Integration principles

Application to Agility

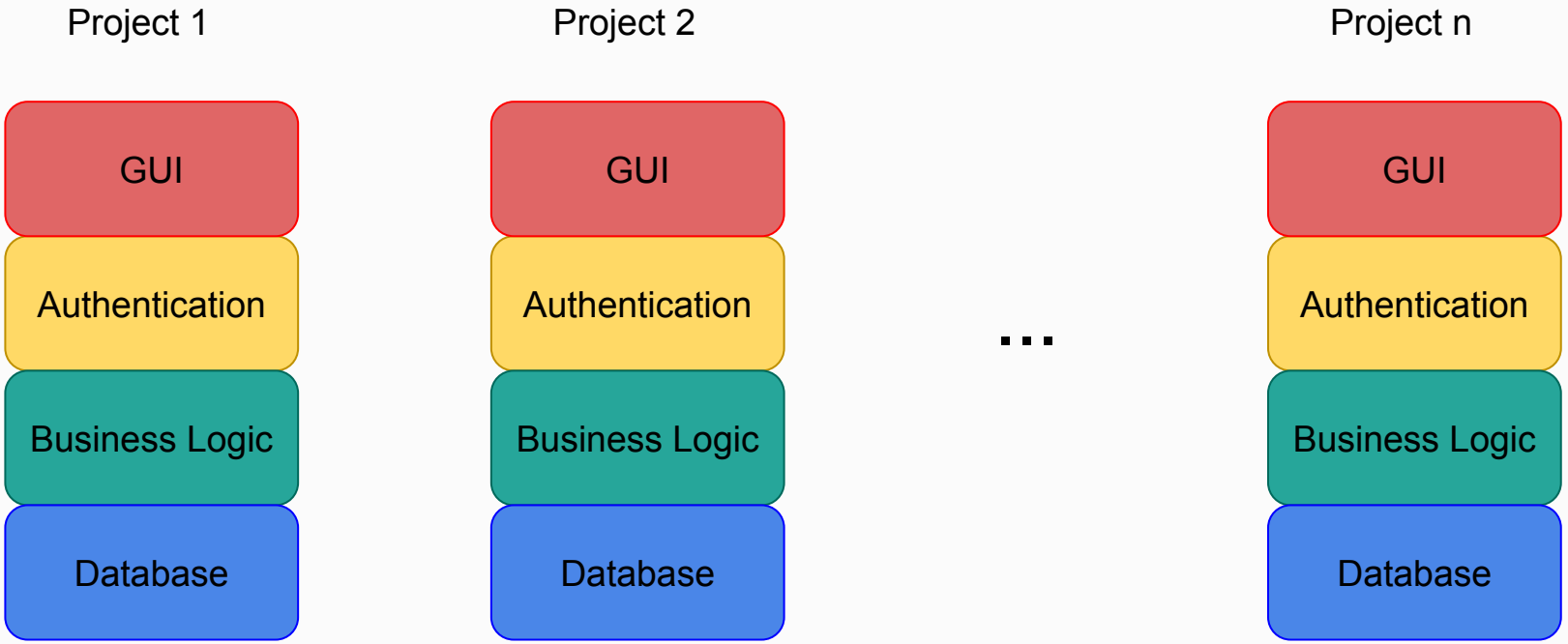
10/01/2017

Cécile Camillieri/Clément Duffau

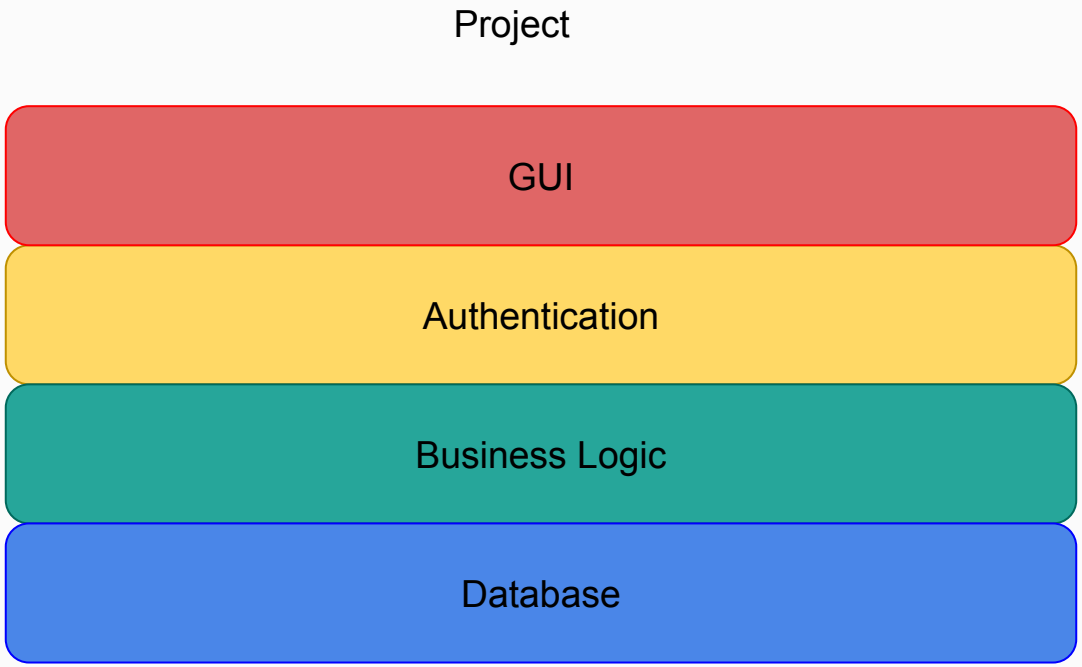
iut Nice Côte d'Azur

Université Nice Sophia Antipolis

State of play

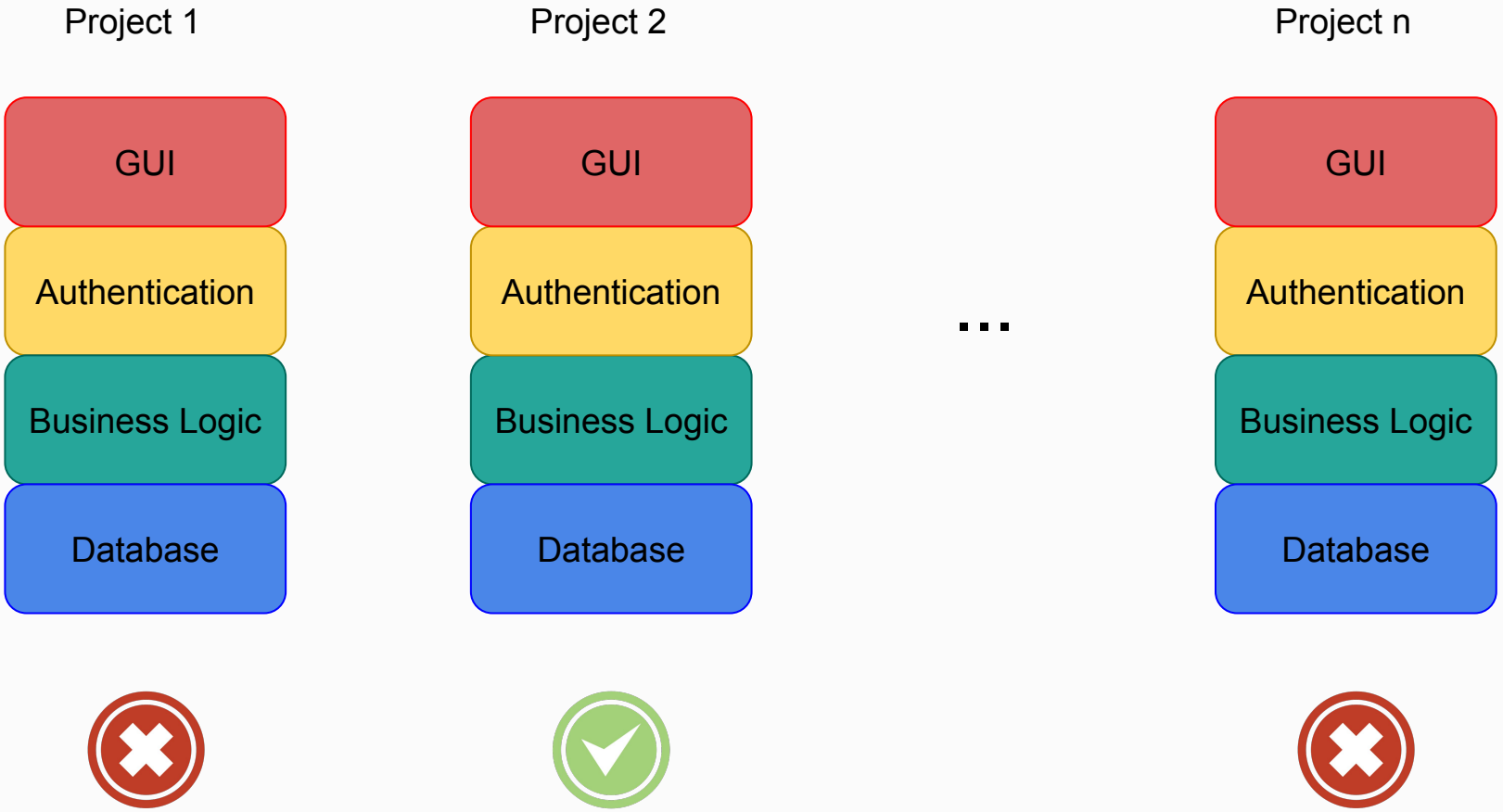


At the end

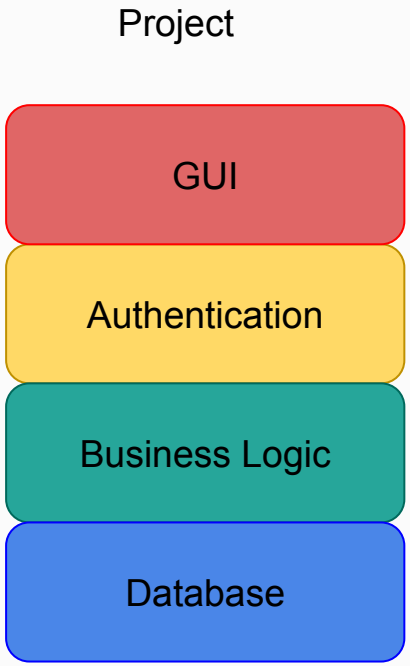


Different methods to do that

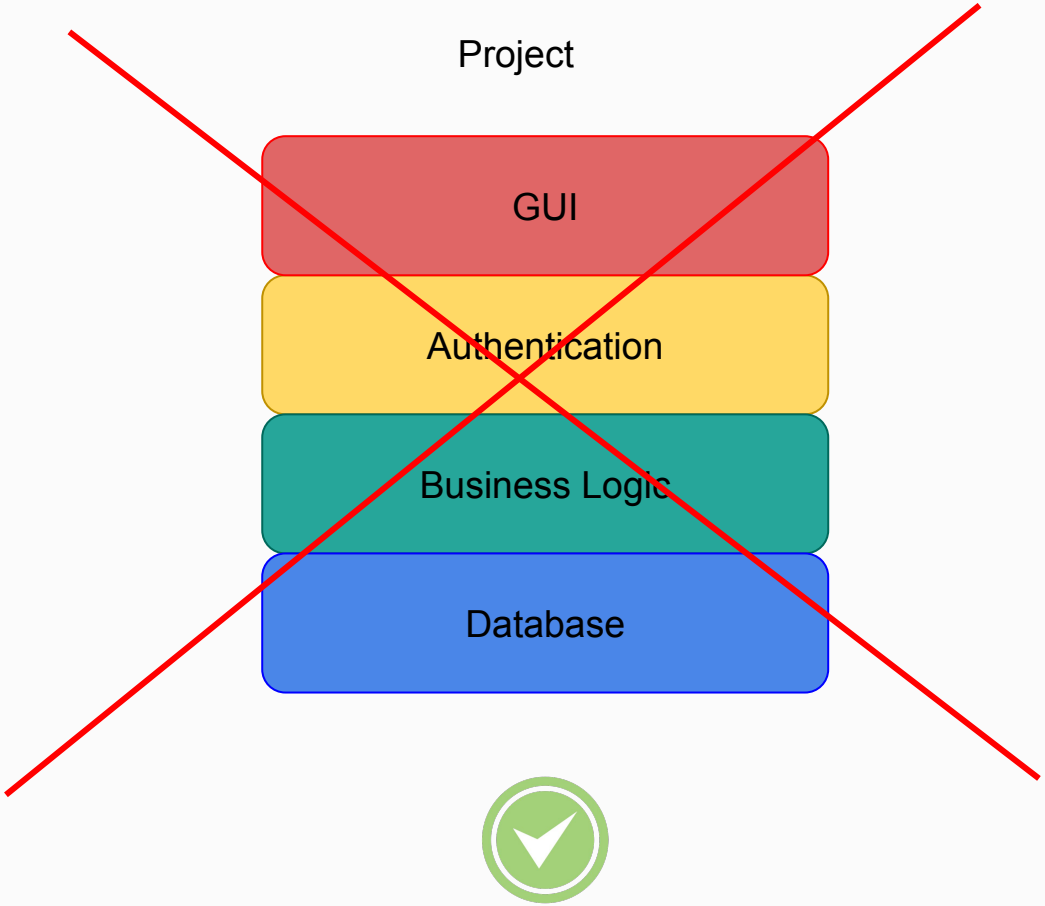
The “no way to integrate” method



The “no way to integrate” method



The “no way to integrate” method



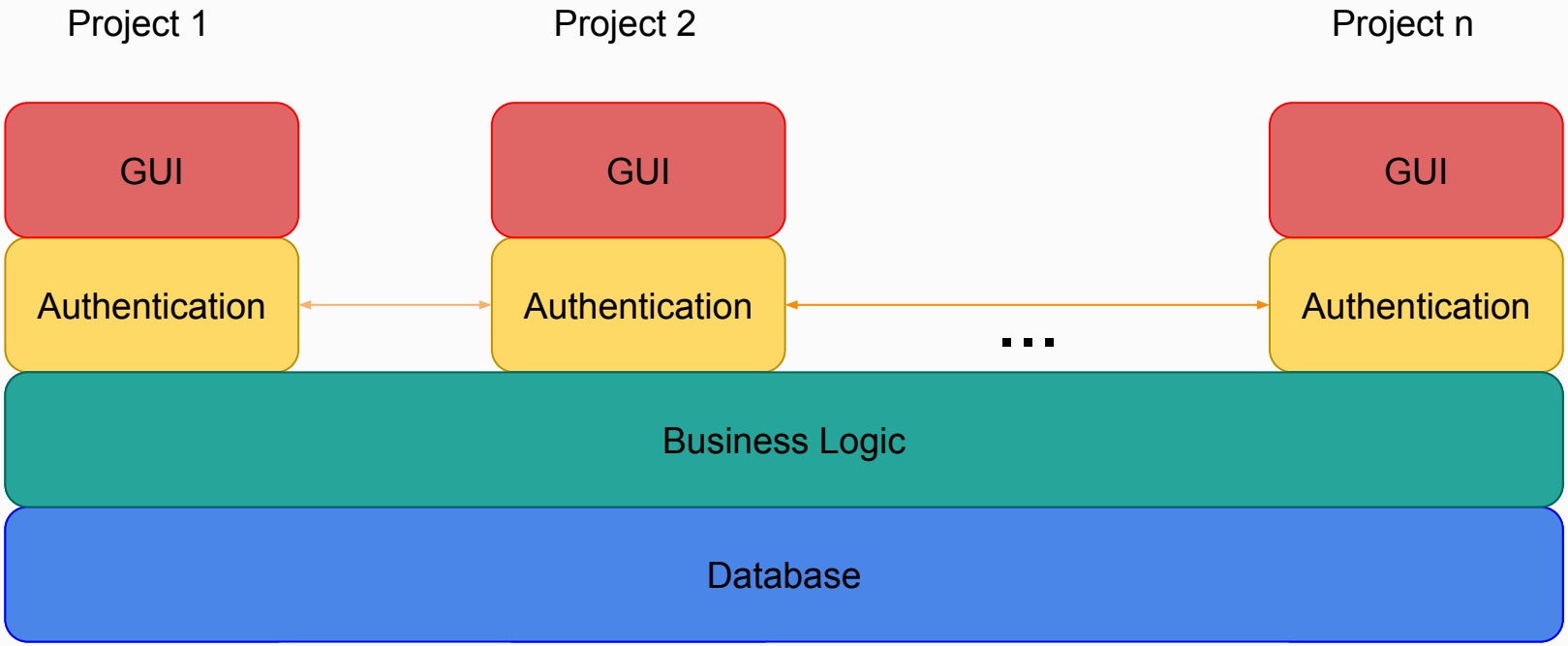
Costly

No reuse

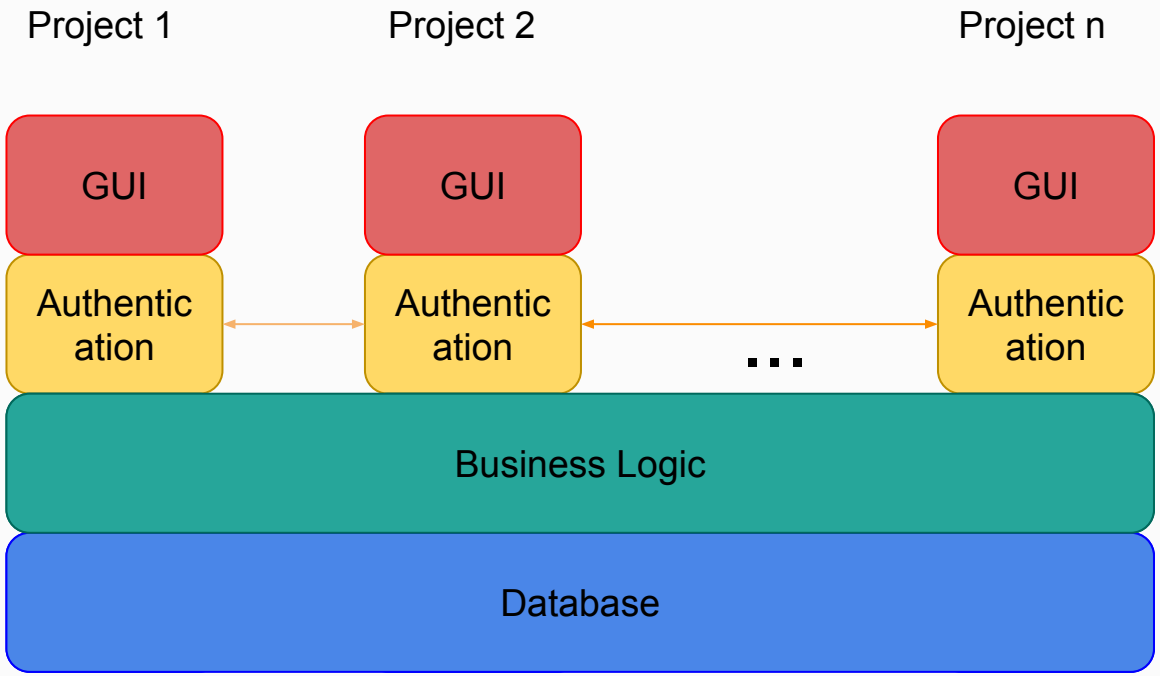
Rediscussion about the client needs

Not oriented continuous client value

The “client will have to wait” method



The “client have to wait” method



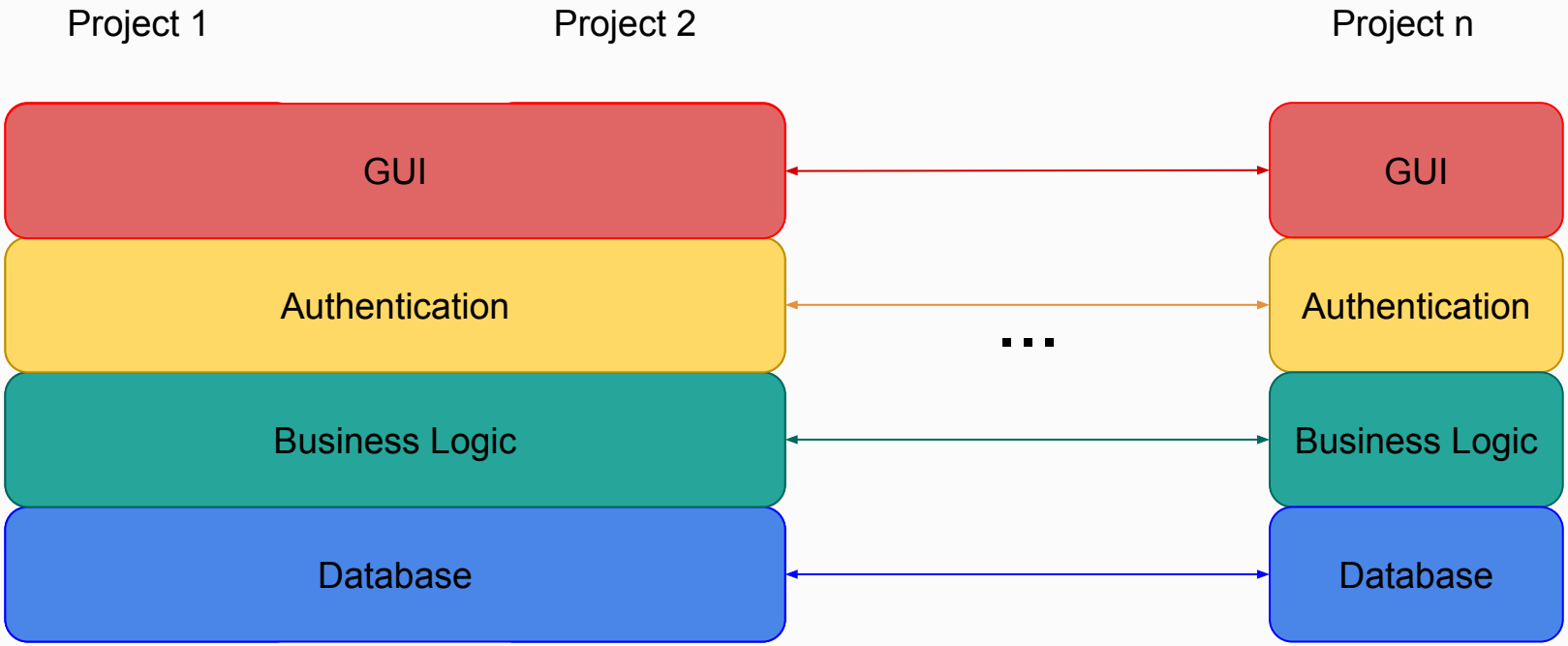
Very not oriented continuous client value

Risky integration

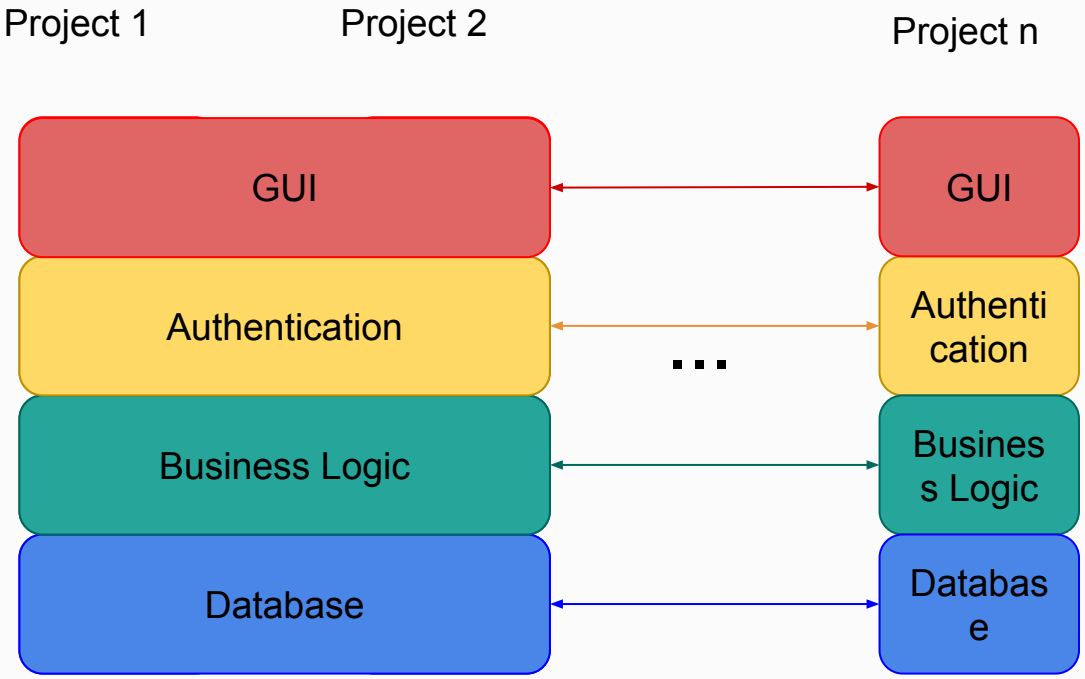
Costly

Reuse

The “step by step” method



The “step by step” method

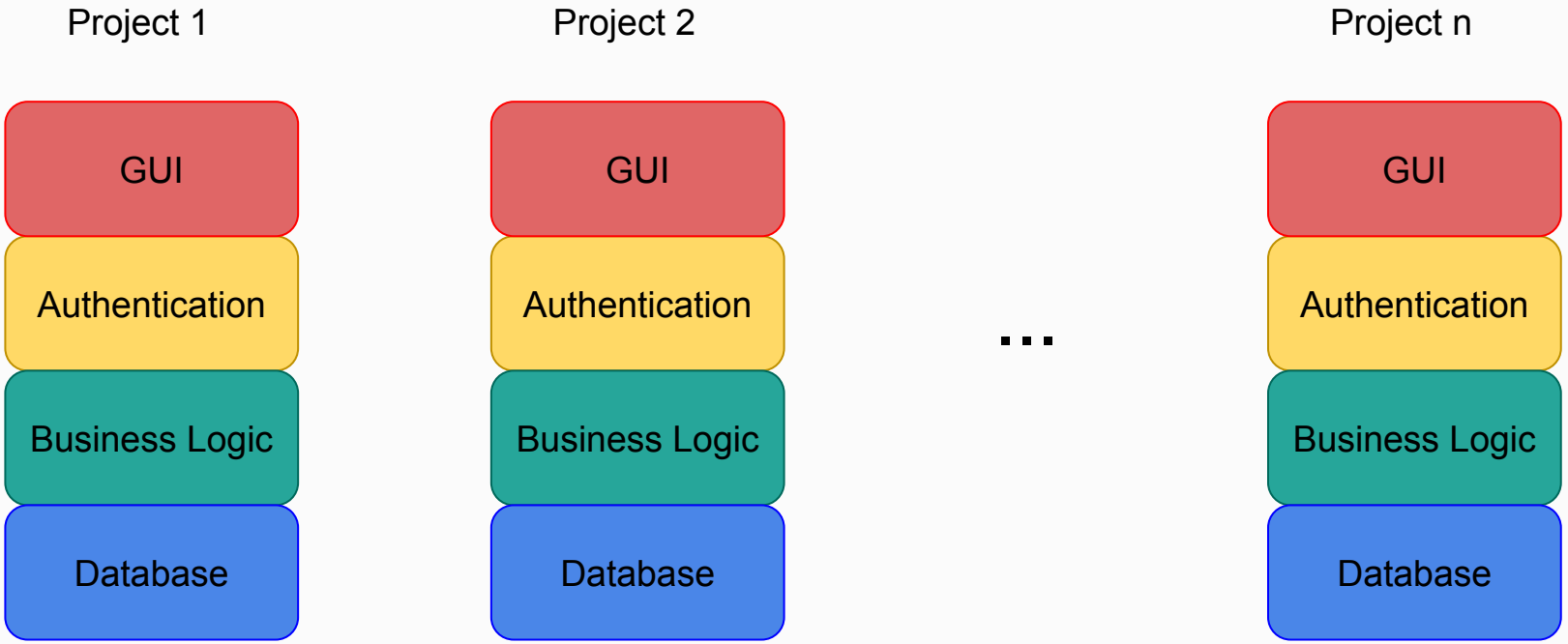


Oriented continuous client value

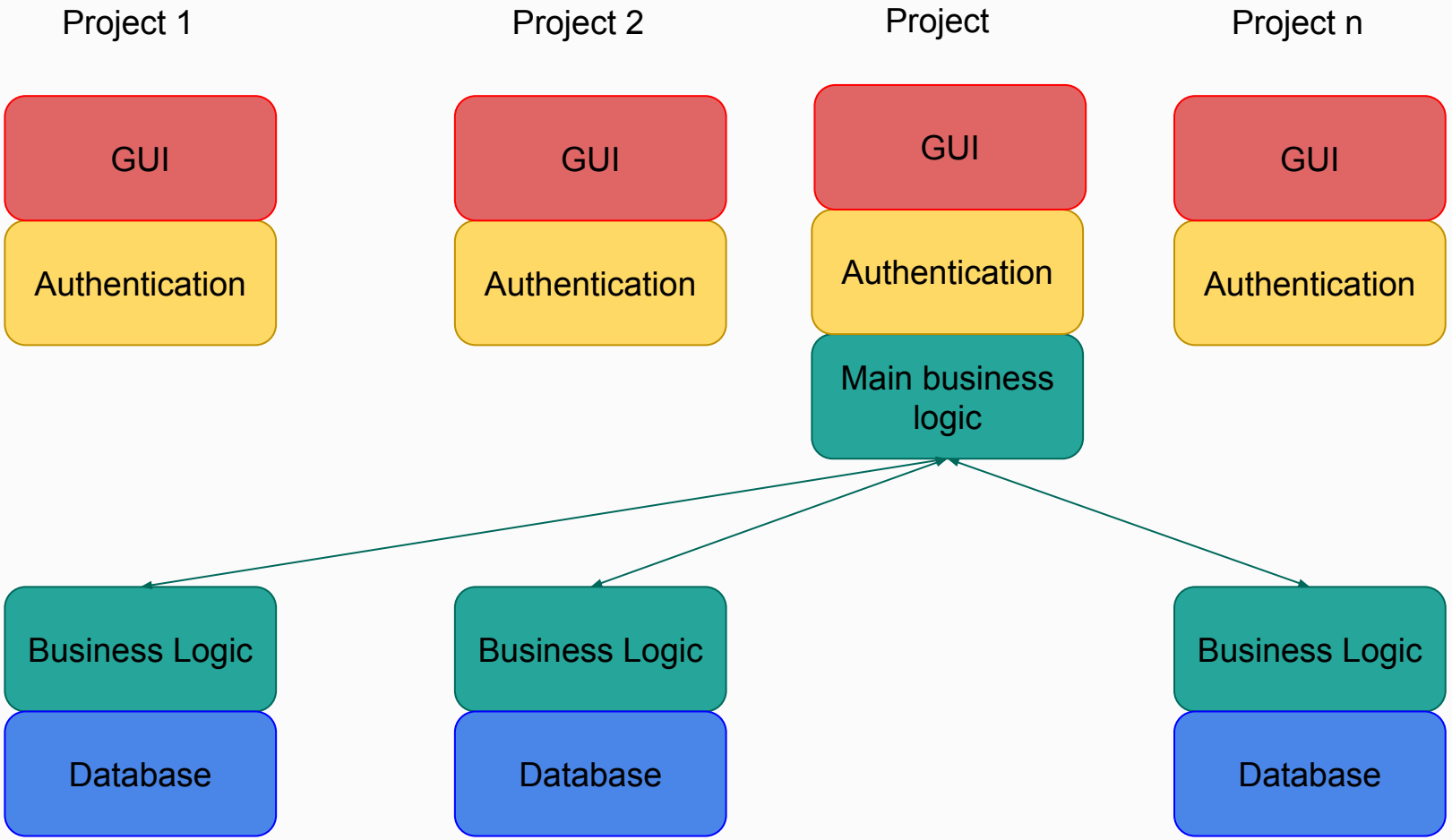
Risky integration

Reuse

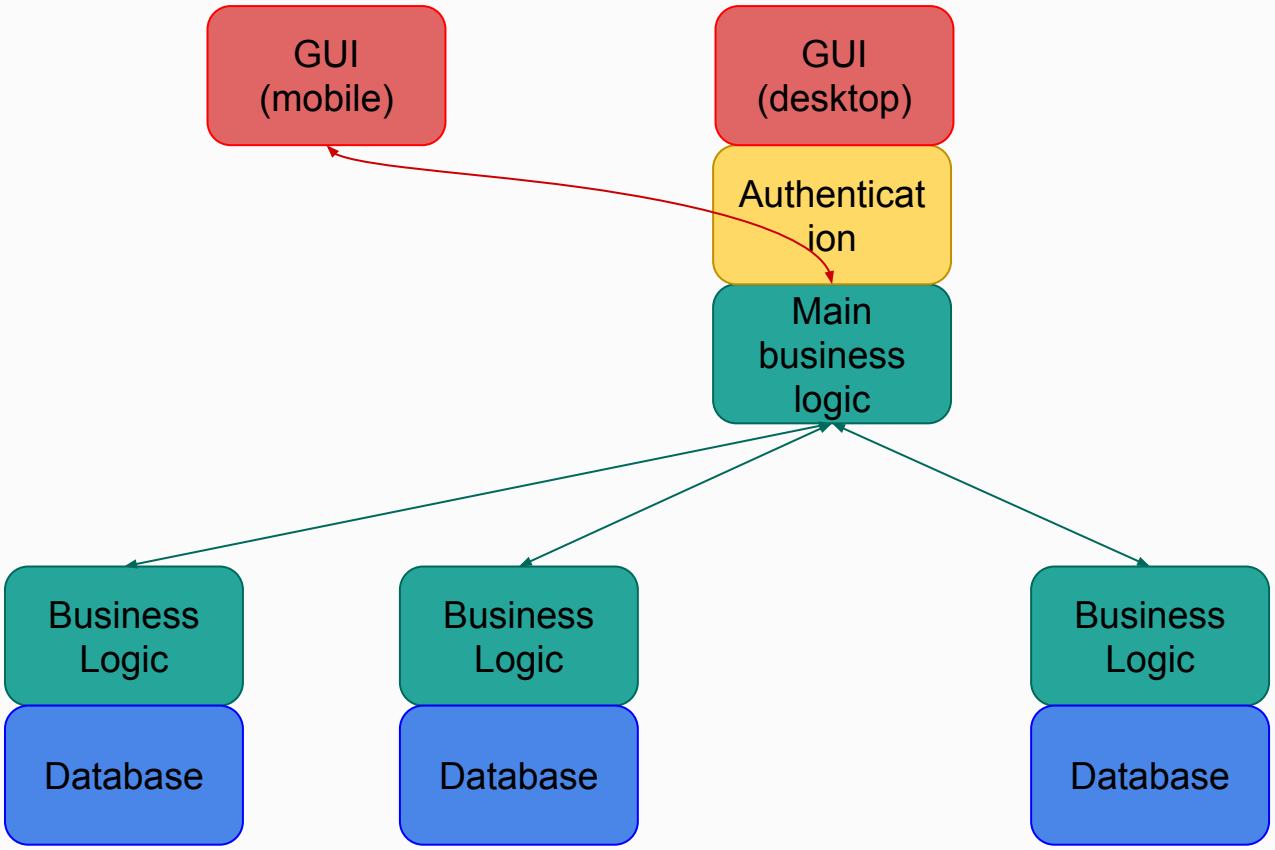
The “service oriented” method



The “service oriented” method



The “service oriented” method



Oriented continuous client value

Low risky integration

Reuse

Evolutionary

Example

```java
/**
 * Created by Jordan Dejoux on 22/12/2016.
 */
public interface JoueurService {

    @GET
    @Path("/{idPartie}")
    @Produces(MediaType.APPLICATION_JSON)
    Response getJoueurs(@PathParam("idPartie") int idPartie, @QueryParam("token") String token);

    @GET
    @Path("/chef/{idEquipe}")
    @Produces(MediaType.APPLICATION_JSON)
    Response getChefEquipe(@PathParam("idEquipe") int idEquipe, @QueryParam("token") String token);

    @POST
    @Path("/position/{idJoueur}")
    @Produces(MediaType.APPLICATION_JSON)
    Response setPositionJoueur(@PathParam("idJoueur") String idJoueur, @FormParam("lat") double lat, @FormParam("lng") double l
}
```

```java
/**
 * Interface pour un service web de gestion
 */
public interface AreaService {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    Response getAllAreas();

    @GET
    @Path("/capture/{areaId}")
    @Produces(MediaType.APPLICATION_JSON)
    Response getAreaGame(@PathParam("areaId") Integer areaId, @QueryParam("capturerId") String capturerId);

    @POST
    @Path("/capture/{areaId}")
    @Consumes(MediaType.APPLICATION_JSON)
    @Produces(MediaType.APPLICATION_JSON)
    Response captureArea(@PathParam("areaId") Integer areaId, @QueryParam("capturerId") String capturerId, String postParams);

}
```

Integration tips for your project

1. Discuss between you at
   a. conceptual level with UML
   b. code level with service interface

2. Don’t have to understand everything, just the interaction/integration point

3. You have designed the same main core, compare them. Choose the best or refactor one to be your good one

4. Don’t do everything at the same time
   a. Choose 2 services to integrate and work in pair
   b. Integrate step by step and focus on client value

Not a database course

- For the ones who have not legacy database, we give you the
  - database schema
  - mocked data

- Now, focus on what matters

- Here : https://github.com/LP-IDSE-16-17/authentication-ws/tree/master/db

(Thanks to Team A)

Organize your sprints

- Think vertically
  - No client value at the end of a sprint = not the good way

- Multiple way of splitting the work, choose yours

Front-end oriented	Back-end oriented
1. GUI for mobile app with main core features	1. Iteratively integrate service and maintain desktop app
2. Iteratively integrate services	2. Portability to mobile app
3. Develop your own feature	3. Develop your own feature