JAXR

Java API for XML Registries

Introduction to JAXR

What is JAXR?

- Java API for XML Registries (JAXR) is a tool that allows Java applications to interact with various XML registries in a standard, unified way.

Purpose of JAXR

- Enables seamless interactions with different registries for business-to-business (B2B) applications.
- Simplifies registry access by providing a single, consistent interface for multiple registry types.

Why Use JAXR?

- Standardizes registry access, making it easier for applications to find and publish services.

Registries and Repositories

Definitions

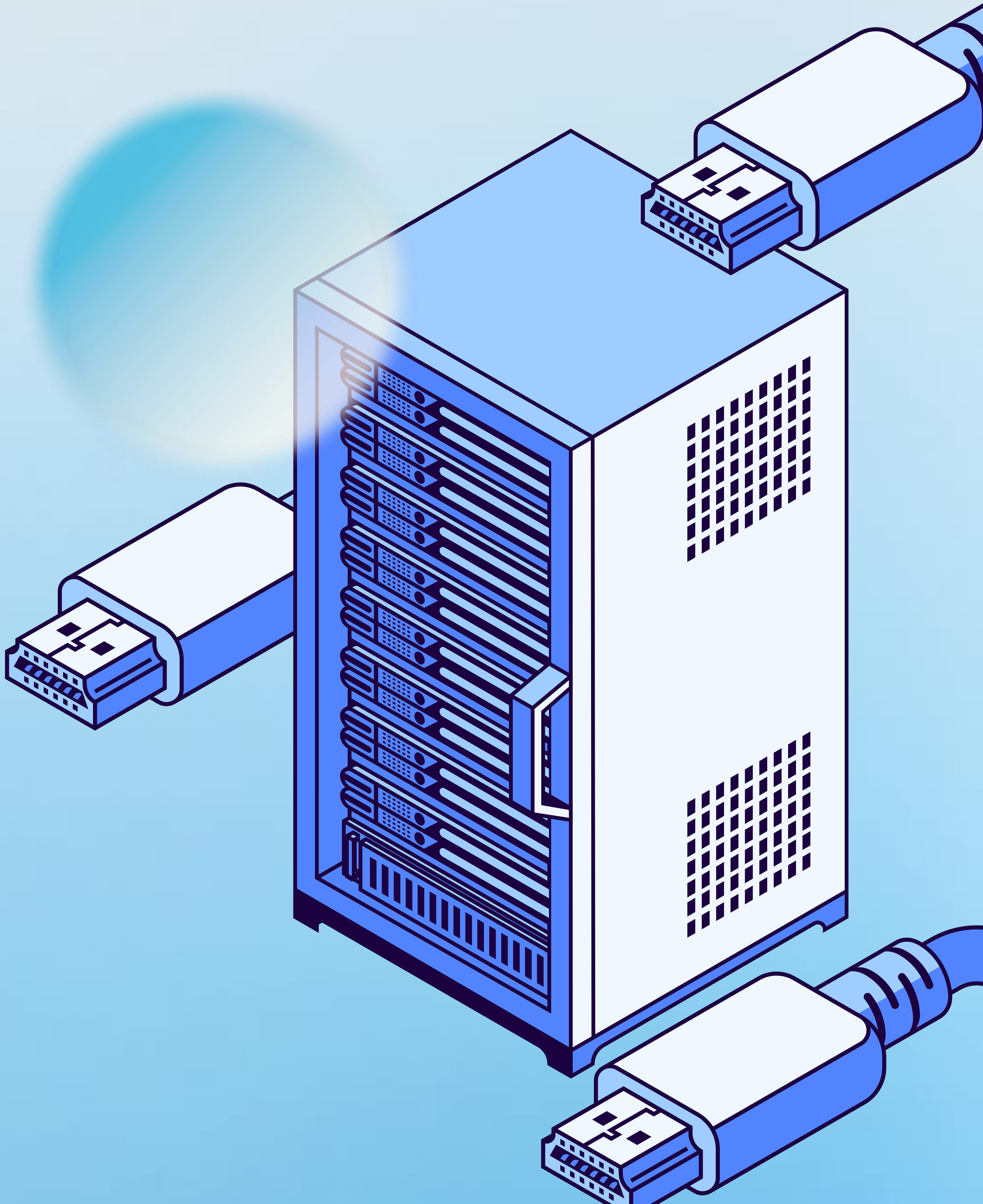
- Registry: A directory that lists services and other information, like UDDI (Universal Description, Discovery, and Integration).
- Repository: Stores both data and metadata about services, such as ebXML (Electronic Business XML).

Purpose of XML Registries

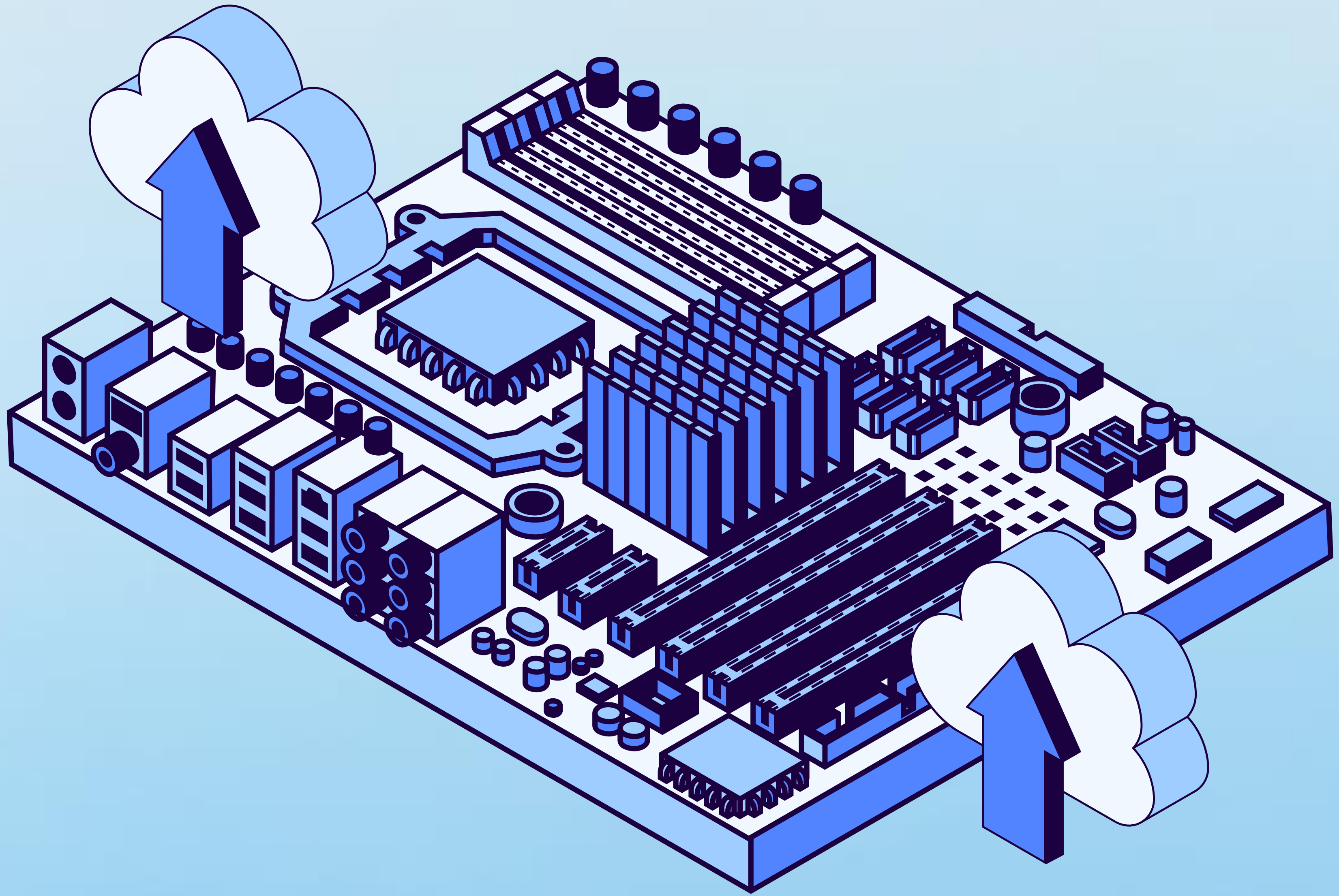
- XML registries allow for the discovery, storage, and management of web services, crucial for B2B transactions.

Shared Resource

- Registries provide a neutral, third-party space for businesses to share information and services dynamically and in a loosely coupled way.



Types of XML Registries Supported by JAXR



ebXML Registry

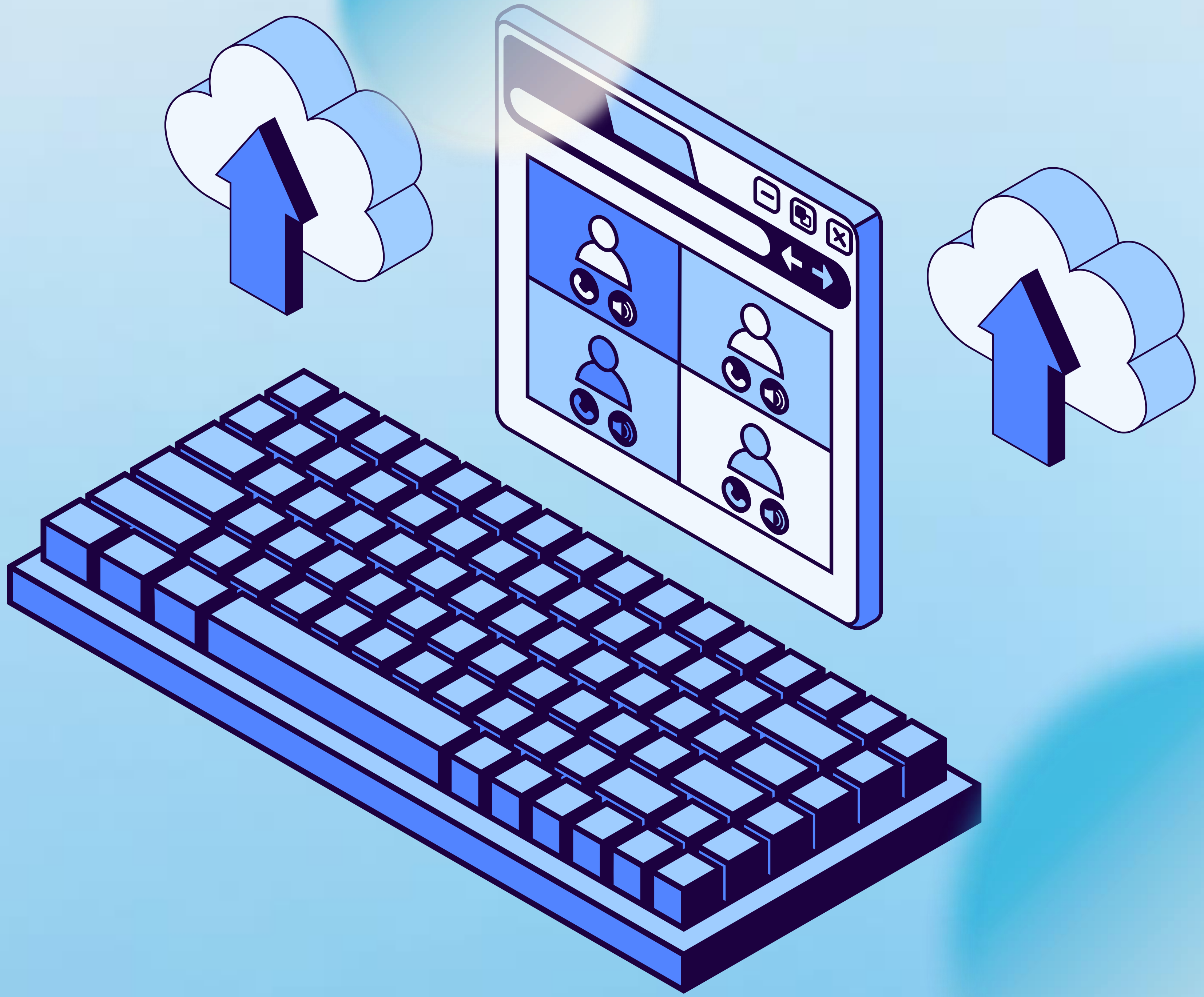
- Developed by OASIS and UN/CEFACT, used for broader content management beyond service listings.
- Acts both as a registry (listing of services) and repository (storage of data and metadata).

UDDI Registry

- Created by a consortium of technology vendors.
- Stores information specifically about businesses and the services they provide.

JAXR Overview

- JAXR provides a single API interface for interacting with both UDDI and ebXML registries.
- Applications written using JAXR can work across different types of registries, ensuring flexibility.
- Provides additional functions beyond what's directly available in UDDI and ebXML, such as an abstracted information model.



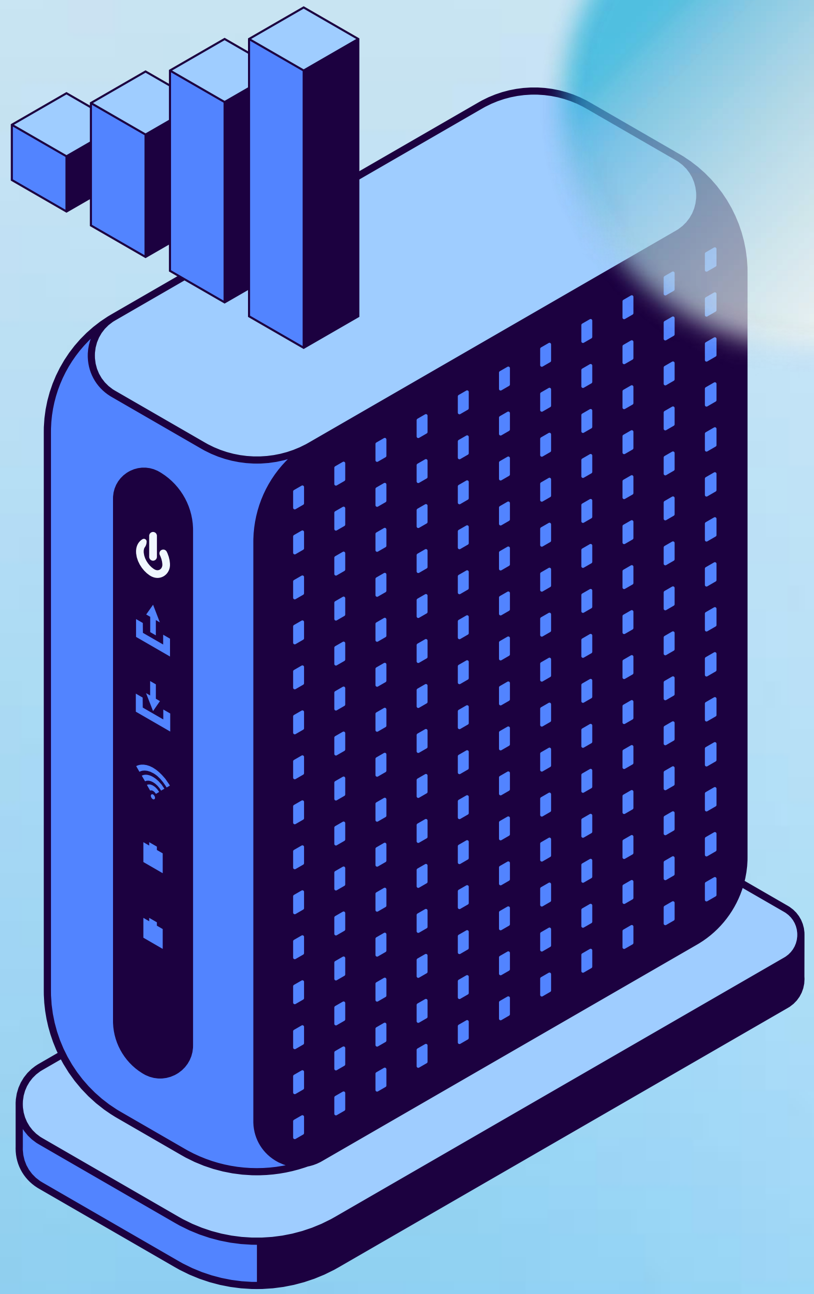
JAXR Architecture Overview

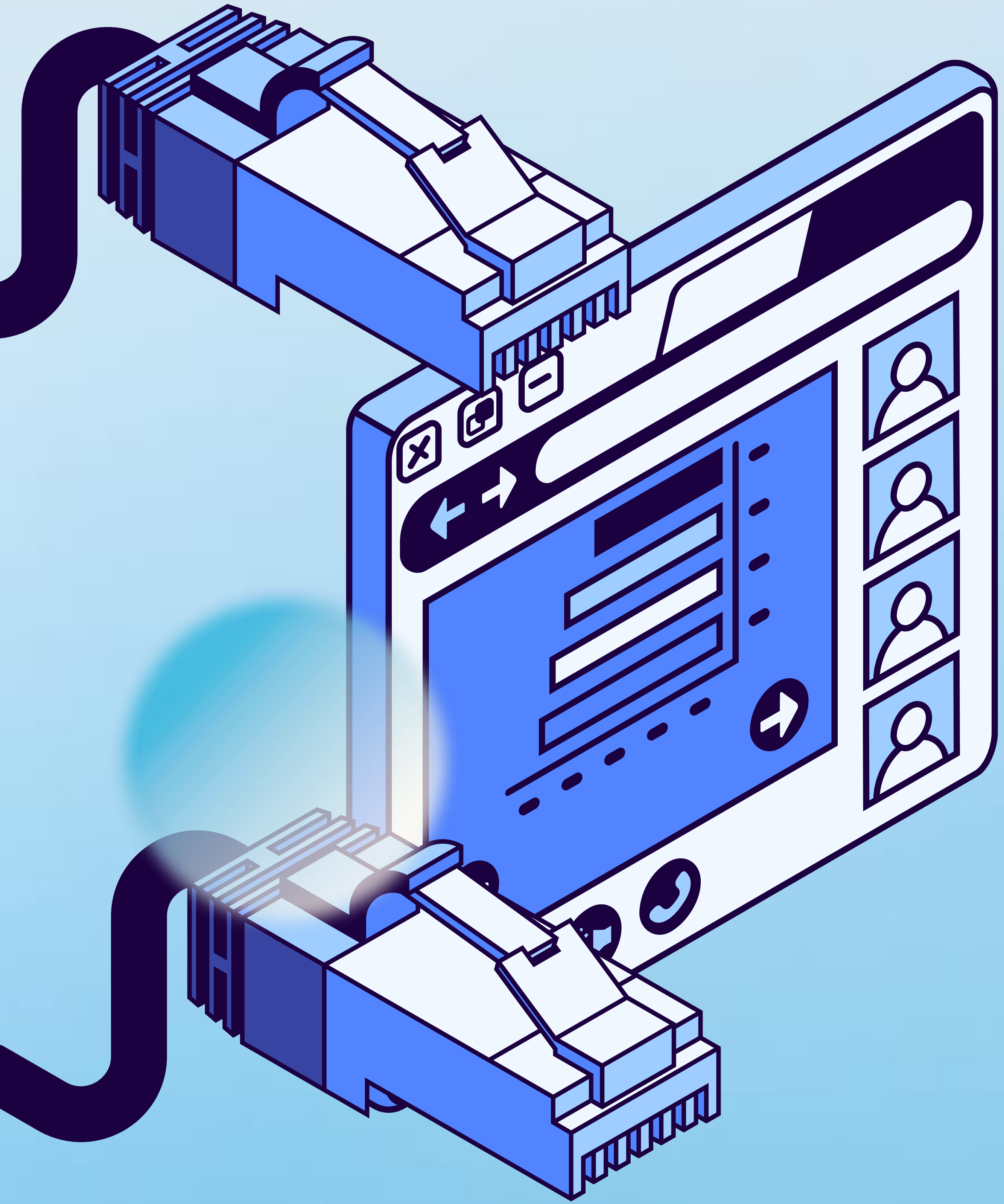
JAXR Client:

The JAXR client is the application or program that utilizes the JAXR API to interact with an XML registry. This client is responsible for sending requests to the registry, performing actions such as publishing services or querying for information. By using JAXR, developers can create applications that can connect to multiple types of registries without needing to understand the underlying complexities of each one.

JAXR Provider

The JAXR provider is an implementation of the JAXR API that facilitates communication between the JAXR client and specific XML registries, such as ebXML or UDDI. The provider acts as an intermediary, translating the requests made by the JAXR client into operations that the registry can understand. This allows for greater flexibility and interoperability, as the same client can work with different registries through their respective providers.





Key Interfaces in JAXR API

javax.xml.registry:

- Defines registry access interfaces and classes.
- Main interfaces:
- Connection: Establishes a client session with a registry.
- RegistryService: Allows the client to access the registry's functions.

javax.xml.registry.infomodel:

Defines the information model, which describes data types and relationships in the registry.

JAXR’s Use of Capability Levels

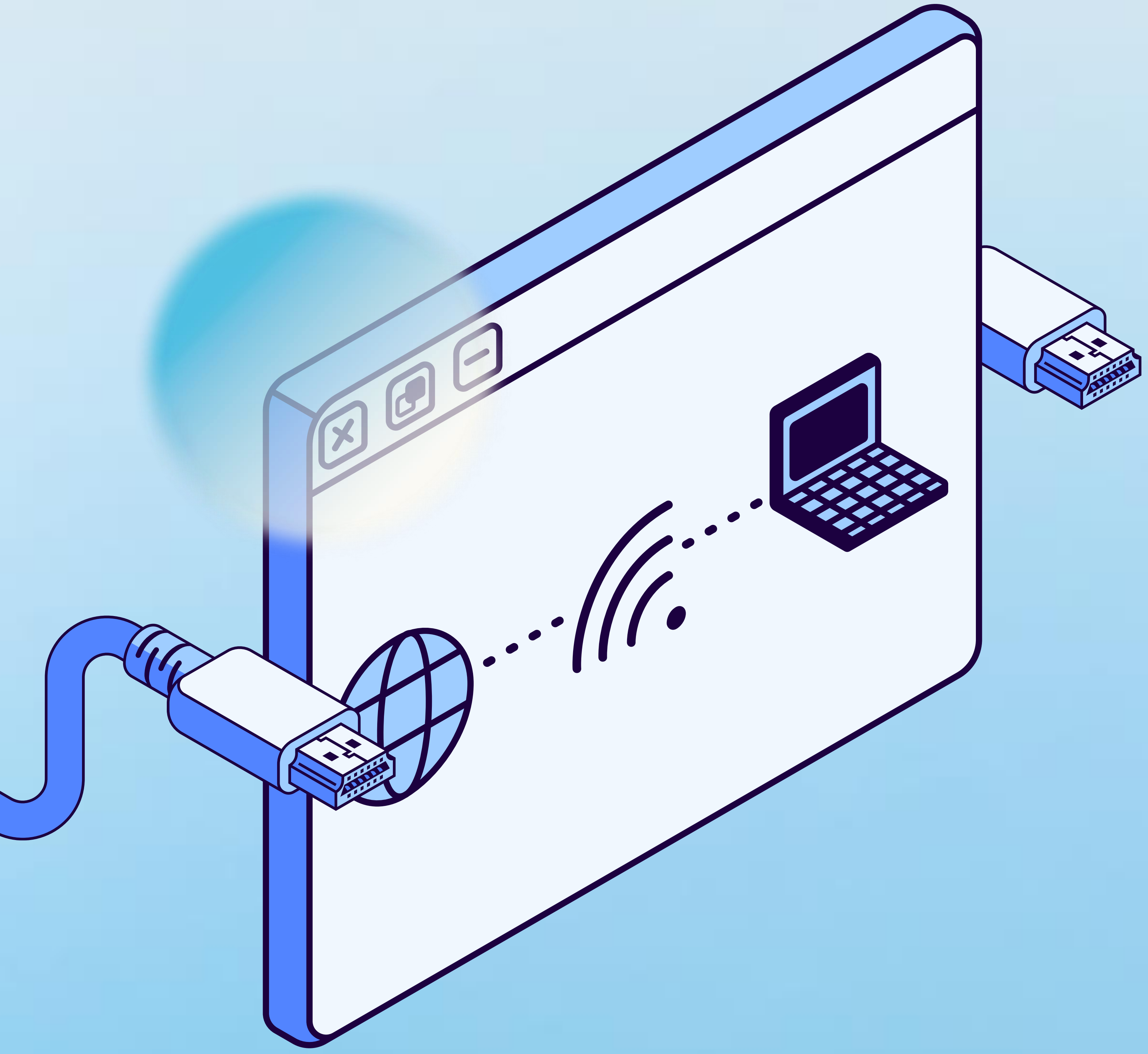
Level 0 and Level 1 Capability Profiles

- Level 0: Basic capability; suitable for simpler registry tasks.
- Level 1: Advanced capability; provides full access to ebXML registries.

Compatibility and Future Enhancements

- JAXR Level 1 can access enhanced registry functions, particularly relevant for ebXML.
- JAXR’s future versions will likely improve alignment with ebXML's evolving standards.

Benefits and Limitations of JAXR



Benefits:

- Standardization: Provides a unified interface across registry types.
- Flexibility: Supports a wide range of registry tasks, including search and data management.
- Interoperability: Allows Java-based applications to interact with registries on different platforms.

LIMITATIONS:

- Complexity: Some functions can be complex for simpler applications.
- Legacy Usage: Modern alternatives may offer more efficient, simpler solutions for registry access.

Thank You!

ROSHAN JEET KUMAR
2021UCS1670
SERVICE ORIENTED ARCHITECTURE