

## Java Message Service (JMS): Architecture

- API Standard from Oracle (Sun Microsystems)
  - Not an implementation standard
- Hub and Spoke Architecture
  - JMS Server
    - Message access control
  - JMS Clients
    - Use JMS API
    - Send/receive messages

```
graph TD; JMS[JMS Server] <--> JMSClientA((JMS Client A)); JMS <--> JMSClientB((JMS Client B)); JMS <--> JMSClientC((JMS Client C)); JMS <--> JMSClientN((JMS Client N)); JMS <--> ACL[ACL];
```

The diagram illustrates the JMS architecture. At the center is a blue rectangular box labeled "JMS Server". Four arrows point from four green circular boxes labeled "JMS Client A", "JMS Client B", "JMS Client C", and "JMS Client N" towards the central server. Above the central server is a small grey cylinder labeled "ACL". Two arrows point from the "ACL" cylinder to the central server: one from the top and one from the bottom.

## Java Message Service (JMS): Architecture

JMS is a standard API for building centrally-managed messaging applications.

The standard specifies the API, but it does not define how to implement services within the API. The standard does not address standards above the API level, such as specifying the wire format. Interoperability between vendor JMS implementations is not guaranteed.

JMS is a client/server ("hub-and-spoke") architecture: JMS requires a central server component where all produced messages are sent and where all consumers get their messages from. Even though the JMS architecture is hub and spoke, JMS clients communicate in peer-to-peer fashion: producers and consumers don't know about each other, they only communicate with the server.