# Endpoint and URI

1. Endpoint is an abstraction that models the end of a message channel through which a system can send or receive messages.
2. Endpoint is defined using URIs in Camel most of the time.
3. URI identifies the component and how that component is configured.  
   Then we can send/receive msgs to/from this Component.

# Message Channel

1. **Message Channel = Queue**.
2. Input Message Channel and Output Message Channel.
3. Usage:
   1. Used to group messages by production area, department or even project.
   2. Channels are also responsible for defining who receives messages.

# Message Router

1. A message router consumes messages from an input channel and, depending on a set of conditions, sends messages to one of a set of Output Channels.  
   A diagram of a diagram

   Description automatically generated
2. d

# Component Registry

1. Take one URI for example: from("**ftp**://rider.com/orders?username=rider&password=secret")  
   here ftp is component name (technical ftp is scheme here) which will be mapped to FtpComponent by looking up in Component Registry which is part of Camel Context.

# Component

1. A **Component** is defined by URI like this 🡪 from("**ftp**://**rider.com/orders**?**username=rider&password=secret**")  
   **ftp**: scheme, **rider.com/orders:** Context Path, **username=rider&password=secret** : Options  
   Here ftp is component name (technical ftp is scheme here) which will be mapped to **FtpComponent** by looking up in Component Registry which is part of Camel Context.
2. Component then works as factory for creating Endpoint based on the context path and options.