Immordino Alessandro – 969549

Pala Riccardo – 969598

Polvanesi Giacomo – 971083

POLITECNICO
MILANO 1863

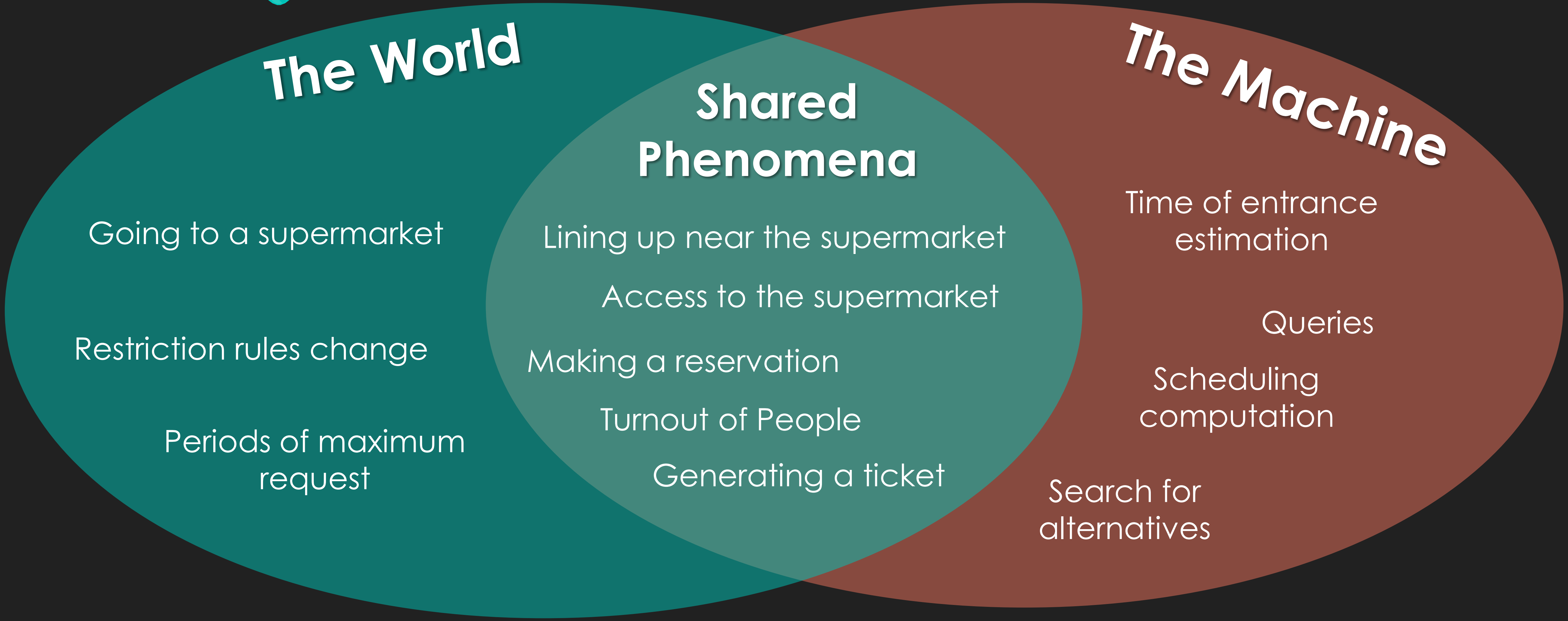
CLup

Project Discussion

RASD – Goals

- **Booking service** for grocery shopping.
- Easing the **access** to the supermarkets during **coronavirus emergency**.
- Prevent people forming crowds outside the supermarkets.
- Respect of **restrictions** about maximum accesses in a building.

RASD – World, Machine & Shared Phenomena



RASD – Features

- **Real-Time Reservation**:
  - Retrieve a ticket to virtually line up.
  - System provides the expected time of entrance.
- **Planned Reservation**:
  - Advanced service to book a visit.
  - Priority over a real-time reservation.
  - Enter the store at the time fixed by the reservation.

RASD – Main use cases

- **Real-time reservation via Ticket Generator**, a fallback option for those who do not have a smartphone.
- **Real-time reservation via app**, a simple way to virtually line-up from home.
- **Planned reservation with alternatives**, basing on the closest available timeslot or supermarket.

RASD – Most significant requirements

- A **ticket number** for each real-time reservation.
- Allow entrances only if **maximum capacity not reached**.
- Keep track of the number of **clients within the store**.
- Provide a client with **expected time of entrance**.

RASD – Most significant assumptions

EACH STORE HAS A **MAXIMUM CAPACITY**.

REAL-TIME RESERVATIONS: NO LINING UP UNTIL **EXPECTED TIME OF ENTRANCE**.

PLANNED RESERVATIONS: **NO NEED TO LINE-UP**.

RASD – Alloy model

1

Allow an entrance only if a user got a reservation.

2

Allow an entrance only if maximum capacity is not reached.

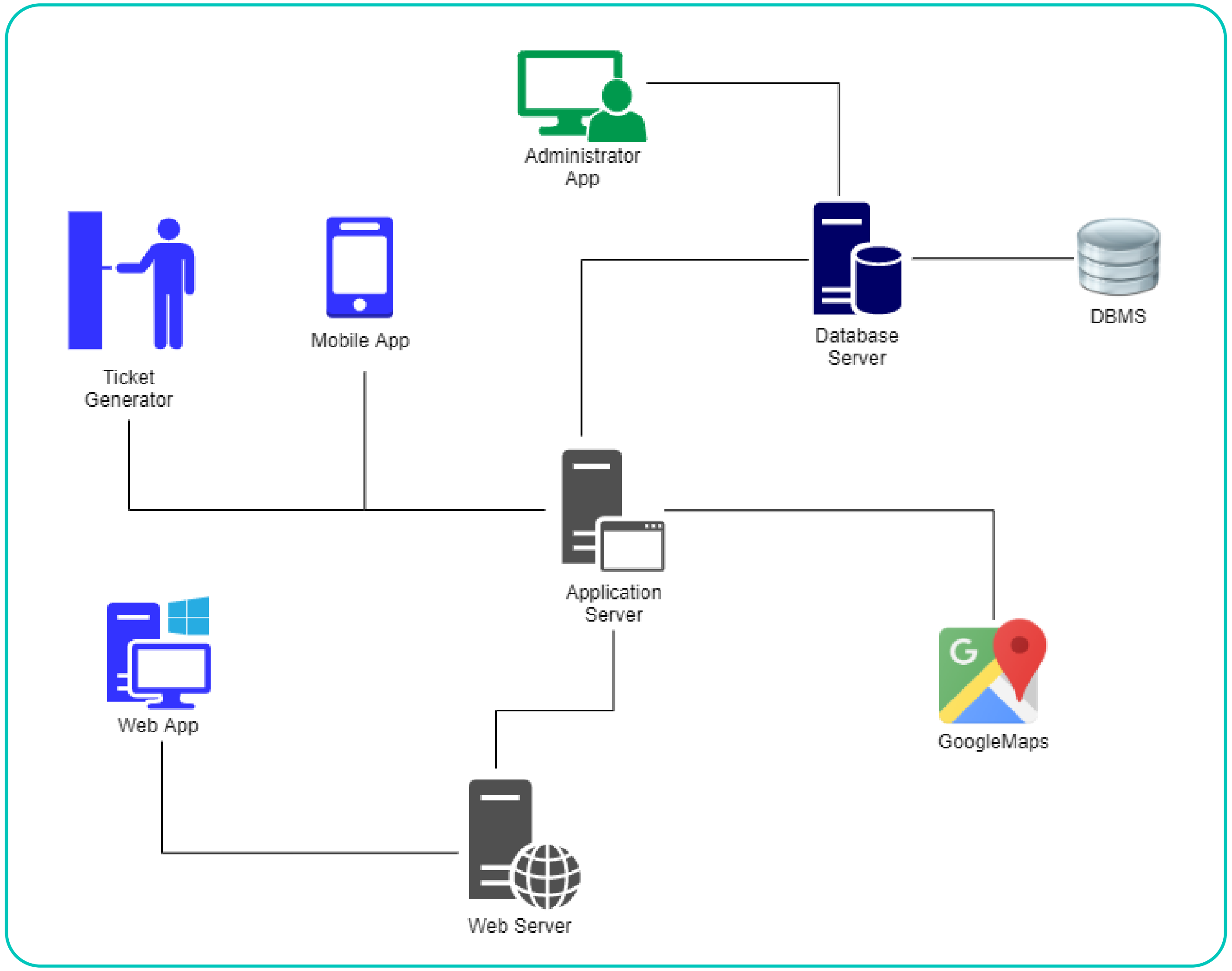
3

Maximise users inside in each moment to minimise the lined-up users.

DD – Architectural styles and paradigms

Three-layered **architecture**:

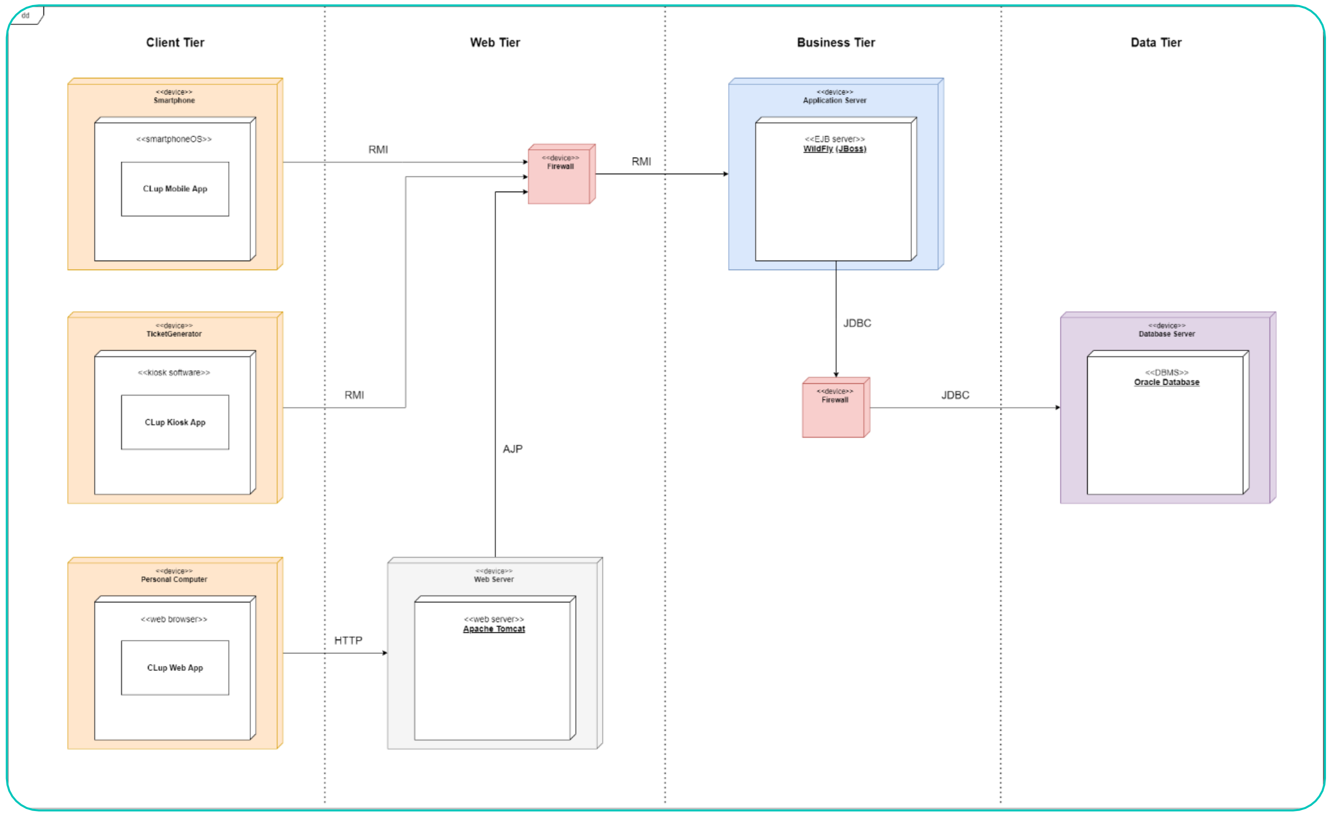
- Presentation layer
- Business or Application layer
- Data layer



DD – Architectural styles and paradigms

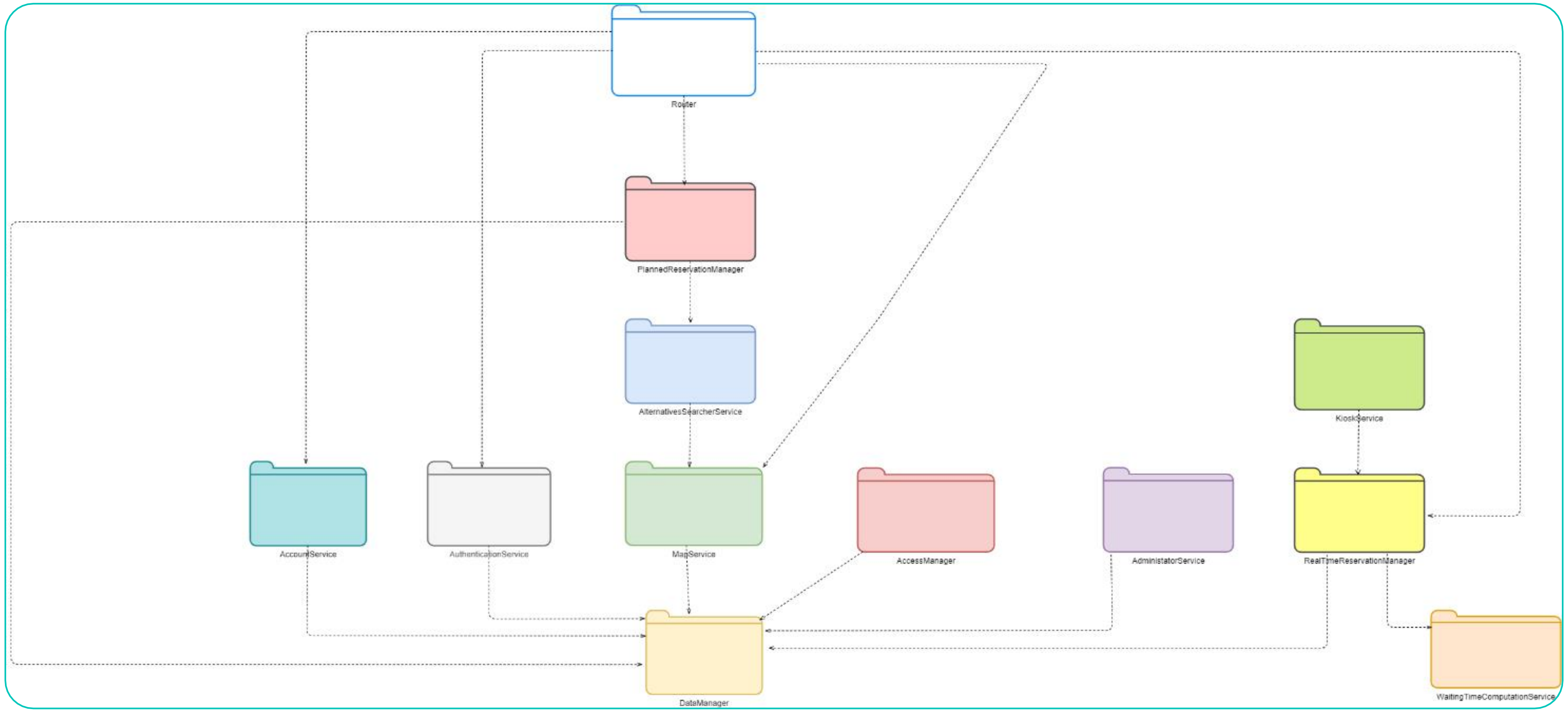
Architectural **pattern** based on 4 tiers:

- Client tier
- Web tier
- Business tier
- Data tier



DD – Components & Interfaces

Router («façade»)	**Reservation manager**	**Access manager**
Waiting time computation service	**Alternatives searcher service**	**Data manager**


Router
PlannedReservationManager
AlternativesSearcherService
KioskService
AccountService
AuthenticationService
MapService
AccessManager
AdministatorService
RealTimeReservationManager
DataManager
WaitingTimeComputationService

DD – Implementation strategy

Bottom-up approach to guarantee a proper interaction among components during implementation, carried out by means of **drivers**.

I&T – Adopted frameworks & APIs

Frameworks:

- **Java Enterprise Edition** (JEE)
    - **Enterprise Java Bean API** (EJB API)
    - **Java Persistence API** (JPA)
    - **Java Transactional API** (JTA)
- **JUnit** (for testing)

Useful APIs:

- **Thymeleaf** (HTML 5 template engine)
- **Zxing** (QR codes processing API)

I&T – Algorithmic solutions

COMPUTATION OF EXPECTED TIME OF ENTRANCE

- Recursive computation of all real-time reservations ETE
- Planned reservations are also considered

CHECK AVAILABILITY IN TIMETABLE

- Working hours are taken into account
- Planned reservations have a priority over real-time ones

I&T – Code structure

Two Java projects:

- CLupEJB
  - EJB Entities
  - EJB Services (Components)
  - Utilities
  - Exceptions
  - Tests
- CLupWEB
  - Controllers (Servlets)
  - HTML pages
  - CSSs

I&T – Test cases

Unit tests:

- Entities
- Services

Integration tests between different **services**:

- User + Database
- Supermarket + Database
- User + Supermarket + Database
- Router + User + Database
- Router + Supermarket + Database
- Router + Reservation + Supermarket + ETEComputation + Database