即时通讯工具

简要介绍

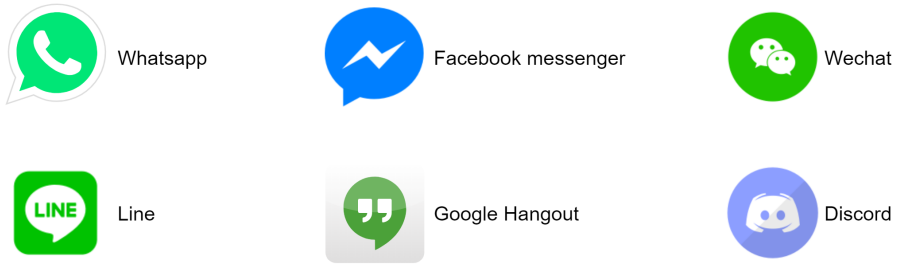


Figure 12-1

几乎每个人都使用聊天应用程序。
显示了市场上一些最流行的应用程序

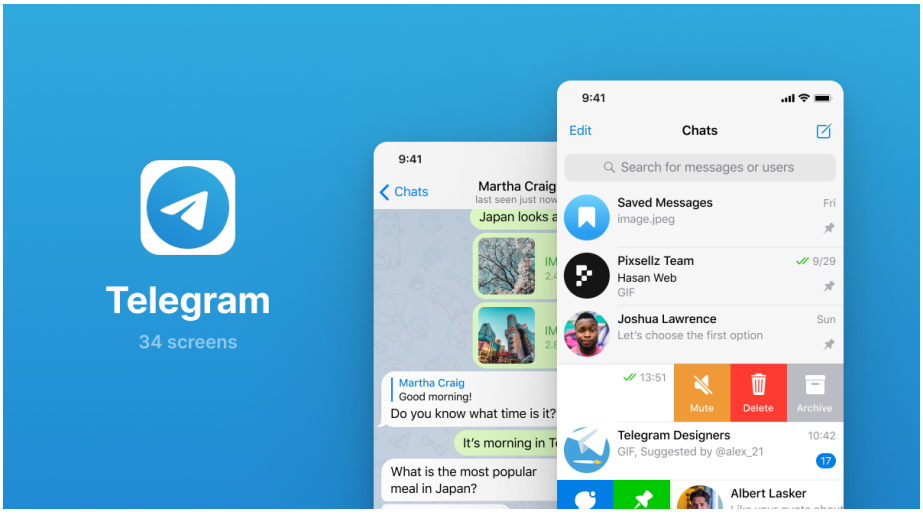
功能列表

- 具有低交付延迟的一对一聊天
- 小群聊天
- 在线状态
- 只支持消息
- 多设备支持。同一个账号可以同时登录多个账号。
- 推送通知

原型设计

整体

完全参考 Telegram 设计



消息列表

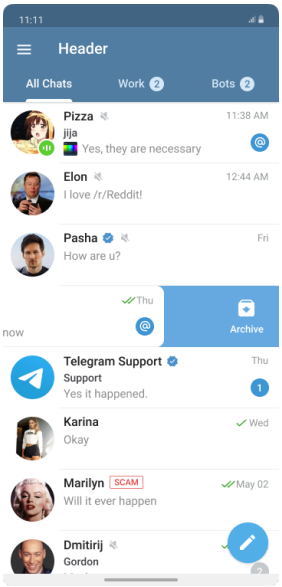


Figure 1: chat-list

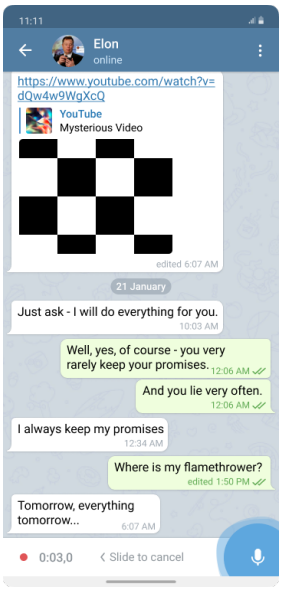


Figure 2: chat

消息框

主菜单

个人信息

系统架构设计

客户端和服务器如何通信

在聊天系统中，客户端可以是移动应用程序或 Web 应用程序。客户端之间不直接通信。相反，每个客户端都连接到支持上述所有功能的聊天服务。让我们专注于基本操作。聊天服务必须支持以下功能：

- 接收来自其他客户端的消息。
- 为每条消息找到正确的收件人并将邮件转发给收件人。
- 如果收件人不在线，则将该收件人的消息保留在服务器上，直到在线。

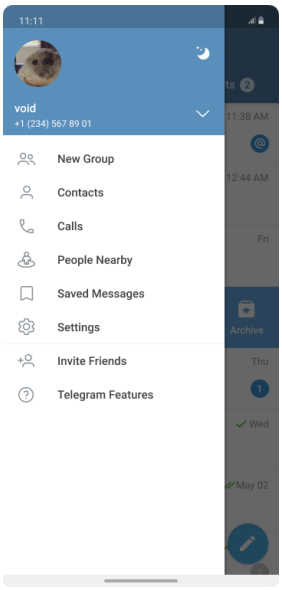


Figure 3: main-menu

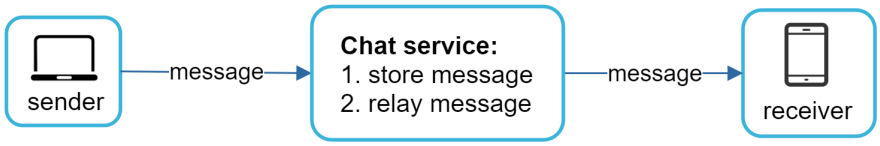


Figure 12-2

客户端（发送者和接收者）与聊天服务之间的关系。

当客户端打算开始聊天时，它会使用一个或多个网络协议连接聊天服务。 对于聊天服务，网络协议的选

发送端

- 大多数客户端/服务器应用程序的请求由客户端发起。
- 当发送者通过聊天服务向接收者发送消息时，它使用了久经考验的 **HTTP** 协议，这是最常见的 **Web** 协议。
- 客户端打开与聊天服务的 HTTP 连接并发送消息，通知服务将消息发送给接收方。keep-alive 对此非常有效，标头允许客户端保持与聊天服务的持久连接还减少了 TCP 握手的次数。

接收端

由于 HTTP 是客户端发起的，因此从服务器发送消息并非易事。

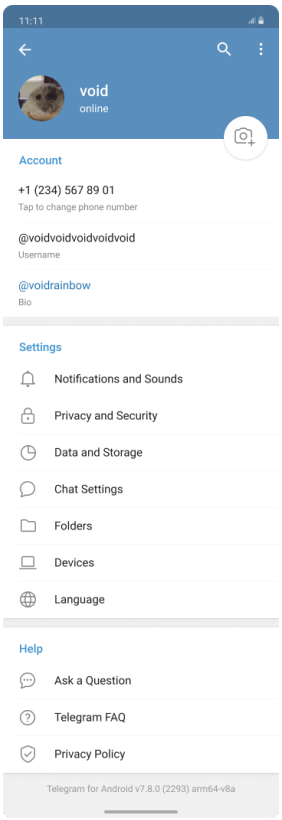


Figure 4: profile

多年来，许多技术用于模拟服务器发起的连接：轮询、长轮询和 WebSocket。

WebSocket

- WebSocket 是从服务器向客户端发送异步更新的最常见解决方案。

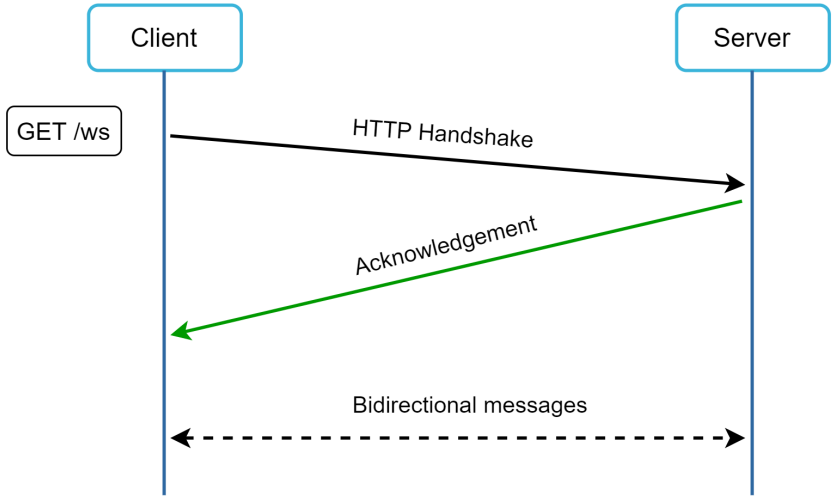


Figure 12-5

- WebSocket 连接由客户端发起。它是双向的和持久的。它从 HTTP 连接开始，可以通过一些定义明确的握手“升级”到 WebSocket 连接。通过这种持久连接，服务器可以向客户端发送更新。即使有防火墙，WebSocket 连接通常也能正常工作。这是因为它们使用 HTTP/HTTPS 连接也使用的端口 80 或 443。
- 在发送方，HTTP 是一个很好的协议，但由于 WebSocket 是双向的，所以没有理由不将它也用于发送。

WebSockets (ws) 如何用于发送方和接收方。

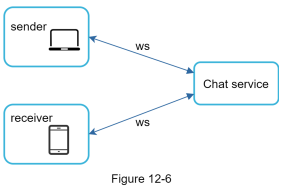


Figure 5: websocket

- 通过使用 WebSocket 进行发送和接收，它简化了设计并使客户端和服务器上的实现更加直接。由于 WebSocket 连接是持久的，因此有效的连接管理在服务器

端至关重要。

- WebSocket 作为客户端和服务器之间的主要通信协议是为了进行双向通信，，其他一切都不一定是 WebSocket。
- 聊天应用程序的大多数功能（注册、登录、用户配置文件等）都可以使用 HTTP 请求/响应方法。

无状态服务、有状态服务和第三方集成

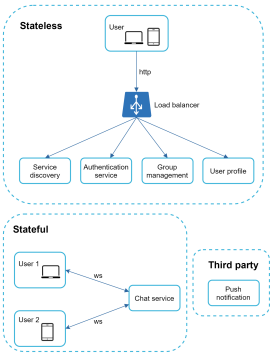


Figure 12-7

推送通知是最重要的第三方集成。 这是一种在新消息到达时通知用户的方式，即使应用程序没有运行也

储存

- 两种类型的数据。
  1. 通用数据，例如用户个人资料、设置、用户好友列表。这些数据存储在健壮可靠的关系数据库中。
  2. 聊天历史数据。键值对存储。允许轻松的水平扩展。键值存储为访问数据提供了非常低的延迟。

数据模型

message	
message_id	bigint
message_from	bigint
message_to	bitint
content	text
created_at	timestamp

Figure 12-9

group_message	
channel_id	bigint
message_id	bigint
user_id	bigint
content	text
created_at	timestamp

Figure 12-10

技术选型

前端

Flutter 是 Google 的一个开源框架，用于从单个代码库构建漂亮的、本地编译的多平台应用程序。

后端

Django 是一个高级 Python Web 框架，它鼓励快速开发和简洁、实用的设计。

Django REST framework 用于构建 Web API 的强大而灵活的工具包。

数据库

- 关系型数据库
  1. **MariaDB** 是 MySQL 关系数据库管理系统的社区开发、商业支持分支。旨在 免费和开源软件 下 GNU 通用公共许可证。MariaDB 旨在保持与 MySQL 的高度兼容性，确保具有库二进制奇偶校验和与 MySQL API 和命令的精确匹配的直接替换功能。
- NOSQL
  1. **MongoDB** 是一个 源代码可用 的跨平台 面向文档的数据库 程序。被归类为 NoSQL 数据库程序，它使用 JSON 的文档模式。MongoDB 根据服务器端公共许可证 (SSPL) 获得许可。