

IF3 I 203 / IF4 I 203 PENGEMBANGAN SITUS WEB (PSW) II

Oleh: Togu Novriansyah Turnip (TNT)
Program Studi D3/D4 Teknik Informatika IT Del



WEB APP AND LANGUAGE INTRODUCTION

OUTLINE

- Introduction
 - Web App,
 - Multi-tiered architecture,
 - Web Components
 - Static vs Dynamic Webs
- PHP Basic
 - Variables, constants, operators and arrays
 - Conditional, loops, type casting, and Built-in function

WEB APPLICATION

- **An application that is accessed with a web browser over a network.**
- No software distribution and installation.
- Accessible by numerous amount of clients.
- Mostly, its contents are dynamically generated in a standard format(HTML) and transferred through reliable connection(TCP/IP).

MULTI-TIER ARCHITECTURE

- a client-server architecture which separates the presentation, business logic, and data into different physical locations.
- By doing so, changes in a specific tier won't directly affect the other tiers.
- For example, you moved the database from a server to another dedicated location, it won't make a real pain.

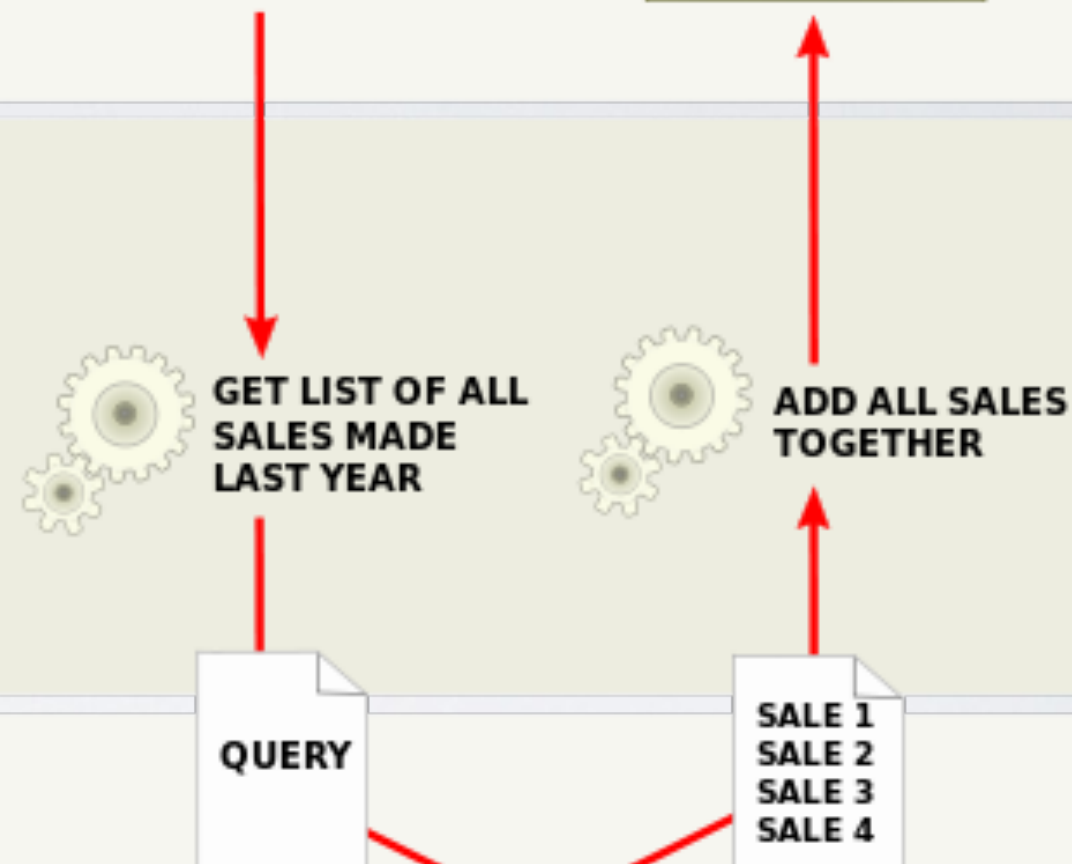
Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.



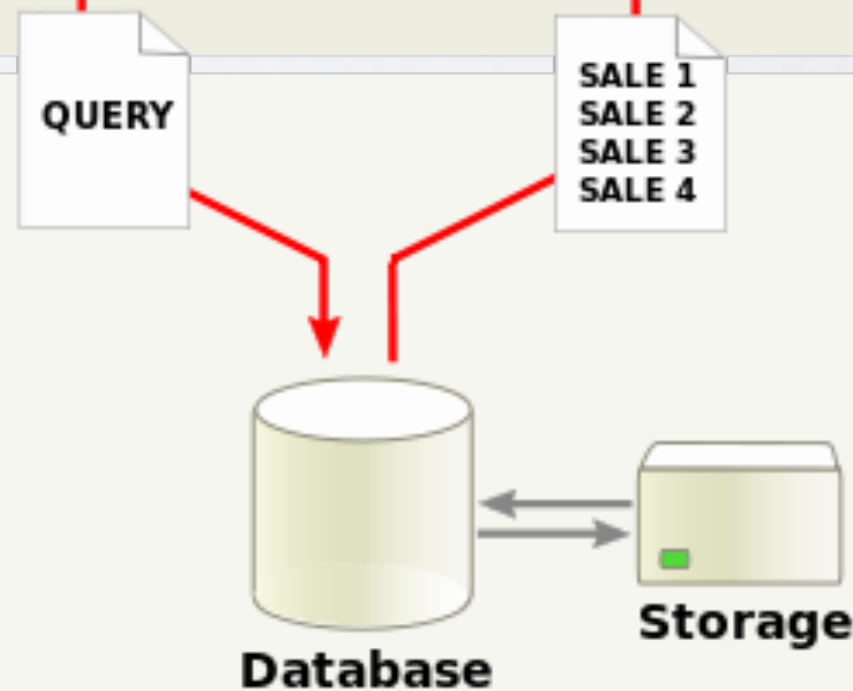
Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.



Data tier

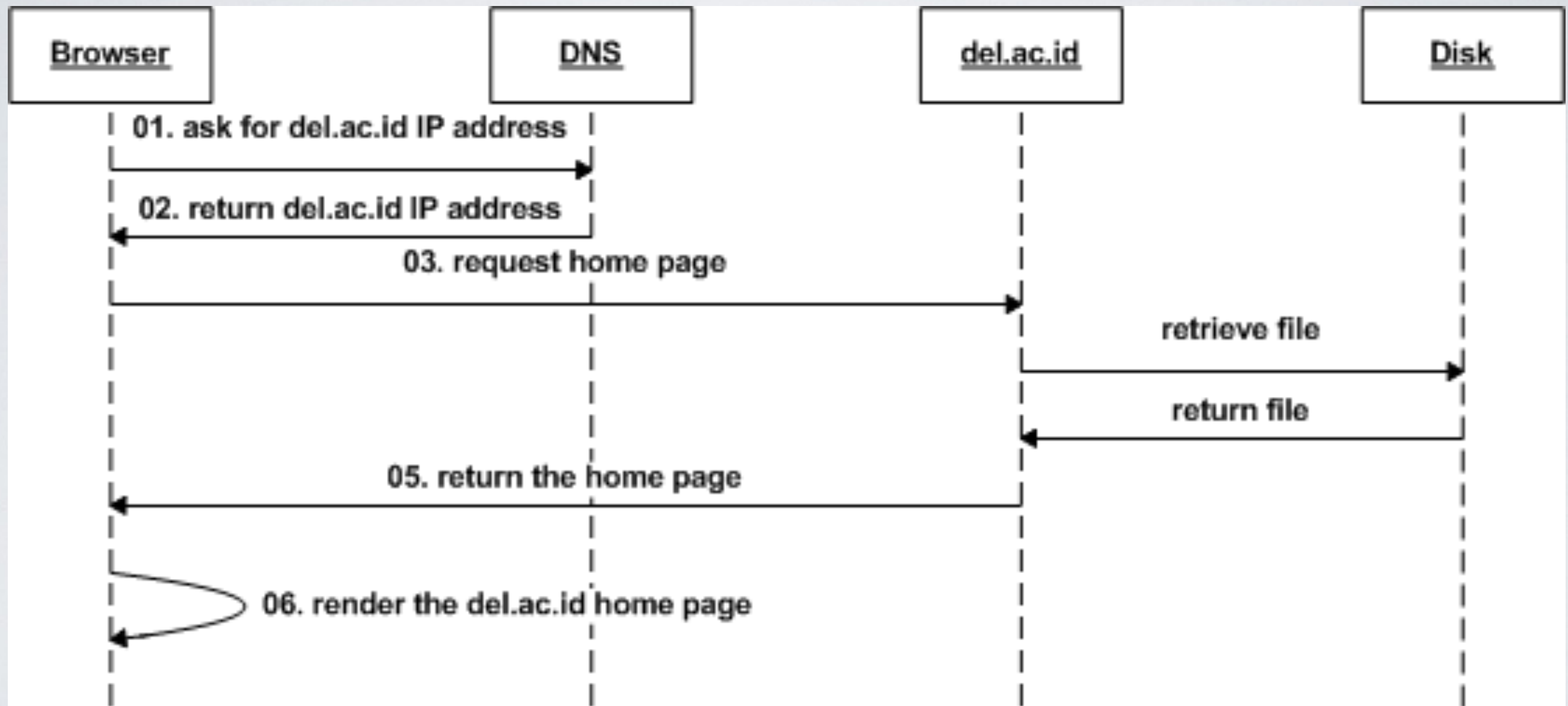
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.



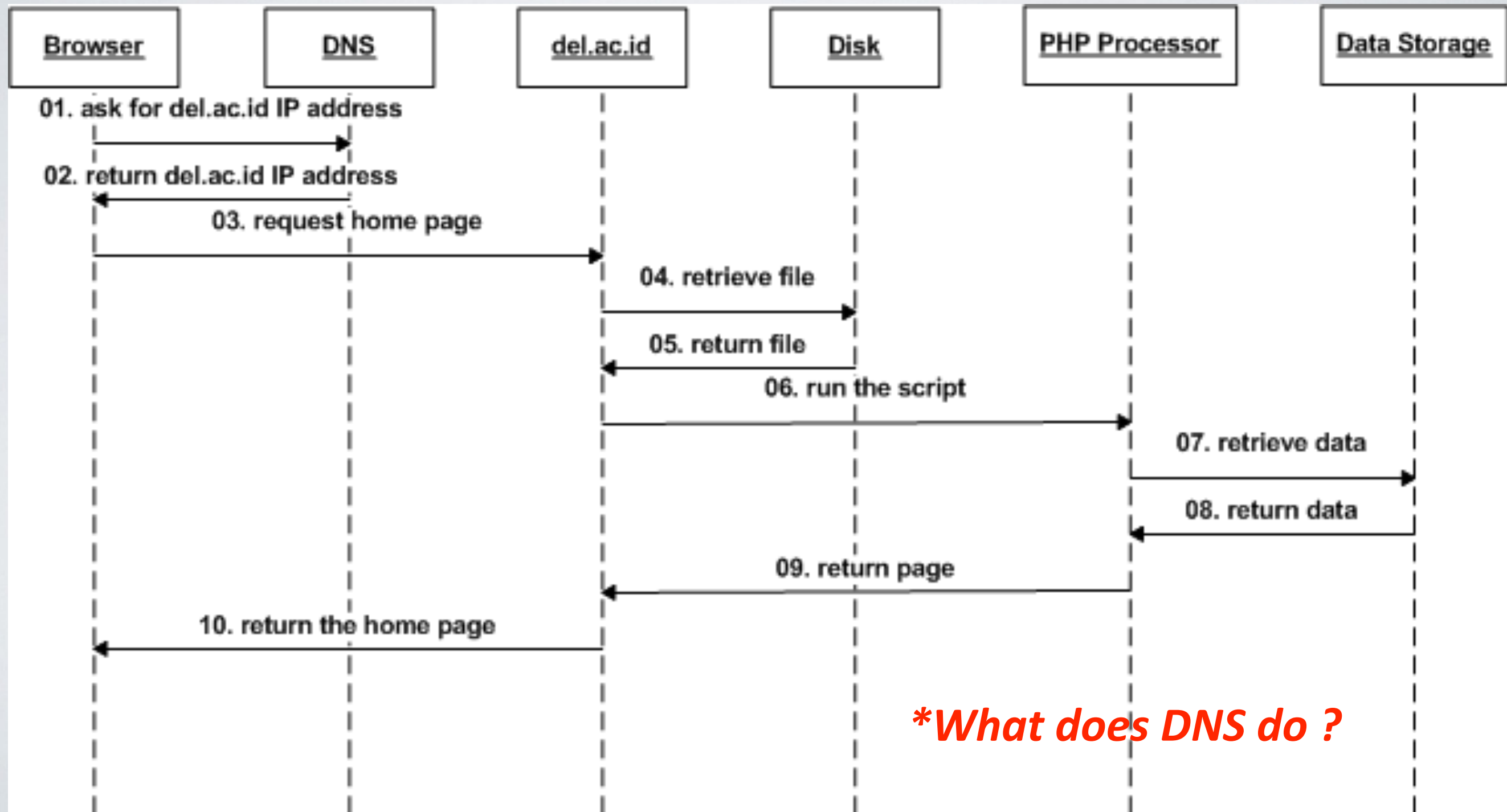
WEB COMPONENTS

- **Server:** a computer which resources (like page , images, documents etc.) are stored.
- **Web server:** an application that accepts client's request, process it & return the result.
- **Client:** a computer which connected to server through a certain network and able to request resource resides on the server.
- **Browser:** a client-side application required to interpret the web.
- **Protocols:** HTTP & TCP/IP

STATIC WEB: HOW DOES IT WORK ?



DYNAMIC WEB: HOW DOES IT WORK ?



STATIC TO DYNAMIC

- At the beginning, web technology was one-way, where client couldn't create web content, no 'conversation' at all, or simply static web (also called as web 1.0).
- A dynamic content is required, to let client to contribute content, interact with other client, and so on.
- Server-side programming language is the answer **to insert additional processing into web page** to make it dynamic (called web 2.0).

SERVER SIDE PROGRAMMING LANGUAGE

- There are tons, such:
 - Java Enterprise Edition,
 - C# (.NET)
 - Ruby,
 - Perl,
 - PHP,
 - And many more ...

PHP

- PHP (PHP: Hypertext Preprocessor)
- One of the most popular server side prog. language, a scripting language to be precise.
- Using PHP, it is possible to add logical activities into a web page, such fetching data, doing some calculations, and so on.
- PHP is claimed to be the simplest yet fast and flexible language which make it popular.

WHY PHP?

- It is widely used, supports OO paradigm.
- Working on most architectures (X86, X64) & platforms (Linux, Windows, Mac, etc.).
- Working with most web servers (Apache, NGINX, IIS, etc.).
- Compatible to most databases servers (Oracle, MS. SQL Server, MySQL, Postgre, etc.)

PHP: FIRST LOOK [I]

- Imagine you are developing a web which informs its visitor about the current time ?

“now is: 01 January 2015, 09:30 AM”

- Well, you can create an HTML page contains:

```
<html>
  <head><title>PHP: first look [1]</title></head>
  <body>
    now is: 01 January 2015, 09:30 AM
  </body>
</html>
```

PHP: FIRST LOOK

[2]

The problem is, will you **edit**
the time **every minute** for
the sake of a single page ?
– **that's insane !**



PHP: FIRST LOOK [3]

- Hey! remember, PHP is here to help you out
- You just need to add a PHP script on the code.
- Then change the file extension to **.php**.

```
<html>
  <head><title>PHP: first look [2]</title></head>
  <body>
    now is: <?php echo(date('d F Y H:i a')); ?>
  </body>
</html>
```

- Voila,!!! You did it.

PHP: VARIABLES [1]

- **PHP is a dynamic, loosely typed language.** Variables do not have to be declared before used & PHP always converts variables to the type required by the context when they are accessed.
- In PHP, there is **no explicit type definition** in a variable declaration; a variable's **type is determined by the context** in which the variable is used.
- For example: if a string value is assigned to variable `$var`, then it becomes a string. If an integer value is then assigned to `$var`, it becomes an integer.

PHP: VARIABLES [2]

- A variable is prefixed with the dollar sign (\$) then followed by the variable name.
- It is **case-sensitive**, means `$myName` is not `$MYNAME`.

```
<?php
    $name = 'Wiro Sableng';
    $height = 170;
    $weight = 72.6;
    $weapon = "the 2-1-2 dangerous axe";
    $motto = 'me + PHP = kicks the problem away!';
?>
```

PHP: ARITHMETIC OPERATORS

Operator	Description	Example
+	Addition	<code>\$j + 1</code>
-	Subtraction	<code>\$j - 6</code>
*	Multiplication	<code>\$j * 11</code>
/	Division	<code>\$j / 4</code>
%	Modulus (division remainder)	<code>\$j % 9</code>
++	Increment	<code>++\$j</code>
--	Decrement	<code>--\$j</code>

PHP: ASSIGNMENT OPERATORS

Operator	Example	Equivalent to
=	\$j = 15	\$j = 15
+=	\$j += 5	\$j = \$j + 5
-=	\$j -= 3	\$j = \$j - 3
*=	\$j *= 8	\$j = \$j * 8
/=	\$j /= 16	\$j = \$j / 16
.=	\$j .= \$k	\$j = \$j . \$k
%=	\$j %= 4	\$j = \$j % 4

PHP: COMPARISON OPERATORS

Operator	Description	Example
==	Is equal to	\$j == 4
!=	Is not equal to	\$j != 21
>	Is greater than	\$j > 3
<	Is less than	\$j < 100
>=	Is greater than or equal to	\$j >= 15
<=	Is less than or equal to	\$j <= 8

PHP: LOGICAL OPERATORS

Operator	Description	Example
&&	And	<code>\$j == 3 && \$k == 2</code>
and	Low-precedence and	<code>\$j == 3 and \$k == 2</code>
 	Or	<code>\$j < 5 \$j > 10</code>
or	Low-precedence or	<code>\$j < 5 or \$j > 10</code>
!	Not	<code>! (\$j == \$k)</code>
xor	Exclusive or	<code>\$j xor \$k</code>

PHP: CONSTANTS

- its main job is to store value, like variable.
- Unfortunately, unlike variable, you can only define a constant value once & can't be altered.

```
<?php
    define("PI", 3.1415927);
    $radius = 7.0;
    echo PI * ($radius * $radius);
?>
```

CONDITIONAL

- An expression that evaluates to either **TRUE** or **FALSE** (Boolean value). They are mostly used to determine the **program flow**.
- The expression value decide which flow will the program proceed.
 - **if**
 - **else**
 - **else if**
 - **switch**

CONDITIONAL: FROM IF TO SWITCH

```
<?php
$name = 'Wiro';

if($name == 'Wiro'){
    echo('your weapon is the 2-1-2 axe. ');
}else if($name == 'Wolverine'){
    echo('your weapon is the cat-paw. ');
}else{
    echo('you have no weapon. ');
}

?>
```

```
<?php
$name = 'Wiro';

switch($name){
    case 'Wiro':
        echo('your weapon is the 2-1-2 axe. ');
        break;
    case 'Wolverine':
        echo('your weapon is the cat-paw. ');
        break;
    default:
        echo('you have no weapon. ');
        break;
}

?>
```

LOOPS

- Doing a defined repetitive tasks.
- e.g. counting the number of book in the library which are categorized as programming textbook.
 - **for**
 - **while**
 - **do-while**
 - **foreach**

LOOP: FOREACH

- Used to loop an object that holds multiple values, such as array.

```
1 <?php
2     $arrName = array(
3         'Wiro Sableng', 'Jaka Sembung', 'Saras 008',
4         'Milkyman', 'Ranger Merah', 'Pikachu'
5     );
6 ?>
7 <html>
8     <head><title>arrays</title></head>
9     <body>
10         <h2>my super hero</h2>
11         <ul>
12 <?php
13     foreach($arrName as $name){
14         echo('<li>' . $name . '</li>');
15     }
16 ?>
17         </ul>
18     </body>
19 </html>
```

TYPES OF ARRAYS

Indexed Arrays

Arrays with a **numeric index**

Indexed can be assigned **automatically** or **manually**

Associative Arrays

Arrays with **named keys**

Using **foreach** loop to print all the values of an array

Multi dimensional Arrays

Arrays containing **one or more arrays**

Two-Dimensional Array is **an array of arrays**

PHP: ARRAYS[I]

- The index can be assigned automatically (index always starts at 0), like this:

```
1 <?php
2     $arrName = array(
3         'Wiro Sableng', 'Jaka Sembung', 'Saras 008',
4         'Milkyman', 'Ranger Merah', 'Pikachu'
5     );
6 ?>
7 <html>
8     <head><title>arrays</title></head>
9     <body>
10         <h2>my super hero</h2>
11         <ul>
12 <?php
13     echo('<li>' . $arrName[0] . '</li>');
14     echo('<li>' . $arrName[1] . '</li>');
15     echo('<li>' . $arrName[2] . '</li>');
16 ?>
17         </ul>
18     </body>
19 </html>
```

- To loop through and **print all the values of an indexed array**, you could use a **for loop**

PHP: ARRAYS[2]

- Unlike other programming languages, PHP support any type as either the key or value.

```
<?php
$age = array(
    "Peter"=>"35",
    "Ben"=>"37",
    "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```

- To loop through and **print all the values of an associative array**, you could use a **foreach loop**

PHP: ARRAYS[3]

- sometimes you want to store values with more than one key. This can be stored in multidimensional arrays.

Name	Stock	Sold
Volvo	22	18
BMW	15	13



```
<?php
$cars = array
(
    array("Volvo",22,18) ,
    array("BMW",15,13) ,
);
echo $cars[0][0].": In stock: " .
$cars[0][1].", sold: ".$cars[0]
[2].".<br>";
echo $cars[1][0].": In stock: " .
$cars[1][1].", sold: ".$cars[1]
[2].".<br>";
?>
```

- We can also put a For loop inside another For loop to get the elements of the \$cars array

TYPE CASTING

- Converting a value from one type to another. In PHP it's also called as type juggling, mostly it's done automatically.
- See the documentation for a complete rules: <http://php.net/manual/en/language.types.type-juggling.php>

FUNCTION

- Functions are "self contained" modules of code that accomplish a specific task. Functions usually "take in" data, process it, and "return" a result.
Once a function is written, it can be used over and over and over again. Functions can be "called" from the inside of other functions.
- function vs procedure ?

FUNCTION: BUILT-IN

- In default, PHP provides tons of ready to use function available.
- To display value: `echo()`, `printf()`, `var_dump()`, etc.
- To manipulate strings: `substr()`, `trim()`, etc.
- To manipulate array: `array_keys()`, etc.
- And many more ...

“Any Questions?”

“Nine People can’t make a baby in a month.” Then,
more programmers doesn’t mean the project will be
completed faster.

-HAPPY CODING-
;)