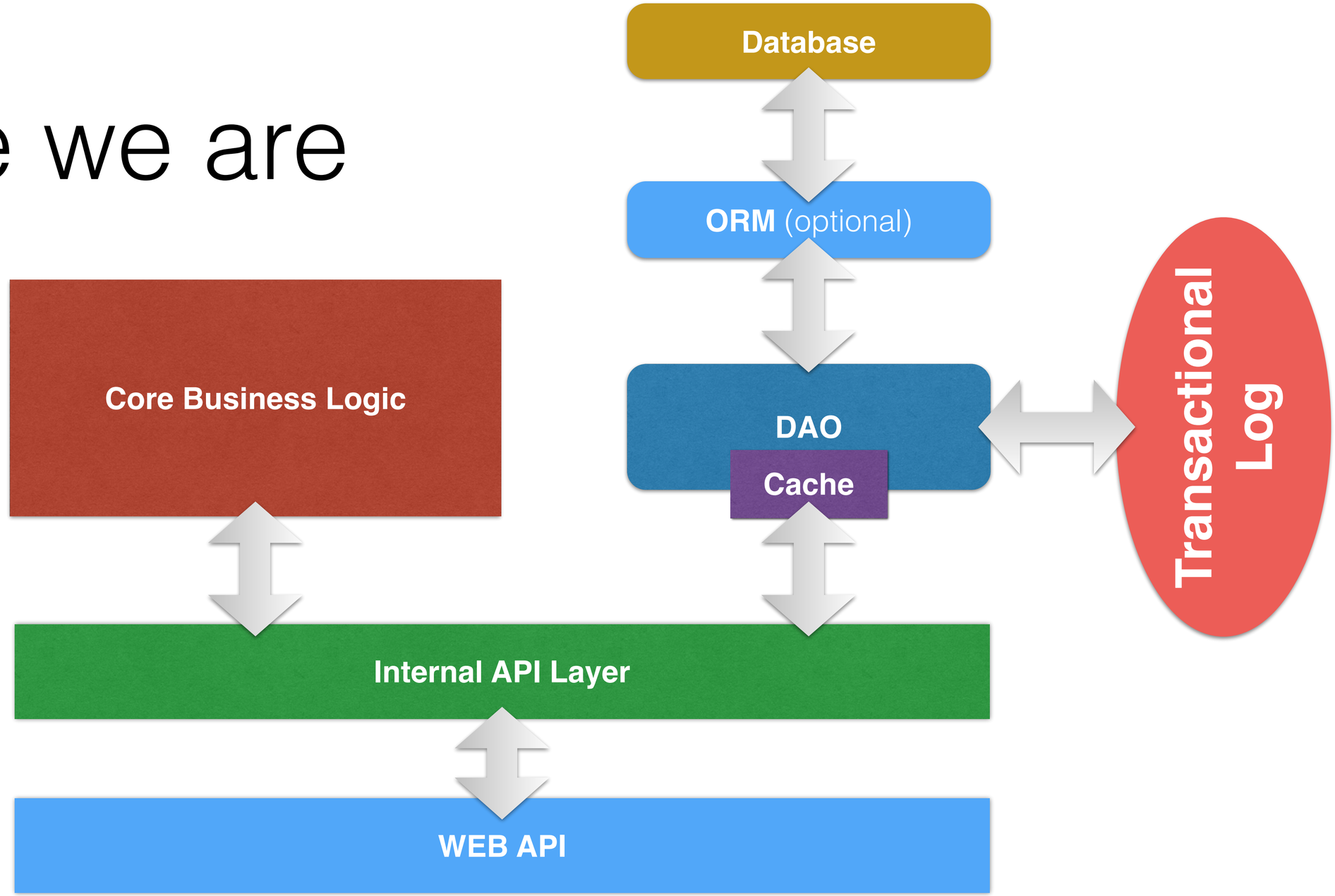
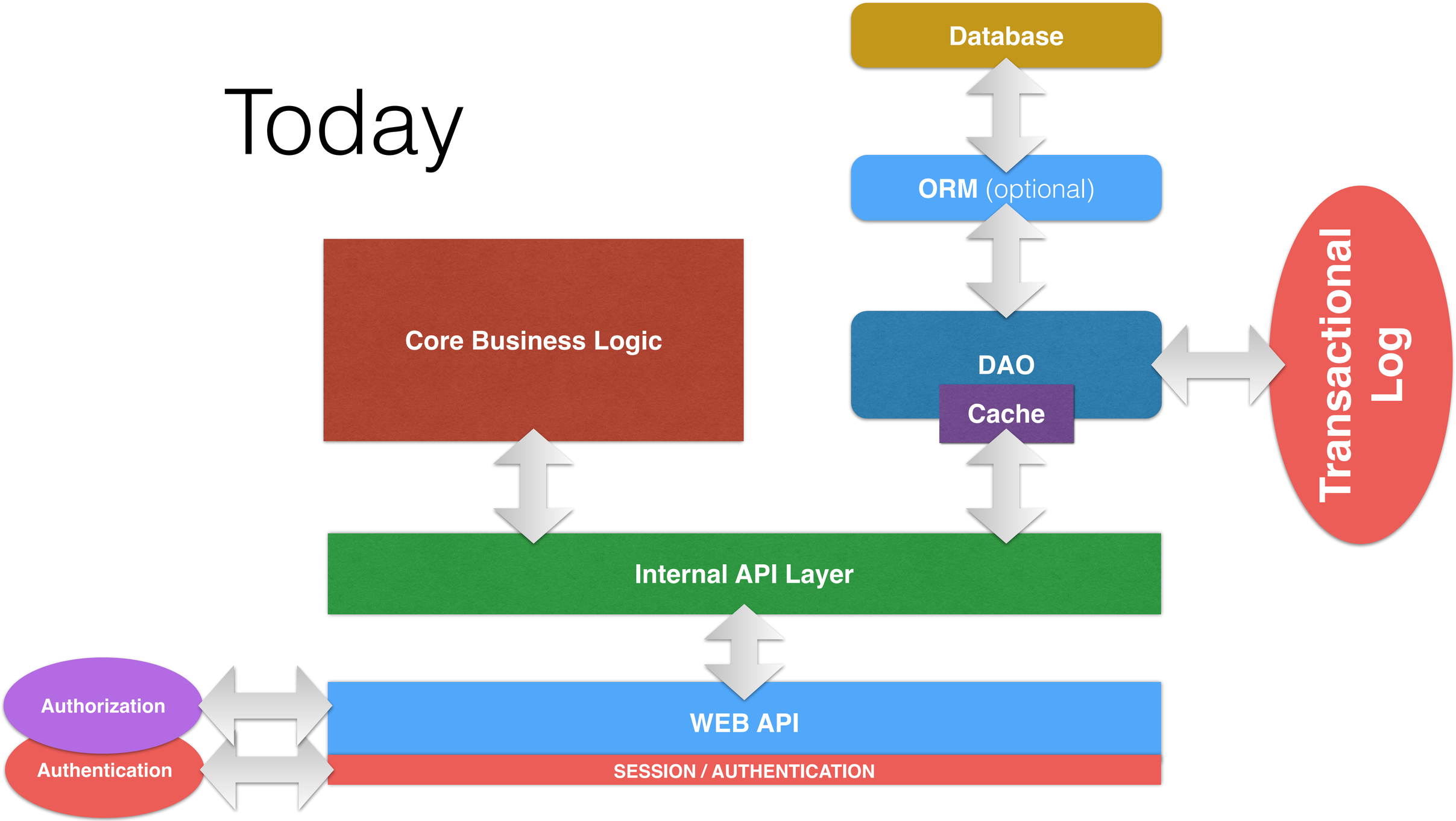
Lecture 5

CSCI 6907.12 - Full Stack Application Engineering

Where we are


Database
ORM (optional)
Core Business Logic
DAO
Cache
Transactional Log
Internal API Layer
WEB API

Today


Database
ORM (optional)
Core Business Logic
DAO
Cache
Transactional Log
Internal API Layer
Authorization
Authentication
WEB API
SESSION / AUTHENTICATION

HTTP

- HTTP is a session less protocol
- HTTP has a basic authentication but not useful

Authentication

- **Authentication** - act of confirming the truth of an attribute of a single piece of data (a datum) claimed true by an entity
- In more practical words:
  Are you who you say you are?

Authentication Architecture

- Monolithic
- Micro-service
- Decentralized aka Social Login
  - (OpenID, OAuth, Facebook Connect, etc.)

Monolithic

- Your application handles the authentication logic
- Keep a database of who is who and how they authenticate themselves
- **Pros**: You own how things are done
- **Cons**: Data liabilities and non-modular design

Micro-service

- Separate Authentication Service
- Same idea as monolithic but as a separate service apart from your main app
- **Pros**: You still own all the method and data
- **Cons**: Data liabilities and user has to learn your method of authentication

Decentralized

- Let some other entity provide authentication
- Similar to micro-service but owned by someone else
- **Pros**: no data liability, less work for you and your user
- **Cons**: your service is now dependent on someone else's product. No control over how things are done.

Authentication Methods

- Username and password
- Email verification
- Two factor authentication
- Smart card
- Biometric authentication
- And more!

Authorization

- **Authorization**: specifying access rights to a particular resources
- **Example**:
  - Accessing someone else's account
  - Paid services

Architecture

- Database of groups and its users
- A mapping of functionality to group of users

```
def authorized?(user, functionality)
def group(user)
def authorized?(group, functionality)
```