small number of users and simple logic and can easily be made accessible across the company with an HTML presentation - data source is a few tables in a database.
- May have to be built quickly and may grow as people want to calculate reimbursement checks, feed them into the payroll system, understand tax implications, provide reports for the CFO, tie into airline reservation Web services, etc...
- Trying to use the architecture for B2C or B2B may slow down the development of B2E service. However, do not want to make decisions now that will hamper future growth.
- If flexibility added, but later proved inappropriate, the added complexity may actually make it harder to evolve in the future.
- However, although such systems may be small, enterprises may have many of them so the cumulative effect of an inappropriate architecture can be significant.



# Patterns: Problems/Solutions in Context

---

- Each of these three enterprise application examples has difficulties, and they are different difficulties...
- .. a single architecture may not be right for all three.
- Choosing an architecture requires understanding of the particular context of the system...
- .. patterns inherently reflect the context in which they may be suitable



Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE

