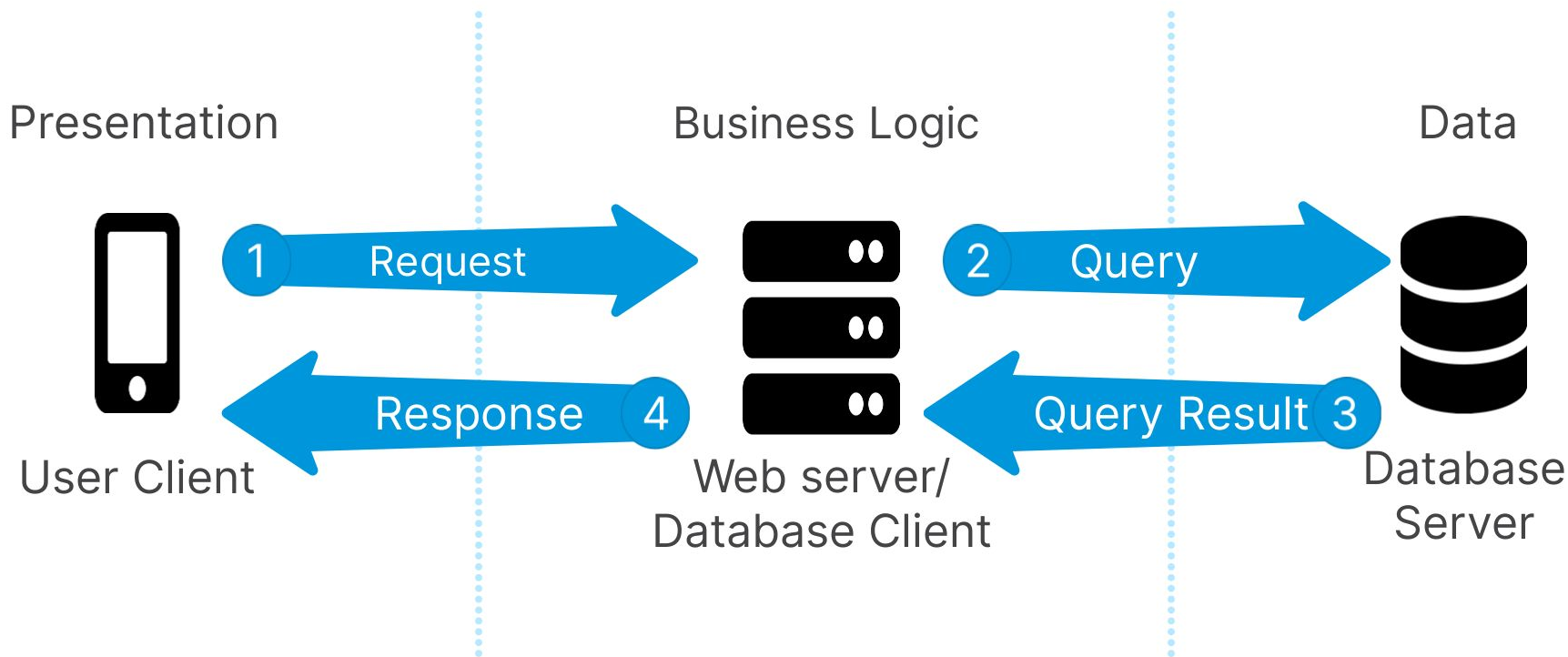
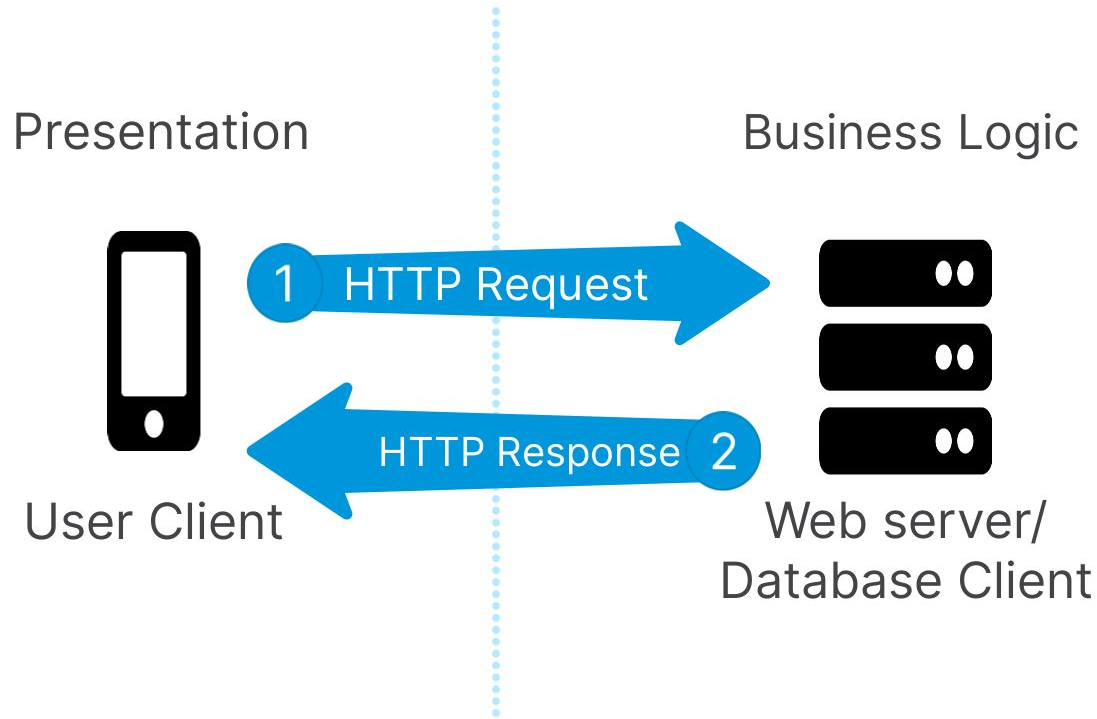


URL



Recall: Client-server architecture



HTTP: Protocol used for data transmission between clients and servers on world wide web

Uniform

Resource

Locator

`https`://www.google.com/search?q=waffles



Specifies how to transmit data

HTTP = HyperText Transfer Protocol

HTTPS = HyperText Transfer Protocol Secure

Uniform

Resource

Locator

`https://www.google.com/search?q=waffles`



Specifies location of server
(alias for IP address)

Uniform

Resource

Locator

`https://www.google.com/search?q=waffles`



Specifies requested resource

Uniform

`https://www.google.com/search?q=waffles`

Specifies parameters
as key-value pairs



Resource

`https://www.google.com/search?q=waffles&junk=value`

Locator

Multiple parameters joined with **&** symbol

Unused parameters are typically ignored

Uniform Resource Locator

<https://www.google.com/search?q=waffles#top-of-page>

Examples



Permalink to this headline

In addition to the examples below, more examples are given in [How to Use urllib The urllib Package](#).

This example gets the python.org main page and displays the first 3

```
>>> import urllib.request
>>> with urllib.request.urlopen('http://www.python.org/') as f:
...     print(f.read(300))
...
b'<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<?xml
xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en
<meta http-equiv="content-type" content="text/html; charset
<title>Python Programming'
```


Uniform

`http://localhost:5433` (pgAdmin example)

Resource

`http://www.google.com:80/search?q=waffles`

Locator

`https://www.google.com:443/search?q=waffles`

Uniform

`http://username:password@example.com`

Resource

`http://username@example.com`

Locator

Syntax:

`protocol://username:password@domain.name:port/path?query=parameter&q2=p2#fragment`

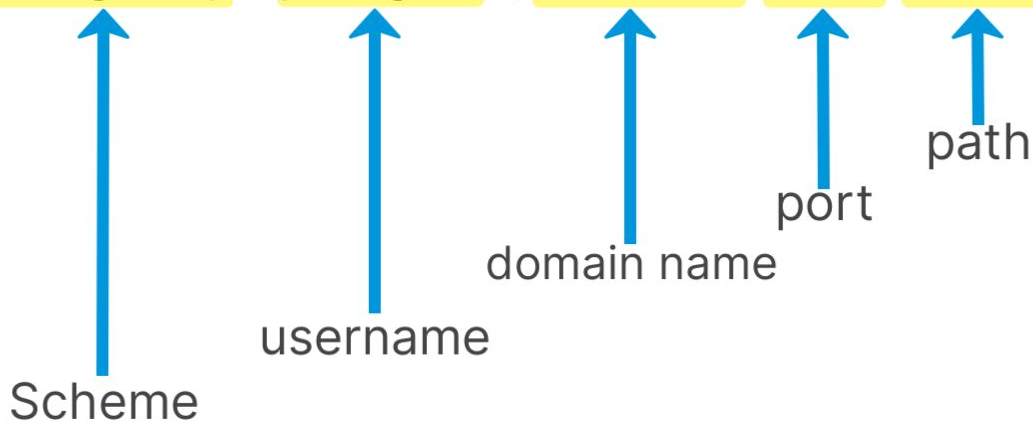
Postgres connection string

Connect using psql:

```
bash-5.1# psql postgresql://postgres@localhost:5432/week3
```

Postgres connection string

postgresql://postgres@localhost:5432/week3



```
bash-5.1# psql postgresql://postgres@localhost:5432/week3
```

Query parameter encoding

Some characters not valid including space, newline, @

Invalid characters must be encoded

Example: "blueberry waffles" has a space,
which is encoded in a URL as **%20**

<https://www.google.com/search?q=blueberry%20waffles>

URL encoding: Python example

```
>>> import urllib.parse
>>> params = {'name': 'Alice Anderson', 'email': 'alice@example.com'}
>>> urllib.parse.urlencode(params, quote_via=urllib.parse.quote)
'name=Alice%20Anderson&email=alice%40example.com'
```

Raw	Encoded
space	%20
@	%40
#	%23
\$	%24
/	%2F
&	%26