

Using with

By [Raul Maldonado](#)

Motivations

You are creating a web framework using **Flask** to present some data findings through either visualizations, tables, or some other data related operation.

You would like to work with a database. However, there is no pre-built Object Relational Mapping (**ORM**) with Flask...because it's a “no batteries included” minimalistic python framework.

To resolve our pain-points, we use **SQLAlchemy**.

This is Not...

- a [Flask-SQLAlchemy](#) module overview, but for both [Flask](#) and [SQLAlchemy](#)

```
from flask import Flask, render_template, redirect, request
```

```
from sqlalchemy import create_engine
```

```
from sqlalchemy.ext.automap import automap_base
```

```
from sqlalchemy import inspect
```

```
from sqlalchemy.orm import Session
```

- In-depth of technical analysis and discussion of both modules
- Complex application to present data to web framework. Only a “Lightweight” application. [See here](#)**

Combining Flask and SQLAlchemy

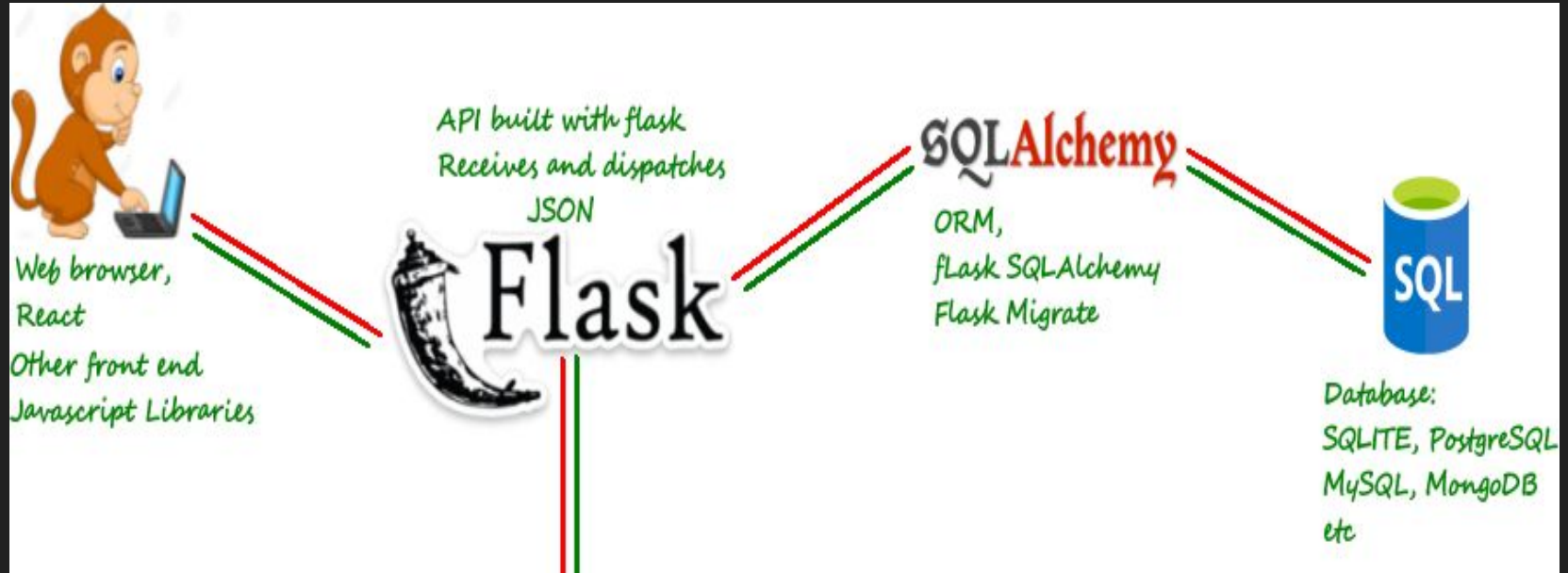


Image by Daniel Osaetin

SQLAlchemy

- SQLAlchemy is a popular ORM SQL toolkit for database manipulation
- Use with web frameworks to support SQL-like queries from an Object Oriented perspective with either CORE or ORM options
- Benefits:
 - Security
 - Convenience
 - DRY principle
 - Persistence
 - Performance
 - And more! (e.g. Start being one of the cool kids)

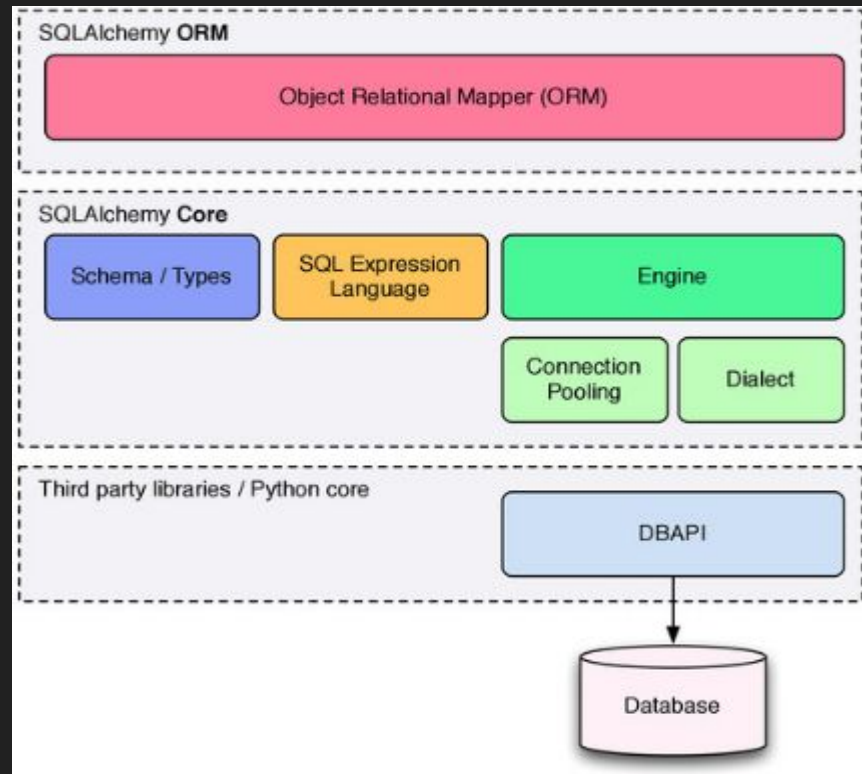


Image by SQLAlchemy website

Flask

- Popular lightweight/minimalist Python web framework
 - Not developed based on “Batteries-included Approach”
- Micro but extensible web framework by customization of open and compatible tools and libraries
 - Developer friendly because of the minimal framework + added features approach
 - Micro: “Keep core simple but extensible”
- Can have option to add database layer, form validation, upload handling, or other framework-like tools

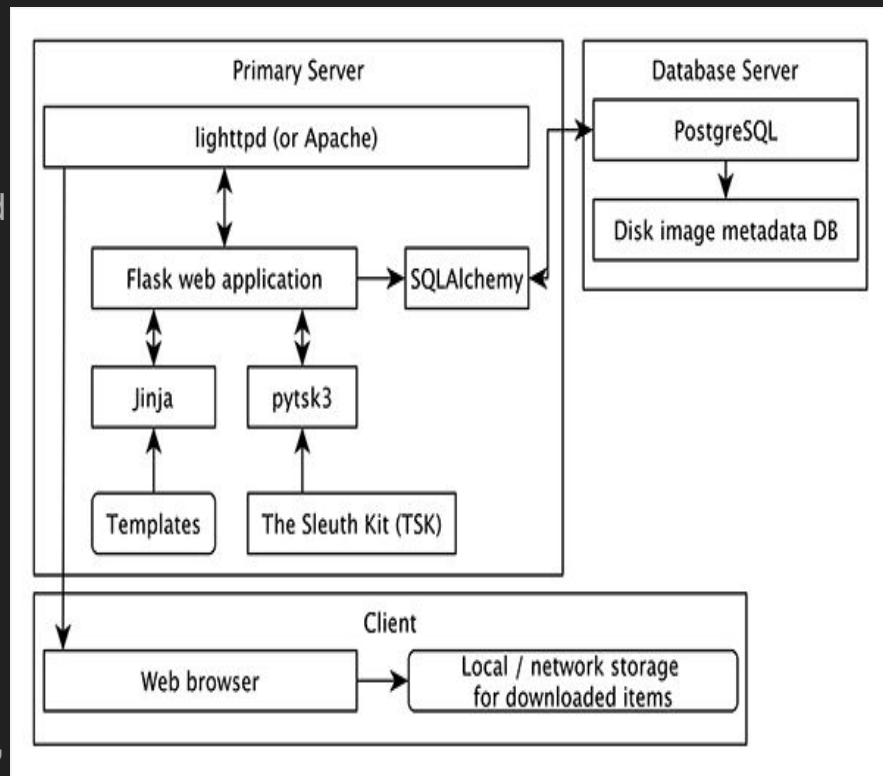


Image by Code{4}Lib