URI, URL, URN & DOI

Understanding Resource Identifiers

URI (Uniform Resource Identifier)

- **General concept**: String identifying a resource
- **Defined by**: RFC 3986 *(Generic Syntax)*

```
URI = scheme ":" hier-part [ "?" query ] [ "#" fragment ]
```

- **Parent category** for URL and URN
- **Example**: `https://example.com`, `mailto:user@example.com`

URL (Uniform Resource Locator)

- **Where** & **How** to access a resource
- Specifies location + protocol
- Example: `https://www.example.com/path/file.html`
- *"Where to find it?"*

URN (Uniform Resource Name)

- **Persistent name** independent of location
- Identifies by unique name
- Example: `urn:isbn:978-0-123456-78-9`
- *"What is it?"*

DOI (Digital Object Identifier)

- **Specialized URN** for digital objects
- Persistent identifier for academic/scientific content
- It explicitly conforms to RFC 3986 syntax
- Example: `doi:10.1000/182` → `https://doi.org/10.1000/182`
- Used for papers, datasets, reports

- But DOI is not just an identifier...

Handle System: The Resolution Engine for DOI

- The Handle System is a distributed infrastructure for resolving persistent identifiers, developed by the Corporation for National Research Initiatives (CNRI).
- The Handle System manages the mapping between identifiers and URLs + metadata

```
[ Persistent ID ] → [ Handle System ] → [ Current Info ]
 doi:10.1000/182          (CNRI)          URL + Metadata
```

Component	Role
Handle	Persistent identifier (`prefix/suffix`)
GHR	Global Handle Registry: routes queries to correct server
LHS	Local Handle Service: stores actual mappings
Resolver	Client or HTTP proxy (`https://doi.org/...`)

> - **Generic infrastructure**: works for any digital resource
> - **Multi-value**: one handle → multiple typed records (URL, email, metadata)
> - **Decoupled**: identifier stays fixed, target can change

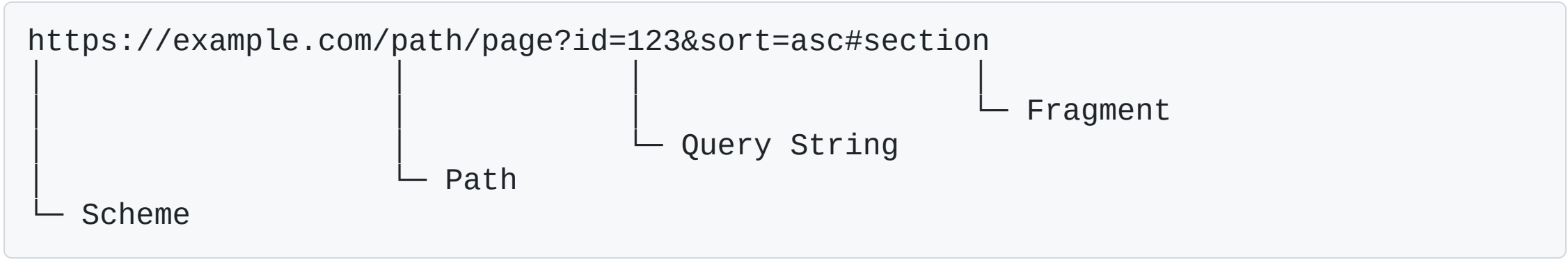
DOI = Handle System + Governance + Standards

Hierarchy

```
URI (umbrella term)
├── URL → Location + Access method
└── URN → Persistent name
     └── DOI → Scientific digital objects
```

Rule: All DOIs are URNs, all URLs/URNs are URIs

URI Anatomy



- **Query (`?`)**: Sends data to the server
- **Fragment (`#`)**: Client-side reference only

Query & Fragment: URI vs URL

Component	Generic URI	Practical Use
Query `?`	✅ Valid syntax	🌐 Mostly HTTP URLs
Fragment `#`	✅ Valid syntax	🌐 URLs; ⚠️ Rare in URN/DOI

> A URN *can* have `?` or `#`, but resolution depends on the scheme.
>
> **Example** (valid but rarely supported):

```
urn:isbn:978-0-123456-78-9?#chapter-3
```

Query vs Fragment

Feature	Query String (`?`)	Fragment (`#`)
Sent to Server?	✅ Yes	❌ No
Purpose	Filter, search, params	Navigate within page
Example	`?search=uri&page=2`	`#chapter-1`
Reload?	Changes content	Scrolls position

> **Note**: Fragments are processed by the browser, not the server.
> ⚠️ **URN/DOI**: Always check scheme specification before using `?` or `#`.

Quick Comparison

Type	Purpose	Example
URI	Identify any resource	`https://...`, `urn:...`
URL	Locate & retrieve	`https://site.com/file.pdf`
URN	Unique persistent name	`urn:isbn:1234567890`
DOI	Stable citation	`doi:10.1038/nature12345`