`

北京大学

硕士研究生学位论文

题目：基于 REST 的 Smart Meter 控制与显示系统的设计与实现

姓　　名：李敬杰

学　　号：1301210720

院　　系：软件与微电子学院

专　　业：嵌入式软件工程

研究方向：嵌入式软件

导师姓名：吴中海教授

二〇一五年　九月

版权声明

摘要

物联网（IOT）通过互联网络和终端设备网络延伸和扩展互联网的范围。随着人们对物联网意识的提高，越来越多的应用可以应用于各个领域。特别是，如何开发出智能系统的节能成为各界新的挑战[1]。智能电表是微控制器技术、专用集成电路技术、工业控制网络技术和软件等多种技术共同发展的结果。其应用领域十分广阔，涉及电力系统，工业生产控制、智能小区、网络化家电等各行各业，具有良好的发展前景[2]。

为了更好的共享和使用智能电表中的数据，就要首先将智能电表数据集成到网络中，由于不同组成结构和接口的智能电表网络，这变得很不容易。本论文主要介绍一种使用 RESTful API 软件平台模型，来分享并利用智能电表中的数据。首先，文章列举了传统普通电表和智能电表的实现方式，发现其中的不足。之后，介绍了项目的技术需求项目的基本描述。进而，论证了使用 RESTful API 的意义和优势，并指出来本平台体积结构的好处和兼容性。最后，本论文通过智能电表硬件平台，进行了实际测试和数据分析。

对于 RESTful API 前台显示与安全性访问，本论文分析研究现有流行的前台设计框架和安全模型，提出了一种结合目标比较流行的 Oauth 验证登陆和用户注册登录相结合的方式。

关键字：智能电表；RESTful；Jersey；网站服务器，Oauth2.0 客户端；

Design and implementation of Meter Smart control and display system based on REST

Jingjie Li(Embedded software)
Directed by Zhonghai Wu

ABSTRACT

The Internet of Things (IoT) extends and expands the range of the internet by interconnecting Internet and end device networks. As the raising of awareness about IoT, more and more application may applied for various areas. Especially, how to develop intelligent systems for energy saving becomes a new challenge in all circles.[1]Smart Meter is the result of the development of many kinds of technologies, such as micro controller technology, special integrated circuit technology, industrial control network technology and software. Its application area is very broad, involving power systems, industrial production control, intelligent community, network home appliances and other industries, has a good development prospects.[2]

While due to different structures and interfaces of the smart meter networks, it’s not that easy to fully integrating their data to the web then share and reuse it later.This paper introduces a platform using RESTful Web of Things API to help share smart meter data.First of all, the article lists the traditional common meter and the intelligent electric meter to realize the way, found the shortage. After that, describes the basic description of the project's technical requirements for the project. And then, it demonstrates the significance and advantages of using API RESTful, and points out the benefits and compatibility of the volume of the platform. Finally, this paper carries out the actual test and data analysis through the smart meter hardware platform.

For API RESTful front display and security access, this thesis analyzes and studies the existing popular foreground design framework and security model, and proposes a combination of Oauth verification and user registration login method.

KEY WORDS: Smart Meter , RESTful , Jersey , web server , Oauth2.0 Client

目录

摘要.....1
ABSTRACT.....1
第一章 绪论.....1
1.1 项目来源与介绍.....1
1.1 应用背景.....2
1.2 价值和意义.....4
1.3 国内外软硬件研究现状.....4
1.3.1 硬件现状.....4
1.3.2 软件现状.....6
1.4 论文组织结构.....9
第二章 技术介绍与需求分析.....10
2.1 Java 多线程方式.....10
2.1.1 Future 多线程模型.....12
2.2 服务器搭建方式.....13
2.2.1 三层架构模型.....13
2.2.2 M-V-C 模型.....14
2.2.3 M-V-P 模型.....15
2.2.4 M-V-VM 模型.....16
2.3 数据库存储.....16
2.3.1 oracle.....16
2.3.2 SQL Server.....17
2.3.3 MySQL.....17
2.3.4 数据库比较.....17
2.4 网络安全模型.....18
2.4.1 网络攻击分析.....18
2.4.2 服务器安全设计.....19
2.5 RESTful 设计.....21
2.6 项目需求.....22
2.7 本章小结.....22
第三章 项目描述.....24
3.1 硬件部分.....25
3.1.1 CC2538 芯片.....25
3.1.2 硬件通信方式.....25
3.2 软件部分.....25
3.3 软件通信方式.....26
3.3.1 数据同步.....26
3.3.2 数据显示.....27
3.3.2 安全验证.....28

3.4 本章小结 29
第四章 研究内容和重点 30
4.1 数据同步模块 30
4.1.1 数据同步方式 POLL 与 PUSH 31
4.2 服务器设计架构 32
4.2.1 Server-Sent-Event 的机制 32
4.2.2 服务器架构模型 32
4.2 URL 设计方案 33
4.3 OAuth2.0 安全客户端 34
4.4 前台显示模块 36
4.5 数据库设计 36
4.6 本章小结 38
第五章 结论及展望 39
5.1 结论 39
5.2 展望 39
参考文献 41
致谢 44
北京大学学位论文原创性声明和使用授权说明 45

第一章　绪论

在以数字化和网化为特征的信息时代，数据的采集、存储和分析逐渐成为人们首要关心的对象。其中，电力工业的发展面临着新一轮的挑战，即如何快捷有效准确地提取和利用分布式用电资源的信息。智能电网的概念应运而生，它实现了分布广泛的电能资源的整合利用，在电能传输的过程中应用数字技术节约能源、降低成本，提高了供电部分和用户单位的电能利用率，成为现代电网的主要发展模式。智能用电实现了传统用电方式的改变，客户不再是单纯地从电网进行购电，而是实现了在信息、业务等方面实时互动的全方位供电关系[3]。

1.1 项目来源与介绍

本文中项目来自于美国 International Technological University 实验室的 Green Campus 项目，旨在以建设绿色节能型校园为出发点，并最终推广到居民区、工厂等社会地区中使用。目前在本校区进行测试和试用，其可靠性和实时性都得到了很好的验证。

本次项目与传统的智能电表有很大的不同之处，不仅可以实现采集数据的基本功能，还融入了智能家居的理念。主要体现在以下 4 个方面：

（1）多通道采集。传统电表只是负责采集以家庭为单位的电流和电压总量，但是用户不能了解主要家用设备的电量消耗，也就无法采取有效的节能措施。从智能家居的角度出发，本次项目采用多通道采集方法，可以不同的设备进行采集和监控，让用户了解哪些是主要耗能设备，才能有针对性的做出合理的节能方案。而且可以增加传感器来测量有需求的数据，比如室内温度、湿度等，极大的增加了智能电表的可扩展性。

（2）实时控制。传统电表虽然也提供了一些控制方法，但都是针对数据采集和重发的处理机制，并没有给用户提供可以控制用电设备的内容。本次项目增加了用户参与的环节，可以对特定的设备，比如空调、保湿器等进行远程操控，增强了用户对私人设备的掌控能力，营造舒适的办公和居住环境。

（3）采用 RESTful 结构设计。RESTful 是一种软件架构设计风格，它主要用于客户端和服务器交互类的软件。基于这个风格设计的软件可以更简洁，更有层次，更易于实现缓存等机制。目前，RESTful 结构开始在嵌入式应用领域流行，本次设计应用在本地控制系统和云端存储服务器，提供了众多常用功能的接口，比如调取某个时间段设备的数据采集量，实时反映设备运行状态等。厂商可以根据需求自己取舍功能接口，

增加了设备的可扩展性。

除了具备以上特点之外，考虑到将来市场的需求，本次项目采用了低成本、低功耗的设计。在硬件方面使用 TI 生产的 CC2538 芯片来做研究开发，CC2538 是在智能能源基础设施领域的主流芯片，通信方式使用 RS485，功耗小，但是可靠性能高，可以适应比无线通信复杂的周边环境。在软件方面，采用 Linux 和 eclipse 等免费的开源环境，不能可以满足设计的功能需求，也极大的减少了开发成本。

1.1 应用背景

随着信息产业的蓬勃发展，电子设备深入到社会的各个领域，医疗、工厂、学校等。与之俱来的是电力能源消耗日益剧增，节能减排也日益受到世界各国的重视。建设资源节约型、环境友好型社会，是党中央、国务院在新形势下作出的具有全局性、战略性的重大决策[2]。世界各国从高效节能的目标出发对用电系统、智能电网和智能电表等都做了较深地探索和实践。

在美国，国家电网正经历着巨大的变革，其的关注点在于如何提高信息技术在智能电网中所占的比例，包括鼓励可再生能源和分布式能源发展政策推动下的数据技术应用，加快电力网络的基础架构升级换代。美国能源部 2014 年公布的《2014 智能电网系统报告》，电网更强调对极端气候的应对能力，而消费者和各行业在电力管理和生产两方面的参与度是不断提高的。智能电网的核心技术是数字化电网、分布式能源系统和信息家电化，美国在此已经进行了大量技术准备，如飞思卡尔，德州仪器等公司都在尝试应用新技术实现互联。美国能源部西北太平洋国家实验室的研究结果表明，仅使用数字工具设定家庭温度及融入价格信息，能源消耗每年可缩减 15%。这一技术革命整合不仅会改变目前所有电器设备、信息产品的基本概念，使人类全面重新设计制造各种各样基于新信息技术的电器产品，清洁能源开发、能效经济、绿色配额等交易将融为一体，实现多维度整合优化。

在欧洲，其智能电网的发张主要以欧盟为主导。欧盟负责制定整体目标和方向，并提供政策和资金支撑。2006 年欧盟理事会公布的能源率绿皮书《欧洲可持续的、竞争的和安全的电能策略》明确指出，智能电网技术是保证欧盟电能电网质量的关键技术和发展方向，是进入新能源时代中的重要规划。欧盟还号召其成员国利用新兴的信息技术提高能效，既可以应对气候变化，也可以促进欧洲经济的复苏。欧洲电网的根本出发点在于减少能源消耗和温室气体排放，推动欧洲的可持续发展，其实现目标是能够支撑可再生能源以及灵活接入分布式能源，向用户提供双向互动的信息交流。目前，各发达国家取得了巨大的成就。针对智能电网发展过程中的不足，欧盟并出台和建立了完善的标准体系：制定欧盟层面的通用技术标准，保证不同系统的兼容性，优化电

力消费和生产，实现本地电力需求调整和负荷控制的一体化。

我国一直都是电力应用大国，在很长的一段时间内都存在着电力分布不均、能源浪费多和抄表收费难的问题。随着世界环境压力与日俱增，数据网络和计算机的蓬勃发展，电力发展的市场化呼声越来越高，国家电网公司在“2009 年特高压输电技术国际会议”上表示我国将逐步建设具有信息化、自动化、互动化和数字化为特征的智能电网，标志着高效利用和节能的电力网络在全国范围铺展开来。越来越多的电子设备和软件开发将融入到智能电网建设中。到目前，我国各类电能表的主要产品均已达到或接近国际技术标准。智能电网的总体目标是突破大规模间歇式新能源电源并网与储能、智能配用电、大电网智能调度与控制、智能装备等智能电网核心关键技术。据国家电网 2015 年计量工作推进会议，今年的计量工作目标包括安装智能电表 6060 万只，可见中国智能电表市场潜力巨大。未来两大电网合计智能电表和用电管理系统市场约每年 160 亿元以上。2015 年，全国智能电表用户数将超过 1.4 亿。到 2020 年，国网、南网将全面建成智能（绿色）电网。目前，国网挂网电表 2.2 亿台、南网挂网电表 5000 万台，以及新增用户都将更换成智能电表。

由以上分析可知，世界各大主要国家都确立了智能电网建设目标，投资方案和行动计划，当前整个电力行业都在面对着巨大挑战，如何在满足经济发展要求下建立安全可靠、经济环保的电力环境成为大家共同的目标。其中，电能计费的准确性和可靠性直接关系到用户的经济利益，是决定智能电网技术顺利发展的重要因素。同时智能电网的网络结构比传统电网更合理，它可以根据不同的拓扑结构最优的智能控制系统。但是，传统电能表主要为感应式电表，设计原理与仪表结构相对简卑，成本也较低，导致了其测量精度低、可靠性差的缺点，已经很难适应工业现代化和供电管理现代化的发展需求。

因此，智能电表作为一种全新的电子式电能表，将成为智能用电中不可或缺的设备[4]。它可以省去不必要的人力，及时准确地把用户所用电量，远程传送到中心计算机，通过数据库来计算收费，实时地向用户、能源供应商或政府监管部分提供准确的功耗数据和设备状态信息，大大简化了抄表的繁琐工作。为供电公司的负荷预测提供相关的依据，实现负荷预测准确度的提高，从而降低电网备用容量，让更多的电力资源能够提供给用户使用[5]。

同时，用电信息采集系统的产品发展，如本地控制系统和数据存储平台必须跟电能表产品的发展同步。人们会越来越重视产品的可靠性，并且在技术上也会要求产品向多功能化、模块化、智能化方向发展，而且能兼容更多的一些新技术，使电力公司进行各种远程的运用成为可能。现阶段，电力系统实现信息化过程会遇到如异构性、信息孤岛、多信息源等诸多问题。但是，物联网作为互联网的延伸[6]，它包括互联网以及互联网上所有的资源，[7]将智能感知、识别技术与普适计算广泛应用于网络的融合中，

形成人与物、物与物的互联，实现智能化、信息化的远程管理、控制网络，为整合分散资源提供了极大的便利和可能，智能电表势必也会不断地更新，往智能家居方向发展。

1.2 价值和意义

Roy Fielding 博士在 2000 年他的博士论文“Architectural Styles and the Design of Network-based Software Architectures”[8]中提出来的一种软件架构风格 REST(表述性状态传递) 架构。它是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。经过一段时间认可之后，REST 设计架构开始被很多大公司使用，例如，Amazon.com 提供接近 REST 风格的 Web 服务进行图书查找；雅虎提供的 Web 服务也是 REST 风格的。

本论文使用目前非常流行并在迅速发展的 REST 体系结构，来实现智能服务器端，将会在下面方面对系统设计有很大改进：

易用性：将服务器抽象为一组离散资源的集合，客户端只需要向服务器提供的不用资源的接口发出请求便可以得到想要的数据，对于以前的 RPC 面向过程的执行方式更加方便有效。

安全性：这里仅指应用层次的安全性(数据库安全性不在此讨论)，主要通过用户权限、角色分配来实现。对于普通查询用户来说，通过登录来完成用户身份的认证；对信息维护用户和系统管理用户来说，还需要通过密钥加密等方式提供附加的安全性；

部署简便性：系统用户分布广泛，因此本文使用 B/S 模式，用户只需要在有网络的地方访问服务器接口便可以得到数据；

可扩展性：对于任何服务器端的资源，都是用过接口的方式访问，如果需要添加任何一项功能，只需要在服务器端添加相应的接口，然后既可以选择冷部署也可以选择热部署，视情况而定；

可配置性：对于不同的数据传输方式，只需要在 REST 架构中加入不同的传输协议便可以使用操作。

1.3 国内外软硬件研究现状

1.3.1 硬件现状

在美国，“智能电网”这样的概念，是由 IBM 公司提出的[9]。国外在自动抄表系

统的理论和技术研究比较早，而且到目前的发展也已经比较成熟，美国在 1986 年就已经成立了自动抄表研究协会 AMR，欧洲和英国也相继成立了各自的自动抄表技术协会，目的是推动 AMR 技术的发展，扩大 AMR 技术的应用领域。欧美一些发达国家早在二十世纪 70 年代就已经开始对远程抄表技术进行研究，实现数据的远程采集与控制，在这期间，也开始研究低压电力线载波技术在数据抄表中的应用并投入使用。随着计算机技术、集成电路的飞速发展，远程抄表技术已成为各国 AMR 系统生产厂商和科研机构关注的热点课题，通过不断的交流与合作，各种技术的不断发展也促进抄表技术的不断进步。

在架构上，国外已经大规模应用 SOC (System On Chip)技术。所谓 SOC 技术，从广义上理解，是应用高集成度的 SOC 芯片，并围绕 SOC 芯片进行二次集成开发采取的一系列技术手段。SOC 芯片推动了电表主控芯片的集成化发展，例如 Teridian 的 654x 系列单片机，飞思卡尔的 MZ 系列单片机，德国的英飞凌也开始了电表行业 SOC 芯片的研究。而随着网络技术和通信技术的发展，数据通讯问题也成了远程抄表中的关键技术，也是在抄表器设计时考虑的首要问题。远程抄表系统也不断地向前进步，数据通讯问题也成了集中器在抄表系统中首要解决的问题发达的国家如日本、美国、英国等国家基本都解决了数据通讯的问题。到目前为止，集中器在抄表系统中按通讯方式划分主要分为以下几种：有线信道通讯、无线信道通讯。有线抄表系统需要进行综合布线，增加了抄表系统的费用和难度，但是可以保证数据通讯的稳定性和可靠性。无线抄表系统则无需要进行布线，只是利用无线收发装置设备进行数据的传输，系统结构设计相对比较简单，能够节省大量的人力和物力资源，但是无线抄表存在接收数据不稳定的现象，受环境的因素影响很大。随着通讯方式的发展远程集抄表系统也在不断地向前进步，逐步取代人工抄表，大大提高了电业局售电管理业务的自动化程度。在整个抄表的系统中，协调器是连接整个系统的枢纽，它能够将采集器或多功能电能表的数据发送给本地控制系统，又可以将本地控制系统发送的命令传至给采集器或多功能电能表。

相比之下，国产智能化仪表无论是设计还是制造都明显弱于国际先进水平，国内电力自动化用户在智能电表的使用经验方面也相对积累较晚、较少，这些现状表明我国智能电表的应用还只是处于一个初级阶段但是在我国远程抄表系统的研究起步比较晚。自 20 世纪到了 90 年代中期以来，在国内的部分企业和研究机构带领下，但发展非常迅速，开发潜力巨大，市场极其广阔。特别是近几年来的发展与进步，都说明远程抄表系统具有发展的潜力。现在智能电表往往两种架构来进行设计:

其中一大类是采用“专用计量芯片+MCU+专用 RTC”的解决方案，随着 MCU、ARM 等通用硬件以及硬件新用法的引入，这种架构设计方式的电能表虽然组合的方式多种多样，无论是在功能上还是在性能上均为系统设计获得了很大的灵活与选择余地，使

得智能电表架构设计发生了很大的变化。Atmel、飞思卡尔、凌阳等生产的 8 bti、16 bit MCU 占据了大量的中低端电能表市场。单其中不可避免的问题是核与核之间匹配以及通信传输的可靠性问题都是不可避免的，而且多核方案的印制板电路也相对复杂，由于使用了数量较多的芯片，不可避免的造成电能表价格昂贵、维修不便等问题，这都给智能电表的发展带来了阻碍。

另一大类是 MCU 加专用集成芯片(ASICs)架构，MCU 负责控制并读取专用集成芯片采集和处理的数据结果。像珠海炬力、ADI 等许多国内外芯片制造商纷纷结合自己的技术优势定制出许多高性能的电能专用计量芯片模块，如炬力的 ATT7021、ATT7022B、ATT7023 等三相多功能计量芯片，ADI 的 ADE7754、ADE7755、ADE7758 等大量专用集成芯片的引入，有利于智能电表的快速开发和生产降低的成本。

在智能电表的需求巨大的市场下，我国各种各样的生产预付费电能表仪表厂商应运而生，由于我国在这方面没有监督体制，使得种类繁多的生产厂商都自行拟定了开发仪表的管理协议。这就造成基于仪表的各预付费电能表管理系统使用的协议是不同的，制约了数据的通信，造成了不能统一管理的诸多问题[10]。鉴于当前我国电力系统的现状是，不同的区域采用的电能表不完全相同。当各个区域需要并网时，会造成系统计量主站无法自动化的对多种电能表进行收费。

1.3.2 软件现状

我国的信息化已经历了数字化和局域应用两个主要阶段.数字化带来了信息技术的初步普及和推广，使信息资源的概念深入人心；局域应用引入了网络的概念，开发并应用了大量的网络信息系统[11]．这两个阶段开发的信息系统一般基于 C／S 或 B／S 模式，都有各自的优缺点．随着对信息化要求的提高，人们希望将两种模式的优点进行结合[12]。

自微软推出.NET 战略以来，智能客户端(Smart Client)技术得到广泛的应用[11]．智能客户端是一种新型的客户端技术，它结合了胖客户端[13-14]和瘦客户端[15]的优点。

如图 1 所示。

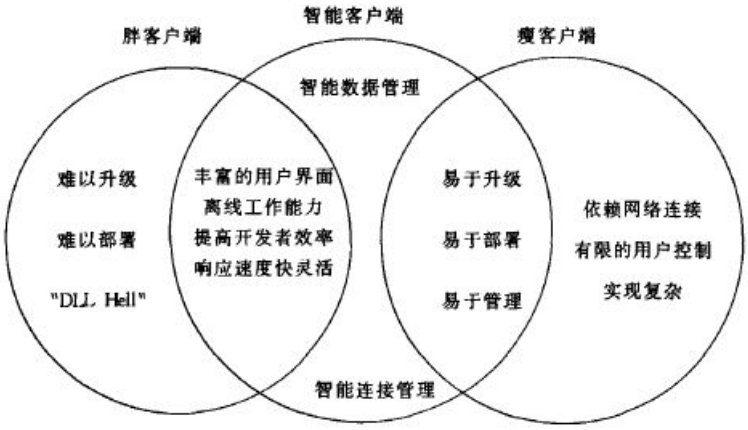

图 1 智能客户端结合胖客户端和瘦客户端的优点

通过采用智能客户端设计方案，充分利用服务器端 WEB 服务提供的功能，基本满足了上述用户需求[16]，现在大部分公司都开始采用 Web 服务器加智能客户端的体系结构，如图 2 进行智能设备上位机系统的开发与设计[17]。

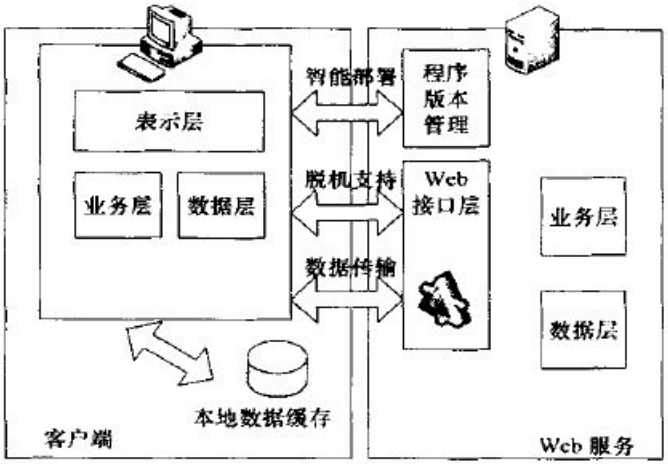

图 2 智能客户端体系结构

微软. NET 平台下已形成 Windows 智能客户端，Office 智能客户端[18]以及移动智能客户端[19-20]等解决方案. 在 2005 年，. NET 平台的智能客户端应用程序的比例达到了 60%. 在国内，新中大推出了自己的智能客户端技术，已有两个基于智能客户端技术的系列产品：Gsoft / G6 产品采用了符合 J2EE 标准的应用平台和工具；A3 产品采用了 Microsoft. Net 平台. 有从事管理软件研究的人士认为，智能客户端将会成为目前 ERP 产业技术创新的最佳切人点，向“智能客户端”的迁移将打破现有 ERP 的格局，摆脱 ERP 产业的恶性竞争。对于服务器端的设计，2000 REST 的主要框架已经开始出现，但

仍然在开发中，REST 的架构的出现和已有的 RPC 和 SOAP 等面向过程的服务器端开发完全不同 REST 架构将整个服务器中的所有资源抽象化。REST 从以下三个方面资源进行定义：

直观简短的资源地址：URI，比如：http://example.com/resources/。

传输的资源：Web 服务接受与返回的互联网媒体类型，比如：JSON，XML ，YAML 等。

对资源的操作：Web 服务在该资源上所支持的一系列请求方法。

资源	GET	PUT	POST	DELETE
一组资源的 URI，比如 http://example.com/resources/	列出 URI，以及该资源组中每个资源的详细信息（后者可选）。	使用给定的一组资源替换当前整组资源。	在本组资源中创建/追加一个新的资源。该操作往往返回新资源的 URL。	删除整组资源。
单个资源的 URI，比如 http://example.com/resources/142	获取指定的资源的详细信息，格式可以自选一个合适的网络媒体类型（比如：XML、JSON 等）	替换/创建指定的资源。并将其追加到相应的资源组中。	把指定的资源当做一个资源组，并在其下创建/追加一个新的元素，使其隶属于当前资源。	删除指定的元素。

表 1：HTTP 请求方法在 RESTful API 中的典型应用[21]

REST 架构可以支持各种传输方式，例如现在移动应用端非常流行的 JSON 传输方式，当然也支持基本的 XML 传输，REST 的传输方式是可以配置的，如果你需要添加新的传输方式，例如 Google 近期开源的 ProtoBuf 格式，只需要配置上具体的 Provider 类，便可以像其他的传输方式一样使用。

1.4 论文组织结构

各章节内容安排如下：

第一章对智能电网和智能电表的发展现状进行了概括，世界各国在智能用电的道路上进行深入的探索。提出论文的选题意义和价值，说明论文的结构安排和所做的工作。

第二章说明项目的主要技术背景和需求分析，并根据比较选择适合本项目的技术。

第三章从整体上把握智能电表的设计，分析硬件设备的选取，以及项目实现中数据传输所使用的具体方式。

第四章详细说明 Web 服务器的设计与实现，分别阐释了各个模块的关系和应用，以及在实际操作中遇到的问题和解决方案。

第五章总结本文所做的工作，对设计完成的智能电表本地控制系统进行概括，并作出新的前景展望。

第二章 技术介绍与需求分析

根据中华人民共和国节约能源法的规定：每年能耗耗电万千万时以上或每年耗能相当于万吨以上标准煤的企事业单位均属于重点能耗单位，高校则位列其中[22]。本文设计了一种基于物联网的分布式能耗采集控制系统，为校园的能耗监管提供了信息化手段。 此系统通过对校园各建筑的用能系统实时计量，建立校园能耗数据库，实现对校园的用水、气、暖、冷和电等能源资源消耗量的存储[23]。

2.1 Java 多线程方式

目前的计算机操作系统都可以同时执行多个程序，也就是多进程的处理模式。在操作系统的发展历史中，进程的概念源于多任务处理的思想。为了提高程序的运行效率，在编程时也要考虑多任务并行执行[24]。但是，进程是相互独立的具有自己专业的全程环境，它们占用保留自己的 CPU、数据和程序代码，这样造成 CPU 的负担也会造成内存的极大浪费。在这种情况下，产生了线程的概念。同一个进程的多个线程共享同一块内存空间，共享该进程的全程环境，提高了 CPU 利用率[25]。线程唯一的专用部分是执行时使用的寄存器和堆栈。所以，利用线程操作可以大大减少系统在任务切换上花费的时间，从而提高程序和系统的运行效率[26]。

Java 虚拟机提供了一个多线程机制，可以在同一份程序中产生多个进程的效果，在不同的线程之间具备某种程度的资源共享。由此引发的最重要问题就是线程的管理、并发执行和共享数据的协调保护。当多个线程操作共享数据时，为了保证数据的一致性和完整性，通常要求一个线程对共享数据操作时，另一个线程则不能同时操作该共享数据，否则会导致线程共享数据发生冲突[27]。如表 2.1 所示是 Java 线程在不同状态时的标识和说明。

Java 线程状态表	
状态	功能描述
NEW	新建线程，线程尚未启动
RUNNABLE	线程就绪
RUNNING	线程运行中
BLOCKED	其他线程占用资源，本线程等待执行中
WAIT	释放本线程占用资源，进入等待中
SLEEP	线程在规定时间内进入睡眠状态
DEAD	线程运行结束

表 2.1 Java 线程状态表

参照表 2.1，图 2.2 所示是在 Java 多线程中各个主要状态之间的转换流程图。从结构上划分，Java 中的每个线程具有五个基本状态：线程创建，线程就绪，线程运行，线程等待和线程结束。其中最有创造性和复杂性的就是线程等待，它关系到如何协调好线程之间的时间片分配，是为了解决线程冲突问题而设立的，也就是解决多个线程共享同一资源的问题。

线程同步是指在多线程环境中，可能出现两个或多个线程同时使用同一个有限资源的情况，为了避免出现资源冲突问题，实现在临界区上的互斥访问，Java 提供了同步机制来保证任一时刻只有一个线程能够进入临界区[28]，采用的机制是对象锁机制[29]。这个机制的原理是 Java 的每个对象都有一个内置同步锁以及等待队列。Java 中可以使用 synchronized 关键字来取得一个对象的同步锁。用 synchronized 关键字修饰的方法和程序块只允许一个线程访问，其他线程需要进入等待队列，直到进入的线程退出，释放对象锁，这就实现了互斥[30]，也就是表 2.1 中 BLOCKED 的含义和作用。

除此之外，在线程等待中还有以下几个内容，sleep、wait 和 join 等。这些也都是为了防止线程冲突而设置的函数。sleep 使当前线程（即调用该方法的线程）暂停执行一段时间，让其他线程有机会继续执行，但它并不释放对象锁。也就是说如果有 synchronized 同步快，其他线程仍然不能访问共享数据。join 方法使调用该方法的线程在此之前执行完毕，即等待该方法的线程执行完毕后再往下继续执行。wait 方法经常和 notify 方法配合使用，其作用是当前线程暂停执行并释放对象锁，让其他线程可以进入 synchronized 数据块，当前线程被放入对象等待池中。当调用 notify()方法后，将从对象的等待池中移走一个任意的线程并放到锁标志等待池中，只有锁标志等待池中

线程能够获取锁标志；如果锁标志等待池中没有线程，则 notify()不起作用。

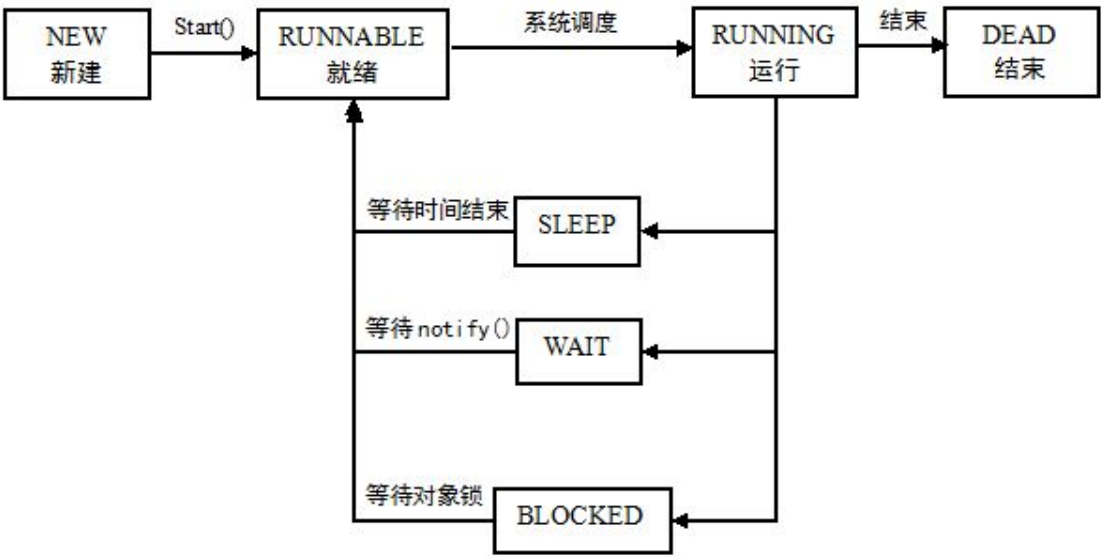

图 2.2 Java 多线程状态转换图

2.1.1 Future 多线程模型

Future 模式的核心在于：去除了主函数的等待时间，并使得原本需要等待的时间段可以用于处理其他业务逻辑。Future 模式有点类似于商品订单。在网上购物时，提交订单后，在收货的这段时间里无需一直在家里等候，可以先干别的事情。类推到程序设计中时，当提交请求时，期望得到答复时，如果这个答复可能很慢。传统的时一直等待到这个答复收到时再去做别的事情，但如果利用 Future 设计模式就无需等待答复的到来，在等待答复的过程中可以干其他事情。

例如图 2.3 的请求调用过程时序图。当 call 请求发出时，需要很长的时间才能返回。左边的图需要一直等待，等返回数据后才能继续其他操作；而右边的 Future 模式的图中客户端则无需等到可以做其他的事情。服务器段接收到请求后立即返回结果给客户端，这个结果并不是真实的结果（是虚拟的结果），也就是先获得一个假数据，然后执行其他操作。

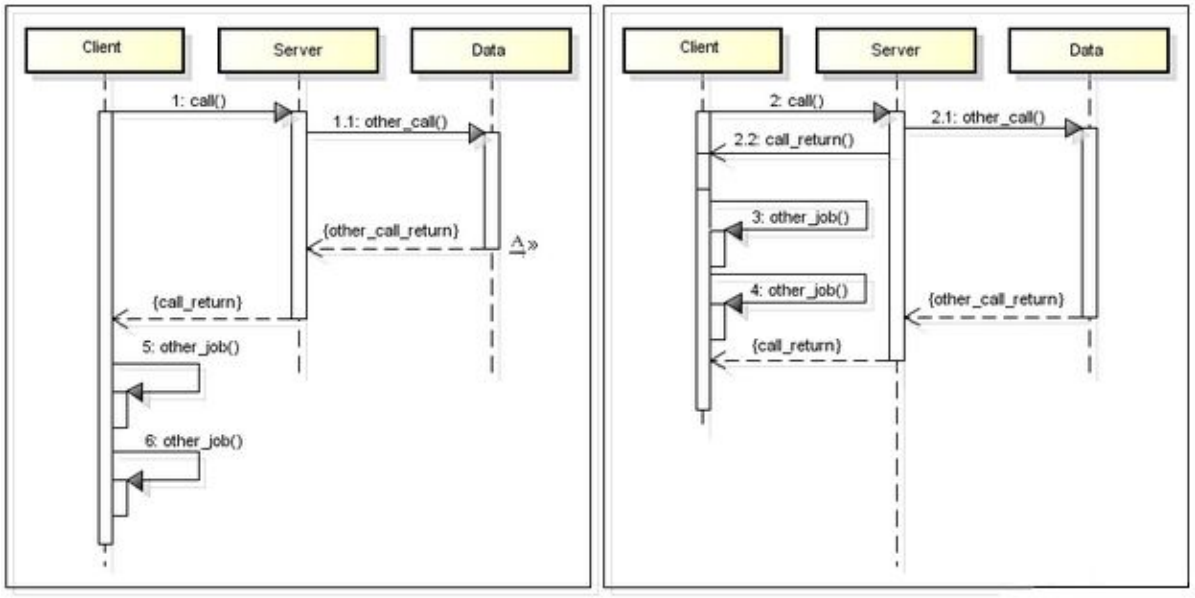

图 2.3 请求调用过程

客户端发送一个长时间的请求，服务端不需等待该数据处理完成便立即返回一个伪造的代理数据（相当于商品订单，不是商品本身），用户也无需等待，先去执行其他的若干操作后，再去调用服务器已经完成组装的真实数据。该模型充分利用了等待的时间片段。

2.2 服务器搭建方式

随着 Web 应用的商业逻辑包含逐渐复杂的公式分析计算、决策支持等，使客户机越来越不堪重负，因此将系统的商业分离出来。单独形成一部分，这样三层结构产生了。 其中‘层’是逻辑上的划分。

2.2.1 三层架构模型

通常意义上的三层架构就是将整个业务应用划分为： 表现层（UI）、业务逻辑层（BLL）、数据访问层（DAL）。区分层次的目的即为了“高内聚，低耦合”的思

想。

1、表现层（UI）：通俗讲就是展现给用户的界面，即用户在使用一个系统的时候他的所见所得。

2、业务逻辑层（BLL）：针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。

3、数据访问层（DAL）：该层所做事务直接操作数据库，针对数据的增添、删除、修改、更新、查找等。

三层体系结构对 Web 应用的软件架构产生很大影响，促进了基于组件的设计思想，产生了许多开发 Web 层次框架的实现技术。较之两级结构来说，三层结构修改和维护上 更加方便。目前开发 B/S 结构的 Web 应用系统广泛采用这种三层体系结构。 MVC（Model-View-Controller）是最常见的软件架构之一，业界有着广泛应用。它本身很容易理解，但是要讲清楚，它与衍生的 MVP 和 MVVM 架构的区别就不容易了。

2.2.2 M-V-C 模型

Model-View-Controller，严格说这三个加起来以后才是三层架构中的 UI 层，也就是说，MVC 把三层架构中的 UI 层再度进行了分化，分成了控制器、视图、实体三个部分，控制器完成页面逻辑，通过实体来与界面层完成通话；而 C 层直接与三层中的 BLL 进行对话。MVC 模型如图 2.4 所示

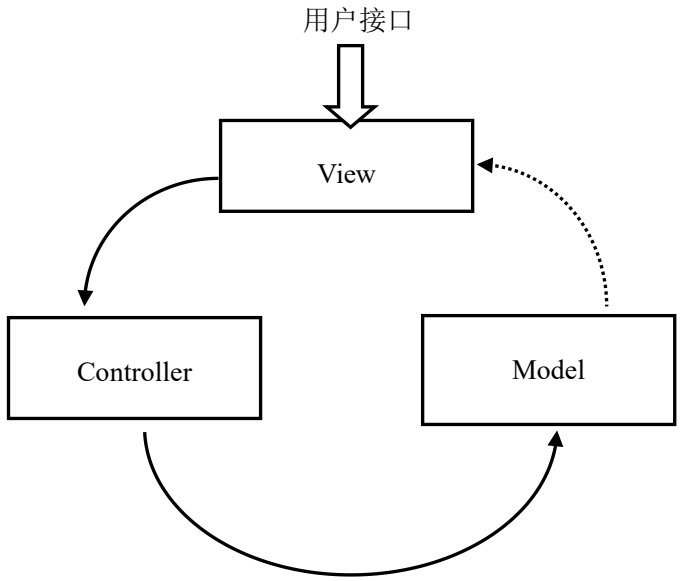

图 2.4 MVC 服务器搭建模型

MVC 模型之中数据流的通信处理过程如下：

1、View 接受用户的交互请求；

2、View 将请求转交给 Controller；
3、Controller 操作 Model 进行数据更新；
4、数据更新之后，Model 通知 View 数据变化；
5、View 显示更新之后的数据；、

View 和 Controller 使用 Strategy 模式实现，View 使用 Composite 模式，View 和 Model 通过 Observer 模式同步信息。Controller 不知道任何 View 的细节，一个 Controller 能被 多个 View 使用。MVC 的一个缺点是很难对 Controller 进行单元测试。

2.2.3 M-V-P 模型

如图 2.5 所示

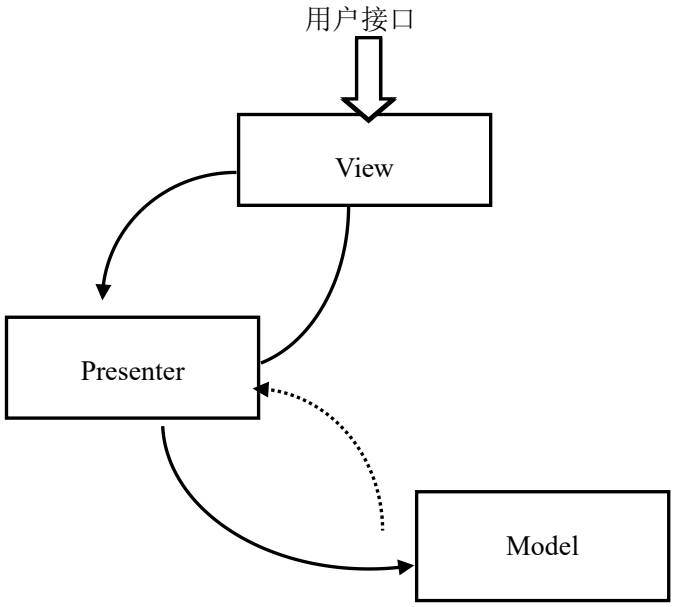

图 2.5 MVP 服务器搭建模型

MVP 模型之中数据流的基本处理过程如下：
1、View 接受用户的交互请求；
2、View 将请求转交给 Presenter；
3、Presenter 操作 Model 进行数据库更新；
4、数据更新之后，Model 通知 Presenter 数据发生变化；
5、Presenter 更新 View 的数据；

Presenter 将 Model 的变化返回给 View。和 MVC 不同的是，Presenter 会反作用于 View，不像 Controller 只会被动的接受 View 的指挥。正常情况下，发现可以抽象 View，暴露属性和事件，然后 Presenter 引用 View 的抽象。这样可以很容易的构造 View 的 Mock 对象，提高可单元测试性。在这里，Presenter 的责任变大了，不仅要操作数据，而且要更新 View。

在现实中，MVP 的实现会根据 View 的充、贫血而有一些不同，一部分倾向于在 View 中放置简单的逻辑，在 Presenter 放置复杂的逻辑；另一部分倾向于在 presenter

中放置全部的逻辑。这两种分别被称为：Passive View 和 Superivising Controller。

2.2.4 M-V-VM 模型

MVVM 是在原有领域 Model 的基础上添加一个 ViewModel，这个 ViewModel 除了正常的属性意外，还包括一些供 View 显示用的属性。例如在经典的 MVP 中，View 有一个属性 IsCheck，需要在 Presenter 中设置 View 的 IsCheck 值。但是在 MVVM 中的 Presenter 也会有一个 IsCheck 属性来同步 View 的 IsCheck 属性，可能会用到 Observer 模式同步 IsCheck 的值。在 MVVM 中，Presenter 被改名为 ViewModel，就演变成了你看到的 MVVM。在支持双向绑定的平台，MVVM 更受欢迎。MVVM 模型 2.6 所示

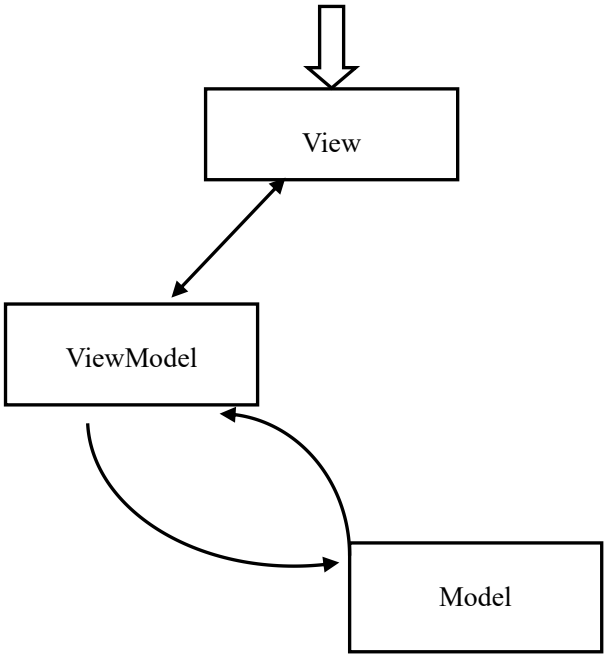

图 2.6 MVVM 服务器搭建模型

2.3 数据库存储

2.3.1 oracle

Oracle 是美国甲骨文公司开发的一种关系型数据库管理系统[31]。它以语言为基础，是目前最流行的 client/server 体系结构的数据库之一。数据库可支持多种不同的硬件和主流操作系统平台，采用开放的策略目标，从大型和超级计算机到台式机，具有高度的可伸缩性[32]，并提供广泛的国际语言支持，可以集中不同的计算机、不同的操作系统、不同的网络于一个信息处理系统中。Oracle 数据库是一个多用户数据库系统，除

了可以完成数据查询、操作、定义和控制等基本数据库管理功能以外，还具有完善的批处理和在线环境系统故障恢复功能。的产品覆盖了大、中、小型机等几十种机型，在数据库领域一直处于领先的地位，但是其属于企业级数据库，价格非常昂贵。

2.3.2 SQL Server

SQL Server 是美国 Microsoft 公司推出的一种关系型数据库管理系统[33]。它具有安全、可靠、可伸缩和集成度高等特点，是分布式 client/server 体系结构的数据库，可用于 Windows XP、Windows Server2003 等多种操作系统平台，支持 Windows 图形化管理工具，并可以管理和配置本地和远程的系统。SQL Server 具有强壮的事务处理功能，支持存储过程、ODBC，有自主的 SQL 语言。SQL Server 在 Windows 平台上和 Windows 操作系统有着很高的整体结合程度，并且使用方便，但是 SQL Server 却只能在 Windows 上运行。

2.3.3 MySQL

MySQL 是目前最受欢迎的开源数据库管理系统，在 Windows 平台和大多数 Linux 平台上都可运行的数据库，并且在包括 Linux 在内的任何 UNIX 平台下都是免费的。有一些不同于 SQL 标准的转变，这些转变大多是针对 SQL 脚本语言不足的补充，但也有一些扩展使得 MySQL 不同于其他的数据库，如 LINK 子句搜索时忽略大小写的区别，允许用户依照需要自定义 SQL 函数。MySQL 的插入和查询特别高效，并且支持存储过程和 ODBC，具有简单、高效、可靠的特点[34]。MySQL 是一个轻量级的数据库，在资源的适用方面伸缩性非常大，它可以在资源非常充裕的环境下运行，也可以在资源非常少的环境下运行。MySQL 是现在最流行的开源数据库管理系统，它不仅体积小，速度快，成本低，而且容易使用[35]。

2.3.4 数据库比较

数据库系统能够方便人们有效的组织和存储数据，并可以对数据进行高效的操作和维护。一个成熟的数据库管理系统，为满足不同的业务需求，其功能的实现是一个重点参考对象。MySQL 完全可以完成一些基本的结构化数据存储读取操作，它还基本具备所有通用数据库管理系统所具有的基本功能，并提供了一些简单的可编程支持。虽然与 Oracle 这样大型商业数据库管理系统相比，功能上还比较有限，但已完全满足无线远程抄表系统对数据信息的存储和管理的需要。所以，虽然作为一个通用的数据库管理系统 MySQL 还无法与其他几个大型商用数据库管理系统并驾齐驱，但其在功能上已基本满足企业在实际中的商业需求。

从系统易用性方面来看，与其他数据库相比，尤其是与一些大型的商业数据库管

理系统相比，易学易用的优势就更为凸显出来了。对于普通用户来说，它们的操作难易程度差距很大。MySQL 的简单易用，也直接节省了人力成本，公司不用再花费很多时间和金钱来培训人员或者花大价钱来雇佣专业的商业级数据库开发者。另外，在满足基本需求的前提下，MySQL 中没有多余的功能来拖累 CPU 或占用内存[36]。即使有些时候确实需要额外的功能，也可以通过一些其他的解决方案来满足需求。

毋庸置疑，几大商业级数据库经过大量的实践检验，已经得到了业界的认可，在可靠性方面几乎没有太大的问题，不过，值得一提的是 MySQL 作为开源数据库的代表，在可靠性方面也具有非常优异的表现。从这样的大型网站也使用数据库就可以看出，在稳定可性方面，并不比商业级数据库管理系统逊色太多。不仅如此，在全球排名前位的大型网站里，大部分都有部分业务是在数据库环境上运行的，例如 Yahoo、 Google 等。在国内，也有部分网站的数据库正在往上迁移，如阿里巴巴。

综上述可知对于本次以中小型项目为目标的设计，MySQL 可以在保证自身足够稳定性的前提下，尽可能地提高自身的处理能力，其性能高也一直是引以为豪的一个特点[37]。

2.4 网络安全模型

Web 安全需要新的思想和方法.1996 年,M. Blaze 等人为解决 Internet 网络服务的安全问题首次使用了“信 任管理(trust management)”的概念[43],其基本思想是承认开放系统中安全信息的不完整性,系统的安全决策需要 依靠可信任第三方提供附加的安全信息.信任管理的意义在于提供了一个适合 Web 应用系统开放、分布和动 态特性的安全决策框架架.与此同时,A. Adul-Rahman 等学者则从信任的概念出发,对信任内容和信任程度进行划分,并从信任的主观性入手给出信任的数学模型用于信任评估[39~42] 。

信任管理将传统安全研究中,尤其是安全授权机制研究中隐含的信任概念抽取出来,并以此为中心加以研 究,为解决 Web 环境中新的应用形式的安全问题提供了新的思路.信任管理研究可用于安全协议分析,并可结合 加密技术等研究成果,指导新的应用系统安全机制的建立。

2.4.1 网络攻击分析

网络攻击主要分为 1、利用 Web 服务器的漏洞进行攻击。如 CGI 缓冲区溢出，目录遍历漏洞利用等攻击；2、利用网页自身的安全漏洞进行攻击。如 SQL 注入，跨站脚本攻击等。攻击的主要方式有：

1、缓冲区溢出——攻击者利用超出缓冲区大小的请求和构造的二进制代码让服务器执行溢出堆栈中的恶意指令。

2、Cookie 假冒——精心修改 cookie 数据进行用户假冒。

3、认证逃避——攻击者利用不安全的证书和身份管理。

4、非法输入——在动态网页的输入中使用各种非法数据，获取服务器敏感数据。

5、强制访问——访问未授权的网页。

6、隐藏变量篡改——对网页中的隐藏变量进行修改，欺骗服务器程序。

7、拒绝服务攻击——构造大量的非法请求，使 Web 服务器不能响应正常用户的访问。

8、跨站脚本攻击——提交非法脚本，其他用户浏览时盗取用户帐号等信息。

9、SQL 注入——构造 SQL 代码让服务器执行，获取敏感数据。

10、URL 访问限制失效——黑客可以访问非授权的资源连接强行访问一些登陆网页、历史网页。

11、被破坏的认证和 Session 管理——Session token 没有被很好的保护 在用户推出系统后，黑客能够盗窃 session。

12、DNS 攻击——黑客利用 DNS 漏洞进行欺骗 DNS 服务器，从而达到使 DNS 解析不正常，IP 地址被转向导致网站服务器无法正常打开。

2.4.2 服务器安全设计

目标针对不同的网络攻击方式，主要的防范措施有一下几种：

1、登录页面加密

在登录之后实施加密有可能有用，这就像把大门关上以防止马儿跑出去一样，不过他们并没有对登录会话加密，这就有点儿像在你锁上大门时却将钥匙放在了锁眼里一样。即使你的登录会话被传输到了一个加密的资源，在许多情况下，这仍有可能被一个恶意的黑客攻克，他会精心地伪造一个登录表单，借以访问同样的资源，并访问敏感数据。通常加密方式有 MD5 加密、数据库加密等。

2、专业工具辅助

在市面目 前有许多针对于网站安全漏洞的检测监测系统，但大多数是收费的，但也有一些免费的网站安全检测平台，利用他们能够迅速找到网站的安全隐患，同时一般也会给出相应的防范措施。

3、加密连接管理站点

使用不加密的连接(或仅使用轻度加密的连接)，如使用不加密的 FTP 或 HTTP 用于 Web 站点或 Web 服务器的管理，就会将自己的大门向“中间人”攻击和登录/口令的嗅探等手段敞开大门。因此请务必使用加密的协议，如 SSH 等来访问安全资源，要使用经证实的一些安全工具如某人截获了你的登录和口令信息，他就可以执行你可做的一切操作。

4、兼容性加密

根据目 前的发展情况，SSL 已经不再是 Web 网站加密的最先进技术。可以考虑 TLS，即传输层安全，它是安全套接字层加密的继承者。要保证你所选择的任何加密方案不会限制你的用户基础。

5、连接安全网络

避免连接安全特性不可知或不确定的网络，也不要连接一些安全性差劲的网络，如一些未知的开放的无线访问点等。无论何时，只要你必须登录到服务器或 Web 站点实施管理，或访问其它的安全资源时，这一点尤其重要。如果你连接到一个没有安全保障的网络时，还必须访问 Web 站点或 Web 服务器，就必须使用一个安全代理，这样你到安全资源的连接就会来自于一个有安全保障的网络代理。

6、不共享登录信息

共享登录机要信息会引起诸多安全问题。这不但适用于网站管理员或 Web 服务器管理员，还适用于在网站拥有登录凭证的人员，客户也不应当共享其登录凭证。登录凭证共享得越多，就越可能更公开地共享，甚至对不应当访问系统的人员也是如此；登录机要信息共享得越多，要建立一个跟踪索引借以跟踪、追查问题的源头就越困难，而且如果安全性受到损害或威胁因而需要改变登录信息时，就会有更多的人受到影响。

7. 采用基于密钥的认证而不是口令认证

口令认证要比基于密钥的认证更容易被攻破。设置口令的目的是在需要访问一个安全的资源时能够更容易地记住登录信息。不过如果使用基于密钥的认证，并仅将密钥复制到预定义的、授权的系统（或复制到一个与授权的系统相分离的独立介质中，直接需要它时才取回），你将会得到并使用一个更强健的难于破解的认证凭证。

8. 维护一个安全的工作站

如果你从一个客户端系统连接到一个安全的资源站点，而你又不能完全保证其安全性，你就不能保证某人并没有在监听你所做的一切。因此键盘记录器、受到恶意损害的网络加密客户以及黑客们的其它一些破坏安全性的伎俩都会准许某个未得到授权的个人访问敏感数据，而不管网络是否有安全措施，是否采用加密通信，也不管你是否部署了其它的网络保护。因此保障工作站的安全性是至关重要的。

9. 运用冗余性保护网站

备份和服务器的失效转移可有助于维持最长的正常运行时间。虽然失效转移可以极大地减少服务器的宕机时间，但这并不是冗余性的唯一价值。用于失效转移计划中的备份服务器可以保持服务器配置的最新，这样在发生灾难时你就不必从头开始重新构建你的服务器。备份可以确保客户端数据不会丢失，而且如果你担心受到损害系统上的数据落于不法之徒手中，就会毫不犹豫地删除这种数据。当然，你还必须保障失效转移和备份方案的安全，并定期地检查以确保在需要这些方案时不至于使你无所适

从。

10. 确保对所有的系统都实施强健的安全措施，而不仅运用特定的 Web 安全措施

在这方面，可以采用一些通用的手段，如采用强口令，采用强健的外围防御系统，及时更新软件和为系统打补丁，关闭不使用的服务，使用数据加密等手段保证系统的安全等。

11、利用防火墙防护网站安全

例如使用操作系统自带的 Internet 连接防火墙（ICF），检查出入防火墙的所有数据包，决定拦截或是放行那些数据包。防火墙可以是一种硬件、固件或者软件，例如专用防火墙设备、就是硬件形式的防火墙，包过滤路由器是嵌有防火墙固件的路由器，而代理服务器等软件就是软件形式的防火墙。

12、运用网站监控措施

随着互联网的迅速成长，个人网站、企业网站、社区网站……越来越多，同时网站竞争也越来越强，从而衍生出来的对网站的监控，网站监控是通过软件或者网站监控服务提供商对网站进行监控以及数据的获取从而达到网站的排错和数据的分析。

2.5 RESTful 设计

REST(Representational State Transfer，表述性状态转移)，这个概念出自 Roy Fielding 所写的博士论文，代表的是一种架构风格，并非一种标准。这种架构风格目的在于简单化互联网软件的分布式设计，提高系统的可伸缩性。REST 设计风格有如下几点特征[44]：

1) 网络上的所有事物都被抽象为资源（resource）；
2) 每个资源对应一个唯一的资源标识（resource identifier）；
3) 通过通用的连接器接口（generic connector interface）对资源进行操作（通常为 HTTP）；
4) 对资源采取的各种操作不会改变资源标识；
5) 所有的操作都是无状态的（stateless）。

RESTful 作为一种基于 HTTP 协议和 REST 架构策略的简单 web 服务，已经成为一个普遍承认和遵守的 API 设计标准。知名的 RESTful API 有 GoGrid API、Rackspace、Amazon EC2 和 Sun，以及国内的 TOP（Taobao Open Platform，淘宝开放平台）。RESTful 风格可以说是一种轻量级的方法。在 RESTful 系统中，服务器利用 URI(Universal Resource Identifier) 暴露资源，客户端使用四个 http 谓词来访问资源[36]。因此需要显式地使用 CRUD 方法，即创建、读取、更新和删除 CRUD(create，read，update，and delete) ，从而与 HTTP 服务建立四种映射:

1) 若要在服务器上创建资源，应该使用 POST 方法。

2) 若要检索某个资源，应该使用 GET 方法。

3) 若要更改资源状态或对其进行更新，应该使用 PUT 方法。

4) 若要删除某个资源，应该使用 DELETE 方法。

总体来说，通过 RESTful 风格设计 Web 资源管理系统能够赋予其高伸缩性和高灵活性[37]。因为被指向和操作的资源是客户端通过 http 谓词来访问的; 而每个 Web 资源都通过唯一的 URI 进行开放，并且通过标准的语法进行统一的表述，从而简化了整个系统架构，改进了子系统之间交互的可见性，也简化了客户端和服务器的实现。

2.6 项目需求

本项目以实现绿色节能校园为目标，能够给学生提供舒适愉快的生活环境，主要从以下几个方面考虑：

（1）准确性。智能电表要准确获得电流电压等信息，这样返回的数据才有意义，为供后续阶段的数据分析提供保障。同时智能电表设备数量多，本地控制系统要准确快速的找到对应的协调器和智能电表，地址找不对得到的数据也没有意义。

（2）可靠性。数据丢包错包在传输过程中都是不能忽视的问题，所以通信质量不能忽强忽弱。无论是从本地控制系统到协调器，还是从协调器到智能电表，有线通信才是可靠地保证。

（3）实时性。本地控制系统应该能够实时反映智能电表采集的数据，后者也要及时执行本地控制系统发出的控制信息。在一些关键指令，比如控制开关、应急报警，需要软件和硬件协调配合完成实时控制。

（4）错误反馈机制。本地控制系统要包含数据包校验位，如果数据包出现错误，应该包含重试机制，否则及时记录在数据库中并提醒本地管理员哪只智能电表出现故障，防止出现用电事故。

（5）多功能性。和一般的智能电表相比，本项目中的智能电表加入了智能家居的一些特点，可以采集温度，湿度等信息并发出调控指令，改善学生生活环境的同时也会节能减排。

2.7 本章小结

本章主要分析对比在项目中应用到的一些关键技术和项目的需求分析。详细的讲解了项目中将会使用的技术要点，对数据的同步模块，远程服务器系统接收采用 Future 多线程模型，服务器的设计方式使用 MVVM 开发模型，简洁高效；web 安全采用了目

前非常流行的 Oauth2.0 授权认证方式，同时支持本地注册和登录两种方式，方便用户使用。数据的存储使用目前开源，开发成本低，开发速度快的 Mysql 数据库。

第三章 项目描述

本文中所描述的项目来自美国 International Technological University 实验室的 Green Campus 工程, 旨在推进绿色节约能源建设。本项目中智能电表和传统的智能电表最大的区别在于不仅能够获取基本的电流电压数据，还可以获取温度、湿度等其他重要信息，为学生提供舒适安全的校园环境。 整个智能系统包括五部分，分别是智能电表，协调器，本地控制系统，云端存储和客户端显示。智能电表和协调器基于德州仪器出产的 CC2538 SoC 芯片，而软件部分考虑到软件平台的可移植性，采用 Linux 操作系统。各部分之间的联系如图 2.1 所示。其中采集数据流程可以做如下描述：智能电表负责直接通过传感器采集用电数据或其他有关生活舒适度的数据，然后经过 RS485 传输给协调器；协调器收到其连接区域中的有效数据包，将继续传递数据到本地控制系统；本地控制系统负责存储和分析大部分的数据，考虑到本地计算机资源有限和为以后的大数据分析做准备，所有的数据信息会备份到云端存储器；客户端显示将通过浏览器访问，直接从云端存储部分提取数据，获得用户想要的信息。

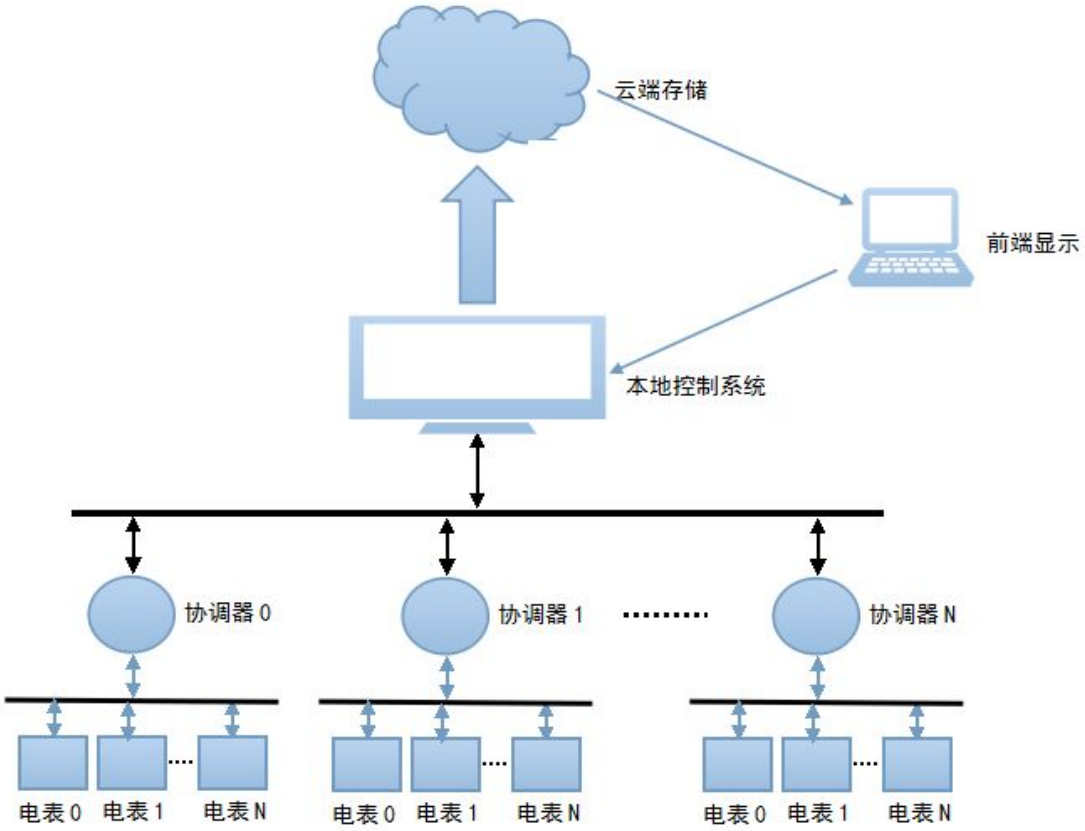

图 3.1 智能系统结构

3.1 硬件部分

3.1.1 CC2538 芯片

2013 年，德州仪器宣布推出 CC2538 片上系统，是一款简化支持 ZigBee 无线连接功能的智能能源基础设施、家庭楼宇自动化以及智能照明网关开发。本次项目中智能电表和协调器部分目前都是基于 CC2538 芯片来是实现用户需求。

CC2538 很重要的一个特性在于支持 Z-stack 操作系统。2017 年，德州仪器推出业界领先的 ZigBee 协议栈 Z-Stack，支持多种平台并包含了网状网络拓扑的几近于全功能的协议栈，在日益竞争激烈的物联网领域占有重要地位。Z-Stack 操作系统主要工作分为两步，其一是系统初始化，即系统启动代码需要完成初始化硬件平台和软件架构所需要的各个模块，为操作系统的运行做好准备工作，主要分为初始化系统时钟，检测芯片工作电压、初始化堆栈、初始化各个硬件模块等。其二是操作系统的执行，启动代码为操作系统的执行做好准备工作后，就开始执行操作系统入口程序，并由此彻底将控制权移交给操作系统，它的工作就是不断查询每个任务中是否有事件发生。

3.1.2 硬件通信方式

本项目中硬件通信分为两部分：智能电表到智能电表集线器的通信以及集线器到 Web 服务器的通信。

根据第二章分析可知，在智能电表和协调器之间采用符合工业标准要求的 RS-485 半双工连接。RS485 有线连接系统组织网络,实时读取终端数据到服务器端,并进行相应的数据处理,得到电力消耗的高峰期和低谷期,以便做出调整更合理的使用能源。智能电表会一直检测传输线的状态，当要发送数据且线状态为空是，写信号置高，发送数据到协调器。数据发送完毕后读信号置高，默认处于接收状态。

集线器到 Web 服务器之间采用网络通信。即可以示无限通信方式如 WIFI，也可是使用双绞线的网络通信方式。如果没有将 Web 服务器放在广域网中，则集线器只需要跟 Web 服务器在同一个局域网内，否则集线器也要连入广域网才能通信。

3.2 软件部分

考虑到开发的便捷性和系统的可移植性，本次项目是基于 Linux 平台，采用 Java 来编写本地控制系统和云端存储，JavaScript 等来编写客户端显示部分。本系统功能多样，后期会面临不断的升级和扩充，因此为了兼容系统更新更新，在设计之初就要符

合 OCP 的原则，对扩展开放。在软件部分用到了大量的流行技术，保证系统的稳定性和可重构性。

整个系统主要有三大模块构成，智能电表数据同步端、前台展示端、安全防护端。其中前台显示的整体设计是个相对复杂的过程[45]，它也可以直接访问本地数据库中的用户信息[46]，所以相关的需求分析、界面的排列方式等等都要提前考虑好。从控制指令的角度来讲，用户通过浏览器发送请求[47]也可以通过前台服务器调用本地控制系统的 API 实现控制功能和参数设定，达到用户预期的目的。Web 服务器采用轻量级的 Jetty web 容器，简洁易用。当用户想登录前台服务器时，会通过 Google Oauth2.0 认证，在客户端访问受保护资源之前，它必须先从资源拥有者获取授权，然后用访问许可交换访问令牌。客户端通过向资源服务器出示访问令牌来访问受保护资源[48]。保证用户信息的隐私和安全，防止用户习惯泄露带来不必要的麻烦。对于智能电表数据同步和安全性防护将在第四章详细介绍。

3.3 软件通信方式

JSON 是 JavaScript 对象表示法（JavaScript Object Notation）的简称，JSON 协议源自 JavaScript 脚本语言的对象持久化表示方法，由于这种表示法比较简单易懂，而且传输的数据也比较小巧（相对于 XML 来说应该算是非常小巧了），因此，近年来被广泛地用于互联网应用的数据封装。本项目中软件通信方式统一使用 JSON 格式进行通信。

软件通信，在本项目中主要是指，从硬件获取到电表数据之后，电表数据在整个项目中的流动，主要有三种方式：1.数据同步；2.数据显示；3.安全验证。

3.3.1 数据同步

数据同步模块主要功能是将从电表获得的数据，通过网络，保存到服务器数据库的过程，在数据同步过程中，数据包的格式不能由设计人员个人凭喜好设定，需要符合工业界使用的标准。下面是本项目中在同步数据时候使用的数据包格式。这里用中文表述不透漏具体字段信息。

1.同步电表本身参数格式

{所属系统 ID：value，设备 ID：value，设备名称：value，设备类型：value，设备状态：value}

2.同步电表数据格式

{所属系统 ID：value，设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}

3.同步系统本身参数格式

{所属系统 ID：value，系统名称：value，系统描述；value}

4.集线器同步电表数据格式

{所属系统 ID：value，[{设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}，... ，{设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}]}

3.3.2 数据显示

当服务器获得了电表数据之后，就需要以图表的形式将数据从数据库中存取出来，显示给需要使用的用户，比如：电表系统管理员，电表使用用户等。对管理员来说有可能从图表中看到电表取数据的异常值从而做出正确的判断，比如用户用电一直为零，可以判断此处线路损坏或者用户偷电漏电等。相应的本文中电表数据是以 RESTful API 的方式提供给前台页面显示，也需要符合工业界标准，本项目中用于数据显示的数据包类型如下所示：

1.获取电表本身参数格式

{所属系统 ID：value，设备 ID：value，设备名称：value，设备类型：value，设备状态：value}

2.获取系统本身参数格式

{所属系统 ID：value，系统名称：value，系统描述；value}

3.获取电表数据格式

[{所属系统 ID：value，设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}，... ，{所属系统 ID：value，设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}]

4.集线器同步电表数据格式

[{所属系统 ID：value，设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}，... ，{所属系统 ID：value，设备 ID：value，取值时间：value，电压值：value，电流值，value，相位值：value，温度值：value}]

5.控制电表状态

{所属系统 ID：value，设备 ID：value，RelayOn：value}

6.控制电表采集频率

{所属系统 ID：value，设备 ID：value，采集频率：value}

7.错误处理

{错误信息：value}

当 JSON 格式的数据包通过上述方式进行编码之后，只要使用者严格遵守这种格式的使用便不需要对系统实现细节做过多了了解，却可以灵活的使用，到后面结合 URL 的设计将会有更深的体会。

3.3.2 安全验证

Web 服务器的安全一直是众多软件开发者关注的比目标，对于用户身份的验证主要有两种方式：1.用户在本网站注册，使用密码和密钥访问；2.用户通过第三方授权，自动访问。本项目使用 Oauth2.0 授权客户端和 网站注册相结合的两种方式实现之智能电表 web 服务器的安全。

其中网站注册登录这种方式主要是如何 实现用户密码的安全性传输，目标比较流行的两种方式是 1.在登录之前，先把密码用 DES/SHA1 等加密，到服务器解密。2.数据库存的是密码的 MD5 散列值，每次登录前先 MD5 散列。第一种方式一旦被非法者截获 DES/SHA1 使用的 key ，密码便会被解密，第二种方式如果非法者模仿用户发送相同的请求到服务器同样可以假冒用户登录。本项目对注册登录安全方式应用了一种加盐的方式，并对其做了一定的修改，数据流如下：

1. 用户输入用户名 U 和密码 P，提交到服务端
2. 服务端通过 U 找到数据库中的相应的 salt 和 hashpass
3. 使用 salt 计算出用户输入密码的 hash 值，H = HASH(salt, P)，

或者是 HMAC(L_CASE(U), salt, P)

4. 对比计算结果 H 和数据库中取出来的 hashpass，如果一致，则验证成功。

其次对于 Oauth 2.0 客户端授权的方式，其主要原理是 OAuth 在"客户端"与"服务提供商"之间，设置了一个授权层（authorization layer）。"客户端"不能直接登录"服务提供商"，只能登录授权层，以此将用户与客户端区分开来。"客户端"登录授权层所用的令牌（token），与用户的密码不同。用户可以在登录的时候，指定授权层令牌的权限范围和有效期。"客户端"登录授权层以后，"服务提供商"根据令牌的权限范围和有效期，向"客户端"开放用户储存的资料。运行数据流如下：

（1）用户打开客户端以后，客户端要求用户给予授权。

（2）用户同意给予客户端授权。

（3）客户端使用上一步获得的授权，向认证服务器申请令牌。

（4）认证服务器对客户端进行认证以后，确认无误，同意发放令牌。

（5）客户端使用令牌，向资源服务器申请获取资源。

（6）资源服务器确认令牌无误，同意向客户端开放资源。

3.4 本章小结

本章主要介绍了智能电表项目的整个框架，无论是硬件设备还软件平台的使用都是基于用户需求所考量的结果。然后着重介绍了项目中软硬件数据交互的通信方式，同时对服务器安全性提出了自己的见解。

第四章 研究内容和重点

通过第三章的分析可以了解到，数据是否真实有效的反应设备的运行状态以及用户发出的命令能否被准确地执行都由 eb 服务器进行调度和管理。所以，Web 服务器是整个智能电表项目的核心，是连接了被监测设备和用户之间的桥梁。按照功能差异，可以将 Web 服务器划分为：数据同步模块、前台显示模块、安全验证授权模块和数据存储模块。如图 4.1 所示，其中，不同的模块通过与数据库的连接而紧密的结合在一起，例如数据同步模块接收数据写入到数据库，而前台显示模块读取数据库的电表数据，并以图表形式展示给前台用户。

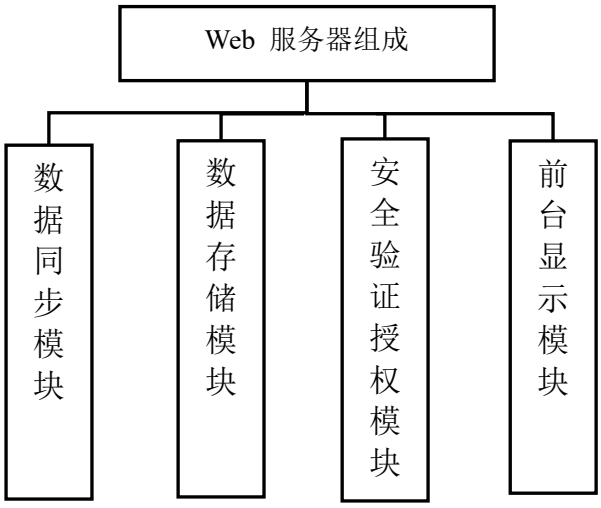

图 4.1 模块关系图

4.1 数据同步模块

跟据第三章中数据同步模块的介绍，我们知道在服务器完成其他工作之前需要将电表数据保存到 Web 服务器的数据库中，本项目中对于电表数据的接收是通过 RESTful API 的方式进行发送，客户端用到的手段，只能是 HTTP 协议。具体来说，就是 HTTP 协议里面，四个表示操作方式的动词：GET、POST、PUT、DELETE。它们分别对应四种基本操作：GET 用来获取资源，POST 用来新建资源（也可以用于更新资源），PUT 用来更新资源，DELETE 用来删除资源。在数据同步模块中主要是使用 POST 方式来新建电表节点，或者一个节点可能含有很多电表这种在工业中比较常用，同时使用 PUT 方式将电表中的数据发送到 Web 服务器的特定数据库中进行保存。本模块中需要解决的问题主要有以下几种：

1.符合工业界要求的 RESTful API 设计；

2.符合工业界要求的数据包格式；

3.发送数据安全性的保证；

4.对于不同的智能电表接收频率的控制；

对于第一点，在后面的 4.2 URL 设计方案本文会详细的阐述并给出实例，对于第二点请参考第三章软件通信方式；如何在硬件数据流传输上保证数据传输的安全，首先在创建智能电表节点的时候，是超级管理员通过 Web 服务器的 RESTful API 接口访问，本身就保证了创建节点的安全性和封闭性，同时在本系统的设计中，对去数据同步模块使用了一种 Server Event 驱动的模型，会在各个电表和服务器之间保持一种常链接，通过这个常链接可以让 Web 服务器时刻得知智能电表的状态，同时，通过这个链接服务器也可以解决第四个问题。

4.1.1 数据同步方式 POLL 与 PUSH

在数据同步方式中主要有 POLL 和 PUSH 两种方式：

POLL 方式，也称为轮循，是大家都比较熟悉的一种数据同步方式，客户端定期去 ping 查询服务器，确定是否有需要的数据。例如，软件更新模块，客户端软件需要定期去查询官方网站，判断当前是否有更新的版本，如果有就提醒用户进行升级。邮件客户端，需要定期查询邮件服务器，查询是否有新的邮件。RSS 阅读器，也是需要不断的查询 rss 地址的状态，如果有更新，就将数据拿回来。当服务器没有数据的时候，poll 方式会浪费大量的带宽。 为了降低带宽，通常是采用减低 poll 的频率来实现的，这就导致了消息的长延迟，实时性不高。像 gmail 的 POP3 邮件检查间隔从 10 分钟到 1 小时不等。

PUSH 方式，POLL 的问题在于很多情况下，通信信道是单向的。为了解决 poll 的问题，可以将通信信道设计为双向的，这样就可以服务器采用 push 的方式主动向客户端进行数据同步了。双向通信信道设计，考虑到要穿透 NAT 和防火墙，很多实现采用长连接。例如各种 IM 的实现：MSN 是 TCP 的长连接，QQ 是 UDP 模仿的长连接，GTALK 是 HTTP 模仿的长链接。

服务器主动向客户端推送数据。当前实现 PUSH 方式有两种方法：

1. 客户端首先连接到服务器，并维持长连接

2. 服务器能够直接访问到客户端，不需要长连接

在国内 NAT 和 firewall 遍地都是的情况下，第 2 种方法不是很可行，但是对于一些企业应用，这种方法还是不错的。

但是 push 方式，通常需要长连接，对于服务器端其实也是一个不小的压力，虽然现在 C10K 问题得到了比较好的解决，但是对于一些大规模互联网应用来讲，用户数是数以亿计的，单单是维持 TCP 连接，就需要太多的服务器。因此，除了一些实时性

要求比较高的应用，现在 push 方式使用的范围还不是很广，例如 push 方式在 IM、服务器监控等领域都有应用。

4.2 服务器设计架构

在第二章中详细的介绍了目前比较流行的服务器设计架构，本文基本架构设计遵循分层模块化的思想，基本符合 MVVM 设计方式，服务器与底层智能电表终端，前天页面显示模块和 Oauth 验证模块完全解耦和，只是通过相应的 RESTful API 机制进行数据的交互。

4.2.1 Server-Sent-Event 的机制

Server-Sent-Event 使服务器到客户端的数据传输变得高效——例如基于文本的事件流，实时通知或向客户端发送服务器上的更新。使用 Server-Sent-Event 基本工作流程如下：

1.客户端（Jersey Client）向 Web 服务器申请建立连接；

2.服务器（Web Server）向客户端反馈链接已经打开响应，此后链接一直保持；

3.服务器 前台接收到用户或者超级管理员请求；

4.服务器通过广播 将数据发送给 jersey Client。

Server-Sent-Event 的机制只是实现 PUSH 机制中的一种除此之外还有很多比较流行的技术如 Long-polling 和 Websocket 机制。

不同于 Long-polling 机制，Server-Sent-Event 不是每个连接发送一个消息。客户端发送一个请求，服务器端给客户端一个反馈并保持这个链接，一旦服务器端有新的消息，这个消息便会通过初始连接发送到客户端。客户端在处理完消息之后并不关闭这个链接，因此就实现了重复使用一个初始连接进行通信的方式。同时，定义了一种专用的媒体类型用于描述从服务器发送到客户端的单个事件。

4.2.2 服务器架构模型

再结合目前比较流行的服务器设计模型的基础上，本论文结合项目中实际需求与调查研究，设计的服务器架构模型如下图所示：

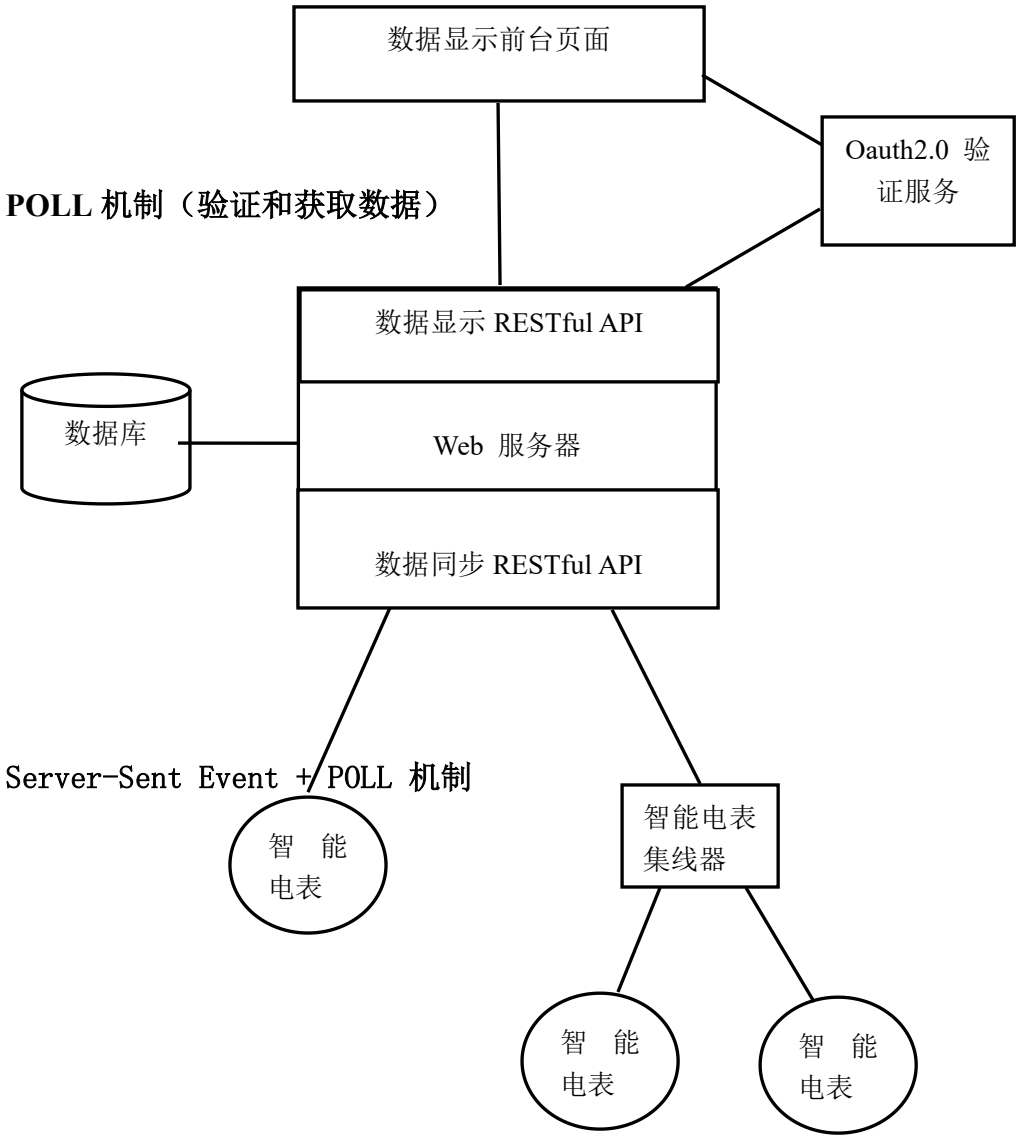

图 4.2 服务器架构模型图

4.2 URL 设计方案

我们先前提到的，HTTP 的访问使用以下四个 HTTP 动词确定在一个特定的资源上采取的行为[49]：

POST /uri 创建；

DELETE /uri/xxx 删除；

PUT /uri/xxx 更新或创建；

GET /uri/xxx 查看；

同时本文中参考 Lei Gao[50] 的方式对于获取同一层次资源信息时候定义了一个 LIST 方式，详细的 URL 设计请参考表 4.3 RESTful API 的设计。

访问方法	智能电表系统	智能电表	数据
LIST	/systems	/systems /{sys_id} /smartmeters	/systems /{sys_id} /smartmeters /{smartmeter_id} /datas
POST			
GET	/systems /{sys_id}	/systems /{sys_id} /smartmeters /{smartmeter_id}	/systems /{sys_id} /smartmeters /{smartmeter_id} /datas /{time_period}
PUT			
DELETE			

表 4.3 RESTful API 的设计

需要注意的是本项目实现过程中给予 Google Chrome 浏览器开发设计实现，支持 URL 设计过程中所有的方法对于一些个别浏览器不支持 PUT 或者 DELETE 方法的情况，可以通过发送 POST 请求并且在浏览器 URL 参数后面写上”_method”方式来使用。

4.3 OAuth2.0 安全客户端

前面已经提到并且说明 web 服务器安全的重要性，并且列举出一些列安全防范措施，本节主要讲述 Oauth2.0 安全客户端的实现。OAuth 2.0 客户端控制着 OAuth 2.0 保护的其它服务器的资源的访问权限。 配置包括建立相关受保护资源与有权限访问资源的用户之间的连接。 客户端也需要实现存储用户的授权代码和访问令牌的功能。本项目中 Oauth2.0 所使用的认证服务器是 Google 认证服务器，授权方式是授权码的方式，图 4.4 详细展示了授权的流程时序

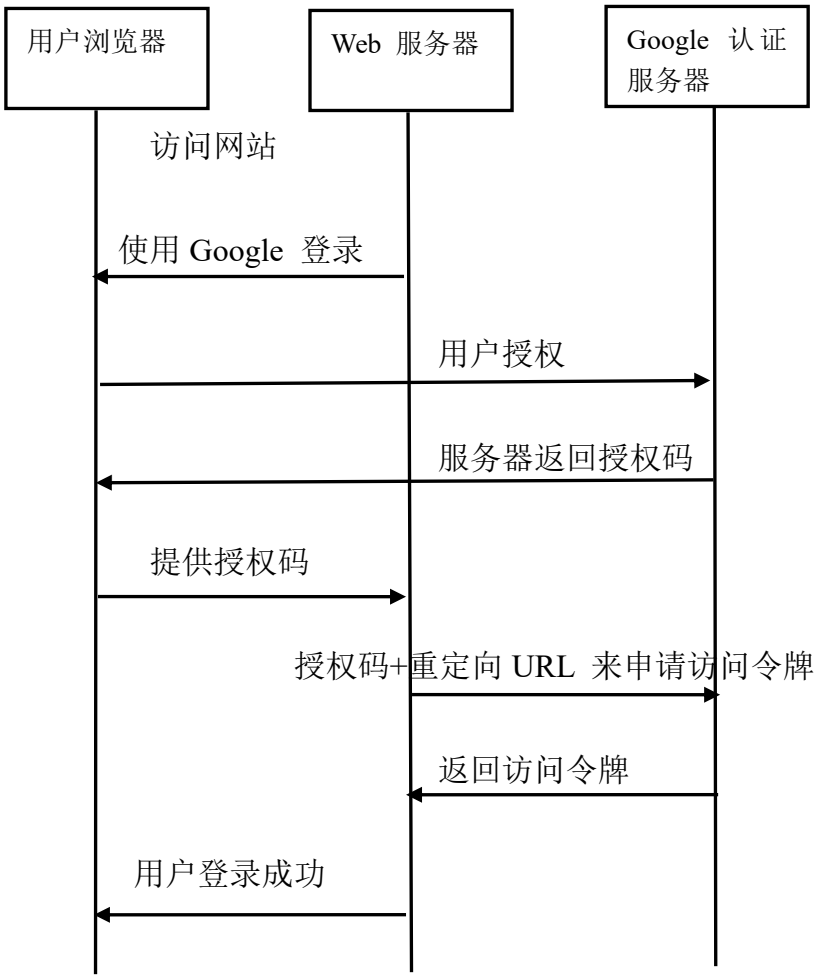

图 4.4 Oauth2.0 授权流程时序

第一步，用户访问客户端 web 应用。应用中的按钮”通过 Google 登录”。

第二步，当用户点击了按钮后，就被重定向到授权的应用(如 Google)， 然后用户登录，并被询问是否要授权应用中的数据给客户端应用，用户接受后。

第三步，授权应用将用户重定向到客户端应用提供的 URI，提供这种重定向的 URI 通常是通过注册客户端应用程序与授权应用程序完成。在注册中，客户端应用的拥有者组注册该重定向 URI，在注册过程中认证应用也会给客户端应用客户端标识和密码。在 URI 后追加一个认证码。该认证码代表了授权。

第四步，用户在客户端应用访问网页被定位到重定向的 URI。在背后客户端应用连接授权应用，并且发送在重定向请求参数中接收到的客户端标识，客户端密码和认证码。授权应用将返回一个访问口令。

一旦客户端有了访问口令，该口令便可以被发送到 Google 来访问登录用户的资源。

4.4 前台显示模块

本项目前台实现使用了 MVVM 模式中 View-Model 和 View 结合的方式，将页面和服务器的交互用 View-Model 隔离开来，更好的体现了搞内聚低耦合的思想，其中 View-Model 使用 Javascript 来实现实现，而对于 html 页面的设计本项目使用了开源了 bootstrap 鉴于其既能支持 PC 浏览器端也能自适应调整支持手机端，对于前台模块整个使用技术与功能划分如下：

1.使用开源的 bootstrap 实现 HTML 页面布局简洁高效；

2.使用 Javascript 的 AngularJS 开源开发框架，本身包含 Jquery 和 Ajax，容易使用；

3.使用 google visulization 实现图形化的展示，形象直观；

对于前台显示模块虽然都是访问 Web 服务器的 RESTful API 但是在前台不存在需要一直保持连接的情况，只需要使用 POLL 方式来进行客户端和服务器端的交互便已经达到要求，由于前台显示不是本文研究的重点，所以就简单的介绍下，本文使用的技术。

4.5 数据库设计

数据库设计是指对于一个给定的应用环境，设计优化数据库逻辑模式和物理结构，并建立数据库以及应用系统，使其能够有效地存储和管理数据满足各种用户的应用要求，包括信息管理要求和数据操作要求。数据库中的数据是按照一定的数据模型组织、描述和存储的，具有较小的冗余度、较高的独立性和易扩展性，并允许共享。数据库的应用已经越来越广泛，先进的数据库技术被应用到计算机领域，来保证数据的完整性和共享性。后台软件将智能电表的测量数据进行实时存储和处理，完成对冗余数据的过滤和优化，建立实时数据库机制[51]，为系统数据的后续应用提供了保障。

数据库设计将业务对象转换成视图、表等数据库对象，是数据库应用系统开发过程中首要也是最基本的内容。数据库是信息系统的核心和基础。它将系统中的大量数据按照一定的模型组织起来，并提供存储、查询和维护等功能，使信息可以及时、方便、准确的从数据库中提取出来。一个系统的各个功能是否能够准确、紧密的结合实现，关键在于数据库。所以，必须对数据库进行合理的设计。数据库图表如图 4.4 所示。

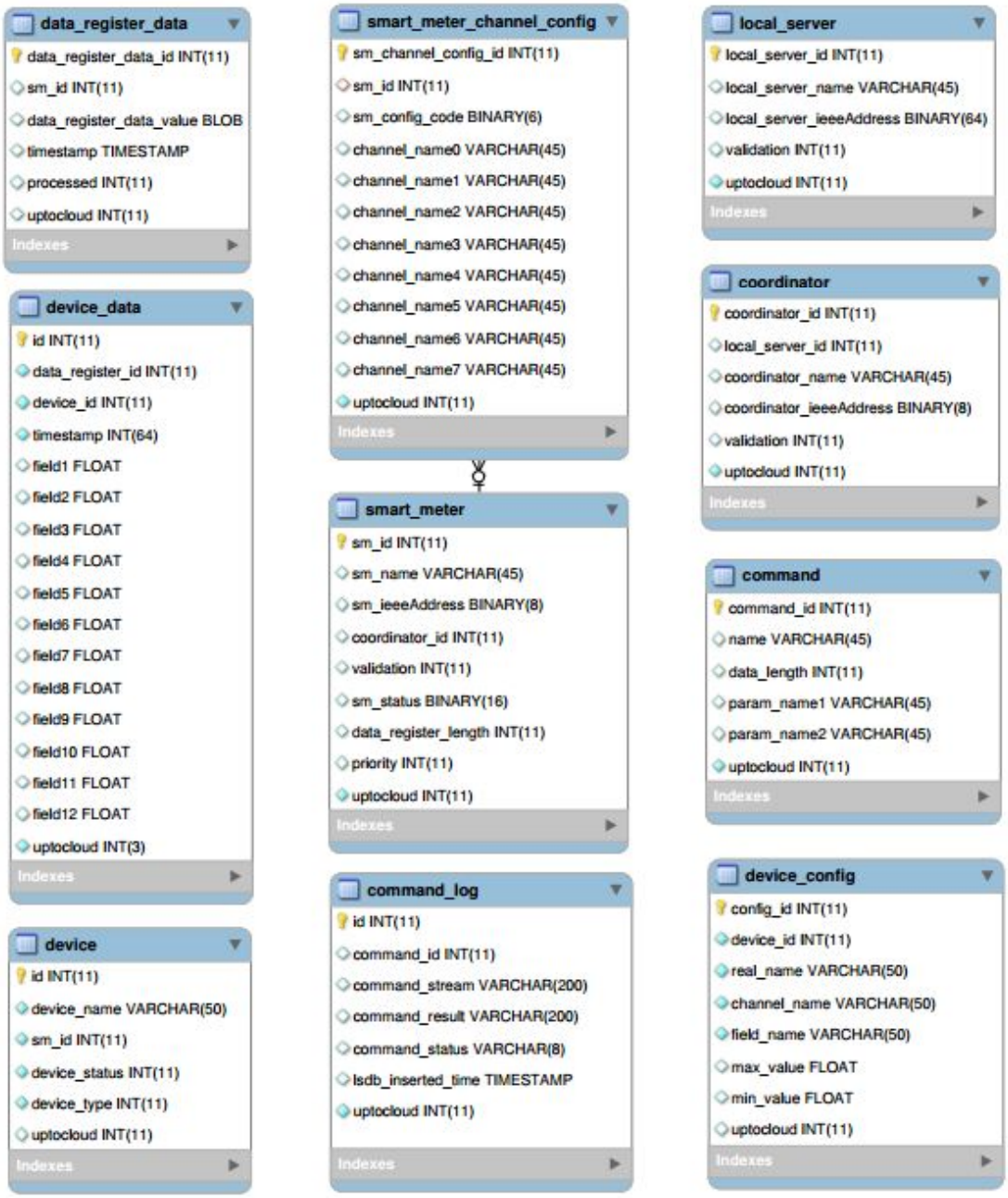

表 4.5 数据库设计

根据项目需求分析，在本项目中设计的数据表可以分为两类：

（1）配置表，即在本地控制系统采集数据之前要准备好的数据表。包括以下四张表格：

1）local_server,本地控制系统表。由于在实际应用中本地控制系统不止一个，为了在云端进行区分，在本表中包含了本地控制系统的编号和地址。

2）coordinator，协调器表。说明在同一个本地控制系统下有多少个协调器，主要内容包括协调器编号、协调器地址、本地控制系统编号等内容。

3) smart_meter,智能电表表。主要包括智能电表地址、协调器编号、智能电表状态和优先级等字段。

4）device，设备表。主要包括设备编号、设备名称、智能电表编号和设备状态等字段。

5）smart_meter_channel_config,智能电表多通道配置表。主要包含多通道配置编码和每个通道的实际内容名称。在采集数据的时候，不同的采集内容会有不同的校准方式，所以智能电表要先接受数据库中的配置编码，并依此处理不同的通道数据。

6）device_config, 设备配置表。多通道情况下，为了保证数据的准确性和后续的能量计算工作，哪些通道属于一个设备是需要明确标注的，所以本表主要包括设备编号、通道编号、实际测量内容等信息。

7）command,指令表。本地控制系统从数据库中读取指令表，然后发送给协调器并执行。主要字段包括指令编号、指令名称和指令长度等。

（2）采集表，即在本地控制系统启动后会频繁写入的表格，作为数据存储或者记录运行日志等工作。

1）data_register_data，原始数据表。本地控制系统收集到的数据会首先存入该表中，保证原始数据的完整性。然后由监测解码模块提取数据逐步处理。主要包含的字段有智能电表编号、原始数据、时间戳、处理状态和同步状态等。

2）device_data，设备数据表。以智能电表监测的设备为单位，原始数据过监测解码处理后会根据设备编号的不同存入设备数据表。主要包括设备编号、采集内容和时间戳等字段，其中采集内容为每个设备预留了 12 个字段，各字段的具体含义可以在 device_config 表中查找到。

3）command_log，指令日志表。为了能及时反应本地控制系统的运行状态，需要将关键的数据传输位置、内容和时间戳记录下来，方便在出现运行异常的时候进行修复工作。主要包括日志级别、指令编号、时间戳、指令返回结果和位置等信息。

4.6 本章小结

本章详细的论证了 Web 服务器系统各个模块的功能和工作原理，数据同步模块中的 Server-Sent-Event 和 POLL 相结合的模式满足了本项目的需求并且提高了项目的安全性和实时性。同时，满足工业界要求的 RESTful API 设计和数据包格式更彰显了本系统的应用型，表 4.3 更是详细的展示了目前 URL 的设计形式。对于网站的安全性本章主要对 Oauth2.0 安全客户端开发机制进行了详细的介绍，还有最重要的一部分就是数据库的设计，图 4.4 展示了目前表格设计结构，并简单说明了各表格的含义和用途。它们提供管理员插入、删除和搜索的功能，直接决定项目应该如何简洁高效的利用数据和存储数据。

第五章 结论及展望

5.1 结论

从绿色校园的方向出发，不仅是符合学校自身的长远发展建设，培育高素质人才，提高师生综合素质的需要，也符合建设资源节约型、环境友好型社会国家战略决策。随着智能电网在我国的大力发展，智能电表设计越来越成为研究的热点，本地控制系统作为远程抄表的通信枢纽，起着承上启下的作用。本文说明了智能电表的应用背景和国内外发展现状，论述了本地控制系在实际应用中的必要性并分析了系统的相关需求和可实现性，最后给出了系统的总体设计目标和思路、系统的层次模型、功能模型。

本文取得的主要成果有：

（1）本项目中实现了以 RESTful API 接口为主的智能电表 Web 服务器设计，是的对于新的智能电表的添加、删除、修改工作只需要在 HTML 页面中由超级管理员轻松修改编辑。

（2）本项目中可以通过 Web 服务器直接控制智能电表的行为，即只要在智能电表中集成相应的功能模块，同学统一的数据包个数，超级用户或者管理员就可以通过网页来直接控制智能电表所控制设备的开关，或者可以控制智能电表采集数据的频率，实现了可修改性电表。

（3）传统的电表只能采集以家庭为单位的用电信息，本项目的目标是获取主要用电设备的用电量，能更好的分析与制定节能方案。传统的电表只负责采集电流电压等基本用电信息，本项目设计的方案还可以采集温度湿度等与人们生活舒适度相关的数据，具有很强的扩展性。

（4）在网路安全性方面本项目研究并使用了一种比较合理的使用注册用户方式和 Oauth2.0 客户端验证相结合的方式。对于之前的注册方式，使用了目前比较流行的加盐的方式来保证密码传输的安全性，对于 Oauth2.0 客户端使用了相对比较安全的授权码的方式。

5.2 展望

本项目是将智能电表和智能家居、物联网相结合的一次尝试，通过在实验室中的测试与检验，其准确性、实效性和可靠性达到了基本的要求，但是还有些内容需要进一步的实验与完善。

1.服务器效率与能耗：本项目测试中仅使用了 5 个智能电表进行测试，数量比较少，所以让每个智能电表和服务器之间保持一个常链接对服务器负担不大，但是我们知道显示生活中有千万级别智能电表，要服务器去保持这么多链接是根本不可能的，虽然本项目中没有采取更加耗时的 Long-polling 机制，但是要 Web 服务器去维持这么多链接对服务器整体性能有很大的影响。这部分功能还需要更加完善。

2.对大数据的分析和处理：本项目中从智能电表采集了很多的电表数据，还没有进行合理的分析，试想本系统投入应用之中，如果能对采集到的电能数据进行大数据分析，将会根据每个电表数据使用地理分布，使用时间分布等等得到很多有价值的信息。

3.本人不是安全方向的，本次设计 Web 服务器安全模块得到很多启发，凡是还是感觉有些地方安全性存在忧患，比如如果不是本系统的智能电表能够模拟本系统中其他智能电表的数据发送，将会对本系统由不可估量的损失，所以这方面还需要加强。

4.对 Oauth2.0 server 的设想，在一个小系统中如果要保持一定的安全性，使用本文中采用的方法已经足够，但是如果真的应用到实际中去，去实现以个 Oauth2.0 Server 不妨是以个很好的选择，安全性会在这种情况下大大增加。

参考文献

[1] Chin-Feng Lai; Ying-Xun Lai; Yang, L.T.; Han-Chieh Chao, "Integration of IoT Energy Management System with Appliance and Activity Recognition," in Green Computing and Communications (GreenCom), 2012 IEEE International Conference on , vol., no., pp.66-71, 20-23 Nov. 2012

[2] 静恩波．智能电网发展技术综述[J]．低压电器，2010(6)：14·18．

[3] 宋春悦, 靳嘉桢. 智能电表 “芯” 保障——记 “智能电表关键芯片研发与应用” 项目[J]. 中国科技奖励, 2014 (4): 22-24.

[4] 李东东, 崔龙龙, 林顺富, 等. 家庭智能用电系统研究及智能控制器开发 [J][J]. 电力系统保护与控制, 2013 (4).

[5] 回海滨，封勇韬，李中伟，等.智能家电管理系统及其上位机软件设计[J].电器与能效管理技术，2014,10(54):75-78

[6] Wu J, Cheng Y, Schulz N N. Overview of real-time database management system design for power system scada system[C]//SoutheastCon, 2006. Proceedings of the IEEE. IEEE, 2005: 62-66.

[7] GONG G J，SUN Y．A study on the application scheme and framework of IoT technology regarding smart power grids［J］．Prot Contr Electr Pow Sys，2011，39(20) : 52-57．

[8] Fielding R T. Architectural styles and the design of network-based software architectures[D].Irvine,California:University of Califomia,Irvine,2000.

[9] Maciej J，Artur B，Maciej S．New approach for management services with a web browser[J]．Computer Networks，1999。31(21)：2227—2236．

[10] 王小雷.基于智能电表的 web 管理系统[J].软件, 2014(4):23-27.

[11] 刘海波，钟志农，陈宏盛，等．智能客户端技术研究及应用[J]．网络与信息技术，2006，25(11)：42—44．Liu Haibo，Zhong Zhinong，Chen Hongsheng，et a1．Research and application on intelligent client technology[J]．Network and Information Technology，2006，25(11)：42—44．(in Chinese).

[12] 冀庆斌,靳祯.基于智能客户端技术的系统升级及应用[J].中北大学学报（自然科学版）,2010,31(1):23-28.DOI:10.3969/j.issn.1673-3193.2010.01.006.

[13] Scott M．Frameworks for component—based client／server computing FJ]．ACM Computing Surveys，1998，30(1)：3—27．

[14] Mary C，Leslie P，Ashok S．A strategic client／server implementation：new technology，lessons from history[J]．The Journal of Strategic Information Systems，1997，6(2)：95-128．

[15] Maciej J，Artur B，Maciej S．New approach for management services with a web browser[J]．Computer Networks，1999。31(21)：2227—2236．

[16] 王介之,陈志刚.利用WEB服务实现智能客户端应用[J].计算技术与自动化,2005,24(1):85-87.DOI:10.3969/j.issn.1003-6199.2005.01.025.

[17] 朱涛,张水平,李云云等.基于智能客户端架构的自助服务系统的设计与实现[J]. 计算机工程,2007,33(16):205-207.DOI:10.3969/j.issn.1000-3428.2007.16.072.

[18] 方睿，郝玉洁．Smart Client 与 Office System 整合应用研究 I-J]．成都信息工程学院学报，2006，21(4)：479—483．Fang Rui，Hao Yujie．Smart client and office system[J]．Journal of Chengdu University of Information Technology，2006，21(4)： 479—483．(in Chinese)

[19] 卢宏基，付瑞峰，谈冉．基于移动协同的智能客户端研究 VJ]．武汉理工大学 学报(交通科学与工程版)，2007，31(4)：723—725．Lu Hongji，Fu Ruifeng，Tan Ran．Smart client research based on mobile CSCW model[J] ． Journal of Wuhan University of Technology(Transportation Science&Engineering)，2007，31(4)：723—725．(in Chinese)

[20] Chang Y F ， Chen C S ． Smart phone——the choice of client platform for mobile commerce[]]．Computer Standards 8L Interfaces，2004，12：329—336．

[21] Richardson, Leonard; Ruby, Sam, RESTful Web Services, O'Reilly, 2007 ((May 8, 2007)), ISBN 0596529260

[22] 张丽, 马宪卫, 王丽丽. 高校节能系统建设研究[J]. 信息系统工程, 2010 (11): 92-92.

[23] TANG Z．Key technology of smart power grids and the fusion with IoT technology[J]．J Shanghai Electr Pow Univ，2011，27(5) : 459-462．

[24] 费翔林, 骆斌, 孙钟秀. 操作系统教程[M]. 4 版. 北京: 高等教育出版社, 2008: 60-69.

[25] 陈益, 童亚拉, 杨晓艳. Java 多线程同步机制的应用分析[J]. 智能计算机与应用, 2012, 2(3): 86-88.

[26] 钱振江, 卢亮, 黄皓. 微内核架构多线程机制的形式化设计研究[J]. 计算机科学, 2013, 40(4): 136-141.

[27] 李娟. Java 多线程同步机制研究分析[J]. 中国科教创新导刊, 2014 (7): 183-184.

[28] 吴红萍. Java 的多线程机制分析与应用[J]. 软件导刊, 2014, 13(1): 114-116.

[29] 路勇. Java 多线程同步问题分析[J]. 软件, 2012, 33(4): 31-33.

[30] 韦庆清, 任卫东. Java 多线程编程中数据安全的应用研究[J]. 现代计算机: 下半月版, 2012 (17): 65-69.

[31] 祁新安，侯清江。SQLServer 数据库的运用研究[j]。制造业自动化，2010 ，32（12）：30-32

[32] 厉建欣, 司青燕. 论 MySQL 开源数据库在中小企业的应用[J]. 商场现代化, 2009 (1): 21-22.

[33] MySQL: the world's most popular open source database[M]. MySQL AB, 1995.

[34] 闪四清. 数据库系统原理与应用教程[M]. 清华大学出版社, 2008.

[35] 栾爽, 高玲, 李晶. MySQL 在商业运行模式下的优势[J]. 电脑知识与技术, 2010 (11): 2582-2583.

[36] 伍志聪. MySQL 数据库在中小型业务系统的应用[J]. 数字技术与应用, 2011 (11): 122-122.

[37] 李荣国, 王见. MySQL 数据库在自动测试系统中的应用[J]. 计算机应用, 2011, 31(A02): 169-171.

[38] Abdul-Rahman, A., Hailes, S. A distributed trust model. In: Proceedings of the 1997 New Security Paradigms Workshop. Cumbria, UK: ACM Press, 1998. 48~60. http://www.ib.hu-berlin.de/~kuhlen/VERT01/abdul-rahman-trust-model1997.pdf.

[39] Abdul-Rahman, A., Hailes, S. Using recommendations for managing trust in distributed systems. In: Proceedings of the IEEE Malaysia International Conference on Communication’97 (MICC’97).

Kuala Lumpur: IEEE Press, 1997. http://citeseer.nj.nec.com/ 360414.html.

[40] Yahalom, R., Klein, B., Beth, T. Trust relationships in secure systems——a distributed authentication perspective. In: Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy. IEEE Press, 1993. 50~164. http://isbn.nu/0818633700.

[41] Beth, T., Borcherding, M., Klein, B. Valuation of trust in open network. In: Gollmann, D., ed. Proceedings of the European Symposium on Research in Security (ESORICS). Brighton: Springer-Verlag, 1994. 3~18.

[42] Blaze, M., Feigenbaum, J., Lacy, J. Decentralized trust management. In: Dale, J., Dinolt, G., eds. Proceedings of the 17th Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society Press, 1996. 164~173.

[43] 刘金涛. 基于云计算的资源开放平台研究与实现[D]. 陕西西安：西安电子科技大学, 2012

[44] 葛管库. MVC 模式下程序设计[J]. 软件, 2013 (2): 49-51.

[45] 王国荣, 熊义君, 万利华. 温度检测系统的上位机软件设计[J]. 硅谷, 2010 (19): 54-54.

[46] 李少辉 . 面向对象与 MVC 框架的融合 [J]. 软件，2013,34（1）：82-84.

[47] 张卫全,胡志远.浅析作用于 Web2.0 安全防范的 OpenID 和 OAuth 机制[J].通信管理与技术,2011(2):15-18.

[48] R. T. Fielding, “Architectural Styles and the Design of Network-based

[49] Software Architectures,” Ph.D. dissertation, University of California,

[50] Irvine, 2000.

[51] Lei Gao; Chunhong Zhang; Li Sun, "RESTful Web of Things API in Sharing Sensor Data," in Internet Technology and Applications (iTAP), 2011 International Conference on , vol., no., pp.1-4, 16-18 Aug. 2011

[52] 贾东梨，孟晓丽 . 实时数据库在用电信息采集系统中的应用研究 [J].

致谢

本论文是在张奇勋老师的悉心指导下完成的。由于在国外实习，和张老师都是通过邮件来联系，来回的邮件中包括了从论文的选题、系统的设计思想以及论文的修改，老师都给予我耐心的指导和无私的帮助，引导我不断开阔思路，增长知识。同时我要感谢在美国给我许多帮助的 Dr Karl Wang 和 Dr May Huang ，是他们在美国对我悉心照顾和教导。

还要感谢在美国给予我帮助的老师和同学，使我能参与到绿色校园这个项目中来，学习和培养了软件设计思想，为以后的发展打下新的基础。最后特别感谢我的女朋友刘博，在我的身边一直支持我。

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名：　　日期：　　年　　月　　日

学位论文使用授权说明

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
- 因某种特殊原因需要延迟发布学位论文电子版，授权学校□一年/□两年/□三年以后，在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名：　　导师签名：

日期：　　年　　月　　日