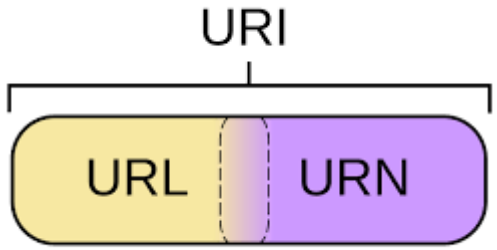
UNIT-5

URI (Uniform Resource Identifier)

- A URI (Uniform Resource Identifier) is a string of characters used to identify a name or a web resource.

- In other words, it is a sequence of characters that identifies a logical or physical resource.

- Examples of resources include
  - Electronic Documents,
  - Elevator Door Sensors,
  - Web Pages,
  - ID microchips for pets



Definition of URI

Each URI begins with a scheme name that refers to a specification for assigning identifiers within that scheme.

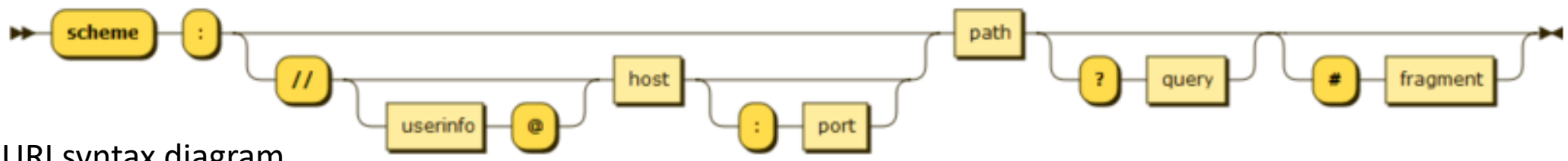
The URI generic syntax consists of a hierarchical sequence of five components:

URI = scheme:[//authority]path[?query][#fragment]

where the authority component divides into **three** subcomponents:

authority = [userinfo@]host[:port]

This is represented in a syntax diagram as:

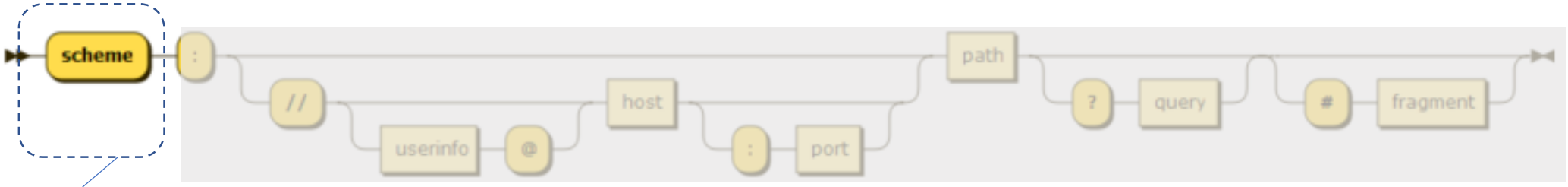


URI syntax diagram

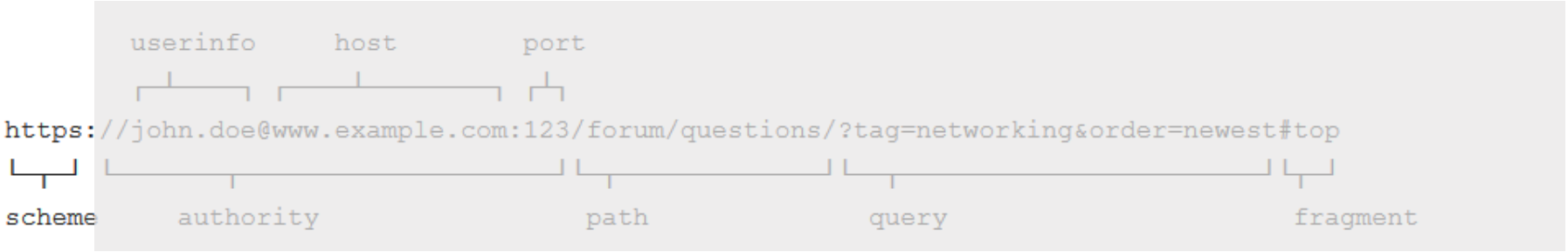
These five components of URI are explained as follows:

1) Scheme: (Eg: http, https, ftp, mailto, file, data, news, tel, telnet, urn)

- A non-empty **scheme** component is followed by a colon (:)
- It consists of a sequence of characters beginning with a letter and followed by any combination of letters, digits, plus (+), period (.), or hyphen (-).
- Examples of popular schemes include http, https, ftp, mailto, file, data.



http
https
ftp



2) Authority: An optional non-empty **authority** component **preceded by two slashes (//)**, comprising:

[userinfo@]host[:port]

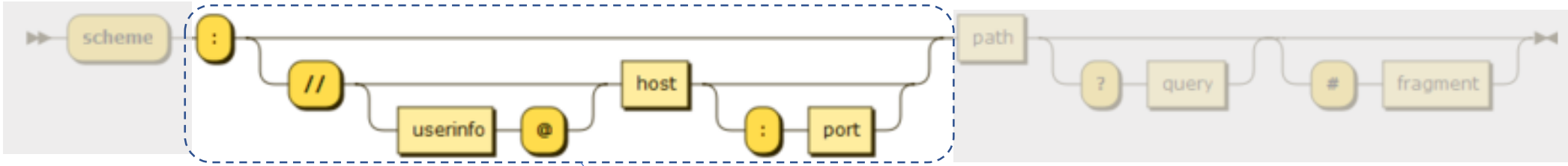
An optional **userinfo** subcomponent that may consist of a user name and an optional password preceded by a colon (:), followed by an at symbol (@).

In fact, use of the format username:password in the userinfo subcomponent is avoided for security reasons.

A non-empty **host** subcomponent, consisting of either a registered name (including hostname), or an IP address.

IPv4 addresses must be in dot-decimal notation, and IPv6 addresses must be enclosed in brackets ([]).

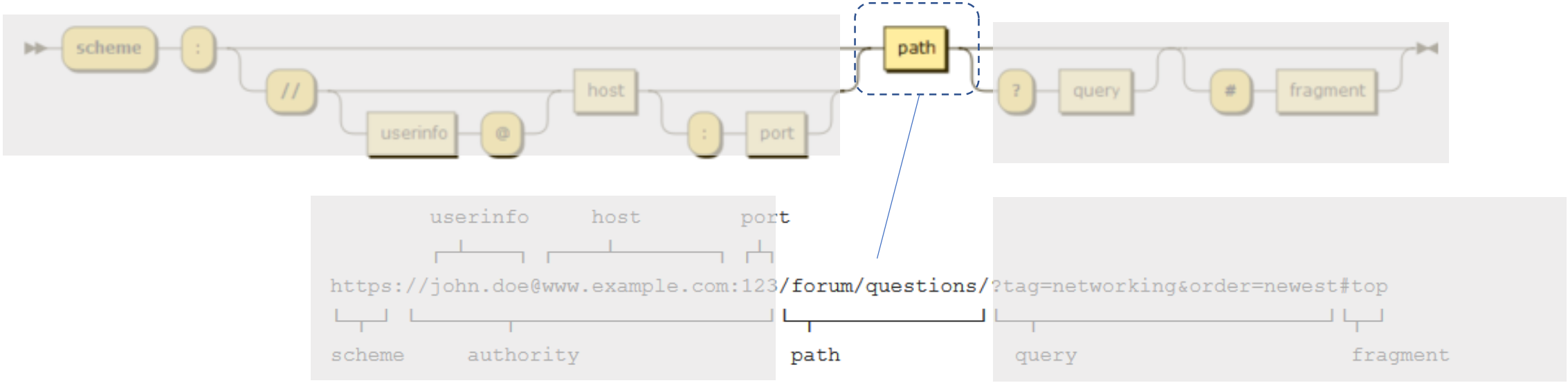
An optional **port** subcomponent preceded by a colon (:)



3) Path

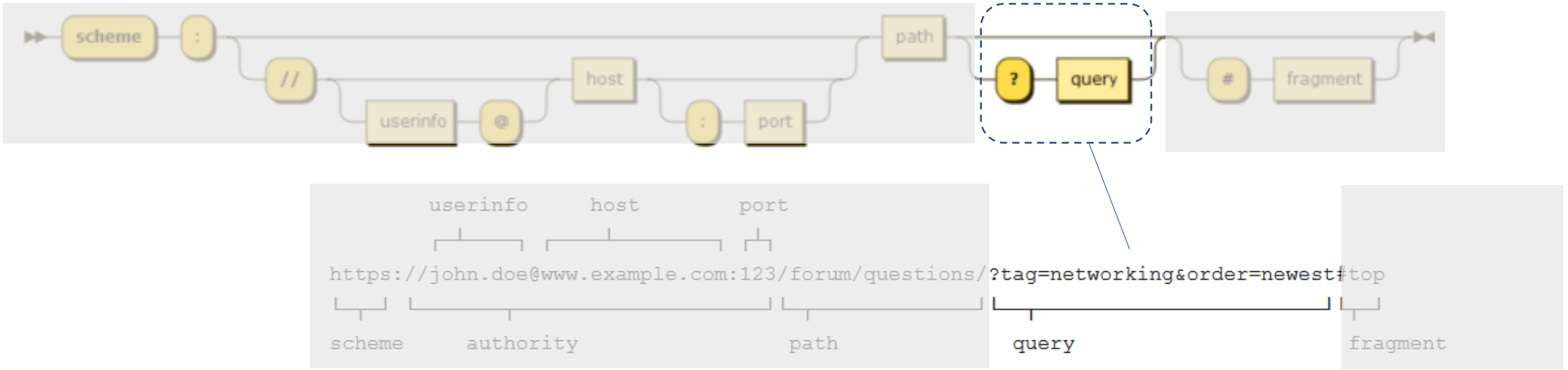
- A path component, consisting of a sequence of **path segments** separated by a slash (/).
- A path is always defined for a URI, though the defined path may be empty (zero length).
- A **path segment may also be empty**, resulting in two consecutive slashes (//) in the path component.
- If an **authority component is present**, then the path component must either be empty or begin with a slash (/).
- If an **authority component is absent**, then the path cannot begin with an empty segment, that is with two slashes (//)

Example: https://ums.lpu.in/lpuums/LoginNew.aspx



4) Query

- An optional **query** component is preceded by a question mark (?)
- It consists of a query string of non-hierarchical data.
- Its syntax is not well defined, but by convention is most often a sequence of attribute–value pairs separated by a delimiter.

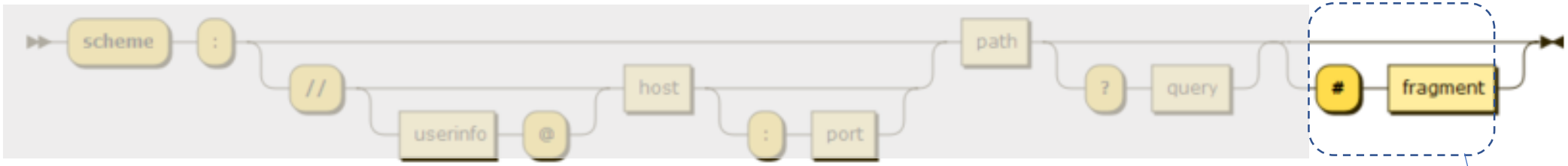


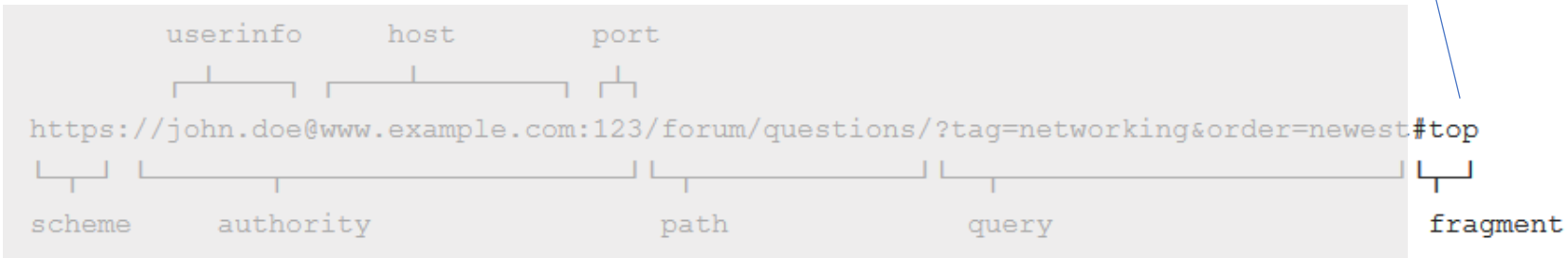
5) Fragment

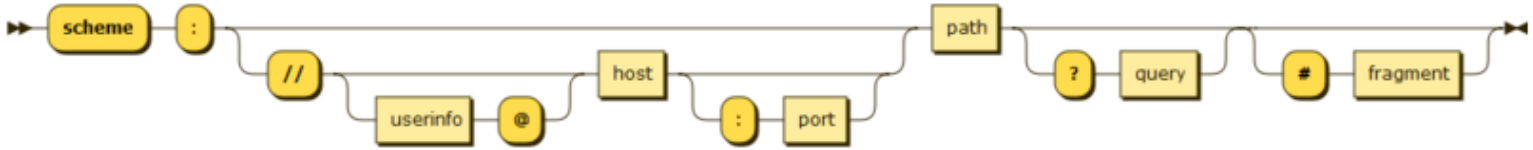
- An optional **fragment** component is preceded by a hash (#).

- The fragment contains a fragment identifier providing direction to a secondary resource, such as a section heading in an article identified by the remainder of the URI.

- When the primary resource is an HTML document, the fragment is often an id attribute of a specific element, and web browsers will scroll this element into view.







Examples: The following figure displays example URIs and their component parts as per above syntax diagram:

Example 1:

```
          userinfo      host        port
          ┌──┴───┐ ┌────┴─────┐ ┌──┴┐
  https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
  └─┬─┘ └─────────┬──────────┘└─────┬─────┘└─────────┬─────────┘ └──┬──┘
  scheme      authority                path                query                   fragment
```

Example 2:

```
  ldap://[2001:db8::7]/c=GB?objectClass?one
  └─┬─┘ └──────┬──────┘└─┬─┘ └──────┬──────┘
  scheme     authority  path        query
```

Example 3:

```
  mailto:John.Doe@example.com
  └──┬──┘ └─────────┬────────┘
  scheme          path
```

Example 4:

```
  news:comp.infosystems.www.servers.unix
  └─┬─┘ └──────────────┬──────────────┘
  scheme              path
```

Example 5:

```
  tel:+1-816-555-1212
  └─┬─┘ └──────┬──────┘
scheme      path
```

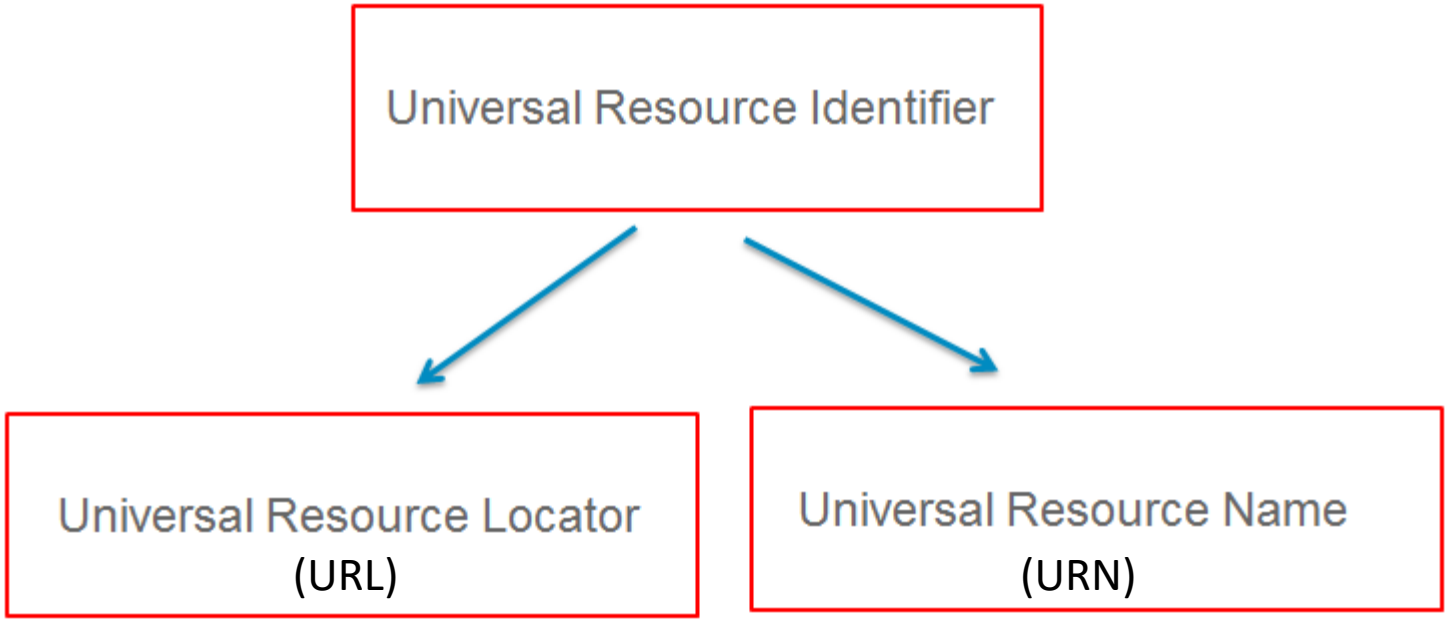
Example 6:

```
  telnet://192.0.2.16:80/
  └──┬──┘ └─────┬─────┘|
  scheme     authority  path
```

Example 7:

```
  urn:oasis:names:specification:docbook:dtd:xml:4.1.2
  └─┬─┘ └──────────────────┬──────────────────┘
scheme                   path
```

There are **Two Types of URIs**:



1. Uniform Resource Locators (URLs) : a means of locating the resource

2. Uniform Resource Names (URNs): – persistent, location-independent identifiers for resources

1. Uniform Resource Locator (URL)

This type of URI begins by stating which protocol should be used to locate and access the physical or logical resource on a network.

- If the resource is a web page, the URI will begin with the protocol HTTP.
- If the resource is a file, the URI will begin with the protocol FTP
- If the resource is an email address, the URI will begin with the protocol mailto.

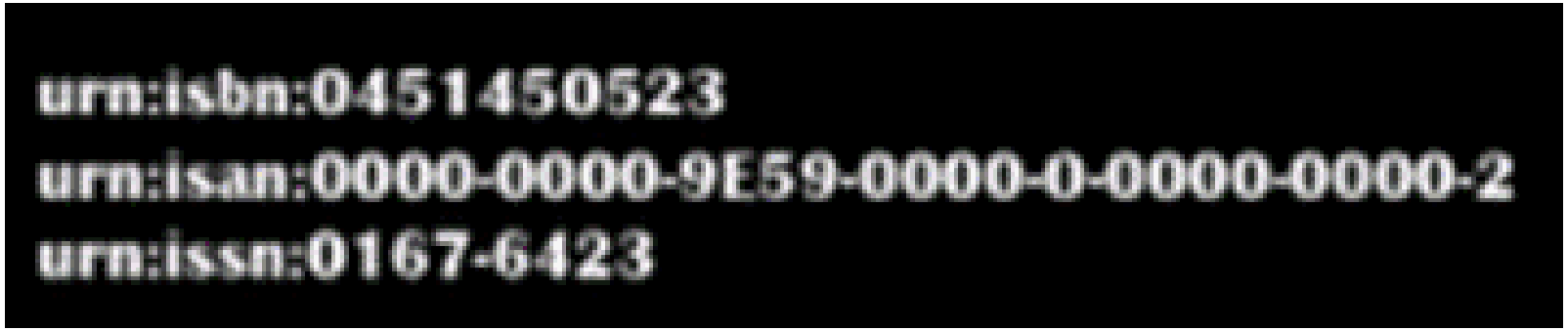
- It is important to remember that URLs are not persistent.
- This means that if the resource's location changes, the URL also needs to change to point to the resource's new location.

2. Uniform Resource Name (URN)

This type of URI does not state which protocol should be used to locate and access the resource.

- it simply labels the resource with a persistent, location-independent unique identifier.
- A URN will identify the resource throughout its lifecycle and will never change.

Each URN has three components: the label “urn,” a colon and a character string that serves as a unique identifier.



Some More Examples of URN

URN	corresponds to
urn:isbn:0451450523	The 1968 book The Last Unicorn, identified by its book number.
urn:isan:0000-0000-9E59-0000-O-0000-0000-2	The 2002 film Spider-Man, identified by its audiovisual number.
urn:ISSN:0167-6423	The scientific journal Science of Computer Programming, identified by its serial number.
urn:ietf:rfc:2648	The IETF's RFC 2648.
urn:mpeg:mpeg7:schema:2001	The default namespace rules for MPEG-7 video metadata.
urn:oid:2.16.840	The OID for the United States.
urn:uuid:6e8bc430-9c3a-11d9-9669-0800200c9a66	A version 1 UUID.
urn:nbn:de:bvb:19-146642	A National Bibliography Number for a document, indicating country (de), regional network (bvb = Bibliotheksverbund Bayern), library number (19) and document number.
urn:lex:eu:council:directive:2010-03-09;2010-19-UE	A directive of the European Union, using the proposed Lex URN namespace.
urn:lsid:zoobank.org:pub:CDC8D258-8F57-41DC-B560-247E17D3DC8C	A directive of the Life Science Identifiers may be resolved to http://zoobank.org/urn:lsid:zoobank.org:pub:CDC8D258-8F57-41DC-B560-247E17D3DC8C .

URIs for Mobile Apps

Under this topic we will cover:

Invoking Native Mobile Applications with URI Schemes

- URI Schemes
- URI Schemes and Mobile Devices

Invoking Native Mobile Applications with URI Schemes

iOS, Android and most A-Grade Mobile platforms implement standard URI Schemes (or parts thereof), which allow for easily launching native applications.

URI Schemes for Mobile Apps

- When considering URI Schemes, more commonly (though incorrectly) referred to as protocols, one typically thinks in terms of the more ubiquitous schemes: http, ftp, file, mailto, about, data:uri and so forth.
- However, as with these and other common URI Schemes, **there are additional schemes** which allow for various practical applications **in the context of Mobile Web Apps**; the most common of which being the ability to invoke a platform's native phone or messaging application, respectively.

URI Schemes and Mobile Devices

In the context of Mobile Web Applications, the **tel**, and **sms** URI Schemes are perhaps the most common and applicable; providing a simple means of invoking their corresponding native applications.

The following are basic examples which work across all major mobile platforms.

1) The `tel` URI Scheme

The tel URI Scheme allows for launching a device's native Phone application in the context of the phone number specified:

```
<a href="tel:1-800-555-1234">Call 1-800-555-1234</a>
```

2) The sms URI Scheme

The sms URI Scheme allows for **launching a device's native Messaging application to send an sms message**, or to send an sms message to the phone number specified:

```
<!-- launch native messaging application -->
<a href="sms:">Send a Text</a>

<!-- launch native messaging application with phone number-->
<a href="sms:1-800-555-1234">Text Us</a>
```