

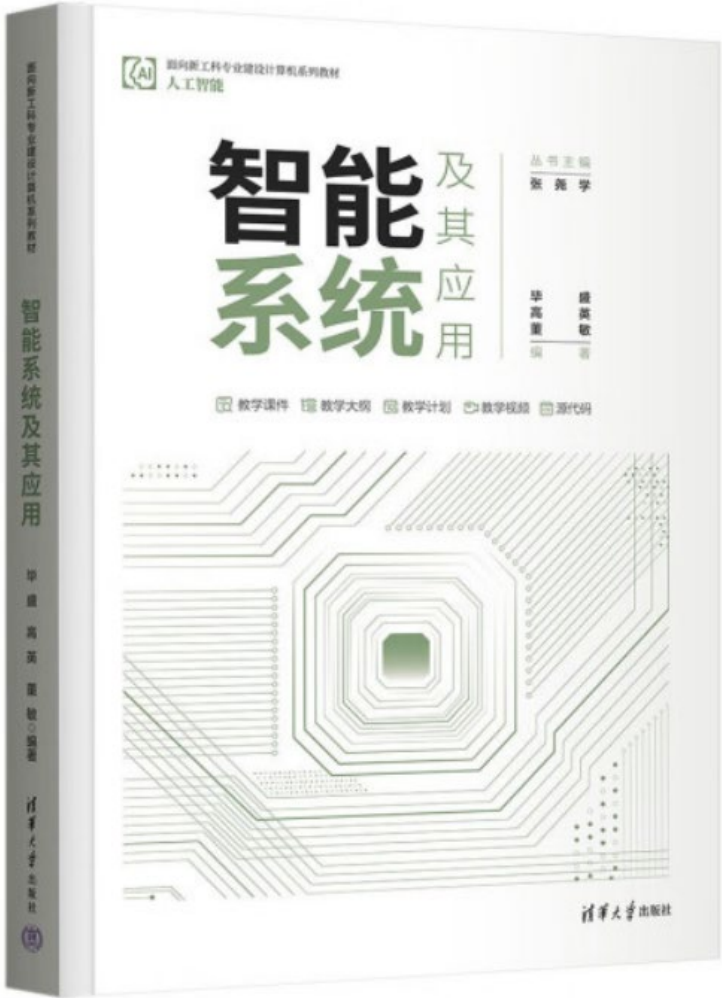


第6章 应用软件开发

智能系统及其应用
配套PPT



书名	书号	作者	出版社
智能系统及其应用	ISBN 978-7-302-60969-8	毕盛 高英 董敏	清华大学出版社





第6章 操作系统软件框架

6.1 程序开发

6.2 网络通信介绍

6.3 软件框架及常见组件

6.1 程序开发

6.1.1 常见开发语言

1.C语言

C是一种通用的编程语言，广泛用于系统软件与应用软件的开发，在实际项目中，C语言主要用于较底层的开发。C语言是一种结构化的语言，提供的控制语句具有结构化特征。C语言能以简易的方式编译、处理低级存储器。C语言是仅产生少量的机器语言以及不需要任何运行环境支持便能运行的高效率程序设计语言。与其他高级语言相比，C语言可以生成高质量和高效率的目标代码，故通常应用于对代码质量和执行效率要求较高的嵌入式系统程序的编写。在编程领域中，C语言的运用非常之多，它兼顾了高级语言和汇编语言的优点，拥有经过了漫长发展历史的完整的理论体系，在编程语言中具有举足轻重的地位。

2.C++语言

C++语言是一种高级语言，进一步扩充和完善了C语言，是一种面向对象的程序设计语言。C++可运行于多种平台上，如Windows、MAC操作系统以及UNIX的各种版本。C++是一种使用非常广泛的计算机编程语言，C++作为一种静态数据类型检查的、支持多范型的通用程序设计语言，能够支持过程化程序设计、数据抽象化、面向对象程序设计、泛型程序设计、基于原则设计等多种程序设计风格。C++的编程领域众广，常用于系统开发，引擎开发等应用领域，深受广大程序员的喜爱。C++不仅拥有计算机高效运行的实用性特征，同时还致力于提高大规模程序的编程质量与程序设计语言的问题描述能力。

3.Python语言

Python语言，它是一个动态类型的面向对象编程语言，在1989年设计出来，Python的语法简单因此非常易学，对初学者很友好。在开发时不用去管一些麻烦的小细节例如指针、内存管理和类型等。Python语言常常用于人工智能和机器学习，大多数人工智能和机器学习的框架和算法都是用Python来实现的。但是由于Python语言是动态类型的编程语言，所以它的稳定性不佳，在编译的时候常常不能告诉出了什么错；由于是一种解释性语言，Python运行速度比C++慢很多。

4.Java语言

Java语言用的人很多，它是一个面向对象而且是静态类型的编程语言，Java语言是1994年Sun公司开发出来，由于Sun公司在2010年被Oracle收购以后就成了Oracle旗下的产品。Java非常健壮和安全，并且是一个完全面向对象的语言，逻辑复用性强，灵活性好；它不像C++有很多需要考虑到的小细节，有垃圾回收器，所以不用去管指针，内存泄漏等问题。Java语言生态非常好，它有很多实践和解决方案可以参考。Java的应用非常广，很多时候可以用来写网站的后端，也可以用来写安卓的App，还可以用在大数据方面例如Kafka软件，用Java的公司也很多，例如阿里巴巴，美团，Uber和Google等。

5.Go语言

Go语言，由于官方网站是Golang.org，所以有时也会称它Golang,2007年Google开始开发并且在2013年正式发布出来，是一个年轻的静态类型编程语言，比较简单易学适合初学者，由于可以直接编译成二进制码在机器上运行，因此运行速度快。Go在语言层面上支持多核并发，这让它在云计算和网络通信方面有优势，由于Go有特别多开箱即用的功能以及对其他技术的支持，它特别适合用来开发网站的后端，现在又很多公司在用Go，比如字节跳动，Google，腾讯和Facebook等。但由于Go语言相对年轻，所以开发生态相对于老的语言来说还没有那么好。

6. SQL语言

SQL语言（Structured Query Language），这个语言专门用在数据库方面，用于对关系型数据库进行交互和管理，每个应用开发都常常有数据库，而大部分数据库又是关系型数据库，所以SQL在数据库管理方面很重要。网站后端会用到数据库，所以SQL也是后端开发很重要的一个语言。SQL语法简洁，所以上手简单。

7. C#语言

C#是2000年微软开发的一个静态类型编程语言，它全面支持面向对象编程，它安全性好和.Net一起可以支持多线程与并发，它具有垃圾回收站，所以不需要担心内存泄漏的问题，由于C#支持lambda表达式所以可以用函数式编程，还可以用C#和ASP.Net一起开发网站的后端，可以开发Windows的桌面App，同时C#也是开发Unity的主要语言，很多公司在用C#语言，例如微软和惠普等。C#的缺点是他比C++运行速度慢，并且C#太依赖.Net框架，所以只能在Windows下工作。

8. JavaScript

JavaScript (JS) 是一种具有函数优先的轻量级，解释型编程语言。虽然它是作为开发Web 页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，例如 Node.js。JavaScript是个动态类型语言，动态类型语言就是在写代码时，每一个对象和每一个变量是并不需要指定它是什么类型。JavaScript语言主要用于前端开发，用于控制网页行为，像 React、Vue 和 Angular 这样的知名架构都使用了JavaScript。

Javascript只是名字里有个Java，语言本身和Java语言一点关系也没有，JavaScript的正式名称是 ECMAScript 。1996年11月，JavaScript 的创造者网景公司将 JS 提交给国际化标准组织 ECMA（European computer manufactures association，欧洲计算机制造联合会），希望这种语言能够成为国际标准，随后 ECMA 发布了规定浏览器脚本语言的标准，即 ECMAScript。ES6 是 ECMAScript 标准十余年来变动最大的一个版本，为其添加了许多新的语法特性，2015年6月， ES6 正式通过，成为国际标准。

9. TypeScript

TypeScript是由微软开发并维护的一门语言，它是JavaScript的超集(也就是JavaScript是TypeScript的子集)，它在JavaScript的基础上加了一层类型系统，所以所有的JavaScript都是合法的TypeScript。它是静态类型语言，增加了项目和代码的可维护性、安全性和健壮性。TypeScript语法友好，表达力强，若已经会JavaScript会很快学会此语言，和JavaScript一样可以制作网页和制作动画效果，目前前端框架也逐渐加入了对TypeScript的支持。

10. HTML和CSS

HTML和CSS语言，并不能被非常严谨地称为编程语言，它们更像是标记语言。HTML（Hypertext Markup Language）超文本标记语言，它专门用来展示网页里面的内容例如文字和图片等。CSS（Cascading Style Sheet），层叠样式表，主要是用来装饰由HTML标记的网页。这两个语言运用范围很广，网页主要是由这两个语言组成的。网站的开发及前端工程师需要掌握这两门语言，HTML和CSS语言入门较为简单，但若是完成写一个网页时候，会有很多代码。

11. Rust

Rust是一个高级的静态类型编程语言，可以看作是C++的竞争者，在2006年被Graydon Hoare设计出来，2011年正式推出，现在主要是由The Rsst Foundation在维护，Rust语言可以操作并且控制一些很底层的计算机资源比如堆和栈，并且在这些资源不再被需要的时候在编译层面去释放这些资源，所以能对内存有个非常高效率的利用。而且Rust并没有垃圾回收期所以会非常的快，由于是一个静态类型编程语言，很多问题会在编译的时候查找出来，所以运行很安全。同时编译器自带的一些功能也确保Rust不会像C和C++那样子容易发生内存泄漏。Rust应用也非常广，它因为原生的支持并发所以非常适合用来网络通信。而且这门语言非常接近底层，所以运行的速度非常快，所以很多时候会被用来开发实时系统，它也拥有直接控制硬件以及内存的能力，因此可以被用来做嵌入式系统开发。用Rust开发的公司挺多的，例如mozilla(开发火狐Firefox浏览器公司)、NPM和Dropbox。但是由于Rust的语法比较晦涩，所以新开发者上手不太容易。

6.1.2 常见开发环境

常见的IDE集成开发环境如下：

1. Visual Studio开发环境

Visual Studio是微软（Microsoft）为以Windows为主的平台开发的一套功能全面而强大的IDE（集成开发环境），支持C#、F#、VB、C/C++等多种语言的开发。VS都有三个主要版本可供选择：Community是免费的社区版，Professional和Enterprise是收费的专业版和企业版，对于初学者三者没有区别。

2. QT开发环境

Qt 是一个跨平台的 C++ 框架（C++库），Qt 虽然经常被当做一个 GUI 库，用来开发图形界面应用程序，但这并不是 Qt 的全部；Qt 除了可以绘制漂亮的界面（包括控件、布局、交互），还包含很多其它功能，比如多线程、访问数据库、图像处理、音频视频处理、网络通信、文件操作等，这些 Qt 都已经内置了。大部分应用程序都可以使用 Qt 实现，除了与计算机底层结合特别紧密的，例如驱动开发。

3. Visual Studio Code（简称 VS Code）

VS Code是一款由微软开发且跨平台的免费源代码编辑器。该软件支持语法高亮、代码自动补全（又称 IntelliSense）、代码重构功能，并且内置了命令行工具和 Git 版本控制系统。用户可以更改主题和键盘快捷方式实现个性化设置，也可以通过内置的扩展程序商店安装扩展以拓展软件功能。VS Code 使用 Monaco Editor 作为其底层的代码编辑器。

VS Code 默认支持非常多的编程语言，包括 JavaScript、TypeScript、CSS 和 HTML；也可以通过下载扩展支持 Python、C/C++、Java 和 Go 在内的其他语言。也支持调试 Node.js 程序。和 GitHub 的 Atom一样，VS Code也基于 Electron 框架构建。VS Code有很好的插件管理功能，可以根据需求插入各种开发插件，从而便于扩展各种新功能。

鸿蒙南向设备开发的IDEDevEco Device Tool就是以插件的形式部署在VS Code上。

4. IntelliJ idea

IntelliJ IDEA是一种商业化销售的Java集成开发环境（Integrated Development Environment，IDE）工具软件，由JetBrains软件公司（前称为IntelliJ）开发，提供Apache 2.0开放式授权的社区版本以及专有软件的商业版本，开发者可选择其所需来下载使用。

Android Studio开发环境就是基于IntelliJ IDEA的社区版本发展而成，用以取代原来提供Android开发者使用的Eclipse ADT。

华为鸿蒙系统的北向应用开发采用IDE DevEco Studio开发环境也是基于IntelliJ IDEA的社区版本发展而成。

同时IntelliJ对个别编程语言所开发的集成环境，如AppCode、CLion、PhpStorm、PyCharm、RubyMine、WebStorm和MPS等，皆可由插件的方式加载IntelliJ IDEA来使用。

5. Eclipse

Eclipse是一款跨平台开源集成开发环境（IDE）。最初主要用来Java语言开发，通过插件使其作为C++、Python、PHP等其他语言的开发工具。Eclipse的本身只是一个框架平台，但是众多插件的支持，使得Eclipse拥有较佳的灵活性，所以许多软件开发商以Eclipse为框架开发自己的IDE。

Eclipse的设计思想是：一切皆插件。Eclipse核心很小，其它所有功能都以插件的形式附加于Eclipse核心之上。Eclipse基本内核包括：图形API（SWT/Jface），Java开发环境插件（JDT），插件开发环境（PDE）等。

Eclipse框架的本质与极高的扩展性，使得厂商可以利用Eclipse作为平台开发各类IDE。甚至一些应用软件也是基于Eclipse的，如Oracle JDK自带的监控程序、Android SDK附带的设备监视工具DDMS。

很多嵌入式系统芯片开发环境都是基于Eclipse开发的。例如众多MCU厂商都在大力发展自家的免费IDE，力图摆脱KEIL和IAR的垄断局面。而这些免费IDE大都基于Eclipse深度定制而来，如迈来芯的MLXIDE、ST的CubeIDE、TruStudio、NXP的S32DS、TI的CCS、再到最近比较火热的Risc-V，国内Risc-V处理器的领军企业芯来科技开发的Nuclei Studio IDE，都是Eclipse的定制产品。

6.1.3 代码版本管理工具

代码管理工具对代码管理有如下的功能：

- （1）能记录一个项目从开始到结束的整个过程。
- （2）追踪项目中所有内容的变化情况，如增加了什么内容，删除了什么内容，修改了什么内容等。
- （3）版本控制，可以清楚的知道每个版本之间的异同点，如版本2.0相比较版本1.0多了什么内容，功能等。
- （4）权限控制，防止代码混乱，提高安全性。
- （5）责任追究，可以清楚的知道谁对哪个文件进行了什么修改，导致了项目无法正常运行。
- （6）回退处理，执行了错误的操作之后还可以有补救的机会，如从版本1.0升级到版本2.0，后来发现版本2.0有一个错误，这时候可以进行回退处理。
- （7）冲突解决，在团队的多人协同开发中，冲突是经常有的事情，比如存在着相同的文件名称，同一个文件中有着相同功能的函数等等，这时候使用源代码管理工具可以比较方便的解决冲突。

1. Git工具

(1) Git是版本控制工具，是一款自由和开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目，是版本控制工具。

(2) GitHub 是一个面向开源及私有软件项目的托管平台，提供给用户空间创建Git仓储，保存用户的一些数据文档或者代码等。GitHub可以托管各种Git库，并提供一个web界面，但与其它像 SourceForge或Google Code这样的服务不同，GitHub的独特优势在于从另外一个项目进行分支的简易性。为一个项目贡献代码非常简单：首先点击项目站点的“fork”的按钮，然后将代码检出并将修改加入到刚才分出的代码库中，最后通过内建的“pull request”机制向项目负责人申请代码合并。

(3) GitLab 是一个用于仓库管理系统的开源项目，使用Git作为代码管理工具，并在此基础上搭建起来的web服务。

Git, GitHub与GitLab的区别:

Git是一种版本控制系统，是一种工具，用于代码的存储和版本控制。

GitHub是一个基于Git实现的在线代码仓库，是目前全球最大的代码托管平台，可以帮助程序员之间互相交流和学习。对于开源项目而言，GitHub用于代码托管从而方便交流和学习。

GitLab是一个基于Git实现的在线代码仓库软件，开发者可以用GitLab自己搭建一个类似于GitHub一样的仓库，但是GitLab有完善的管理界面和权限控制，一般用于在企业、学校等内部网络搭建Git私服。从代码的私有性上来看，GitLab 是一个更好的选择。

2. Gitee工具

使用GitHub时，国内的用户经常遇到的问题是访问速度太慢，有时候还会出现无法连接的情况，可以使用国内的Git托管服务——Gitee（gitee.com）。

和GitHub相比，Gitee也提供免费的Git仓库。此外，还集成了代码质量检测、项目演示等功能。对于团队协作开发，Gitee还提供了项目管理、代码托管、文档管理的服务，5人以下小团队免费。

6.2 网络通信介绍

物联网技术与人工智能相融合，最终追求的是形成一个智能化生态体系，从而实现不同智能终端设备之间、不同系统平台之间、不同应用场景之间的互融互通，万物互融。

应用层协议

MQTT

CoAP

DDS

XMPP

AMQP

HTTP

FTP

.....

网络层、传输协议

IPv4

IPv6

TCP

6LoWPAN

RPL

.....

物理层、 数据链路层协议

近距离通信

Dash7

NFC

Bluetooth

RFID

IRdA

.....

远距离蜂窝通信

GSM (2G)

WCDMA
(3G)

LTE
(3.9G)

TD-LTE
(4G)

NB-IOT

.....

远距离非蜂窝通信

ZigBee

Wi-Fi

Z-Wave

wHART

LoRa

.....

有线通信

MBus

USB

RS232

RS485

Ethernet

.....

设备层

RFID读写器

传感器

可穿戴设备

红外设备

RFID标签

Beacon

摄像机

.....

6.2.1 常见无线通信方案

1. WiFi无线通信

WiFi俗称无线宽带，是WiFi联盟制造商的商标做为产品的品牌认证，是一个创建于IEEE 802.11标准的无线局域网技术。

随着最新的 802.11 ax 标准发布，新的 WiFi 标准名称也将定义为 WiFi6，因为当前的 802.11 ax 是第六代 WiFi 标准了，WiFi 联盟从这个标准起，将原来的 802.11 a/b/g/n/ac 之后的 ax 标准定义为 WiFi6，从而也可以将之前的 802.11 a/b/g/n/ac 依次追加为 WiFi1/2/3/4/5。2.4GHz 频段支持以下标准（802.11b/g/n/ax），5GHz 频段支持以下标准（802.11a/n/ac/ax），由此可见，802.11n/ax 同时工作在 2.4GHz 和 5GHz 频段，所以这两个标准是兼容双频工作。2.4G的优点是穿墙能力不错；缺点是容易受到干扰。5G的优点是抗干扰能力强，带宽宽，吞吐率高，扩展性强；缺点是只适合室内小范围覆盖和室外网桥，各种障碍物对其产生的衰减作用比2.4G大得多。

2. ZigBee无线通信

Zigbee技术是一种应用于短距离和低速率下的无线通信技术，基于IEEE802.15.4无线标准研制开发的，主要用于无线域网，可工作在2.4GHz、868MHz和915 MHz 3个频段上，分别具有最高250kbit/s、20kbit/s和40kbit/s的传输速率，它的传输距离在10-75m的范围内，但可以继续增加。通过IEEE802.15.4协议服务，FFD不仅可以发送和接收数据，还具有路由功能；最终节点负责收集数据，然后发送到协调点或路由节点处理。zigBee支持三种网络拓扑：星形结构，树型结构，网状结构。IEEE 802.15.4定义了PHY（物理层）和MAC（介质访问层）技术规范；Zigbee联盟定义了NWK（网络层）、APS（应用程序支持层）、APL（应用层）技术规范。Zigbee协议栈就是将各个层定义的协议都集合在一起，以函数的形式实现，并给用户提供API（应用层），用户可以直接调用。

一个星型结构的Zigbee网络最多可以容纳254个从设备和一个主设备，一个区域内可以同时存在最多100个ZigBee网络，而且网络组成灵活。网络容量大，网络可以容纳65000台设备。低功耗，小功率待机模式下，两节普通5号电池可使用6-24个月。

3. 蓝牙 (Bluetooth) 无线通信

蓝牙是一种支持设备短距离通信的无线电技术，能在包括移动电话、PDA、无线耳机、笔记本电脑、相关外设等众多设备之间进行无线信息交换。蓝牙是一种短程宽带无线电技术，是实现语音和数据无线传输的全球开放性标准。它使用跳频扩谱（FHSS）、时分多址（TDMA）、码分多址（CDMA）等先进技术，在小范围内建立多种通信与信息系统之间的信息传输。

蓝牙芯片有多个版本，其中蓝牙5.2是蓝牙技术联盟在2020年发布的蓝牙标准，主要的特性是增强版ATT协议、LE功耗控制和信号同步，连接更快，更稳定，抗干扰性更好。一般将蓝牙3.0之前的BR/EDR蓝牙称为传统蓝牙，而将蓝牙4.0规范下的LE蓝牙称为低功耗蓝牙。从蓝牙4.0开始，使用BLE（Bluetooth Low Energy）蓝牙低能耗技术，它是短距离、低成本、可互操作性的无线技术，利用许多智能手段最大限度地降低功耗。蓝牙无线通信传统是服务于点对点的应用场景，借助蓝牙5的网状网络功能，开发人员可以增强无线连接系统（如物联网设备）的通信范围和网络可用性。

蓝牙协议规范包括：传输协议、中介协议和应用协议。传输协议负责蓝牙设备间，互相确认对方的位置，以及建立和管理蓝牙设备间的物理链路；中介协议为高层应用协议或者程序，在蓝牙逻辑链路上工作提供必要的支持，为应用提供不同标准接口；应用协议是蓝牙协议栈之上的应用软件和所涉及到的协议，如：拨号上网、语言功能的应用程序。

4. RFID和NFC无线通信

RFID是Radio Frequency Identification的缩写，即射频识别，俗称电子标签。射频识别技术是一项利用射频信号通过空间耦合（交变磁场或电磁场）实现无接触信息传递并通过所传递的信息达到识别目的的技术。目前RFID产品的工作频率有低频（125kHz~134kHz）、高频（13.56MHz）和超高频（860MHz~960MHz），不同频段的RFID产品有不同的特性。RFID是点对点通信，在传输距离上从几米到几十米都有。

NFC是在RFID的基础上发展而来，NFC从本质上与RFID没有太大区别，都是基于地理位置相近的两个物体之间的信号传输。NFC技术增加了点对点通信功能，可以快速建立蓝牙设备之间的P2P（点对点）无线通信，NFC设备彼此寻找对方并建立通信连接。NFC相较于RFID技术，具有距离近（小于10厘米）、带宽高、能耗低等一些特点。

5. IrDA无线通信

IrDA是一种利用红外线进行点对点通信的技术，IrDA的不足在于它是一种视距传输，2个相互通信的设备之间必须对准，中间不能被其他物体阻隔。

6. UWB无线通信

UWB (Ultra Wideband) 是一种无载波通信技术，利用纳秒至微微秒级的非正弦波窄脉冲传输数据。通过在较宽的频谱上传送极低功率的信号，UWB能在10米左右的范围内实现数百Mb/s至数Gb/s的数据传输速率。UWB具有抗干扰性能强、传输速率高、带宽极宽、消耗电能小、发送功率小等诸多优势，主要应用于室内通信等领域。

同时还可以利用UWB-TDOA定位原理，该技术采用TDOA（到达时间差原理），利用UWB技术测得定位标签相对于两个不同定位基站之间无线电信号传播的时间差，从而得出定位标签相对于定位基站的距离差。

7. LoRa

LoRa是semtech公司创建的低功耗、远距离、无线、广域网的标准。LoRa(Long Range, 远距离)是一种调制技术, 与同类技术相比, 提供更远的通信距离。具有如下特点: 大范围覆盖(5-10公里)、抗干扰、数据率从300bps到50kbps不等、低能耗、双向通信、深度室内渗透和高网络容量 -- 单个网关可以支持数万个终端节点。

由于LoRa调制是物理层(PHY), 因此也可将其用于不同的协议和不同网络架构(如Mesh、Star、点对点)等等。可以将LoRa概括为以下几种协议:

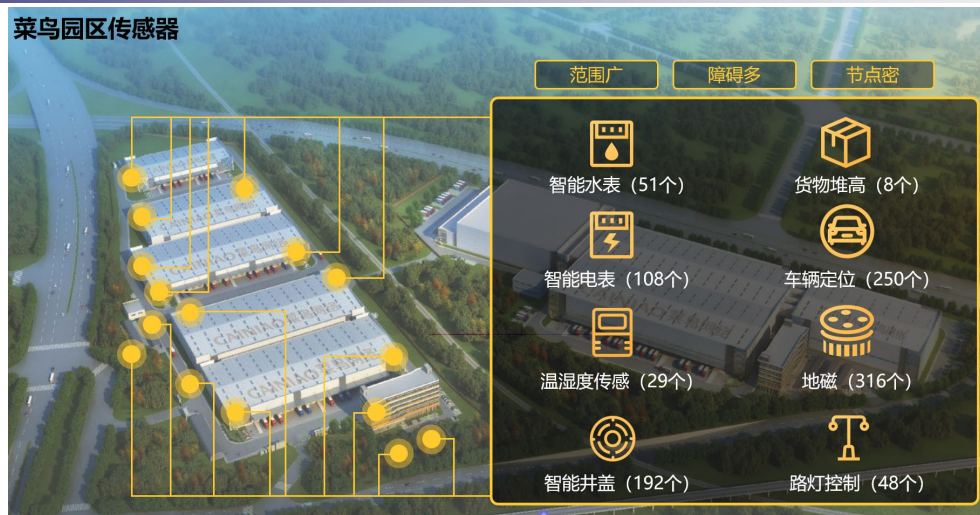
(1) LoRaWAN协议, LoRaWAN协议是由LoRa联盟推动的一种低功耗广域网协议, 针对低成本、电池供电的传感器进行了优化, 包括不同类别的节点, 优化了网络延迟和电池寿命。LoRaWAN 是一种位于媒体访问控制(MAC)层的协议, 可将 LoRa 信号推广到更广泛的应用, 专为单一运营商的大型公共网络而设计。

(2) CLAA网络协议, “中国LoRa应用联盟(China Lora Application Alliance, 简称CLAA)是在LoRa Alliance支持下, 由中兴通讯发起, 各行业物联网应用创新主体广泛参与、合作共建的技术联盟, 旨在共同建立中国LoRa应用合作生态圈, 推动LoRa产业链在中国的应用和发展, 建设多业务共享、低成本、广覆盖、可运营的LoRa物联网。

(3) LoRa私有网络协议, 在面向小范围节点数不多的应用中, 使用LoRaWAN网关部署网络成本就显得高了。用一个或几个SX127x做一个小“网关”或“集中器”, 无线连接上百个的SX127x, 组建一个小的星型网络, 通过自己的LoRa私有通信协议, 就可以实现一个简单的LoRa私有网络, 这也是一种比较灵活方式。当然, 协议也可以是LoRaWAN协议。

(4) LoRa数据透传, 目前市面上LoRa芯片基本上源于美国SEMTECH的SX127x系列, 用LoRa做成透传模块, 只进行简单的发送和接收, 实现点对点数据的传输, 应用相对简单。

菜鸟园区传感器



LoRa联盟企业



8. 蜂窝移动通信

蜂窝移动通信（Cellular Mobile Communication）是采用蜂窝无线组网方式，在终端和网络设备之间通过无线通道连接起来，进而实现用户在活动中可相互通信，3GPP是制定蜂窝技术标准的一个国际化标准组织。

（1）2G

2G 技术可能年代已经比较久远了，2G 的出现标志着数字通信的起源，2G 又叫 GSM，全名为：Global System for Mobile Communications，中文为全球移动通信系统。通过 2G 网络，用户可以打电话，发短信。之后就是 GPRS，又称作是 2.5G，它是 GSM 的延续，同时它的传输速率基于 2G 又有了提升。

（2）3G

3G 用户的数据传输速率就更高了，一般可以达到几百 kbps，3G 用户可以视频通话，同时也可以用手机看电视。在 3G 当中，国内的运营商都使用了不同的标准，它们分别是 CDMA2000、WCDMA 和 TD-SCDMA。其中联通的 WCDMA 最新演进技术 HSPA+ 可以支持最高 42Mbps 的传输速率，目前仍然被应用于大部分视频数据传输领域。

(4) 4G

4G数据速率就更高了，用户可以用手机看视频听歌等一系列多媒体业务。在 4G 网络当中，有不同的等级，也就是 LTE-Cat。其中 Cat 的意思是 Category 等级，它指的是终端设备接入 4G 网络的所支持的不同的速率等级，设计者使用不同的等级方式来规定某一个设备所能达到的最高传输速率。4G刚开始候，市面上常见的各种终端设备使用的是Cat.4等级，但是现如今生活中所使用 的 4G 设备基本都是 Cat.6等级。

在物联网领域，目前使用比较多的是 Cat.1 这个等级，因为除了大部分设备使用低功耗广域网这个场景之外，还有一部分对于网络速率有需求的设备就会使用 Cat.1 等级，尽管 Cat.4 及更高版本的解决方案支持高速率，但是对于物联网终端设备来说，这些模組的成本太高了，所以 Cat.1 等级就成了使用 LTE 网络最具有性价比的等级。Cat.1 支持高达10Mbps 的终端下行链路速率，从而能够将更低功耗和更低成本的 IoT 设备连接到 LTE 网络。

(5) 5G

5G网络是第五代移动通信网络，5G的理论峰值速率可以达到 10Gbps。同时 5G 定义了三大应用场景：增强型移动互联网业务eMBB（Enhanced Mobile Broadband）对应的就是 10Gbps 的速率，海量连接的物联网业务mMTC（Massive Machine Type Communication）对应的是 1 million/km² 的设备连接数，超高可靠性与超低时延业务 uRLLC（Ultra Reliable & Low Latency Communication）对应的是 1ms 的时延。在这三大场景当中，eMBB 场景与用户的关联性比较大，因为它能给用户提供了大带宽的上网速率。但是另外两个场景就与物联网的关联性比较大了，海量的连接与超低的时延能够帮助物联网在很多不同的应用当中进行业务能力的提升。

(6) NBIOT

2015年8月，3GPP RAN开始立项研究窄带无线接入全新的空口技术，称为Clean Slate CIoT，这一Clean Slate方案覆盖了NB-CIoT。NB-CIoT是由华为、高通和Neul联合提出，NB-LTE是由爱立信、诺基亚等厂家提出。随着2015年9月3GPP会议上协商统一，NB-IoT可认为是NB-CIoT和NB-LTE的融合，NB-IoT技术正式写入3GPP协议。

华为、高通、爱立信和VDF等全球主流通信设备商、运营商和芯片厂商均已明确了推进NB-IoT大规模商用的目标。相比Lora技术，NB-IoT可基于运营商现有网络站点升级支持，无需额外的站点/传输资源，且部署方便，便于规模扩展。相比运营商现存的2G/3G/4G网络，专为物联网设计的NB-IoT技术则在技术性能和业务能力上有着绝对优势，200KHz的窄带需求可以充分利用运营商的零散频谱，并能够提供百倍于4G的连接规模、百倍于2G的灵敏度和长达10年的设备电池供电寿命，也可提供成本更低的商用芯片，使蜂窝网络极大地延伸了应用边界。

6.2.2 通信方案及协议

1. 互连参考模型（Open System Interconnect 简称OSI）模型

传统的开放式系统互连参考模型（Open System Interconnect 简称OSI）是一种通信协议的7层抽象参考模型，包括有：物理层，数据链路层，网络层，传输层，会话层，表达层和应用层。其中每一层执行某一特定任务，该模型的目的是使各种硬件在相同的层次上相互通信。

2. 以太网协议(TCP/IP)


通用的以太网通信协议是TCP/IP协议，并被广泛应用于实际工程。相比OSI模型，TCP/IP通讯协议采用了4层结构，每一层都呼叫它的下一层所提供的网络来完成自己的需求。这四层分别为：

（1）应用层：应用程序间沟通的层，如简单电子邮件传输协议(SMTP)、文件传输协议(FTP)、网络远程访问协议(Telnet)等。

（2）传输层：在此层中，它提供了节点间的数据传送服务，如传输控制协议(TCP)、用户数据包协议(UDP)等，TCP和UDP给数据包加入传输数据并把它传输到下一层中，这一层负责传送数据，并且确定数据已被送达并接收。

（3）网络层：负责提供基本的数据包传送功能，让每一块数据包都能够到达目的主机(但不检查是否被正确接收)，如网际协议(IP/ICMP/ARP)。

（4）接口层：对实际的网络媒体的管理，定义如何使用实际网络（如以太网接口等）来传送数据。



TCP/IP 中有两个具有代表性的传输层协议，分别是 TCP 和 UDP。

TCP 是面向连接的、可靠的流协议。流就是指不间断的数据结构，当应用程序采用 TCP 发送消息时，虽然可以保证发送的顺序，但还是犹如没有任何间隔的数据流发送给接收端。

UDP 是不具有可靠性的数据报协议。在 UDP 的情况下，虽然可以确保发送消息的大小，却不能保证消息一定会到达。因此，应用有时会根据自己的需要进行重发处理，例如有人提出KCP协议，KCP 在 UDP 的基础上，实现了用户态的确认、ARQ、流量控制与拥塞控制，它的设计目标是实时性与可靠性，一定程度上破坏了公平性，KCP 不是 RFC 的标准协议，但在实际应用中表现出了较好的效果，目前已经有很多基于 KCP 的开源案例。

3.Socket

Socket是为了方便开发者直接使用更底层协议（一般是TCP或UDP）而存在的一个抽象层。Socket实际上是对TCP/IP协议的封装，本身并不是协议，而是一个调用接口（API）。

Socket使开发人员方便使用TCP/IP协议栈，是对TCP/IP协议的抽象，从而形成一些最基本的函数接口，比如create、listen、connect、accept、send、read和write。

4.Modbus协议

Modbus由MODICON公司于1979年开发，是一种工业现场总线协议标准。1996年施耐德公司推出基于以太网TCP/IP的Modbus协议：ModbusTCP。Modbus协议是一项应用层报文传输协议，包括ASCII、RTU、TCP三种报文类型。标准的Modbus协议物理层接口有RS232、RS422、RS485和以太网接口，采用主/从方式通信。

帧结构PDU由功能码+数据组成。功能码为1字节，数据长度不定，由具体功能决定。Modbus通信协议具有多个变种，其具有支持串口（主要是RS-485总线）、以太网，主要有Modbus RTU（串口），Modbus ASCII（串口）和Modbus TCP（以太网）三种。

5.EtherCat协议

EtherCAT全称EtherNet Control Automation Technology，是由德国倍福（Beckhoff）公司提出的一种实时以太网技术。EtherCAT是一种开放但不开源的技术，意味着您可以任意使用这项技术，但若要进行相关设备的开发，则需要向倍福公司获取相关授权。

相比传统现场总线，EtherCAT的数据传输速率有了极大的提升，可选10Mbit/s或100Mbit/s，甚至依托补充的EtherCAT G技术，传输速率可达1000Mbit/s；同时EtherCAT基于标准以太网帧传输，单帧数据用容量可达1486 Bytes。这使得在传输数据量方面EtherCAT有无比的优越性。

相对于7层OSI模型，EtherCAT使用了物理层、链路层和应用层三层协议，与多数传统的现场总线相同，但相比于其它实时以太网协议，如PROFINET、EtherNet/IP等，其协议栈更加精简。

5. 物联网应用层常用协议

(1) MQTT协议

MQTT 是物联网(Internet of Things, IoT)的 OASIS 标准消息传递协议, 使用 TCP/IP 提供网络连接, MQTT协议是基于TCP。它被设计为一个极其轻量级的发布/订阅消息传输, 非常适合用较小的代码占用和最小的网络带宽连接远程设备。

在MQTT协议中, 一个MQTT数据包由: 固定头(Fixed header)、可变头(Variable header)、消息体(payload)三部分构成。MQTT的传输格式非常精小, 最小的数据包只有2个bit且无应用消息头。发布/订阅模型允许MQTT客户端以一对一、一对多和多对一方式进行通讯。

(2) CoAP协议

CoAP是受限制的应用协议(Constrained Application Protocol)的代名词。由于目前物联网中的很多设备都是资源受限型的, 所以只有少量的内存空间和有限的计算能力, 传统的HTTP协议在物联网应用中就会显得过于庞大而不适用。因此, IETF的CoRE工作组提出了一种基于REST架构、传输层为UDP和网络层为6LowPAN(面向低功耗无线局域网的IPv6)的CoAP协议。CoAP采用与HTTP协议相同的请求响应工作模式。

（3）XMPP协议

XMPP (Extensible Messageing and Presence Protocol: 可扩展消息与存在协议) 是目前主流的四种IM (IM: instant messaging, 即时消息) 协议之一, 其他三种分别为: 即时信息和空间协议 (IMPP)、空间和即时信息协议 (PRIM)、针对即时通讯和空间平衡扩充的进程开始协议 SIP (SIMPLE)。

（4）HTTP协议

HTTP协议是Hyper Text Transfer Protocol (超文本传输协议) 的缩写, 是用于从万维网 (WWW:World Wide Web) 服务器传输超文本到本地浏览器的传送协议。HTTP是一个基于TCP协议来传递数据 (HTML 文件, 图片文件, 查询结果等), 协议标识符是http (如果加密, 则为https), 服务器网址就是 URL。

HTTP是一个属于应用层的面向对象的协议, 由于其简捷、快速的方式, 适用于分布式超媒体信息系统。客户向服务器请求服务时, 只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP协议简单, 使得HTTP服务器的程序规模小, 因而通信速度很快。HTTP使用报文格式对于嵌入式设备来说需要传输数据太多, 太重, 不够灵活。

HTTP 协议有一个缺陷: 通信只能由客户端发起。这种单向请求的特点, 注定了如果服务器有连续的状态变化, 客户端要获知就非常麻烦。我们只能使用“轮询”: 每隔一段时候, 就发出一个询问, 了解服务器有没有新的信息。轮询的效率低, 非常浪费资源。

(5) WebSocket协议

针对HTTP协议通信只能由客户端发起的缺陷，WebSocket 协议在2008年诞生，2011年成为国际标准，目前大多数浏览器都已经支持。最大特点就是服务器可以主动向客户端推送信息，客户端也可以主动向服务器发送信息，是真正的双向平等对话，属于服务器推送技术的一种。它建立在 TCP 协议之上，服务器端的实现比较容易。与 HTTP 协议有着良好的兼容性。默认端口也是80和443，并且握手阶段采用 HTTP 协议，因此握手时不容易屏蔽，能通过各种 HTTP 代理服务器。协议标识符是ws（如果加密，则为wss），服务器网址就是 URL。

6.2.3 常见序列化协议

互联通讯的双方需要采用约定的协议，序列化和反序列化属于通讯协议的一部分。序列化：将数据结构或对象转换成二进制串的过程。反序列化：将在序列化过程中所生成的二进制串转换成数据结构或者对象的过程。

1. XML

XML 指可扩展标记语言（eXtensible Markup Language）是一种通用和重量级的数据交换格式，以文本结构存储。XML是一种常用的序列化和反序列化协议，具有跨机器，跨语言等优点。XML历史悠久，其1.0版本早在1998年就形成标准，并被广泛使用至今。XML的最初产生目标是对互联网文档（Document）进行标记，所以它的设计理念中就包含了对于人和机器都具备可读性。但是，当这种标记文档的设计被用来序列化对象的时候，就显得冗长而复杂（Verbose and Complex）。

XML所具有的人眼可读（Human-readable）特性使得其具有出众的可调试性，互联网带宽的日益剧增也大大弥补了其空间开销大（Verbose）的缺点。对于在公司之间传输数据量相对小或者实时性要求相对低（例如秒级别）的服务是一个好的选择。由于XML的额外空间开销大，序列化之后的数据量剧增，对于数据量巨大序列持久化应用常景，这意味着巨大的内存和磁盘开销，不太适合XML。另外，XML的序列化和反序列化的空间和时间开销都比较大，对于对性能要求在ms级别的服务，不推荐使用。

2. JSON

JSON(JavaScript Object Notation, JS 对象简谱) 是一种通用和轻量级的数据交换格式。以文本结构存储。**Json**作为数据包格式传输时具有更高的效率，这是因为**Json**不像**xml**那样需要有严格的闭合标签，这就让有效数据量与总数据包比有着显著的提升，从而减少同等数据流量的情况下网络的传输压力。

3. PB (protobuf)

Protocol Buffer是Google开发的一种独立和轻量级的数据交换格式，以二进制结构进行存储，用于不同服务之间序列化数据。**PB**是一种轻便高效的结构化数据存储格式，可以用于结构化数据串行化，或者序列化，可用于通讯协议、数据存储等领域的语言无关、平台无关、可扩展的序列化结构数据格式。序列化后体积相比**Json**和**XML**很小，适合网络传输,序列化反序列化速度很快，快于**Json**的处理速度。但是缺点是由于是二进制结构，可读性不高。

6.3 软件框架及常见组件

6.3.1 后端开发

后端开发即“服务器端”开发，主要涉及软件系统“后端”的东西。比如，用于托管网站和App数据的服务器、放置在后端服务器与浏览器及App之间的中间件，它们都属于后端。简单地说，那些在屏幕上看不到但又被用来为前端提供支持的东西就是后端。

1. 常见的服务器

(1) Node.js

Node.js是能够在服务器端运行**JavaScript**的一个开源和跨平台的服务器。**Node.js**在浏览器之外运行 **V8 JavaScript 引擎**，**Node.js** 应用程序在单个进程中运行，无需为每个请求创建新的线程。**Node.js** 在其标准库中提供了一组异步的 **I/O** 原语，以防止 **JavaScript** 代码阻塞，通常，**Node.js** 中的库是使用非阻塞范式编写的，使得阻塞行为成为异常而不是常态。这允许 **Node.js** 使用单个服务器处理数千个并发连接，而不会引入管理线程并发（这可能是错误的重要来源）的负担。**Node.js** 具有独特的优势，因为数百万为浏览器编写 **JavaScript** 的前端开发者现在无需学习完全不同的语言，就可以编写除客户端代码之外的服务器端代码。

（2）Apache

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，由于其多平台 and 安全性被广泛使用，是最流行的Web服务器端软件之一。Apache服务器是一个模块化的服务器，各个功能使用模块化进行插拔，目前支持 Windows, Linux, Unix 等平台。它快速、可靠并且可通过简单的 API 扩展，将 Perl/Python 等解释器编译到服务器中。

（3）Tomcat

Tomcat是由Apache软件基金会属下Jakarta项目开发的Servlet容器，按照Sun Microsystems提供的技术规范，实现了对Servlet和JavaServer Page（JSP）的支持，并提供了作为Web服务器的一些特有功能，如Tomcat管理和控制平台、安全局管理和Tomcat阀等。由于Tomcat本身也内含了HTTP服务器，因此也可以视作单独的Web服务器。Tomcat提供了一个Jasper编译器用以将JSP编译成对应的Servlet。Tomcat的Servlet引擎通常与Apache或者其他Web服务器一起工作。

（4）Nginx

Nginx是异步框架的网页服务器，也可以用作反向代理、负载均衡器和HTTP缓存。Nginx是一款面向性能设计的HTTP服务器，相较于Apache具有占有内存少，稳定性高等优势。Nginx使用异步事件驱动的方法来处理请求。Nginx的模块化事件驱动架构可以在高负载下提供更可预测的性能。Nginx不采用每客户机一线程的设计模型，而是充分使用异步逻辑从而削减了上下文调度开销，所以并发服务能力更强。整体采用模块化设计，有丰富的模块库和第三方模块库，配置灵活。整体采用模块化设计是Nginx的一个重大特点，甚至http服务器核心功能也是一个模块。

2. 常见数据库

(1) MySQL

MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

(2) PostgreSQL

PostgreSQL是一种特性非常齐全的自由软件的对象-关系型数据库管理系统，PostgreSQL支持大部分的SQL标准并且提供了很多其他现代特性，如复杂查询、外键、触发器、视图、事务完整性、多版本并发控制等。同样，PostgreSQL也可以用许多方法扩展，例如通过增加新的数据类型、函数、操作符、聚集函数、索引方法、过程语言等。在灵活的BSD许可证下发行，任何人都可以以任何目的免费使用、修改和分发PostgreSQL。

（3）GaussDB

GaussDB(for openGauss)是华为公司倾力打造的自研企业级分布式关系型数据库，该产品具备企业级复杂事务混合负载能力，同时支持优异的分布式事务，同城跨AZ部署，数据0丢失，支持1000+扩展能力，PB级海量存储等企业级数据库特性。同时华为开源openGauss单机主备社区版本。

（4）Cassandra

Apache Cassandra是一个高度可扩展的高性能分布式数据库，旨在处理许多商用服务器上的大量数据，提供高可用性而没有单点故障，它是NoSQL数据库的一种。NoSQL数据库（有时称为“Not Only SQL”）是一种数据库，它提供了一种存储和检索关系数据库中使用的表格关系以外的数据的机制。这些数据库是无模式的，支持简单的复制，具有简单的API，最终是一致的，并且可以处理大量数据。与关系数据库相比，NoSql数据库使用不同的数据结构。它使NoSQL中的某些操作更快。

（5）MongoDB

MongoDB 是一个面向文档的NoSQL数据库，文档是用BSON写的，是JSON的二进制表示。MongoDB由 C++ 语言编写，旨在为 WEB 应用提供可扩展的高性能数据存储解决方案，在高负载的情况下，添加更多的节点，可以保证服务器性能。MongoDB 是一个介于关系数据库和非关系数据库之间的产品，是非关系数据库当中功能最丰富，最像关系数据库的。

3.常见中间件

（1）Kafka

Kafka是最初由**Linkedin**公司开发，是一个分布式、分区的、多副本的、多订阅者，基于**zookeeper**协调的分布式日志系统（也可以当做**MQ**系统），常见可以用于日志、访问日志，消息服务等，**Linkedin**于2010年贡献给了**Apache**基金会并成为顶级开源项目。主要应用场景是：日志收集系统和消息系统。

（2）Redis

Redis是一个开源（**BSD**许可）的、基于内存存储、采用**Key-Value**（键值对）格式存储的内存数据库，支持多种数据类型，包括字符串、哈希表、列表、集合、有序集合、位图等。**redis**存储在缓存空间临时性的，存储在内存中，虽然有**rdb**、**aof**两种持久化，但也只是为了避免掉电数据丢失，而且内存容量也有限，并不能解决数据库的大量数据的持久化。

（3）RabbitMQ

RabbitMQ是一款目前应用相当广泛、开源的消息中间件，可用于实现消息异步分发、模块解耦、接口限流等功能。特别是在处理分布式系统高并发的业务场景时，**RabbitMQ**能够起到很好的作用，比如接口限流，从而降低应用服务器的压力；比如消息异步分发，从而降低系统的整体响应时间。

(4) Zookeeper

Zookeeper是一个开源的分布式应用程序协调服务，可以为分布式应用提供一致性服务的软件，简称ZK，其提供的功能服务包括：配置维护、域名服务、分布式同步等，提供的接口则包括分布式独享锁、选举、队列等等。

(5) Dubbo

Dubbo 最早诞生于阿里巴巴，随后加入 Apache 软件基金会，是一款微服务开发框架，它提供了 RPC通信 与 微服务治理 两大关键能力。这意味着，使用 Dubbo 开发的微服务，将具备相互之间的远程发现与通信能力，同时利用 Dubbo 提供的丰富服务治理能力，可以实现诸如服务发现、负载均衡、流量调度等服务治理诉求。同时 Dubbo 是高度可扩展的，用户几乎可以在任意功能点去定制自己的实现，以改变框架的默认行为来满足自己的业务需求。

4. 常见后端开发框架

(1) Spring Boot

Spring框架是Java平台上的一种开源应用框架，提供具有控制反转特性的容器。Spring Boot基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。Spring Boot是一个快速开发框架，可以帮助我们快速整合常见的第三方框架（Maven的继承方式），完全采用注解化（使用注解方式启动SpringMvc），简化XML，内置HTTP服务器，最终以java应用程序进行执行。Spring Boot 在简化配置、打包和集成第三方工具方面确实做得很好，可以减低 Spring 开发人员的入门门槛。

(2) Gin和Gorm

Gin是一个Go语言的WEB框架，很轻量，依赖到很少，有些类似Java的SpringMVC，通过路由设置，可以将请求转发到对应的处理器上。

Gorm是Go语言的对象关系映射（ORM）框架，提供一套对数据库进行增删改查的接口，使用它，就可以类似Java使用Hibernate框架一样，可对数据库进行相应操作。

(3) Django、Flask和Tornado

大部分后端业务逻辑开发中都会使用 web 框架，提升开发效率。常用的 python web 框架有 Django、Flask、Tornado。

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。Django 采用了 MVT 的软件设计模式，即模型（Model），视图（View）和模板（Template）。Django是重量级框架。

Flask是一个使用Python编写的轻量级Web应用框架。Flask被称为“微框架”，因为它使用简单的核心，用扩展增加其他功能。Flask没有默认使用的数据库、窗体验证工具。然而，Flask保留了扩增的弹性，可以用Flask-extension加入这些功能：ORM、窗体验证工具、文件上传、各种开放式身份验证技术。

Tornado是一个用Python语言写成的Web服务器兼Web应用框架，是一个轻量级的Web框架，其拥有异步非阻塞IO的处理方式。作为Web服务器，Tornado有较为出色的抗负载能力，有很好的并发处理能力。

6.3.2 前端开发

前端开发是创建**WEB**页面或**APP**等前端界面呈现给用户的过程，通过**HTML**，**CSS**及**JavaScript**以及衍生出来的各种技术、框架、解决方案，来实现互联网产品的用户界面交互。随着**Web**越来越规范和标准的统一，**Web**组件化技术不断革新，移动端开发不断升华，以下是一些常见开源前端框架：

1.Vue.js

Vue.js是一个用于创建用户界面的开源前端**JavaScript**框架，旨在更好地组织与简化**Web**开发。**Vue**所关注的核心是**MVC**模式中的视图层，同时，它也能方便地获取数据更新，并通过组件内部特定的方法实现视图与模型的交互。组件是**Vue**最为强大的特性之一。为了更好地管理一个大型的应用程序，往往需要将应用切割为小而独立、具有复用性的组件。在**Vue**中，组件是基础**HTML**元素的拓展，可方便地自定义其数据与行为。**Vue**使用基于**HTML**的模板语法，允许开发者将**DOM**元素与底层**Vue**实例中的数据相绑定。开发者只需将视图与对应的模型进行绑定，**Vue**便能自动观测模型的变动，并重绘视图。**Vue**在插入、更新或者移除**DOM**时，提供多种不同方式的应用过渡效果。

2.React.js

React.js 是一个开源的前端 **JavaScript** 库，**React**视图通常采用包含以自定义**HTML**标记规定的其他组件的组件渲染。**React**为程序员提供了一种子组件不能直接影响外层组件（**"data flows down"**）的模型，数据改变时对**HTML**文档的有效更新，和现代单页应用中组件之间干净的分隔。**React** 引入了一种组件驱动、函数式和声明式的编程风格，为主要是单页 **Web** 应用创建交互式用户界面。通过“虚拟 **DOM**”，**React** 提供了非常快的渲染速度，只需渲染发生变化的部分，而不用渲染整个页面。

3.AngularJS

AngularJS是一款由**Google**维护的开源**JavaScript**库，用来协助单一页面应用程序运行。它的目标是透过**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。**Angular**在呈现和资料中间，可以简单创建双向的数据绑定。一旦创建双向绑定，用户输入，会由**Angular**自动传到一个变量中，再自动读到所有绑到它的内容，更新它。效果上就是立即的资料同步。在代码中修改变量，也会直接反应到呈现的外观上。不仅内容可以双向绑定，其他诸如类别、宽度、高度等等，都可以和变量与用户的输入，绑定起来。



谢谢！