|  |  |  |
| --- | --- | --- |
| **Aspect** | **XML** | **JSON** |
| Syntax | Uses tags enclosed in angle brackets | Uses key-value pairs separated by commas |
| Readability | Can be verbose due to tags and attributes | Generally more concise and easier to read |
| Data Types | Supports various data types | Limited to string, number, boolean, array, and object |
| Comments | Allows comments within <!-- --> | Does not support comments |
| Namespace | Supports namespaces for avoiding conflicts | Does not have built-in support for namespaces |
| Parsing | Parsing can be slower due to complexity | Parsing is generally faster due to simplicity |
| Schema | Supports DTD and XML Schema for validation | No built-in schema validation, but can be implemented separately |
| Usage | Commonly used in web services, configuration files, and documents | Preferred for data interchange in web APIs and AJAX applications |

**Authentication** serves as the gatekeeper for system access, validating the identity of users or entities seeking entry. It ensures that only authorized individuals gain admittance, typically through a variety of verification methods such as passwords, biometrics, or cryptographic keys. Authentication protocols authenticate users' credentials against stored records, granting access upon successful validation. By confirming "Who are you?" authentication fortifies systems against unauthorized entry attempts, forming the initial line of defense in safeguarding sensitive data and resources.

**Authorization,** in contrast, governs the actions permissible to authenticated users within a system or application. After authentication confirms a user's identity, authorization steps in to determine "What can you do?" This process assigns access rights and privileges based on the authenticated user's role or level of authority. Access control mechanisms, such as role-based access control (RBAC) or attribute-based access control (ABAC), define and enforce these permissions, dictating which resources or functionalities a user can interact with. By strictly regulating user actions and resource access, authorization mitigates the risk of unauthorized use or misuse of sensitive data, maintaining the integrity and security of systems.