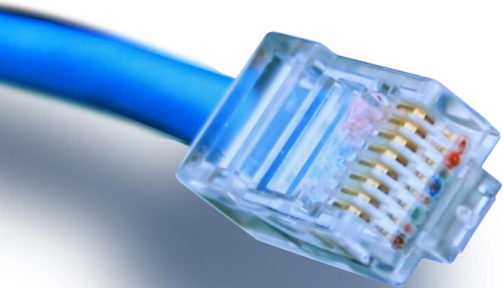


HOUSE OF  
TECHNOLOGY



- en del af **mercantec**<sup>+</sup>

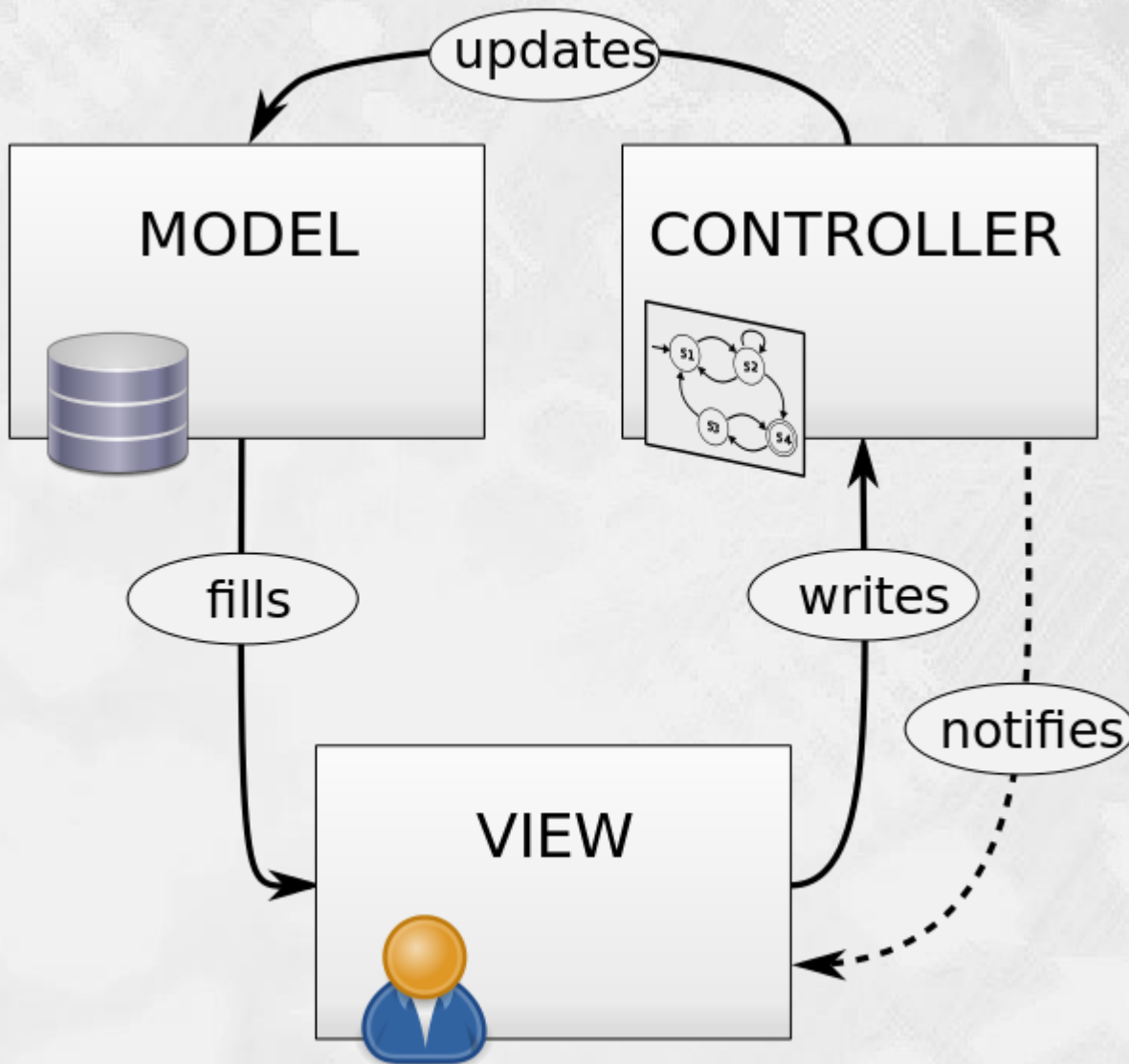


# Model View Control

PHP Object orienteret

**Webintegrator HF1**

# Model View Control



Model View Control er et hyppigt anvendt mønster i programmering. Modellen består af tre dele.

Modellen som indeholder vores data.

Controlleren der opdaterer vores data.

View som styrer hvordan data vises til brugeren.

```
class PersonModel
{
    var $name;
    var $birthDate;
    var $email;
    var $id;

    function person($name, $birthDate, $email,$id)
    {
        $this->name = $name;
        $this->birthDate = $birthDate;
        $this->email = $email;
        $this->id = $id;
    }

    function toJson()
    {
        $arr = array();
        $arr['name'] = $this->name;
        $arr['birthdate'] = $this->birthDate;
        $arr['email'] = $this->email;
        $arr['id'] = $this->id;
        return json_encode($arr);
    }
}
```

I dette tilfælde er Model vores Class person som blot indeholder klassens attributter, en constructor og funktionen toJson.

```
class PersonCtrl
{
  private $connection;
  function PersonCtrl()
  {
    $this->connection = new dbHandle();
  }
  function getPerson($id)
  {
    $sql = "SELECT * FROM [user] WHERE Id='$id'";
    $result = $this->connection->executequery($sql)[0];
    return new PersonModel($result['username'],$result['birthday'],$result['email'],$result['id']);
  }
  function updatePerson($person)
  {
    $sql = "UPDATE [user] SET username='$person->name',birthday='$person->birthDate',email='$person->email' WHERE id=$person->id";
    $this->connection->executequery($sql);
  }
  function deletePerson($id)
  {
    $sql = "DELETE FROM [user] WHERE id=$id";
    $this->connection->executequery($sql);
  }
  function createPerson($person)
  {
    $sql = "INSERT INTO [user] (username,birthday,email) VALUES ('$person->name','$person->birthDate','$person->email')";
    $this->connection->executequery($sql);
  }
}
```

Controlleren indeholder de metoder som skal til for at opdatere Person i databasen.  
Get, Update, Delete og Create.

```
class PersonView
{
  function updateForm($person)
  {
    echo "
      <form action="" method='post'>
        Name<input type='text' name='name' value='$person->name'>
        Email<input type='text' name='email' value='$person->email'>
        BirthDay<input type='text' name='birthdate' value='$person->birthDate'>
        <input type='hidden' name='id' value='$person->id'>
        <input type='submit' name='update' value='Update'>
      </form>
    ";
  }
  function showTable($persons)
  {
    print("<FORM method='post'><table>");
    print("<tr><th>Navn</th><th>Fødselsdag</th><th>E mail</th></tr>");
    foreach ($persons as $person) {
      print("<tr>");
      print("<td>$person->name</td>");
      print("<td>$person->birthDate</td>");
      print("<td>$person->email</td>");
      print("<td><button type='submit' name='edit' value='$person->id'>Edit</button></td>");
      print("<td><button type='submit' name='delete' value='$person->id'>Delete</button></td>");
      print("</tr>");
    }
    print("</table></form>");
  }
}
```

I View laver vi en update form som kan bruges til at editere en enkelt Person

Og en show tables som viser alle Personer

```
$ctrl = new PersonCtrl();
$view = new PersonView();

if (isset($_POST['update'])){
    $ctrl->updatePerson(new PersonModel($_POST['name'],$_POST['birthdate'],$_POST['email'],$_POST['id']));
}
if (isset($_POST['delete'])){
    $ctrl->deletePerson($_POST['delete']);
}
if (isset($_POST['edit'])){
    $view->updateForm($ctrl->getPerson($_POST['edit']));
    die();
}

$persons = ($ctrl->getPersonsAll());
$view->showTable($persons);
```

I index.php laver vi først en controller og et view. Derefter spørger vi på om der er postet update, delete eller edit. Hvis det ikke er tilfældet viser vi alle Personer i en tabel vha view->showTable.