N-Tier (Multi Tier) Software Architecture

MVC — 1-tier

A program → distributed among N separate computers in a distributed network.

1-tier
2
3 ++ (Most Common)

- user interface in client's computer.
- Business logic in a centralized computer.
- Required data in a computer that manages DB.

MVC – Design Pattern
↓ ↓ ↓
"Layer"

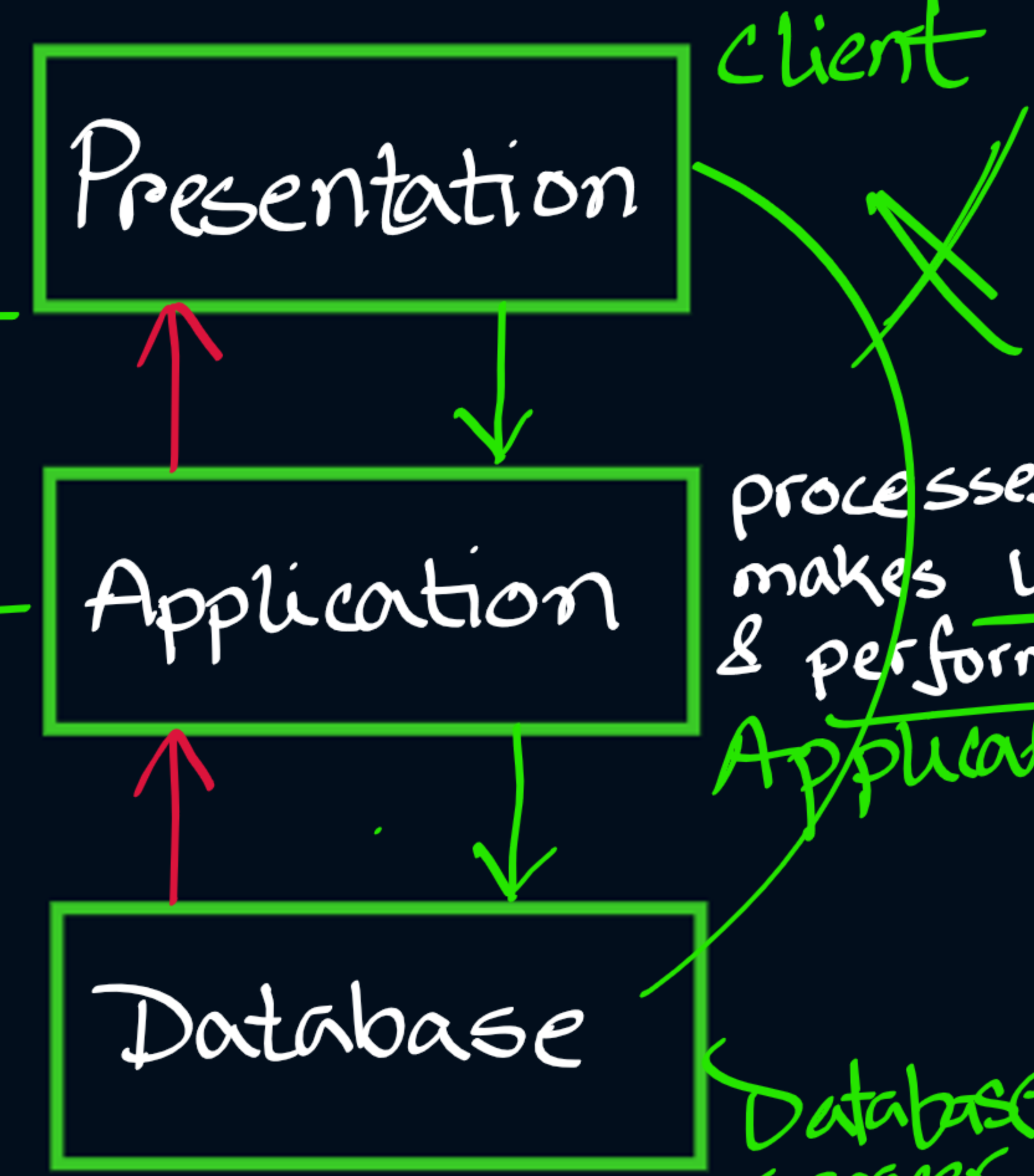
- Physical Separation — N-Tier
- Logical Separation — MVC — Programming Design Pattern

Key-Terms

- Distributed Network
- Client-Server Architecture — Request, Response
- Platform
- Database

~~3ta layer~~

Translates tasks and results to something user can understand

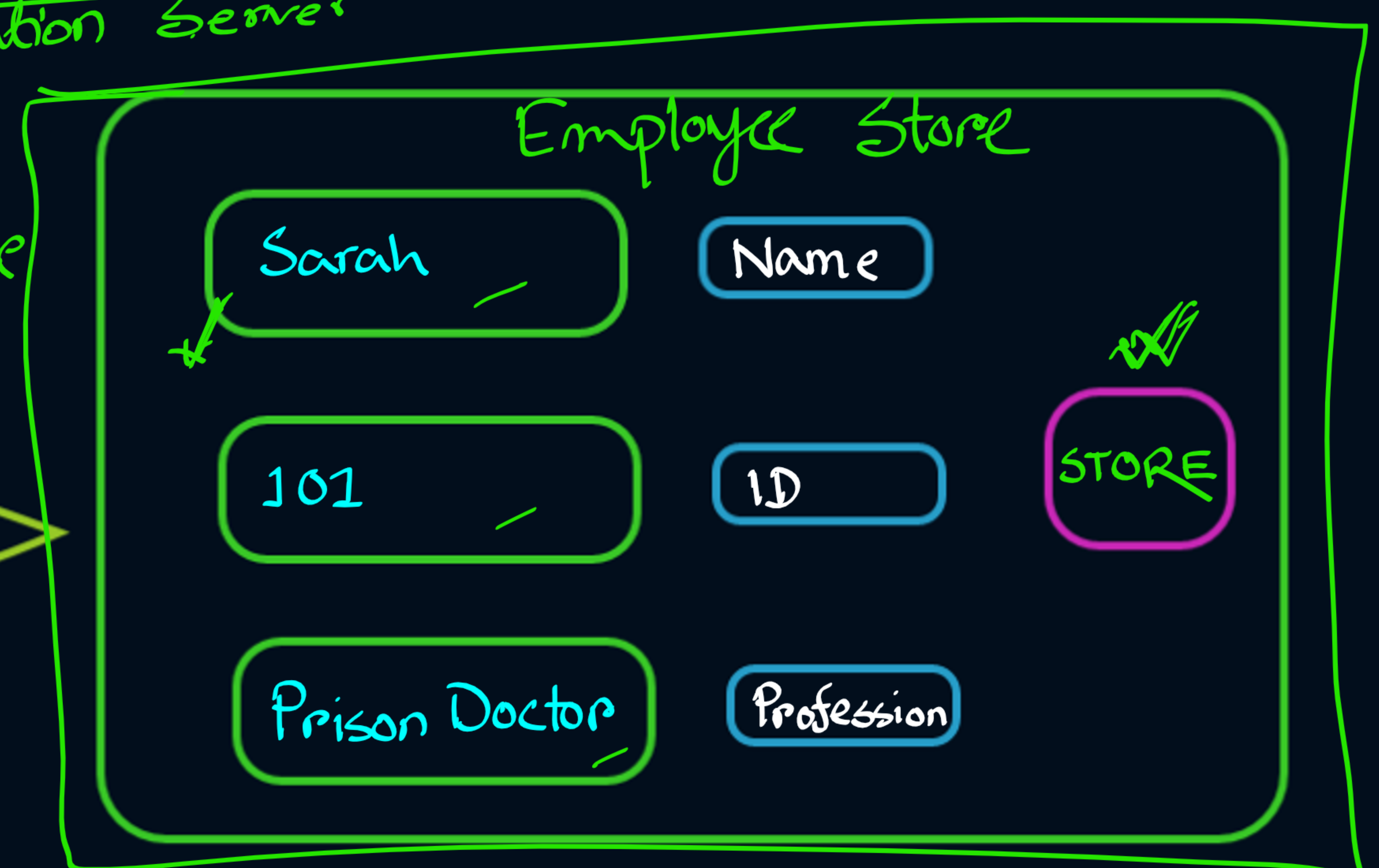


Business Logic Tier

processes commands, makes logical decisions & perform calculations
Application Server

information is stored in an organized way for easy access, manage & update.

Database Server



Presentation →

Application → Creating an instance of Employee class.
employee.name = name_field.value.toString()

Database → Running some SQL or NoSQL query to Store these information.

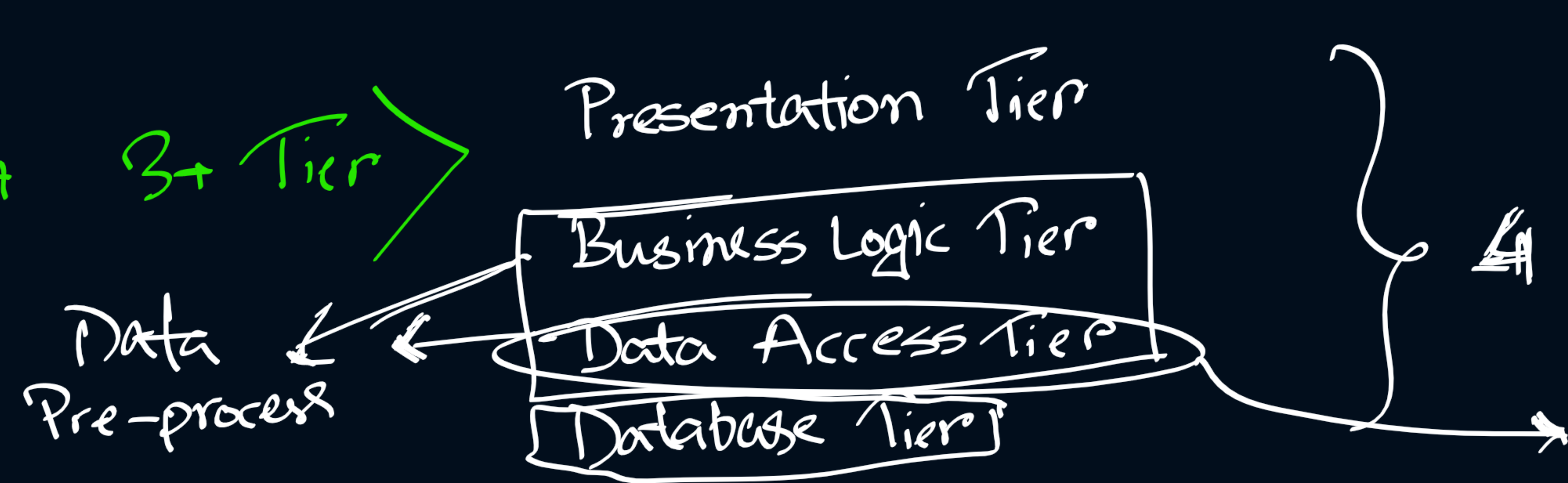
2-Tier Architecture
- client
- Database Server

1-Tier Architecture
- X Network Activity

Standalone Software

3 ↑

4 — 3+ Tier

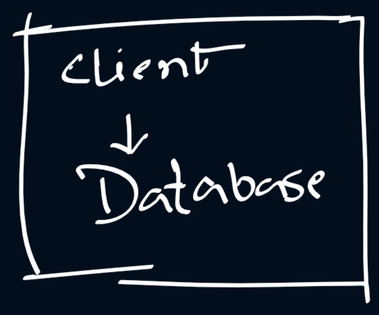


Scalability X
Security X
Fault Tolerance X
Reusability
Maintainability
Data Integrity

Advantages and Disadvantages

1 Tier – 2 Tiers

Simple
fast for a lower number of user



low cost for hardware, maintenance, deployment

Disadvantage

Client ⟶ Database

2

Advantage of Multi Tier

Scalable

database clustering

presentation → group → similar

Application

Client

load balancer

Application

Database

Security

Client →

Business logic

Database

Disadvantage

Higher Bandwidth

Cost

MVC Explained with Code

Controller

function

```
Route::get('/mo_list', 'MathOlympiadController@index')->name('mo_list')->middleware('auth');
```

Route

Database - call

Model

Table define

ictfest2019.com/mo_list

Controller

database

```
public function index()
{
    $mos = MathOlympiad::all();
    return view('math_olympiad/list')->with('mos', $mos);
}
```

MathOlympiadController

mathOlympiad

list.html

php

view

```
<tbody>
    @foreach ($mos as $mo)
    <tr>
      <th>{{$mo->id}}</th>
      <th>{{$mo->name}}</th>
      <th>{{$mo->category}}</th>
      <th>{{$mo->contact}}</th>
      <th>{{$mo->email}}</th>
      <th>{{$mo->institution}}</th>
      <th>{{$mo->total}}</th>
      <th>{{$mo->tshirt}}</th>
      <th>{{$mo->created_at}}</th>
      <?php

      $paid = $mo->total - ($mo->paid);
      if($paid==0)
      {
          echo '<th><font style="color:green">'.'Successful'.'</font></th>';
      }
      else
      {
        echo '<th><font style="color:red">'.'Pending'.'</font></th>';
      }
      ?>
      <?php

      $sel = $mo->selected;
      if($sel=='True')
      {
          echo '<th><font style="color:green">'.'True'.'</font></th>';
      }
      else
      {
        echo '<th><font style="color:red">'.'False'.'</font></th>';
      }
      ?>
a href="/payment_done_mo/{{$mo->id}}" class="glyphicon glyphicon-euro" onclick="return confirm('Are you sure about completing the payment? This cannot be undone.' );"></a></t
a href="/selection_done_mo/{{$mo->id}}" class="glyphicon glyphicon-plus" onclick="return confirm('Are you sure about selecting this participant? This cannot be undone.' );"><
a href="/delete_mo/{{$mo->id}}" class="glyphicon glyphicon-trash" onclick="return confirm('Are you sure about deleting this? This cannot be undone.' );"></a></th>

    </tr>
  @endforeach
</tbody>
```