Enterprise System Integration
(MTAT.03.229)

LECTURE 1: INTRODUCTION

MOHAMAD GHARIB
UNIVERSITY OF TARTU

Based on slides by Marlon Dumas
Part I

Motivating scenario and problem statement
Enterprise system scenario

Information system of a hotel - Three functional areas:

• **Guest management**: reservations, check-in (including room assignment), charges to room (e.g., room service), check-out, closing of bill;

• **Room availability**: how many rooms still available for a given night, forecasting how many guests will stay beyond their reservation, forecasting occupancy of hotel at a future date;

• **Room cleaning & maintenance**: managing maid & maintenance staff, cleaning rooms after late check-out.

Can you see **dependencies** across functional areas?

Based on scenario by Richard Hull
Enterprise system scenario

Change # 1: A restaurant is opened for hotel guests and external customers:

• External customer pays for meal directly.
• Hotel guest has option to pay for meal directly or charge to room
  o In either case, hotel likes to keep a record of the guest eating in the restaurant, if guest is willing to reveal his room number.
  o Restaurant offers breakfast for guests.

⇒ Restaurant billing system has touch-points with hotel guest system.
Enterprise system scenario

**Change # 2:** A loyalty program is introduced:

- Set up new system to keep track of customers who are in loyalty program.
- Loyalty program has multiple touch-points with hotel and restaurant system:
  - At reservation time, inquiring if customer is a member of loyalty program, offer promotions, giving priority for room allocation to club members.
  - At check-in, inquire if customer is in loyalty program, if not offer sign-up, if yes offer better room.
  - Offer discounts at restaurant.
  - Offer discounts on other services at check-out and record length of stay and points earned.
Enterprise system scenario

Change # 3: Hotel is bought out by a franchise:
• Reservations can be made either through the franchise system or by hotel.
  o Room availability have to deal with this dual sourcing;
  o Forecasting & pricing outsourced to franchise’s system.
• Loyalty program merged into franchise’s program
  o Hotel no longer operates the loyalty program, but must interoperate with loyalty program system of franchise;
  o Benefits of existing loyal customer must be honored;
  o The franchise loyalty program gives points for all expenses paid, whereas the original loyalty program looked only at nights stayed in hotel;
• Room cleaning/maintenance still managed locally.
Enterprise system scenario

The problem at glance

Old system

New system
Enterprise system scenario

The problem at glance

Old system

Monolith

New system

Microservice
Part II
Evolution of enterprise software
Classic enterprise “system” layers

Users interact with the system through a presentation layer (aka user interface or UI).

The application logic (aka business logic) determines what the system actually does:
  - Enforces business rules;
  - Coordinates business processes.

The data access layer facilitates the access to persistent data manipulated by the application logic.
  - Includes access to databases, search engines, document managers and/or a file system.
## Classical Enterprise System

### Evolution of computation

<table>
<thead>
<tr>
<th>60’s</th>
<th>70’s</th>
<th>80’s</th>
<th>90’s</th>
<th>00’s</th>
<th>10’</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM mainframes</td>
<td>IBM, DEC Mini-comp.</td>
<td>PC, DOS</td>
<td>Windows</td>
<td>Web 2.0</td>
<td>iOS, Android</td>
</tr>
<tr>
<td>Monolithic</td>
<td>Unix, VAX</td>
<td>Mac, MacOS</td>
<td>Internet</td>
<td>Mac OS X</td>
<td>HTML5</td>
</tr>
<tr>
<td>applications</td>
<td>Dumb terminals</td>
<td>WANs &amp; LANs</td>
<td>Web &amp; HTTP</td>
<td>Windows XP</td>
<td>SaaS, PaaS, IaaS</td>
</tr>
<tr>
<td>Batch</td>
<td>Time-sharing</td>
<td>Client-Server</td>
<td>CORBA &amp; Browsers</td>
<td>Application</td>
<td>Cloud computing</td>
</tr>
<tr>
<td>processing</td>
<td></td>
<td>Relational DBMSs</td>
<td>CGI</td>
<td>servers</td>
<td>Microservices</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SOA &amp; ROA</td>
<td>NoSQL</td>
</tr>
</tbody>
</table>

1-tier architecture

- All layers are bundled in a monolithic entity.
- Typical “mainframe” architecture
  - Users access the system through dumb terminals.
  - All computation happens in a single computer.
With the introduction of computer networks, computation started to be physically distributed.

Application layers are distributed depending on the computing power of clients (e.g., Thin clients or Fat clients).

The concept of API makes its appearance.

Example: Database management systems
  - The separation of data access layer promotes logical independence, reducing the impact of replacing a database technology on the presentation application logic layers.
With the arrival of PCs, the presentation layer moved to the client
- Smartphones.

With this approach, it is possible to have multiple presentation layers
- Text (console) application
- Graphical user interfaces.
- Web applications (e.g., HTML, Javascript, etc.)

Web as the universal platform for computing? Google’s Chrome.
Middleware is a level of indirection between clients and other layers
- Simplifies the design of client applications by reducing the number of interfaces.
- Encapsulates integration logic and global application logic.
- Locates resources, accesses them, and integrates results (mediates between application logic/data access layers).
N-tier architecture

- N-tier architectures result from connecting several 3-tier systems and/or adding a layer to allow clients to access the system through a Web server (“Web layer”).
- The Web layer is hosted in a Web application server: a middleware accessible through the Web.
- Web application servers are taking also parts of the functionality of traditional middleware – the boundary between Web layer and middleware is blurred.
Part III

Micro-services
Microservices

• Unlike monolithic architecture, which consists of one element, microservices architecture is created by cooperating services.
• Microservices were born out of SOA (Service-Oriented Architecture), as a solution to the challenges associated with monolithic applications.
• Microservices is effectively pushing monolithic architecture out of the market.
Microservices
Monolithic VS Microservices architecture

Monolithic architecture

Micro-service architecture

User interface

Data Access layer

Business layer

Database

micro-service

Database

micro-service

Database

micro-service

Database
Monolithic VS Microservices arch.

Discussion:

• What are the key differences between these two architectures?
• What could be the benefits of transforming from Monolithic to micro-services architecture?
• How can we perform such transformation?
Additional resources

• Monolithic architecture vs microservices
  https://www.divante.com/blog/monolithic-architecture-vs-microservices

• 4 Microservices Examples: Amazon, Netflix, Uber, and Etsy
  https://blog.dreamfactory.com/microservices-examples/

• Lessons Learned on Uber's Journey into Microservices (presentation)
  https://www.infoq.com/presentations/uber-darwin/

• The Great Migration: from Monolith to Service-Oriented (presentation)
  https://www.infoq.com/presentations/airbnb-soa-migration/
Thank You for your attention

Mohamad Gharib
mohamad.gharib@ut.ee

unitartu
tartuuniversity