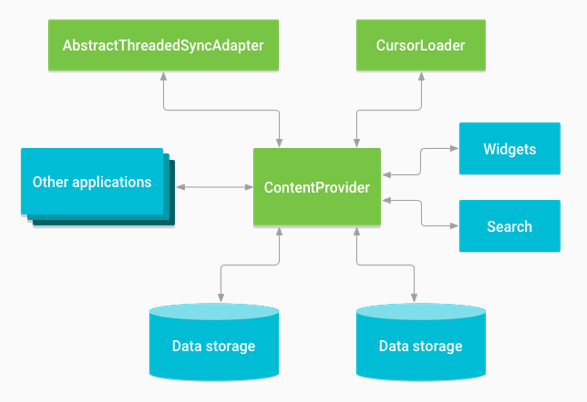
* Content provider có thể được chia sẻ giữa các ứng dụng, cho phép xem và sửa đổi các bản ghi như database, xml, flat file, từ các ứng dụng được cấp quyền.
  + Dùng để cung cấp dữ liệu phức tạp
  + Sao chép dữ liệu giữa các ứng dụng
  + Dùng để gợi ý tìm kiếm
  + Dùng để cung cấp dữ liệu cho widget
  + Không cần thiết nếu chỉ sử dụng trong chính ứng dụng hay cơ sở dữ liệu



* Truy cập dữ liệu từ content provider:
  + Clients -> CursorLoader -> ContentResolver -> ContentProvider
    - Tạo
    - Cập nhật
    - Xóa
    - Truy xuất
* Content URI(Mã định danh tài nguyên thống nhất) sử dụng để truy vấn content provider:
  + content://authority/path
    - VD: content://contacts\_info/users
  + content:// - thể hiện content URI
  + authority: Đại diện cho tên content provider
    - VD: phone, contacts, trinhlk.contactprovider,…
  + Path: Đại diện cho đường dẫn bản ghi
* Tạo content provider:
  + Bước 1: Tạo một lớp content provider kế thừa lớp ContentProvider
  + Bước 2: Xác định content provider URI
  + Triển khai 6 lớp trừu tượng từ lớp Content Provider
    - query() - Nhận được dữ liệu từ bản ghi và trả về dữ liệu dưới dạng đối tượng Cursor
    - insert() - Chèn một hàng mới vào content provider và trả về content URI cho hàng mới được chèn
    - update() - Cập nhật một hàng hiện có trong content provider và trả về số lượng hàng được cập nhật
    - delete() - Xóa các hàng trong content provider và trả về số lượng hàng đã xóa
    - getType() - Trả về loại dữ liệu MIME cho content URI đã cho
    - onCreate() - Khởi tạo provider. Hệ thống Android sẽ gọi phương thức này ngay lập tức sau khi tạo ra provider
  + Đăng ký content provider trong AndroidManifest.xml bằng cách sử dụng thẻ <provider>
* Cụ thể ví dụ như sau:
  + Tạo file UserProvider.java kế thừa lớp ContentProvider
  + Khai báo content provider trong ActivityManifest.xml
  + Tạo giao diện activity\_main.xml
  + Tạo file MainActivity.java xử lý
* Một số content provider có sẵn:
  + Browser: bookmarks, lịch sử duyệt web
  + CallLog: chi tiết các cuộc gọi
  + Contact: thông tin liên lạc
  + MediaStore: các tập tin đa phương tiện ở cả bộ nhớ trong và ngoài
  + Settings: các thiết lập mức hệ thống
  + Telephony: thông điệp SMS và MMS
* Các tham số của query():
  + Content URI xác định dữ liệu muốn truy cập
    - Ví dụ: content://contacts/people/5
  + Các tham số khác:
    - Các côt dữ liệu cần lấy: mProjection
    - Các tiêu chí lựa chọn
    - Tham số của các tiêu chí lựa chọn
    - Thứ tự sắp xếp kết quả trả về
* Lấy dữ liệu từ query dùng cursor:
  + Cursor là đối tượng chứa dữ liệu trả về từ query
* Chèn dữ liệu qua content provider:
  + ContentResolver.insert(Uri url, ContentValues values)
    - Trả về Uri của hàng vừa tạo
  + ContentResolver.bulkInsert(Uri url, ContentValues[] values)
    - Trả về số hàng tạo thành công
* Cập nhật dữ liệu bằng Content Provider
  + ContentResolver.update(Uri uri, ContentValues values, String where, String[] selectionArgs)
    - Tham số là đối tượng ContentValues chứa các giá trị đã cập nhật kèm với tiêu chí chọn các đối tượng cần sửa (selection criteria) giống như lúc truy vấn.
  + Chỉ cần thêm giá trị vào ContentValues đối với các cột có giá trị cần cập nhật.
  + Để xóa dữ liệu của một cột, đặt giá trị của cột đó thành null.
* Xóa dữ liệu bằng Content Provider:
  + ContentResolver.delete(): Tương tự truy vấn dữ liệu
* Một số kiểu dữ liệu của Content provider:
  + Text
  + Integer
  + Float
  + Có thể tự định dạng kiểu dữ liệu MIME riêng:
    - Ví dụ: text/html