D-Bus là giao tiếp liên tiến trình, về mặt cấu trúc nó gồm các layer:

* libdbus: thư viện cấp thấp của dbus
* message bus daemon: được xây dựng trên libdbus, nó tạo môi trường giúp các application có thể giao tiếp với nhau, chuyển tiếp các message từ application này tới các application khác.
* Các thư viện dbus cấp cao hơn như libdbus-glib, libdbus-qt và dbus-python cung cấp các API cho user.

Các thành phần trong dbus:

Bus(daemon D-Bus):

* D-Bus cung cấp nhiều bus để các application có thể giao tiếp với nhau. Không thể truy cập các message được gửi trên 1 bus từ 1 bus khác, các application được kết nối tới cùng một bus có thể trao đổi message với nhau. Nhiều application có thể kết nối tới cùng 1 bus, một application có thể đồng thời kết nối tới nhiều bus.
* 2 loại bus được định nghĩa trước:
  + system bus: sử dụng cho những dịch vụ toàn hệ thống như quản lí truy nhập phần cứng. Những user có thể giao tiếp với nhau qua bus này.
  + session bus: sử dụng bởi các application của mỗi user
* Application có thể tạo thêm các bus nếu cần

Message:

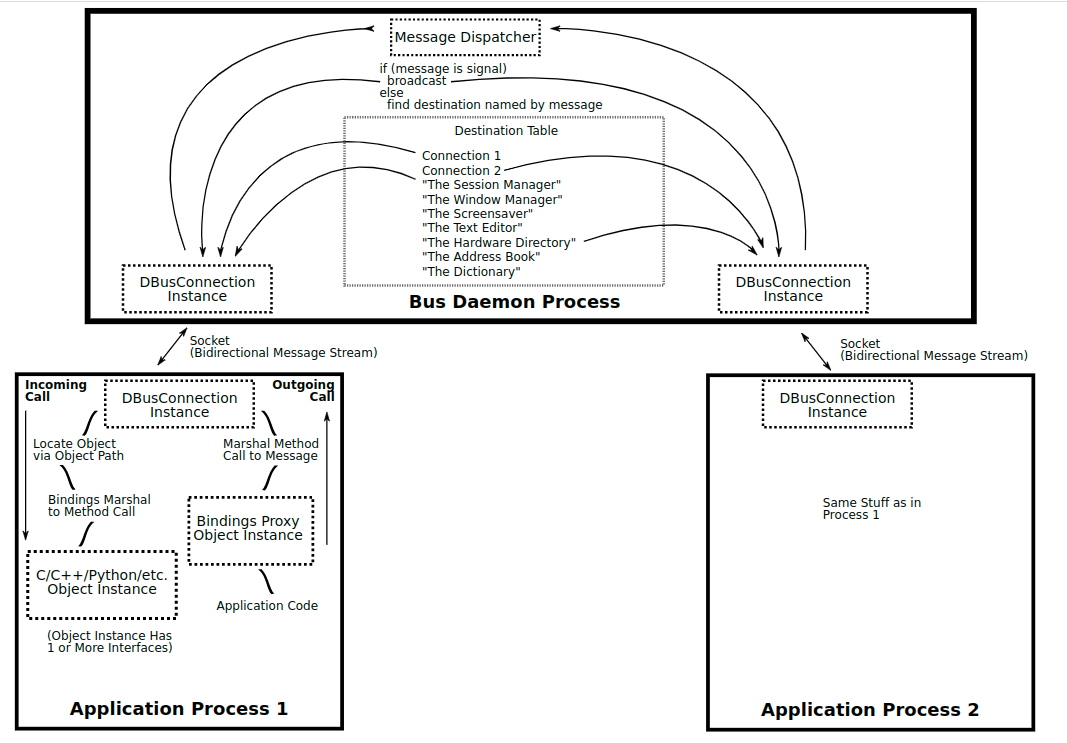
* Các application trên bus giao tiếp với nhau qua việc gửi nhận các message, các message có header xác định loại message, địa chỉ gửi và nhận.
* Các loại message: method calls, signal, respone của method, error respone khi gọi method.

Object:

* Nhiều application có thể nằm trên cùng một bus và một application có thể cung cấp nhiều object để gửi đi message. Các message xác định cụ thể địa chỉ của object bằng cách kết hợp 3 thành phần: interface, service và tên object.
* interface là 1 tập hợp các method và signal có thể được gửi trong bus, các application có thể tạo ra các interface gồm các giá trị: tên, tham số (nếu có) và giá trị trả về (nếu có).
* service đại diện cho kết nối giữa application và bus. Nếu một application có nhiều hơn một kết nối với một bus hoặc nếu nhiều phiên bản của cùng một application có thể hoạt động cùng một lúc, application đó sẽ cần sử dụng một tên service duy nhất cho mỗi kết nối. Thông thường, điều này được thực hiện bằng cách thêm process id vào tên service.
* tên object hay còn gọi là object path là địa chỉ được người dùng tự định nghĩa (ví dụ /Lockdoor/qt , /Lockdoor/rfid,...). Các object được cung cấp quyền truy cập vào các interface, một object có thể truy cập vào nhiều interface cùng một lúc.

Method call và signal:

* Method call là message yêu cầu thực thi method của một object cụ thể.
* Signal là message broadcast signal lên bus, các object theo dõi signal này sẽ đều nhận được thông tin về signal.



Hình ảnh bao quát về các thành phần trong D-Bus