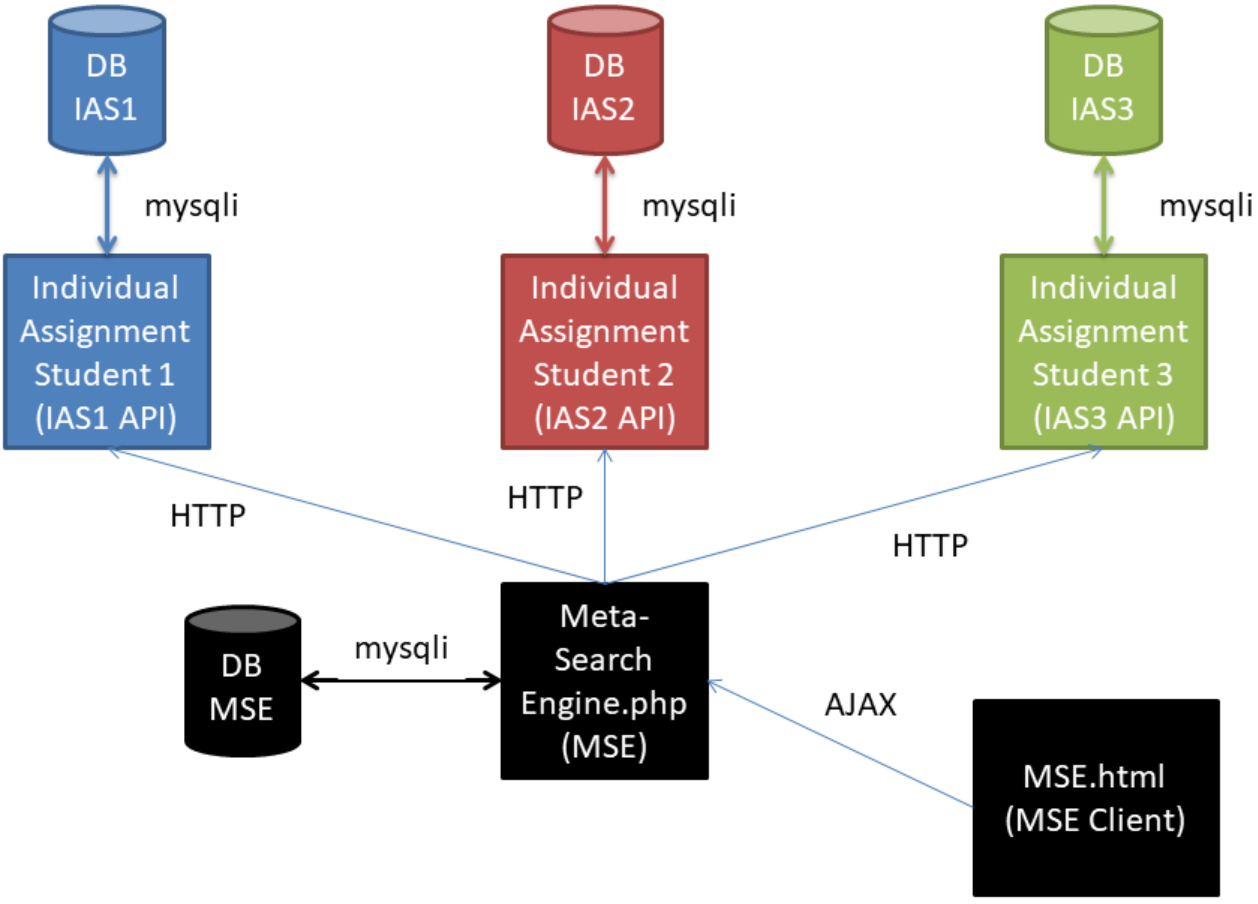
Group Assignment: (15% of marks)

Meta-search engine and e-commerce: Buy "anything" from a web.

The objective of this work is to build a PHP-based e-commerce platform supported by other PHP-based e-commerce platforms (Individual Assignment II). The platform will allow users to buy different types of products with different properties from a common interface (as amazon does). It will be done in groups of 2 or 3 and the meta-search engine will combine the individual assignments (from the members of the group) under a common search and e-commerce facility. As an example, you may think about the different shops inside amazon or the shopping search functionality from google.



The general idea for this last project is to create a fully interactive web-application that will get the data required through AJAX requests to the MSE server part (that will gather the data from other sources using HTTP requests), loading and inserting it without fully reloading the page. To do so you will need to extend your previous individual assignments adding some modifications.

Previous figure shows an outline of the architecture of the system that you will create for the assignment:

- Individual Assignments (IAS1, IAS2, IAS3): They will remain as they are, so the users are able to use them as described in previous practice (creating users, buying, etc). You will need to **extend** them, so they provide data in JSON format (to/from your group project), enabling the interaction with the Meta-Search Engine (MSE).
- Meta-Search Engine (MSE): the core of this project will be responsible of (1) generating an HTML+JS webpage able to request data through AJAX and (2) gather data from IAS (when requested from the HTML+JS) providing a uniform access point to all the IAs. It is important to remark that the MSE should be "fully interactive", you should reduce the number of reloads of the page to the minimum (it can be done only loading the main.php once and doing all the rest through AJAX).

For example, if the three members of the group have furniture, computers and sandwiches the resulting MSE will allow:

- To create an account in MSE (independent from those account existing on the IAs).
- To browse all the products available across all the IAS from a single place.
- To filter out the products following the individual criteria for each of the products from IAs.
- To buy any of the products from any of the IAs (without having to create a specific account in the IAs for that user).
- To review orders performed previously using the MSE account (only those from MSE account).

Main Requirements:

1. **Meta-Search Engine (5 points):** This is the core of the project and will allow the user to:
   - **1.1. Perform Searches (1 point):** The user will be able to search using some words and only products matching the "query" will be shown (from any of the IAs). The search should be triggered through AJAX, without fully reloading the page.
   - **1.2. Filter Results (1 point):** After retrieving a list of products, the user should be able to filter out the list based on the different details of the products (some of them will be common to all like the price and some others will be specific for some type of products). This filtering will be done without reloading the page! Applying it from the JS and hiding the elements out of the search.
   - **1.3. Add products to cart (1 point):** When users click on any of the products it will go to a detailed view where the details of the product will be shown, and the user will have the chance to buy "any" quantity of them. This will ask the MSE (through AJAX) to communicate to the IAs (HTTP) and reduce the stock of those products in the number of units added to the cart. If everything goes ok (the IAs is able to reserve the stock) the product will be added to the users shopping cart.
   - **1.4. Buy products (1 point):** Once the user has finished adding products, he will be able to buy them, "making the payment". This will result in a new order being stored for that specific user (with all the products bought). You need to find a good way of storing that order so it can be later retrieved (it includes products from different IAs and thus different DBs)
   - **1.5. Browse orders (1 point):** the user will be able to browse the orders that has performed in the past. This should include the date, the price of each of the items and the name/description of those items.

2. **User management (2 point):**
   - **2.1. Create new users (0,5 point):** Users will input mail and password into a form to create a new user. Then a confirmation mail will be generated and sent to the user that will need to click on the link provided to activate the account (and be able to use it). Do not allow the same mail registering twice. This user will be only valid to the meta-search platform and will be stored into its own database (does not allow you to login on the individual assignment sites)
   - **2.2. Login/Logout (0,5 point):** The user will be able to login/logout to the platform**,** enabling him to browse their previous orders or buy new ones. All pages will redirect to the login if the user try to access them without login.
   - **2.3. Authenticate to Individual Assignments (1 point):** We want to store all the orders from each user in the Individual Assignments' databases, so the meta-search engine should be able to manage (create

or delete) new orders on the name of the user. To do so you will need to create an internal user account on the Individual Assignments that will be used by your metasearch engine to authenticate itself and perform actions. Also, you will need to distinguish which MSE user has done each action on the IAs so you may need extra fields or a different approach.

3. **Individual Assignment Modifications (3 points):** The individual Assignments can generate complete HTML pages with the content requested embedded (as a list of products, previous orders…). However, this time we want only the data (as a JSON) to be used by our metasearch engine. Therefore, you will need new scripts to return the data as a JSON instead of returning complete HTML pages (consider that those new script are like the old ones, create copies and modify when needed). Among other you will need to:
   - **Request search results:** this script will result in a JSON containing a list of products for that Individual Assignment site. You need to agree on a format before doing this, so the data can be interpreted when used from MSE.
   - **Request to reduce stock:** this script will allow to add products on the shopping cart, reserving them for a particular user.
   - **Request to buy the product:** this script will make the reserve of a product into a real order that has been paid.
   - **Request to show orders:** this script will gather all the orders for a specific user and return them as a JSON.
   - Other requests: those are only the more prominent examples… all the interaction will be done through AJAX so you will eventually need more requests

Additional Requirements:

- Everything that is not totally clear in the specification should behave as close to a real application as possible.
- There is no need for CSS styles (or a little to show the tables).
- The project needs to be presented **properly**, otherwise you will not get all the marks for each of the features not explained properly, so this time you need to spend real time on the presentation. The presentation will behave as a multiplier from 0,5 to 1 (so **a project that is perfect and is not explained properly can have the grade dropped from 10 up to a 5**). If you need further instructions about how to produce a good presentation, ask the teacher.
- You need to make the project work into your **production server** (not only in localhost), so don't wait to the last day to do the testing.
- The site should work properly for **concurrent users**! So pay attention when locking and buying products.
- The site should work simultaneously with the IAs, so don't broke anything from the previous project while building this one! And take this into account when doing the reserve/paying/releasing procedure.
- **Security** in this application is a must:
    - All the data coming from users should be properly treated in regards with security, paying special attention to SQL Injections.
    - All the passwords should be stored hashed and salted into the db, nobody should have access to any of the passwords (like in a real system).
    - Access should be properly controlled in all the scripts that are part of the e-commerce platform.

Extra tips:

- MSE does not access the IAs Databases directly. It will delegate that interaction to the new scripts you will create to extend IAs.
- MSE database only needs to store the user access information; and some information about orders, the rest is stored in each of the IAs databases.
- Design before developing. You should have some diagrams before trying to code anything.

Assessment Criteria:

- The platform follows the requirements (main and additional ones) and behaves as expected.
- Quality and structure of the code. Include proper documentation of the code (comments and naming of functions and variables).
- Presentation of the platform, quality of the answers to questions.
- The division into points is indicative and may vary if we (students and teacher) agree on that.