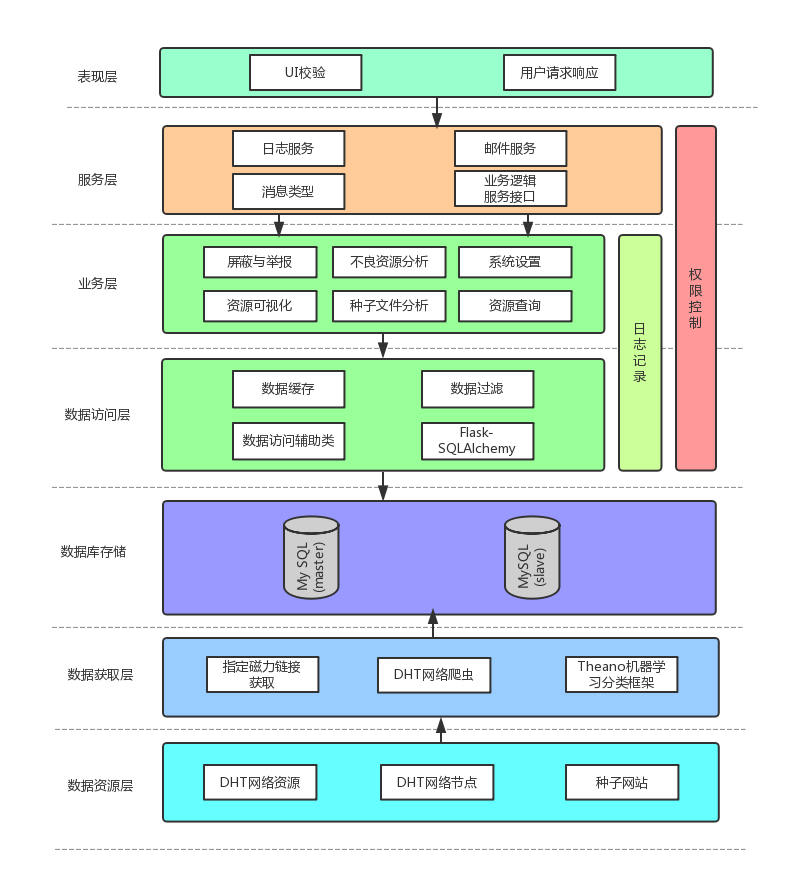
**第二章 实现方案**

**2.1 系统架构**

**2.1.1 系统架构分层视图**

本系统分为表现层、服务层、业务层、数据访问层、数据库存储、数据获取层、数据资源层六个层次，系统总体架构图如图2-1-1：

图2-1 系统架构设计图

**2.1.1.1表现层**

表现层主要负责与用户交互，接收用户的输入，将其转化成业务层能够理解的消息格式，并调用业务层相关的业务逻辑接口来处理用户的输入，最后将得到的结果转换成某种格式呈现给用户。展示层还需要检验用户输入数据的格式，并对错误的数据输入做出响应。**在Web应用中，展示层通常位于浏览器中。**

**2.1.1.2服务层**

服务层是处于应用程序业务层和表现层之间的应用程序边界,边界可能是很薄的一层类设计或者是分布式服务网络跃点。它由表现层直接调用，对业务逻辑层接口很清楚，组织业务逻辑 ，适配表现层。

服务层的职责：

服务层处于应用程序和业务层之间，应用边界，使得他们解耦，利用第三个对象破坏两个对象直接的依赖，并转化适配领域对象(Domain Object ) 和视图对象(View Object) 差异。

服务层隐藏了业务逻辑层的细节，其内部需要组织业务微服务，提供更宏观，面向表现层的服务逻辑，并利用契约接口暴露，包装。系统所有的交互都是从表现层进入。

**本系统的服务层向表现层提供了业务逻辑层的接口，并封装了与业务逻辑无关的日志记录和邮件服务等。**

**2.1.1.3业务层**

* 业务逻辑层的责任：
* 负责处理系统的业务逻辑；
* 负责对用户定义的流程进行建模；
* 负责数据访问层和展示层的通讯；
* 负责将错误信息返回给展示层。
* 业务逻辑层的组成包括：
* 业务实体

业务实体提供对业务数据及相关功能（在某些设计中）的状态编程访问。

业务实体可以使用具有复杂架构的数据来构建。这种数据通常来自数据库中的多个相关表。

业务实体数据可以作为业务过程的部分I/O参数传递。

业务实体可以是可序列化的，以保持它们的当前状态。例如，应用程序可能需要在本地磁盘、桌面数据库（如果应用程序脱机工作）或消息队列消息中存储实体数据。

业务实体不直接访问数据库。全部数据库访问都是由相关联的数据访问逻辑组件提供的。

业务实体不启动任何类型的事务处理。事务处理由使用业务实体的应用程序或业务过程来启动。在应用程序中表示业务实体的方法有很多（从以数据为中心的模型到更加面向对象的表示法）：

1.XML

2.通用Data Set

3.有类型的Data Set

4.自定义业务实体组件（本系统将采用自定义的业务实体组件）

* 业务逻辑组件

业务逻辑组件封装业务逻辑和应用状态。业务逻辑是一种集中于实现业务规则和行为的应用逻辑，同时包括维护全局的一致性，例如数据合法性验证。业务逻辑组件应该设计成容易测试的、独立于表现层和数据访问层。本系统的业务逻辑组件可划分为：

1. 系统管理

2. 种子文件资源分析

3. 指定资源的查询

4. 资源分类学习

5. 资源的可视化处理

6. 不良资源报表生成

**2.1.1.4数据访问层**

数据访问层：又称为DAL层，有时候也称为是持久层，其功能主要是负责数据库的访问。简单的说法就是实现对数据表的Select（查询），Insert（插入），Update（更新），Delete（删除）等操作。如果要加入ORM的元素，那么就会包括对象和数据表之间的 mapping （映射），以及实体对象的持久化。

本系统的数据访问层使用了Flask-SQLALchemy框架，将数据库表映射成数据实体，并封装其持久化的操作。除此之外，数据访问层还封装了DHT网络信息和种子网站获取接口的调用，通过辅助类来实现网络层信息的本地存储。

**2.1.1.5数据获取层**

数据获取层：其功能主要是负责利用爬虫程序进行数据的收集和进行筛选，并将数据填充到数据库表单中，为数据访问层的API进行CRUD操作提供数据源。本系统的数据获取层主要涉及到DHT网络资源数据的爬取，种子网站的数据采样。

此层复杂定义数据质量、围绕数据频率、每次抓取的数据大小和数据过滤器：

 完整地识别所有必要的数据元素

 以可接受的新鲜度提供数据的时间轴

 依照数据准确性规则来验证数据的准确性

 采用一种通用语言（数据元组满足使用简单业务语言所表达的需求）

 依据数据一致性规则验证来自多个系统的数据一致性

 在满足数据规范和信息架构指南基础上的技术符合性

本系统的数据获取主要利用python + bittorrent和 phantomjs 以及Bittorrent 协议实现种子获取程序，其中涉及到文本，图片和视频流等数据类型的获取，该层的最终目的是尽量完整和高效地获取网络中的信息数据，并提高DHT网络中种子文件获取的覆盖率。

**2.1.1.6数据资源层**

数据资源层其主要描述关于系统中的数据来源，例如，在不同系统中的数据处理范围是不相同的，该层的主要是目的是对本系统中进行处理的数据进行限定。

在数据资源层中我们主要考虑到网络数据中的一下特性：

* 数据可用性
* 数据关键性
* 数据真实性
* 数据共享和发布
* 数据存储和保留
* 数据提供程序约束（政策、技术和地区）
* 社交媒体使用条款

**本系统主要考虑DHT网络中的文本，图像，视频和种子网站存储的种子信息，提供系统运行所需的元数据。**

**2.1.2 系统架构物理部署视图**

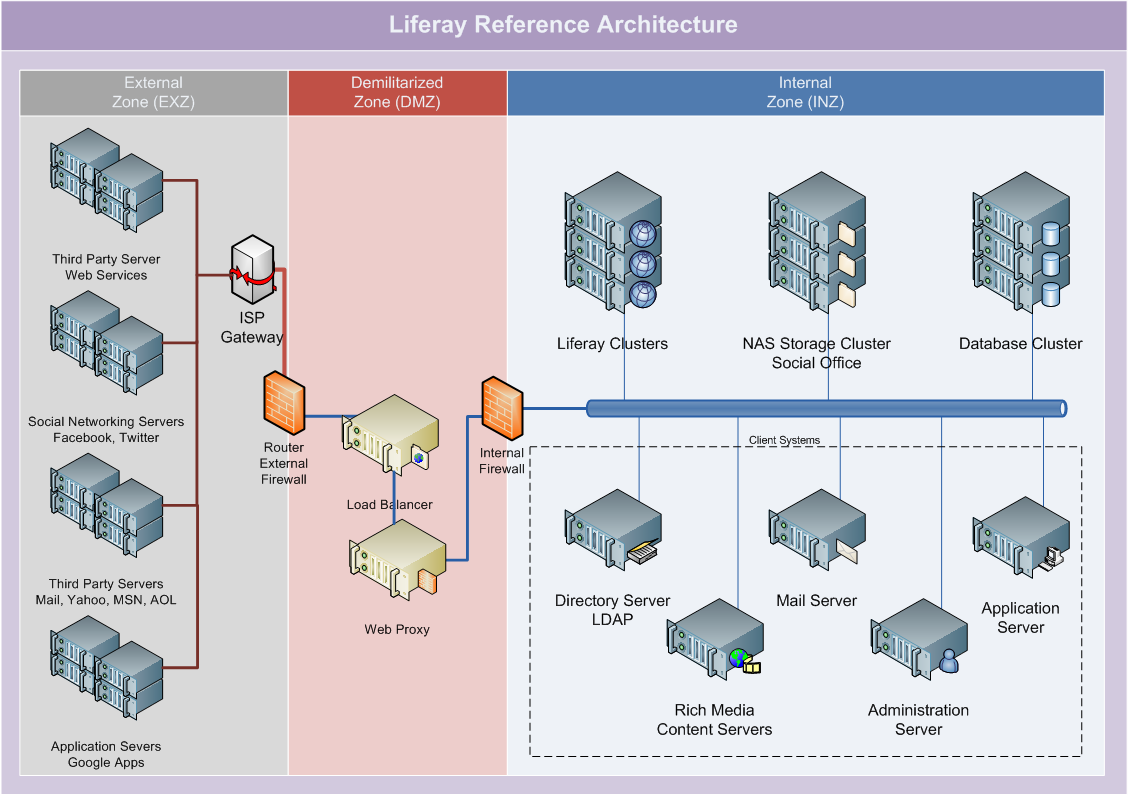


图2-2 系统整体物理部署视图

**2.1.3 技术选择**

**2.1.3.1 前端开发相关技术**

* **HTML5**

HTML5是HTML下一个主要的修订版本，现在仍处于发展阶段。目标是取代1999年所制定的HTML4.01和XHTML1.0标准，以期能在互联网应用迅速发展的时候，使网络标准达到符合当代的网络需求。广义论及HTML5时，实际指的是包括HTML、CSS和JavaScript在内的一套技术组合。

本作品使用HTML的Canvas API进行二维绘图，在浏览器端进行动画渲染，实现数据可视化。

* **CSS**

级联样式表，用来为结构化文档添加样式，进行网页风格设计。

本作品前端样式是基于Twitter的Bootstrap框架(front-end framework)定制的Flat风格UI。Bootstrap是一套基于HTML、CSS、JavaScript的简洁灵活的流行前端框架及交互组件集。利用Bootstrap可较容易实现Web网站跨设备、跨浏览器、响应设计等需求。

* **JavaScript**

JavaScript是一种广泛用于客户端网页开发的脚本语言。它最初由网景公司的Brendan Eich设计，是一种动态、弱类型、基于原型的语言，内置支持类别。除前端的基本验证外，本作品还使用JavaScript进行动画绘制渲染，实现数据可视化。数据可视化主要利用以下两个JavaScript程序库：

* **jQuery**是一套跨浏览器的JavaScript程序库，强化HTML与JavaScript之间的操作。全球前10000个访问最高的网站中，有59%使用了jQuery，是目前最受欢迎的JavaScript库。利用jQuery程序库可以让 JavaScript 操纵 DOM 元素、事件处理等行为更为方便。
* **Chart.js**是一个使用HTML5 Canvas元素实现的图表组件，支持6种不同的图表类型。Chart.js 支持所有主流浏览器，通过polyfills提供对IE7/8的支持，没有依赖其它实现，并提供加载自定义选项。
* **AngularJS** 是一款开源JavaScript库，由Google维护，用来协助单一页面应用程序运行的。它的目标是通过MVC模式（MVC）功能增强基于浏览器的应用，使开发和测试变得更加容易。

**2.1.3.2 服务端开发相关技术**

* **Python**

Python 是一种面向对象、直译式电脑编程语言，具有近二十年的发展历史，成熟且稳定。它包含了一组完善而且容易理解的标准库，能够轻松完成很多常见的任务。它的语法简捷和清晰，尽量使用无异义的英语单词，与其它大多数程序设计语言使用大括号不一样，它使用缩进来定义语句块。

与Scheme、Ruby、Perl、Tcl等动态语言一样，Python具备垃圾回收功能，能够自动管理存储器使用。它经常被当作脚本语言用于处理系统管理任务和网络程序编写，然而它也非常适合完成各种高级任务。Python虚拟机本身几乎可以在所有的作业系统中运行。使用一些诸如py2exe、PyPy、PyInstaller之类的工具可以将Python源代码转换成可以脱离Python解释器运行的程序。

Python支持命令式程序设计、面向对象程序设计、函数式编程、面向侧面程序设计、泛型编程多种编程范式。

本项目采用Python作为主要的开发语言，可以较好地实现DHT网络资源的获取、分类和分析，进而对解析得到的不良信息进行分析和识别。

* **MySQL**

MySQL是一款性能高、成本低、可靠性好的开源数据库，使用C和C++编写，并使用了多种编译器进行测试，源代码具有较好的可移植性。它为多种编程语言提供了API，其中就包括了Python。MySQL还支持多线程、多用户的操作，充分利用本地系统资源。

另一方面，MySQL优化的查询算法，可以有效地提高查询速度，同时还能够处理拥有上千万条记录的大型数据库。MySQL既能够作为一个单独的应用程序在客户端服务器网络环境中运行，也能够作为一个程序库而嵌入到其他的软件中。最后，MySQL提供有用于管理、检查、优化的管理工具，最大程度的简化开发人员的管理与维护。因此，MySQL被广泛地应用在Internet上的中小型网站中。随着MySQL的不断成熟，它也逐渐用于更多大规模网站和应用中。

本项目采用MySQL作为数据库存储系统，存储从DHT网络中获取的磁力链接、分析得到的种子文件摘要等。

* **Sphinx**

一个基于SQL的全文检索引擎，可以结合MySQL做全文搜索，它可以提供比数据库本身更专业的搜索功能，使得应用程序更容易实现专业化的全文检索。Sphinx特别为一些脚本语言设计搜索API接口，如PHP，Python，Perl，Ruby等，同时为MySQL也设计了一个存储引擎插件。

Sphinx单一索引最大可包含1亿条记录，在1千万条记录情况下的查询速度为0.x秒（毫秒级）。Sphinx创建索引的速度为：创建100万条记录的索引只需 3～4分钟，创建1000万条记录的索引可以在50分钟内完成，而只包含最新10万条记录的增量索引，重建一次只需几十秒。

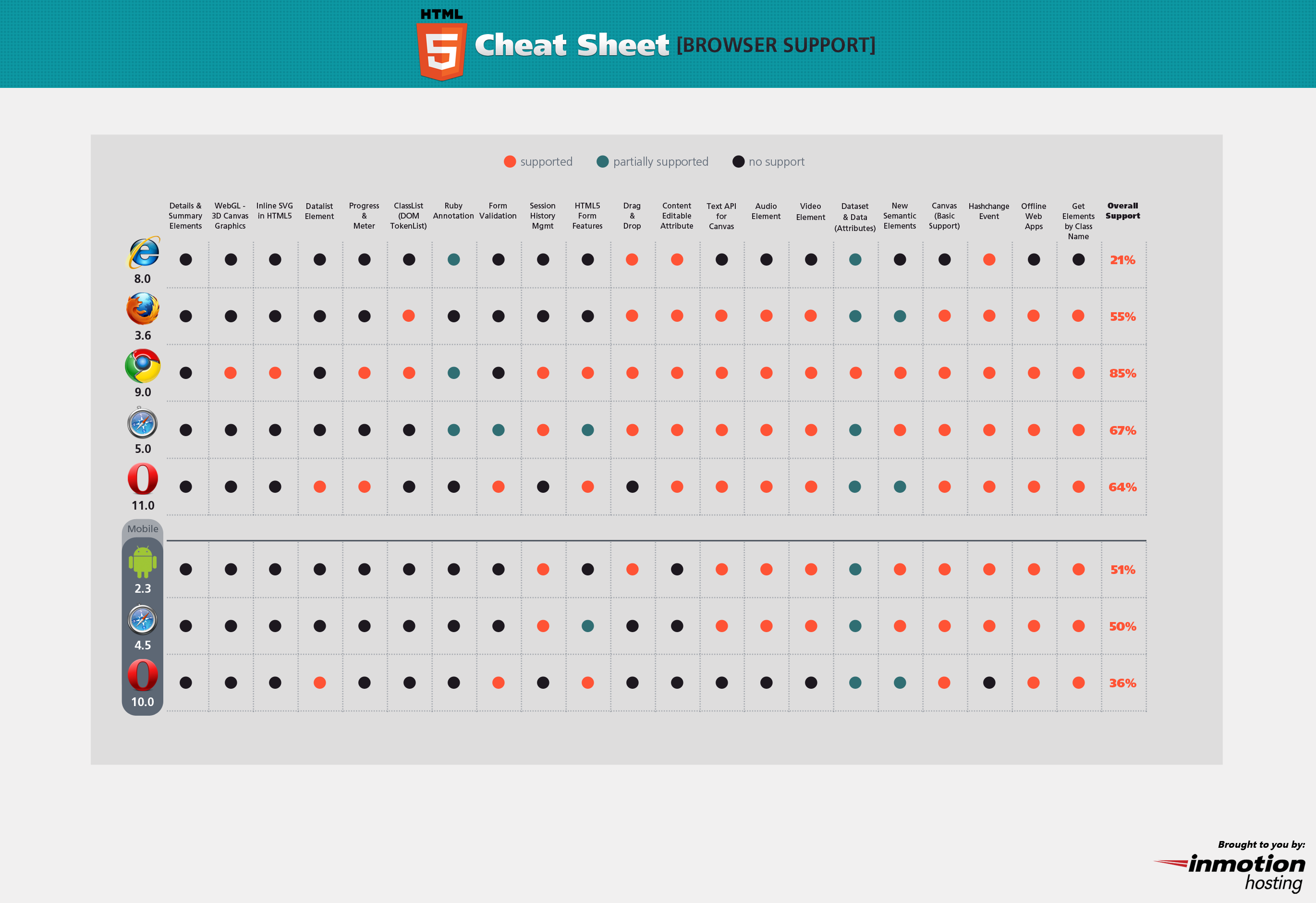
* **Selenium**

elenium 是一个用于Web应用程序测试的工具。Selenium测试直接运行在浏览器中，就像真正的用户在操作一样。支持的浏览器包括IE（7, 8, 9, 10, 11），Mozilla Firefox，Safari，Google Chrome，Opera等。这个工具的主要功能包括：测试与浏览器的兼容性——测试你的应用程序看是否能够很好得工作在不同浏览器和操作系统之上。测试系统功能——创建回归测试检验软件功能和用户需求。支持自动录制动作和自动生成 .Net、Java、Perl等不同语言的测试脚本。框架底层使用JavaScript模拟真实用户对浏览器进行操作。测试脚本执行时，浏览器自动按照脚本代码做出点击，输入，打开，验证等操作，就像真实用户所做的一样，从终端用户的角度测试应用程序。使浏览器兼容性测试自动化成为可能，尽管在不同的浏览器上依然有细微的差别。

**2.1.4系统运行环境**

**2.1.4.1浏览器**

系统采用 HTML5、响应式网页设计等技术，在用户跨设备、跨浏览器访问时能提供当前使用环境最好的用户体验。跨平台、跨浏览器详细支持信息如下图 3-5浏览器兼容性：



**图2-3 浏览器兼容性**

**3.1.4.2 服务器**

|  |  |
| --- | --- |
| 操作系统 | CentOS 7 |
| Python环境 | Python 2.7/3.6 |
| 数据库 | MySQL |
| Flask 环境 | Flask 0.10.2 |

**表2-1 服务器运行环境**

**2.2 功能模块**

待整合