EMC项目远程能耗数据采集与分析系统

设计说明书

2013年7月23日

**修改历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 作者 | 版本 | 修改内容及原因 |
| 2013-7-23 | 王强 | 0.01 | 创建初稿 |

**目 录**

[EMC项目远程能耗数据采集与分析系统 i](#_Toc363564365)

[1. 系统概述 1](#_Toc363564366)

[1.1 项目名称 1](#_Toc363564367)

[1.2 项目用户 1](#_Toc363564368)

[1.3 词汇表 1](#_Toc363564369)

[1.4 项目总体目标 3](#_Toc363564370)

[2. 系统总体设计 4](#_Toc363564371)

[2.1 系统软硬件要求 4](#_Toc363564372)

[2.2 系统部署图 4](#_Toc363564373)

[2.3 系统组件图 6](#_Toc363564374)

[2.4 系统模块结构图 7](#_Toc363564375)

[3. 远程采集客户端设计 9](#_Toc363564376)

[3.1 远程采集客户端概述 9](#_Toc363564377)

[3.2 远程采集客户端结构 9](#_Toc363564378)

[3.3 远程采集客户端冗余 10](#_Toc363564379)

[4. 过程数据库设计 11](#_Toc363564380)

[4.1 过程数据库概述 11](#_Toc363564381)

[4.2 过程数据库组件 11](#_Toc363564382)

[4.3 过程数据库结构 12](#_Toc363564383)

[4.4 过程数据库提供的接口 13](#_Toc363564384)

[5. 报警服务设计 14](#_Toc363564385)

[5.1 报警服务概述 14](#_Toc363564386)

[5.2 报警服务结构 14](#_Toc363564387)

[5.3 报警服务提供的接口 15](#_Toc363564388)

[6. 趋势服务设计 16](#_Toc363564389)

[6.1 趋势服务概述 16](#_Toc363564390)

[6.2 趋势服务结构 17](#_Toc363564391)

[6.3 趋势服务提供的接口 17](#_Toc363564392)

[7. 转储服务设计 18](#_Toc363564393)

[7.1 转储服务概述 18](#_Toc363564394)

[7.2 转储服务结构 18](#_Toc363564395)

[8. 系统平台网站设计 19](#_Toc363564396)

[8.1 系统平台网站概述 19](#_Toc363564397)

[8.2 系统平台网站结构 19](#_Toc363564398)

[8.3 系统平台网站的主要页面 22](#_Toc363564399)

[9. GIS模块设计 23](#_Toc363564400)

[9.1 GIS模块概述 23](#_Toc363564401)

[9.2 GIS模块结构 24](#_Toc363564402)

[10. 组态模块设计 25](#_Toc363564403)

[10.1 组态模块概述 25](#_Toc363564404)

[10.2 组态画面的描述 25](#_Toc363564405)

[10.3 组态模块结构 27](#_Toc363564406)

[11. 实时报警控件设计 29](#_Toc363564407)

[11.1 实时报警控件概述 29](#_Toc363564408)

[11.2 实时报警控件结构 29](#_Toc363564409)

[12. 趋势图控件设计 30](#_Toc363564410)

[12.1 趋势图控件概述 30](#_Toc363564411)

[12.2 趋势图控件结构 30](#_Toc363564412)

[13. 报表模块设计 31](#_Toc363564413)

[13.1 报表模块概述 31](#_Toc363564414)

[13.2 报表模块结构 31](#_Toc363564415)

[14. 身份认证与权限模块设计 33](#_Toc363564416)

[14.1 身份认证与权限模块概述 33](#_Toc363564417)

[14.2 身份认证与权限模块模型 33](#_Toc363564418)

[14.3 身份认证与权限模块结构 34](#_Toc363564419)

[15. 系统业务数据库设计 34](#_Toc363564420)

[15.1 E-R图 34](#_Toc363564421)

[15.2 数据库结构说明 35](#_Toc363564422)

[16. 系统转储数据库设计 40](#_Toc363564423)

[16.1 E-R图 40](#_Toc363564424)

[16.2 数据库结构说明 41](#_Toc363564425)

# 系统概述

## 项目名称

EMC项目远程能耗数据采集与分析系统。

## 项目用户

宝钢节能公司运维部、运改部、财务部等。

## 词汇表

表 ‑1 词汇表

|  |  |
| --- | --- |
| 名 词 | 说 明 |
| B/S架构 | B/S架构（Browser/Server，浏览器/服务器模式），是WEB兴起后的一种网络结构模式，浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。 |
| Web服务器 | Web服务器是可以向发出请求的浏览器提供文档和数据的程序。 |
| Tomcat | Tomcat是Apache 软件基金会的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成，是一个支持标准的Servlet和JSP规范的Web应用服务器。 |
| J2EE | J2EE是Java2平台企业版（Java 2 Platform,Enterprise Edition），J2EE核心是一组技术规范与指南，其中所包含的各类组件、服务架构及技术层次，均有共同的标准及规格，让各种依循J2EE架构的不同平台之间，存在良好的兼容性，解决过去企业后端使用的信息产品彼此之间无法兼容，企业内部或外部难以互通的窘境。 |
| Html | 超文本标记语言，即Hypertext Markup Language，是用于描述网页文档的一种标记语言，具有简易性，可扩展性和平台无关性等特点。 |
| Html 5 | 用于取代Html 4和XHtml 1标准的Html最新标准，在以前的版本的基础上强化了Web网页的表现性能，并且提供了更多能够有效增加网络应用的标准集。 |
| Javascript | JavaScript是一种基于对象和事件驱动并具有一定安全性的客户端脚本语言，也是一种广泛用于客户端Web开发的脚本语言，常用来给HTML网页添加动态功能，比如响应用户的各种操作。 |
| jQuery | jQuery是继prototype之后又一个优秀的Javascript框架。它是轻量级的js库 ，它兼容CSS3，兼容各种浏览器（IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+）。jQuery使用户能更方便地处理HTML documents、events、实现动画效果，并且方便地为网站提供AJAX交互。 |
| Json | JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式，它基于JavaScript标准的一个子集。 JSON采用完全独立于语言的文本格式，但是也使用了类似于C语言家族的习惯（包括C, C++, C#, Java, JavaScript, Perl, Python等）。这些特性使JSON成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成。 |
| XML | XML是可扩展标记语言的缩写 (eXtensible Markup Language) ，可以用来标记数据、定义数据类型，是一种允许用户对自己的标记语言进行定义的源语言。 XML是标准通用标记语言 (SGML) 的子集，非常适合 Web 传输。XML 提供统一的方法来描述和交换独立于应用程序或供应商的结构化数据。 |
| QT | QT是一个开源的跨平台C++应用程序开发框架，经过多年的发展，并且近期还在不断的改进更新。QT可以支持多种平台和操作系统，提供了丰富的API和类供用户使用。 |
| Spring | Spring是Java企业应用开发中最流行的控制反转和面向切面的轻量级容器框架。 |
| iBatis | iBatis是一个基于Java的半自动ORM持久层框架，iBatis  并不会为程序在运行时自动生成 SQL并 执行。具体的 SQL 需要程序员编写，然后通过映射配置文件，将SQL所需的参数，以及返回的结果字段映射到指定的Java对象。iBatis可以支持各种常见的数据库。 |
| SVG | Scalable Vector Graphics，是基于可扩展标记语言（XML），用于描述二维矢量图形的一种图形格式。 |
| Silverlight | Microsoft Silverlight是一个跨浏览器的、跨平台的插件，为网络带来下一代内容丰富、视觉效果绚丽的交互式体验应用。Silverlight提供灵活的编程模型，并可以很方便地集成到现有的网络应用程序中。 |
| 组态 | 组态模块可以使实施人员以灵活多变的组态方式来开发监控画面，并且开发过程方便简易。 |
| SSO | Single Sign On 单点登录。 |
| GIS | 地理信息系统，是一门综合性学科，结合地图学和地理学，是用于输入、存储、查询、分析和显示地理数据的计算机系统。 |
| WMS | WMS（Web Map Service）将地理数据定义为可视化的表现，可根据用户的命令返回不同图片格式的地图，命令通过URL参数传递，返回结果为图片或者GML描述的信息。 |
| WFS | WFS（Web Feature Service）根据用户的命令返回地理要素的信息，通过不同的命令，还可以对地理要素进行增删改，命令可以通过URL参数或GML传递，返回结果为GML描述的信息。 |
| GeoServer | GeoServer是一个社区支持的开源项目，是一个符合OpenGIS Web服务器规范的J2EE实现。GeoServer支持WMS和WFS标准，支持多种GIS数据格式，包括PostGIS、Shapefile、ArcSDE、Oracle、VPF、MySQL、MapInfo，支持上百种投影，能够将网络地图输出为jpeg、gif、png、SVG、KML 等多种图片格式。 |
| OPC | OLE for Process Control的缩写，定义了一个工业标准，使得COM技术适用于过程控制和制造自动化等应用领域。 |
| Modbus | Modbus协议是由Modicon在1979年发明的，是全球第一个真正用于工业现场的总线协议。通过此协议，控制器相互之间、控制器经由网络（例如以太网）和其它设备之间可以通信，此协议支持传统的RS-232、RS-422、RS-485和以太网设备。 |

## 项目总体目标

随着宝钢节能公司的EMC 项目规模不断扩大，EMC 项目的设备运行维护管理成为了宝钢节能公司的一项重要工作内容。因此，实施对各个EMC 项目的设备远程集中监控（设备运行状态和节能数据的采集），建立一个EMC 项目远程能耗数据采集与分析系统平台，实现对EMC 项目节能设备的综合管理，提高管理效率，显得越来越重要，因此提出建设本项目。

本项目的技术目标为：

* 建立EMC项目数据采集、处理、存储的应用基础架构
* 建立多维分析、数据查询工具以及分析报表生成和展示框架

本项目的业务目标为：

* 进行节能设备远程监控及节能数据采集
* 对节能数据进行分析，处理，存储
* 对采集的节能数据进行综合计算和分析，从而快速计算出EMC项目的节能量数据
* 为用户提供多种方式，如报表、图形化组件，使用户可以使用多维度的方式查询所需数据
* 结合工艺变动情况分析节能量数据波动因素，指导用户对节能设备进行优化调整，进一步提高节能效益

# 系统总体设计

## 系统软硬件要求

系统共需要两台服务器，一台为应用服务器，一台为数据库服务器。两台服务器硬件配置相同，如下表：

表 ‑1 硬件配置表

|  |  |
| --- | --- |
| 服务器型号 | 戴尔PowerEdge 12G R720 |
| CPU | Xeon E5-2690 2.9GHz x 2 |
| 内存 | 16GB ECC DDR3 |
| 硬盘 | 300GB x 6 |
| 结构 | 2U |
| 网卡 | Intel四端口千兆网卡/双端口万兆网卡 |

系统所使用的第三方软件如下表，第三方软件均为免费软件，不存在版权问题：

表 ‑2 软件配置表

|  |  |  |
| --- | --- | --- |
| 名称 | 用途 | 备注 |
| MySQL | 关系型数据库 |  |
| Tomcat | Web应用服务器 |  |
| GeoServer | GIS服务器 |  |
| QT | C++开发框架 |  |
| jQuery | Javascript开发框架 |  |
| Spring | Java IOC容器 |  |
| iBatis | Java ORM框架 |  |
| Quartz | Java 定时任务框架 |  |
| Silverlight | RIA前端开发框架 |  |
| ArcGIS Silverlight API | GIS开发控件 |  |

## 系统部署图

系统整体的部署情况如下图：

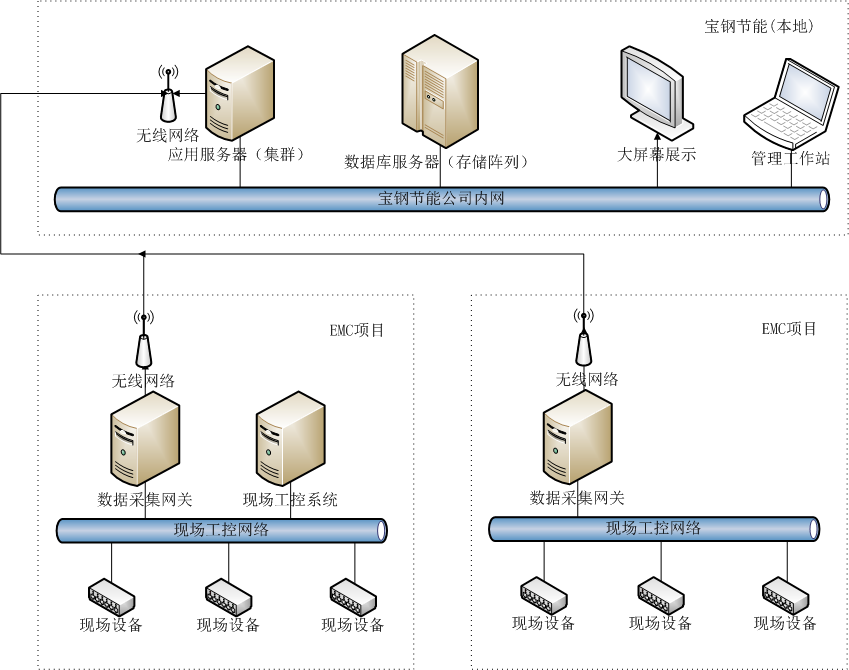


图 ‑1 系统部署图

EMC项目中具有自己的现场工控网络，并且部分项目已有现场工控系统。如果现场没有工控系统，数据采集网络直接通过工业标准协议，如modbus协议，从现场的设备，如DCS、智能仪表等读取实时数据；如果现场已具有工控系统，数据采集网关通过OPC协议从工控系统中读取实时数据，也包括工控系统中产生的报警信息、系统日志和操作日志。数据采集客户端通过无线网络（GPRS、3G）向部署于节能公司的内网的应用服务器中的通讯服务发送数据，应用服务器也可以通过无线网络向具体项目中的网关下发控制参数和指令，最终下发到具体设备。根据现场实际情况，数据采集网关可以与工控系统部署于同一台windows机器上，也可以单独部署于一台独立的windows/linux机器上。

系统中的应用服务器和数据库服务器部署于节能公司内网中，与现场EMC项目通过无线网络交互。应用服务器中包含通讯服务、过程数据库、趋势服务、Web应用服务、GIS服务等服务，通讯服务负责接收/下发EMC项目的数据，由其他服务进行数据的操作和处理。数据库服务器存储项目运行周期内过程数据库转储的历史数据和统计数据，也包括用户录入的数据，供用户访问具体应用页面时查询使用。用户可以通过EMC项目管理工作站PC或笔记本电脑在节能公司内网中访问系统的平台网站。

随着将来接入的EMC项目越来越多，应用服务器可以扩充为应用服务器集群，数据库服务器的存储可扩充为磁盘阵列，以获得更大的存储容量和访问速度。

## 系统组件图

系统组件图如下：

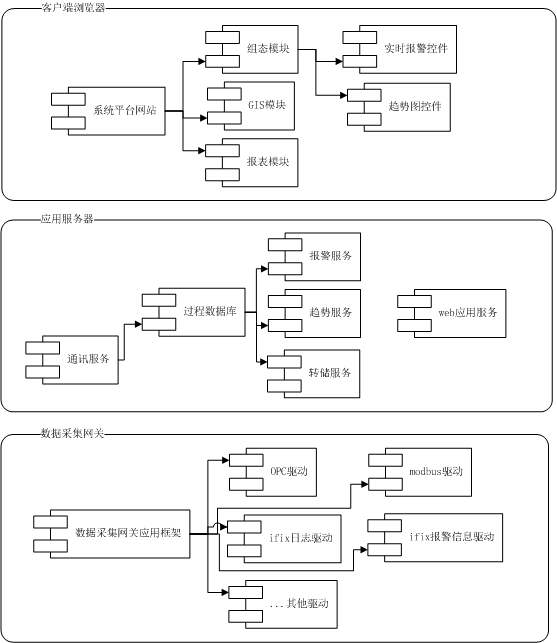


图 ‑2 系统组件图

数据采集网关中的组件包括数据采集网关应用框架，以及采集各种不同数据需要的驱动。需要开发的驱动包括OPC驱动、modbus驱动、iFix日志采集驱动、iFix报警信息采集驱动、WinCC日志采集驱动、WinCC报警信息采集驱动，以及从其他设备采集数据需要的驱动。如果将来有更多类型的设备需要接入，数据采集网关应用框架可以通过配置接入符合框架标准的驱动dll。数据采集网关应用框架和采集驱动使用C++开发，使用QT作为C++开发框架库。

应用服务器中的组件包括通讯服务、过程数据库、报警服务、趋势服务、转储服务以及Web应用服务。通讯服务负责接收数据采集网关发送的数据，并发送到过程数据库中，过程数据库对实时数据进行处理和存储，并提供对外访问的接口。根据趋势配置和转储配置，趋势服务和转储服务从过程数据库中获取数据，并分别保存入趋势数据文件和关系型数据库，趋势服务提供对外访问的趋势服务接口。Web应用服务比较独立，提供了访问网站页面的服务，以及数据查询、管理、录入，报表生成等服务。为保证实时数据处理的效率，通讯服务、过程数据库、报警服务、趋势服务、转储服务使用C++开发；为充分利用已有成熟的Web应用开发框架，Web应用服务使用Java开发，使用Spring + iBatis作为开发框架。

客户端浏览器中最主要的组件为系统平台网站，系统平台网站是用户访问的入口和门户。组态模块、GIS模块、报表模块嵌入在系统平台网站中运行，实时报警控件和趋势图控件又嵌入在组态模块中运行。这些模块从过程数据库和报警服务获取实时数据，从趋势服务中获取趋势数据，从Web应用服务中获取历史数据。组态模块、GIS模块、报警控件和趋势图控件这些具有美观图形展示效果的组件使用Silverlight开发；系统平台网站和报表模块这些主要以文本、表格方式展示项目数据的组件使用JSP、Html和Javascript开发。

## 系统模块结构图

根据系统包含的功能，可以将系统分为两个部分：数据采集、传输及存储子系统和数据展示子系统。

数据采集、传输及存储部分子系统的结构见下图：

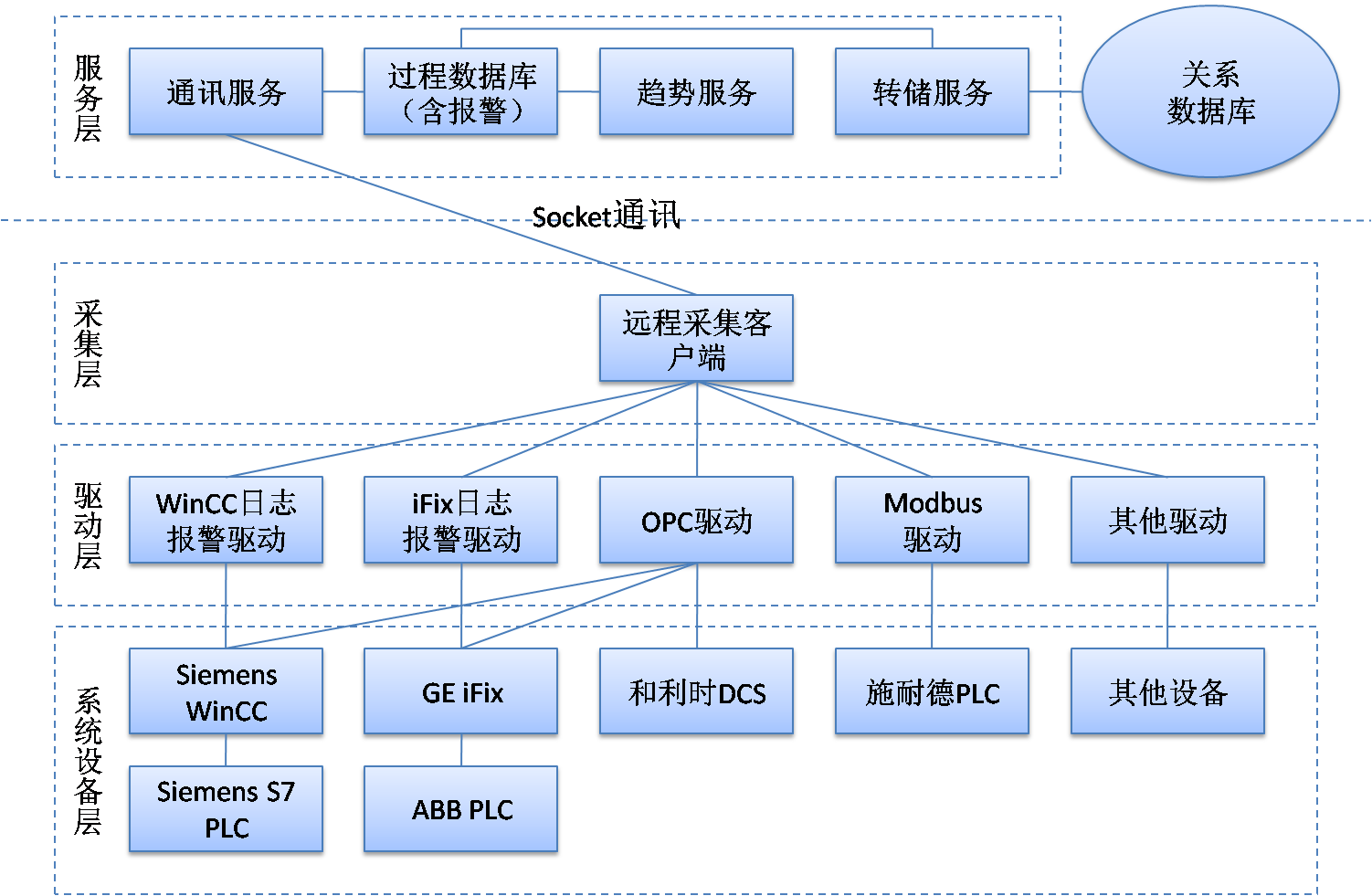


图 ‑3 数据采集、传输、存储子系统模块结构图

数据采集、传输及存储部分子系统可分为四个层次：系统设备层、驱动层、采集层和服务层。

系统设备层包含系统所需要支持的设备及工控软件。

驱动层负责从系统设备层采集实时数据，以及报警信息和日志，针对不同的工控软件和设备，需要开发不同的驱动来进行采集。

采集层的远程采集客户端，可以加载不同的采集驱动，来进行数据采集，并将采集的结果通过socket通讯发送到服务层的通讯服务中。

服务层的入口是通讯服务，通讯服务将接收到的数据发送到过程数据库中，过程数据库进行处理和存储。根据具体配置，趋势服务和转储服务从过程数据库中获取数据并保存，趋势服务将数据保存在趋势文件中，转储服务将数据存储在关系数据库中。

数据展示子系统的模块结构见下图：

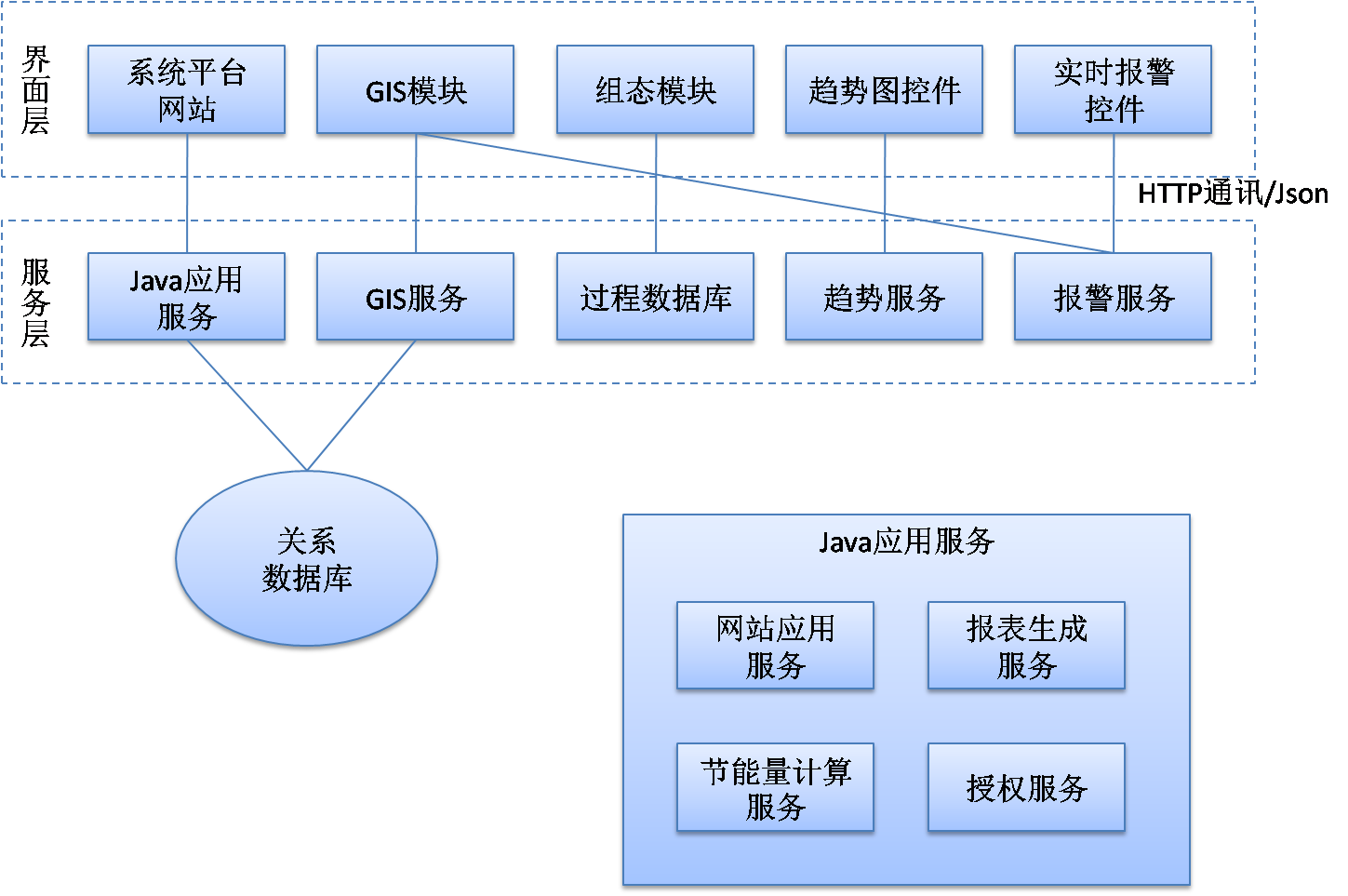


图 ‑4 数据展示子系统结构图

数据展示子系统可以分为服务层和界面层两个层次。每个层次又可以分为几个大的模块，用于完成不同的业务需求。界面层和服务层的通讯都通过http协议进行，数据交换的格式为Json。

用户通过Java应用服务访问系统平台网站中的页面，对数据库中的数据进行查询、分析和录入。Java应用服务中又包含有网站应用服务、报表生成服务、节能量计算服务和授权服务。

用户访问GIS模块时，通过GIS服务获取地图数据信息，通过报警服务获取实时报警信息。

用户访问组态模块时，通过过程数据库获取实时数据信息，下发控制参数和指令。

用户访问趋势图控件时，通过趋势服务获取实时趋势或历史趋势数据。

用户访问实时报警控件时，通过报警服务获取实时报警信息，和确认报警。

# 远程采集客户端设计

## 远程采集客户端概述

远程采集客户端负责远程采集EMC 项目节能设备运行状态参数与节能量等相关数据，如果EMC项目已有工控软件，采集内容也包括工控软件产生的日志和报警信息，如iFix、WinCC等。

## 远程采集客户端结构

远程采集客户端结构如图：

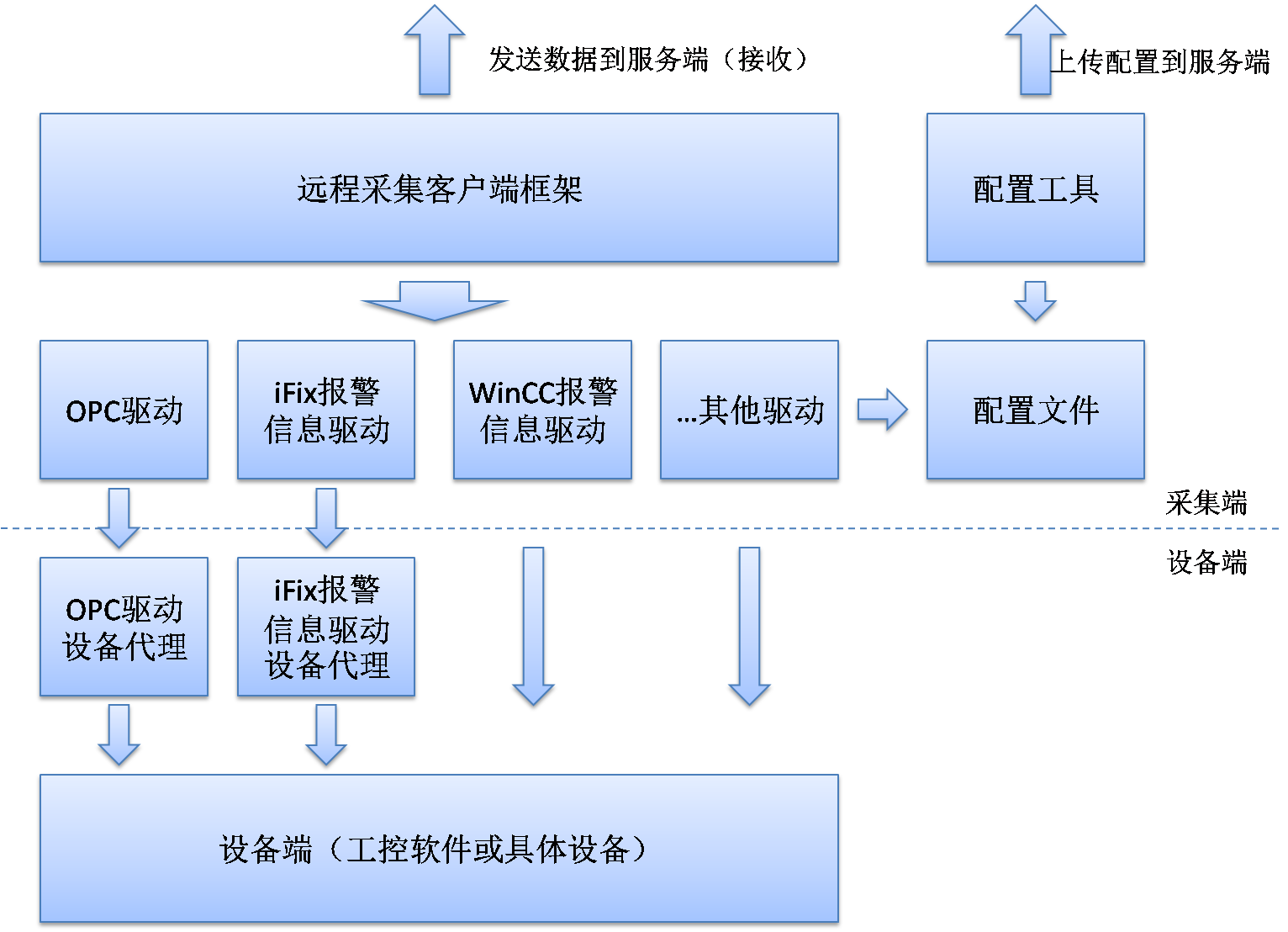


图 3‑1远程采集客户端结构

远程采集客户端总体可以分为两部分，采集端部分和设备端部分，根据现场具体情况，采集端和设备端可以部署于同一台PC中，也可以分别部署。

采集端的核心是远程采集客户端框架，框架程序通过加载不同的驱动从设备端采集数据，再发送到系统的服务端，也可以接收从服务端下发的控制指令和修改的设备参数，采集端和服务端使用socket进行通讯。驱动的种类包括采集实时数据的OPC驱动和modbus驱动，也包括采集工控软件报警信息和日志信息的驱动，驱动可以按照规范进行扩充。远程采集客户端框架，以及驱动的配置文件，通过本地的配置工具在现场配置，配置完成之后，将配置信息同步到服务端，配置工具和服务端也通过socket通讯。用于通讯的网络组件使用QT编写。

设备端包括工控软件和具体的设备。驱动从设备采集数据时，某些数据不需要在设备端部署采集的设备代理，如WinCC的报警信息采集，某些数据需要在设备端部署采集的设备代理，如OPC采集，iFix的报警信息采集。根据具体情况，在设备端部署设备代理，如有需要，驱动从设备代理中采集数据，否则，直接从设备中采集数据。

## 远程采集客户端冗余

为了提高系统的稳定性和健壮性，确保单个远程采集客户端或设备出现故障时，仍然能够向服务端发送数据，设计冗余方案如下，冗余方案包括远程采集客户端冗余和远程采集客户端冗余+设备冗余：

远程采集客户端冗余方案如图：

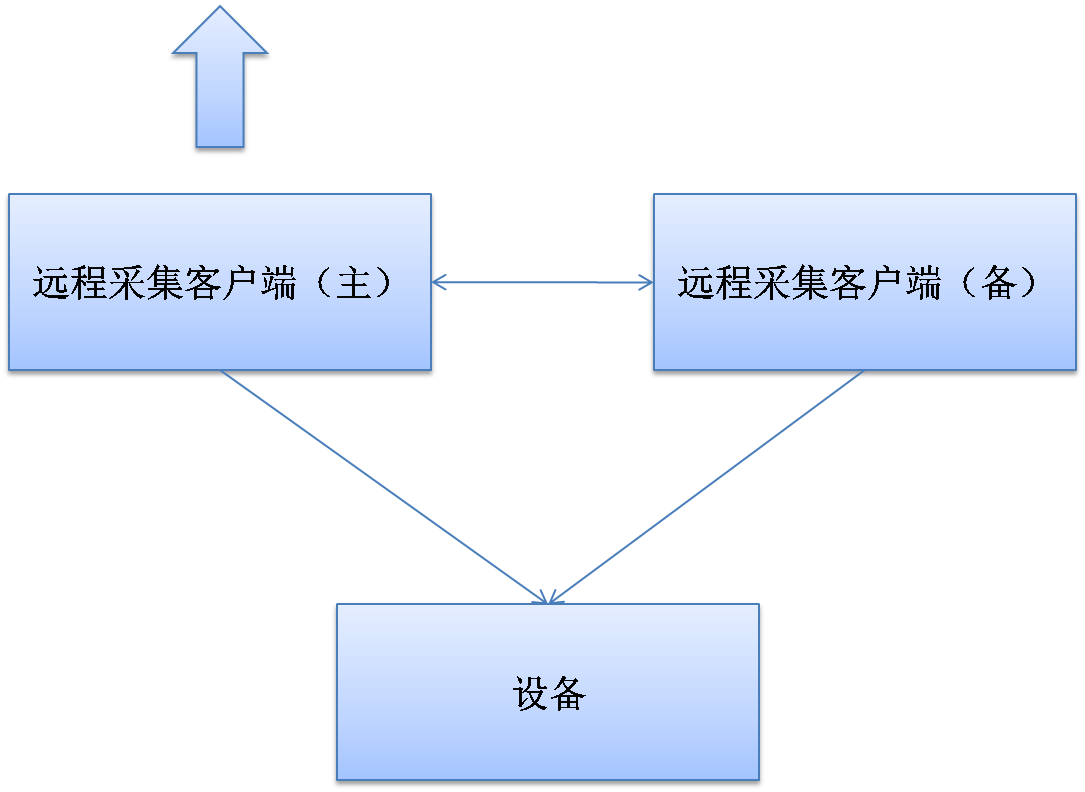


图 3‑2远程采集客户端冗余

部署两个远程采集客户端，一个为主机，一个为备机，主备之间具有心跳通讯。备机检查到主机通讯不正常时，备机自动升级为主机，向服务端发送数据。

远程采集客户端冗余+设备冗余方案如图：

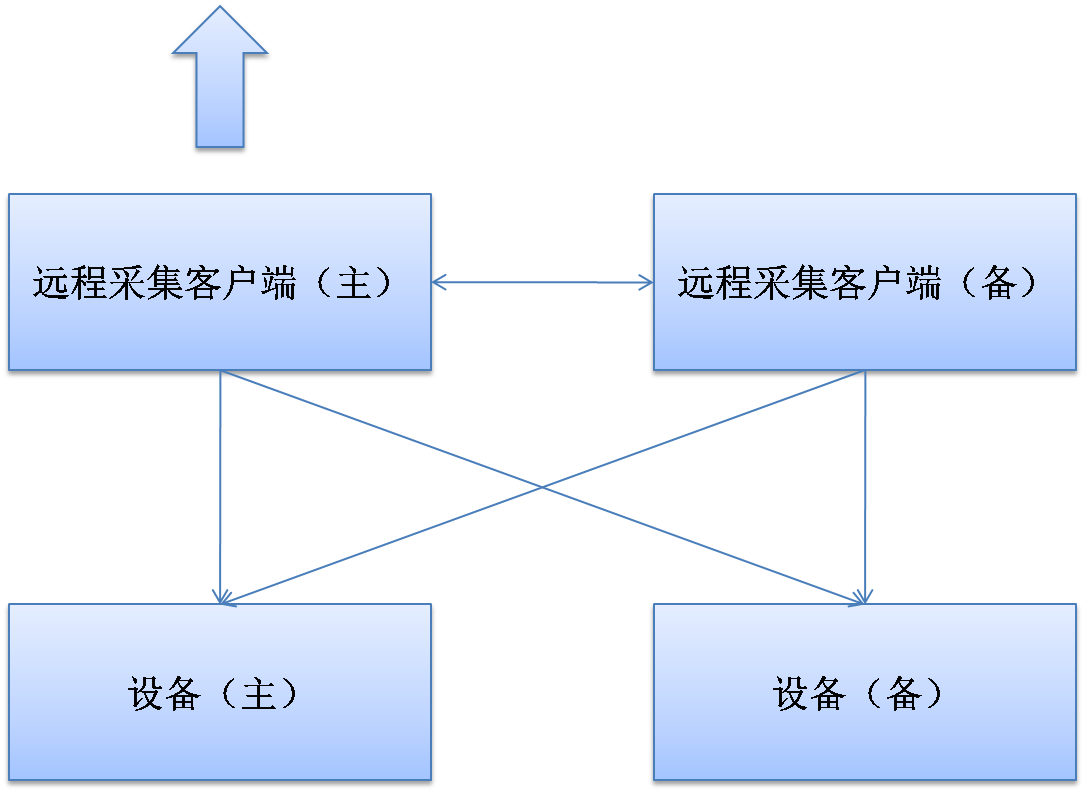


图 3‑3远程采集客户端冗余+设备冗余

这种方案中，除远程采集客户端外，设备也具有主机和备机，远程采集客户端冗余方式与第一种方案相同，备机检查到主机通讯不正常会自动升级为主机。正常工作时，远程采集客户端从设备主机中采集数据，如与设备主机通讯不正常时，远程采集客户端自动切换到从设备备机中采集数据。

# 过程数据库设计

## 过程数据库概述

过程数据库是系统中对应用过程的描述，应用过程由一系列执行不同功能的过程变量组成，过程变量是完成一定功能的过程控制指令的集合。过程变量是一个过程数据库的主要组成部分。在本系统中，过程变量主要完成如下功能：

1. 从其它过程变量或者直接从一个外部设备接收数值；
2. 根据配置执行不同的操作，包括汇总数据，根据其它变量的输入执行计算、监控数据的异常状况等；
3. 向外部设备输出数值。

## 过程数据库组件

过程数据库中包含如下组件：

表‑1过程数据库组件列表

|  |  |
| --- | --- |
| 组件名称 | 组件描述 |
| PBScanner.exe | 过程数据扫描服务。本程序是过程数据库的核心组件，其主要功能是管理采集的实时设备数据，并进行必要的数据处理和报警处理。同时该子模块接收来自于本地实时数据访问接口和远程实时数据访问接口的控制指令。 |
| DataReceiver.exe | 过程数据接收服务。负责接收从远程采集客户端发送来的数据，并发送到过程数据扫描服务。 |
| LRDA.dll | 本地实时数据访问接口。提供了访问同一物理计算机上的实时过程数据的API接口。 |
| RDA.dll | 远程实时数据访问接口。提供了访问位于网络中指定地址的过程数据库上的实时过程数据的API接口。 |
| RDAHttpStub.exe | 过程数据库的http数据转发服务，可以使外部程序通过http协议获取过程数据库中的实时数据。 |

## 过程数据库结构

过程数据库的结构如下图：

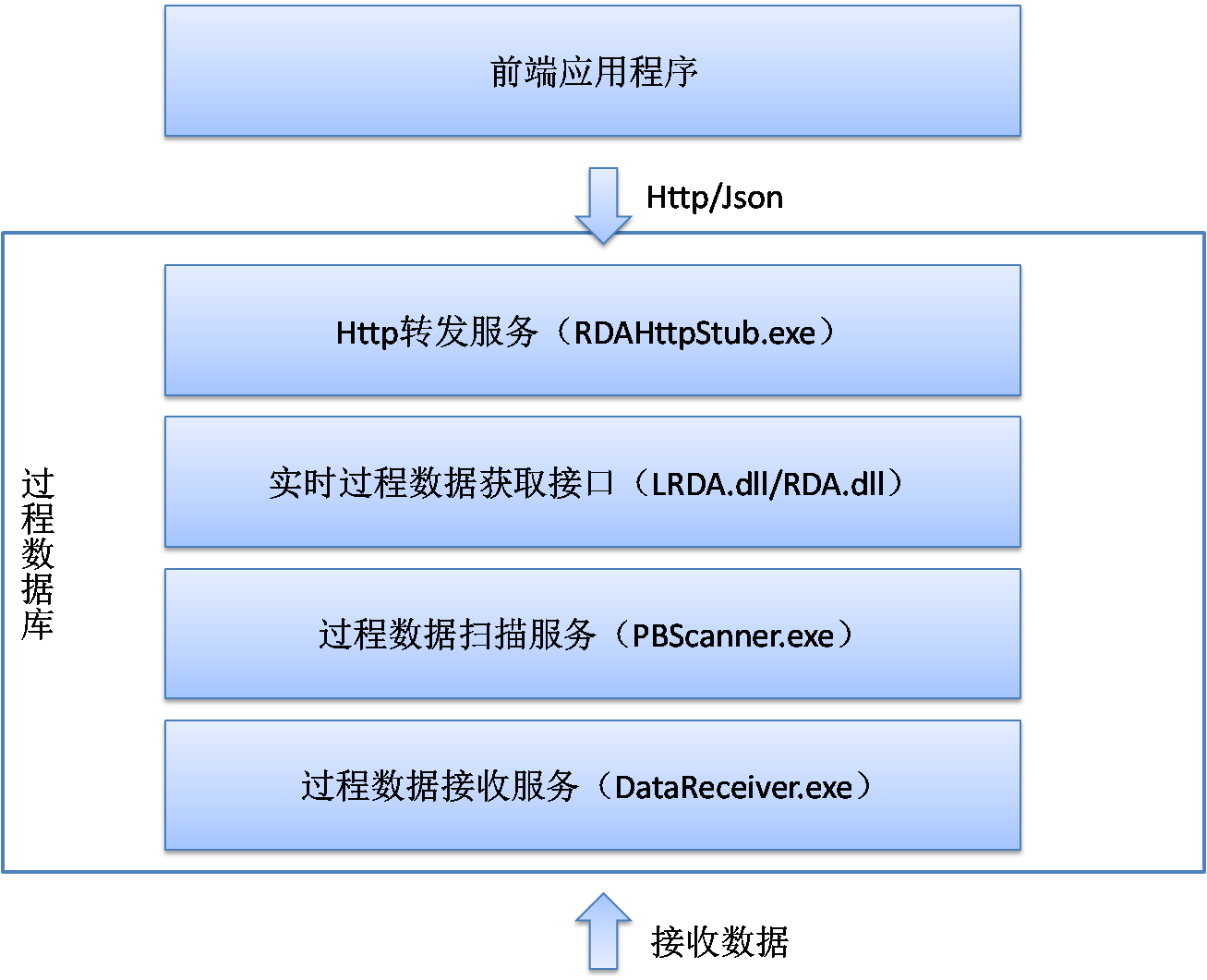


图 4‑1过程数据库结构图

前端应用程序，如组态运行模块等，通过Http请求从过程数据库中获取实时数据。首先将请求发送到Http转发服务中，Http转发服务通过实时过程数据获取接口从过程数据扫描服务中获取实时数据。

过程数据接收服务接收从各个远程采集客户端发送上来的数据，并发送到过程数据扫描服务中，过程数据扫描服务进行必要的数据处理和报警处理。

## 过程数据库提供的接口

本系统中，过程数据库为前端应用程序提供Http访问接口，数据格式使用Json，符合JsonRPC标准，具体接口如下：

1. 获取过程数据服务列表：

输入示例：

{"jsonrpc":"2.0", "method":"getservers", "id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"ServerName":"TestDataServer1"},{"ServerName":"TestDataServer2"}], "id":1}

1. 获取Tag点个数：

输入示例：

{"jsonrpc":"2.0",

"method":"gettagcount",

"params":{"servername":"server1", "tagname":"tag1"},

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":10, "id":1}

1. 获取Tag点列表：

输入示例：

{"jsonrpc":"2.0", "method":"querytags", "params":{"servername":"server1", "tagname":"tag1", "startindex":0, count:100}, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"TagName":"Tag1","TagDesc":"TagDesc1"},{"TagName":"Tag2","TagDesc":"TagDesc2"}], "id":1}

1. 获取Tag点的域列表：

输入示例：

{"jsonrpc":"2.0",

"method":"querytagproperties",

"params":{"servername":"server1", "tagname":"tag1"},

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"PropertyName":"Property1","PropertyDesc":"PropertyDesc1"},{"PropertyName":"Property2","PropertyDesc":"PropertyDesc2"}], "id":1}

1. 注册实时过程变量列表：

输入示例：

{"jsonrpc":"2.0",

"method":"registertags",

"params":["server1.tag1.pv","server1.tag2.pv"],

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":1, "id":1}

1. 注销实时过程变量列表：

输入示例：

{"jsonrpc":"2.0", "method":"unregistertags", "params":1, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":0, "id":1}

1. 获取实时数据

输入示例：

{"jsonrpc":"2.0", "method":"getntfvalues", "params":1, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"NTFName":"server1.tag1.pv","Value":"100","Time":"2013/7/23 10:39:55","Quality":0},{"NTFName":"server1.tag2.pv","Value":"200","Time":"2013/7/23 10:39:56","Quality":0}], "id":1}

1. 直接获取实时数据

输入示例：

{"jsonrpc":"2.0",

"method":"directlygetntfvalues",

"params":["server1.tag1.pv","server1.tag2.pv"],

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"NTFName":"server1.tag1.pv","Value":"100","Time":"2013/7/23 10:39:55","Quality":0},{"NTFName":"server1.tag2.pv","Value":"200","Time":"2013/7/23 10:39:56","Quality":0}], "id":1}

# 报警服务设计

## 报警服务概述

报警服务对系统中的实时数据和设备参数进行监视，当采集的数据达到（超过或低于）系统中设定的阀值时，产生报警信息；或一些设定为时间限制的设备参数达到设定的时间时，产生报警信息。产生报警信息后，根据配置，报警服务会自动将报警信息发送到用户的手机或电子邮箱中。

## 报警服务结构

报警服务的结构如下图：

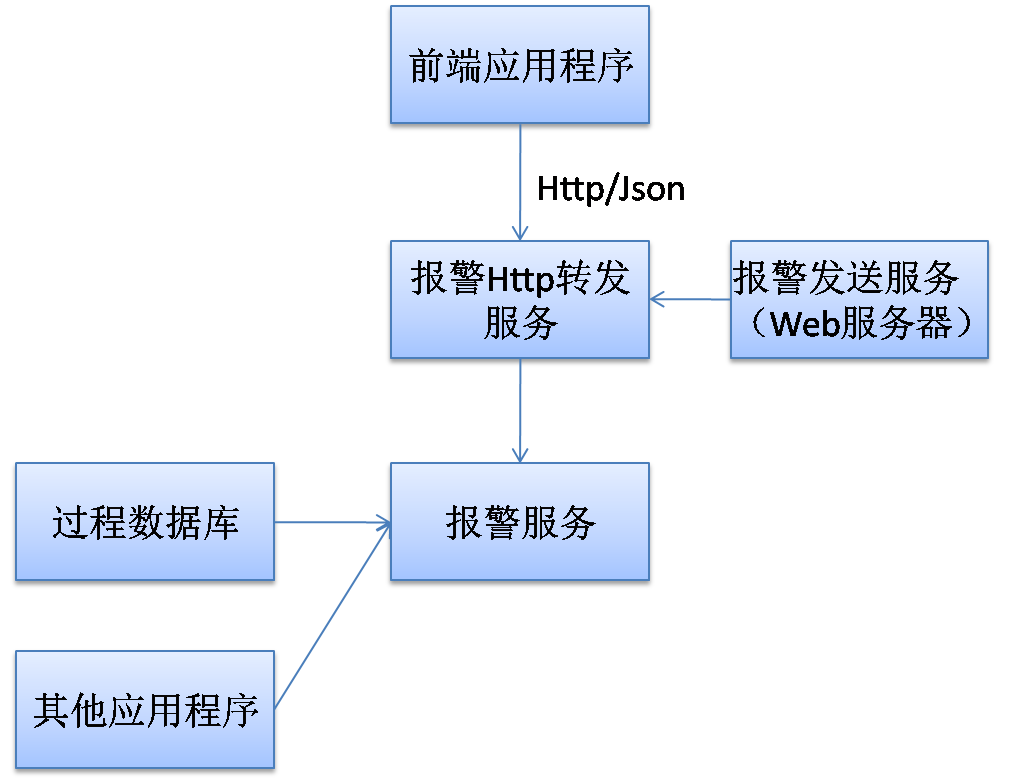


图 5‑1报警服务结构图

报警服务中的报警数据主要来源为过程数据库，过程数据库获取到实时数据后进行报警处理，如果具有报警信息的话，通过报警服务的接口将报警信息传递给报警服务，不仅过程数据库可以产生报警信息，其他应用程序也可以通过接口向报警服务传递报警信息。

前端应用程序通过http请求向报警http转发服务请求实时报警信息，以及进行确认报警等操作。报警http转发服务再调用报警服务的接口获取实时报警信息或进行确认操作。

报警发送服务使用java开发，运行于Web服务器中，报警发送服务向报警转发服务进行了注册，可以获得最新的实时报警信息，取得实时报警信息后，报警发送服务根据配置将报警信息发送到用户的手机或电子邮箱中。

## 报警服务提供的接口

本系统中，报警服务为前端应用程序提供Http访问接口，数据格式使用Json，符合JsonRPC标准，具体接口如下：

1. 注册报警

输入示例：

{"jsonrpc":"2.0", "method":"registeralarm", "id":1}

输出示例：

{"jsonrpc":"2.0", "result":1, "id":1}

1. 注销报警

输入示例：

{"jsonrpc":"2.0", "method":"unregisteralarm", "params":1, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":0, "id":1}

1. 获取最新报警信息

输入示例：

{"jsonrpc":"2.0", "method":"getalarms", "params":1, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":[{"AlarmID":1,"ServerName":"testserver","TagName":"tag1","TagValue":"10000","Desc":"desc","AlarmType":"HiHi","Confirmer":"","ConfirmStatus":false,"ResumeStatus":false,"OccurTime":"2013/7/23 10:55:55","ConfirmTime":"","ResumeTime":null,"AlarmTimes":1,"Priority":1,"Group":null,"Template":null,"SubSystem":null,"Comment":"comment","Commentor":"someone","CommentTime":"2013/7/23 10:55:55"},{"AlarmID":2,"ServerName":"testserver","TagName":"tag2","TagValue":"10000","Desc":"desc","AlarmType":"HiHi","Confirmer":"","ConfirmStatus":false,"ResumeStatus":false,"OccurTime":"2013/7/23 10:55:55","ConfirmTime":"","ResumeTime":null,"AlarmTimes":1,"Priority":1,"Group":null,"Template":null,"SubSystem":null,"Comment":"Comment","Commentor":"someone","CommentTime":"2013/7/23 10:55:55"}], "id":1}

1. 确认报警

输入示例：

{"jsonrpc":"2.0",

"method":"confirmalarm", "params":[{"AlarmID":1,"ServerName":"testserver1","OccurTime":"2013/7/23 11:00:15"},{"AlarmID":1,"ServerName":"testserver1","OccurTime":"2013/7/23 11:00:15"}] ,

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":[0,-1], "id":1}

1. 获取报警区域列表

输入示例：

{"jsonrpc":"2.0", "method":"getalarmareas", "params":"testserver", "id":1}

输出示例：

{"jsonrpc":"2.0", "result":["Area1","Area2","Area3"], "id":1}

# 趋势服务设计

## 趋势服务概述

趋势服务负责从过程数据库中采集数据，经处理之后保存到磁盘中，同时对客户端，如趋势图控件，提供查询接口。当归档的趋势数据文件达到预设的时间长度时，需要将数据文件从磁盘中删除。

## 趋势服务结构

趋势服务的结构如下图：

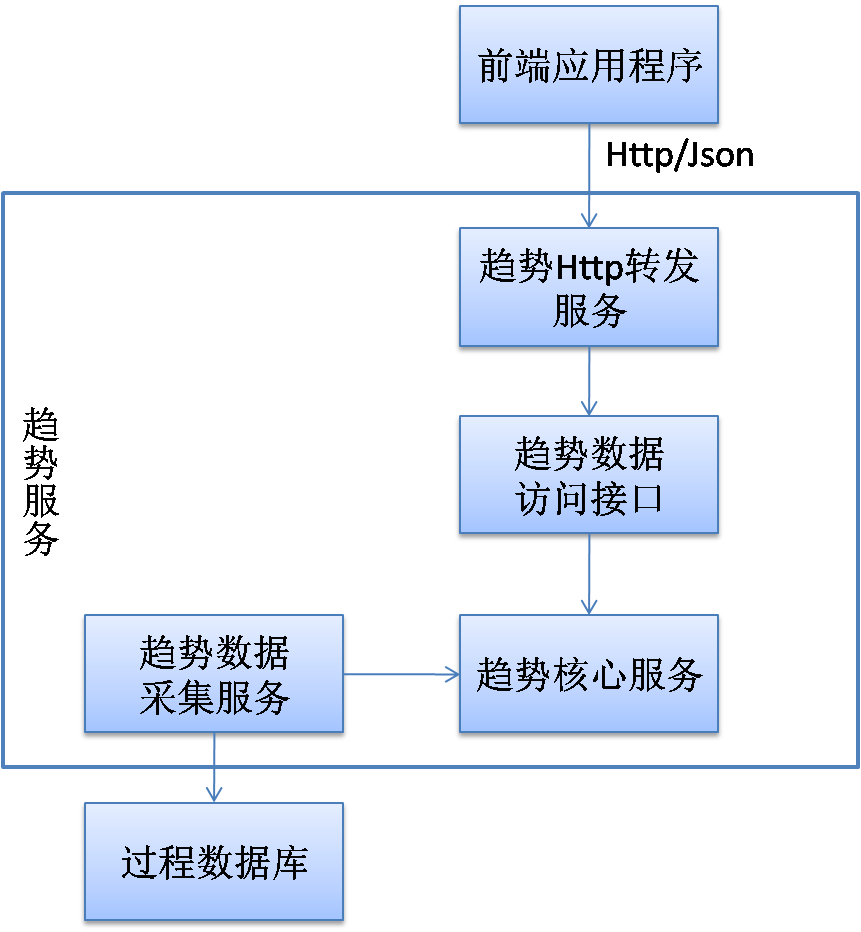


图 6‑1趋势服务结构图

趋势服务的数据来源于过程数据库，趋势数据采集服务定时从过程数据库中采集实时数据，并发送到趋势核心服务中，趋势核心服务将数据进行压缩存储，并删除过期数据。前端应用程序，如趋势图控件，查询趋势数据时，向趋势Http转发服务发送Http请求，趋势Http转发服务通过趋势数据的访问接口到趋势核心服务中查询数据，并返回给前端应用程序。

## 趋势服务提供的接口

本系统中，趋势服务为前端应用程序提供Http访问接口，数据格式使用Json，符合JsonRPC标准，具体接口如下：

1. 查询单个Tag点一段时间内的趋势数据

输入示例：

{"jsonrpc":"2.0",

"method":"getalarmareas",

"params":{"ntfname":"server1.TestTag.pv", "starttime":"2013-8-5 12:00:00", "endtime":"2013-8-5 13:00:00:000", interval:1000},

"id":1}

输出示例：

{"jsonrpc":"2.0", "result":{"ntfname":"server1.tag1.pv","value":["10","20",…,"30"]}, "id":1}

1. 查询一组Tag点一段时间内的趋势数据

输入示例：

{"jsonrpc":"2.0", "method":"getgrouptagrecords", "params":{"ntfname":["server1.tag1.pv","server1.tag2.pv"], "starttime":"2013-8-5 12:00:00", "endtime":"2013-8-5 13:00:00:000", interval:1000}, "id":1}

输出示例：

{"jsonrpc":"2.0", "result":{["ntfname":"server1.tag1.pv","Value":["10","20",…,"30"]], ["ntfname":"server1.tag2.pv","Value":["10","20",…,"30"]]}, "id":1}

# 转储服务设计

## 转储服务概述

转储服务实现历史数据及报警的转储，根据配置的DSN信息将需要转储的数据存储到相应的ODBC数据源。

历史数据转储的主要功能是将实时过程数据作为历史数据进行归档，它从过程数据库中获取需要长期保存的数据，并将其通过ODBC接口存储到关系型数据库中。报警数据转储的主要功能是将报警作为历史数据进行归档，它从报警服务中获取报警数据，并将其通过ODBC接口存储到关系型数据库中。

## 转储服务结构

转储服务的结构如图：

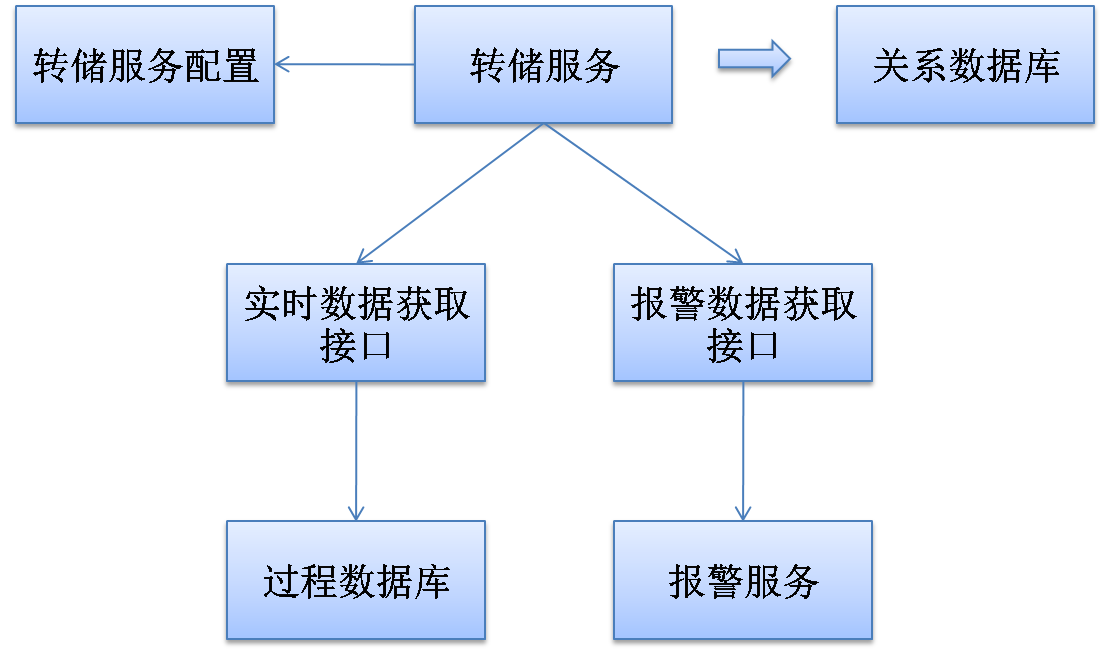


图 7‑1转储服务结构图

转储的历史数据库来源于过程数据库和报警服务。根据转储的配置信息，转储服务定时通过实时数据获取接口从过程数据库中获取实时数据，并转储到关系型数据库中，获取报警数据的方式是通过报警数据获取接口向报警服务注册，如果有新报警产生，报警服务会将新报警数据推送到转储服务中，接收到报警数据后，同样转储服务将数据转储到关系型数据库中。存储到数据库使用ODBC方式。

# 系统平台网站设计

## 系统平台网站概述

为了减少部署成本和复杂度，使用户可以无需安装特定软件，使用浏览器直接访问。本系统中最终用户访问的界面，及其相关的业务和服务组件采用B/S架构开发，以网站的形式提供给用户访问。

系统平台网站采用Java技术开发，原因在于：

1. Java技术非常成熟，在企业级应用系统中具有丰富的应用案例。
2. Java技术具有跨平台的优势，可以部署于各种常见的操作系统。

对应于需求模块，系统平台网站主要包含的需求为：

1. 系统登录
2. GIS系统、组态系统、报表系统的集成
3. 信息编码管理
4. 节能量管理
5. 费用管理
6. 报表配置
7. 身份认证与权限管理

## 系统平台网站结构

系统平台网站的层次结构如下图：

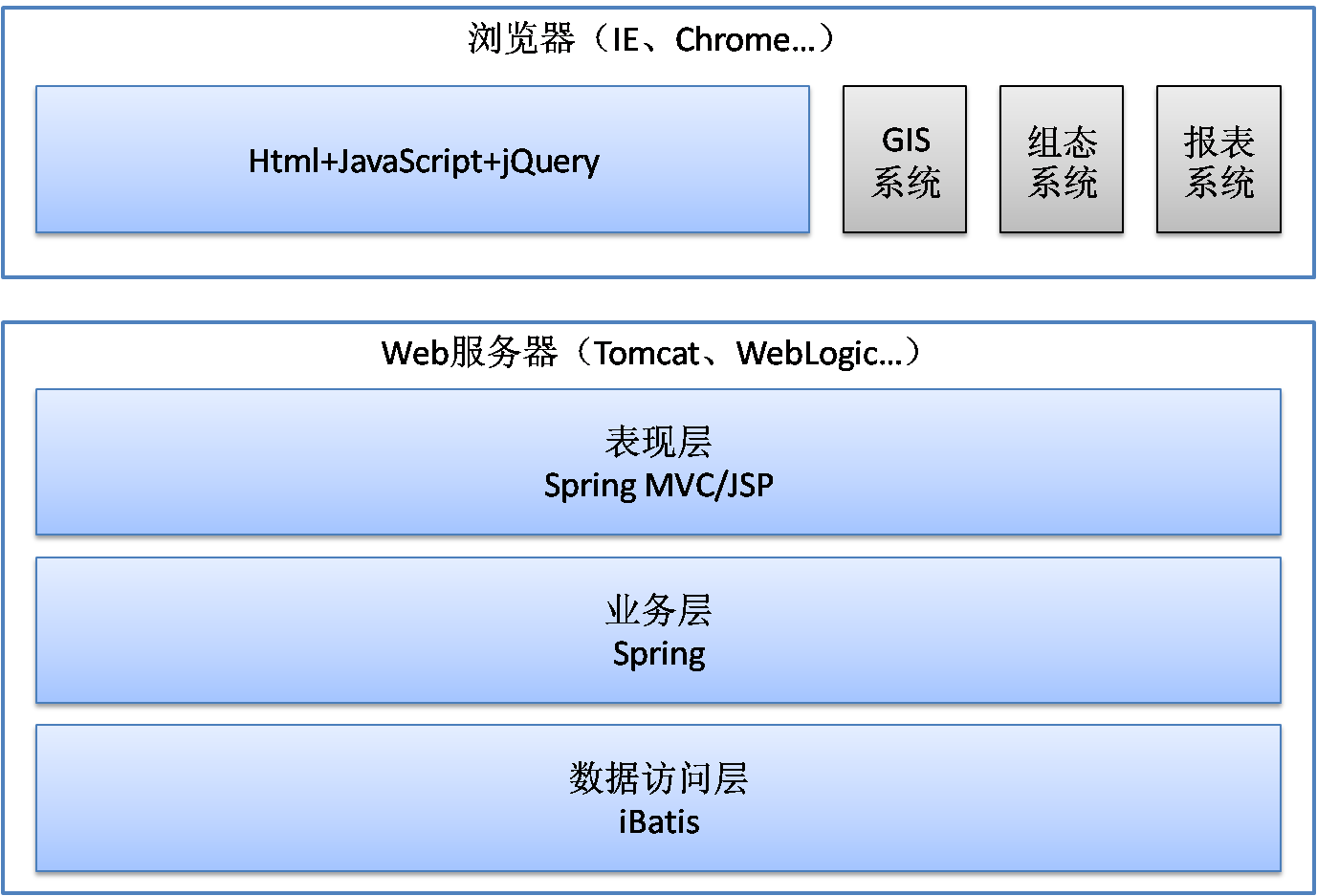


图 8‑1系统平台网站层次结构图

系统平台网站可以分为前端和后端两大部分。

网站前端可以使用常见的浏览器，如IE、Chrome等访问，采用Html+JavaScript开发，使用jQuery库辅助JavaScript开发。

Html+JavaScript是各种常见浏览器都支持的标准，使用Html+JavaScript开发的前端页面可以运行于各种常见浏览器下。

jQuery是一个通用的轻量级JavaScript框架，兼容CSS3和各种常见的浏览器，在网站前端开发中得到了广泛的应用，并且jQuery是免费开源的。通过使用jQuery，可以使前端的代码开发更加简易和稳定，在开发中，主要使用jQuery的选择器，事件处理和Ajax调用等功能。

GIS系统、组态系统和报表系统作为独立的模块，也可以集成入网站前端页面，详细设计见后续章节。

网站后端运行在Web服务器中，如Tomcat、WebLogic等。后端分为三层：表现层、业务层和数据访问层。

后端表现层使用Spring MVC框架进行开发，Spring MVC是一个构建Web应用程序的全功能MVC模块，可支持多种视图技术，例如JSP、Velocity、Tiles、iText和POI。Spring MVC分离了控制器、模型对象、分派器，这种分离让它们更容易进行设计和开发。系统平台网站的页面使用JSP开发，暴露出的服务以Servlet方式提供。

后端业务层使用Spring框架进行开发，Spring是Java企业应用开发中最流行的控制反转和面向切面的轻量级容器框架。通过Spring的使用，可以降低对象之间的耦合性，将系统服务、业务逻辑和业务实体分隔开来，使代码更易于管理，易于测试和调试，也更易于复用。Spring框架的架构如下图：



图 8‑2 Spring框架架构图

后端数据访问层使用iBatis开发，iBatis是一个基于Java的半自动ORM持久层框架。作为一个ORM框架，iBatis通过配置文件自动建立起了业务对象实体与关系型数据表的关系，减少了重复性的开发工作量，并且在业务模型需要修改时，使维护工作更加清晰易行。作为半自动ORM框架，除上述优势外，框架将需要执行的SQL语句开放给开发人员，使开发工作更为灵活，可以支持更灵活的复杂业务对象，对复杂的SQL语句进行充分优化。iBatis框架的架构如下图：

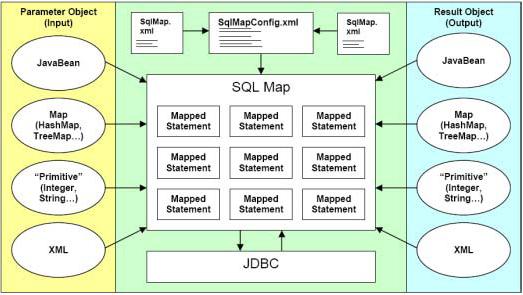


图 8‑3 iBatis框架架构图

系统平台网站的功能结构如下图：

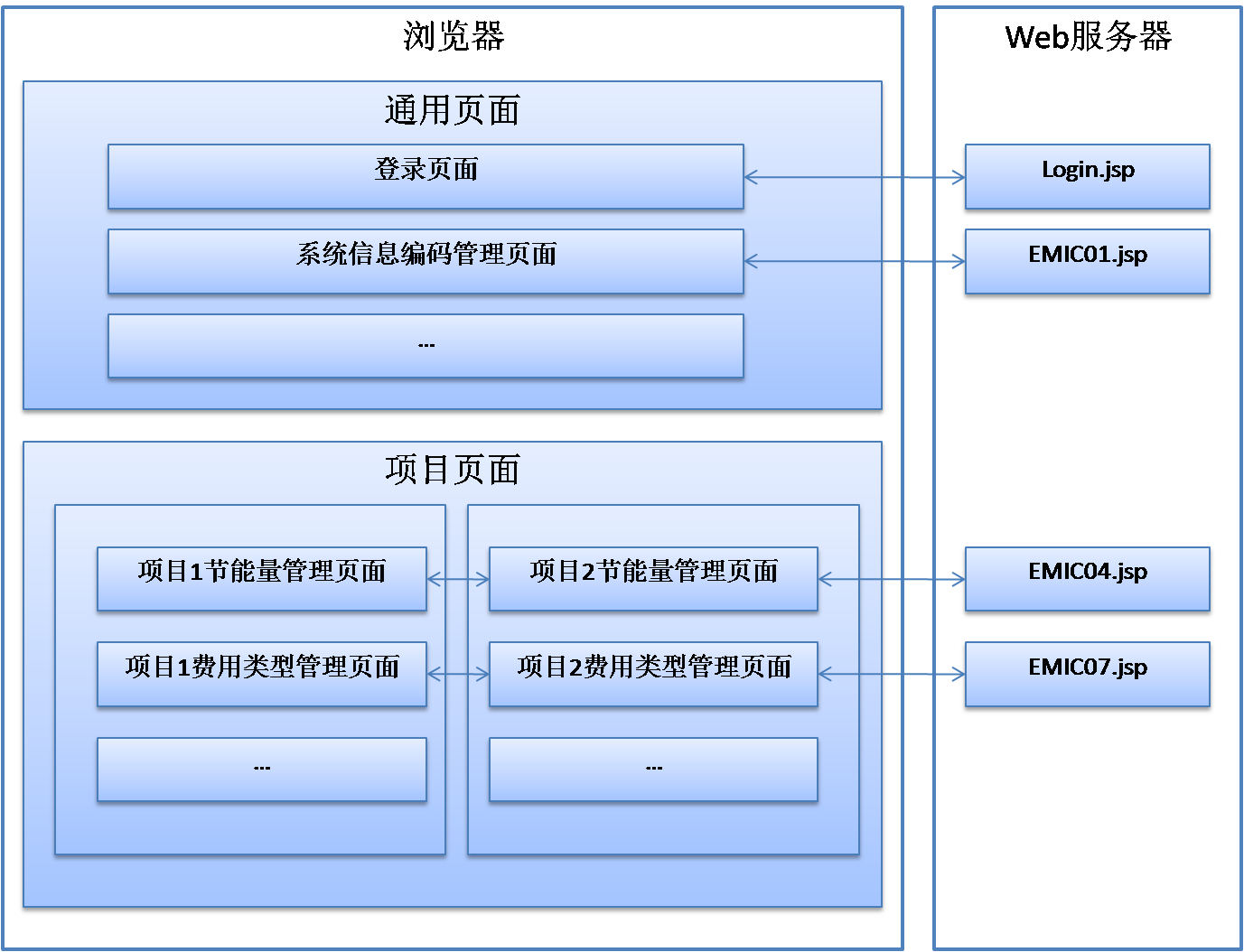


图 6‑4系统平台网站层次结构图

系统平台网站中的页面可以分为两类：一类为通用页面，与具体EMC项目无关，如登录页面、系统信息编码管理页面等，每个通用页面与web服务器中的一个jsp页面对应；另一类为项目页面，与具体EMC项目相关，如节能量管理页面、费用类型管理页面等，每个EMC项目都有自己的项目页面，并且页面展示的内容并不相同，在系统平台网站中，每一类的项目页面都与web服务器中的一个jsp页面对应，由用户访问的URL参数来确定页面所属的项目，并从数据库中查询该项目的数据。如节能量管理页面的URL示例可以为<http://localhost:8080/emc/EMIC04.jsp?projectid=1>。

用户访问系统平台网站时，可以通过网站页眉上的项目选择下拉框来切换当前选择的EMC项目。选择好当前的项目后，系统平台网站主页面左侧的页面菜单树显示当前项目具有的所有可访问页面，可通过点击菜单树来切换不同的应用页面。

## 系统平台网站的主要页面

1. 登录页面（Login.jsp）：用户访问的首页，负责用户的登录。
2. 系统平台网站主页面（Index.jsp）：平台网站的母版页，提供页眉页脚的样式，提供访问其他应用页面的菜单树，加载其他的应用页面。在母版页中，用户可以进行当前访问的EMC项目的切换。
3. 系统信息编码管理页面（EMIC01.jsp）：管理系统信息编码，包括地域信息编码、公司信息编码和项目信息编码。
4. 项目信息编码管理页面（EMIC02.jsp）：管理项目信息编码，包括工艺段信息编码、设备信息编码和设备参数信息编码。
5. EMC项目周期管理页面（EMIC03.jsp）：管理EMC项目的统计周期，例如配置在项目的生命周期内每年划分为几个统计周期（多为季度），每个统计周期的起始日期和结束日期。
6. EMC项目节能量管理页面（EMIC04.jsp）：管理EMC项目关注的节能量，并配置节能量的计算公式。
7. EMC项目节能量抄表值管理页面（EMIC05.jsp）：录入、查看EMC项目节能量的抄表值。
8. EMC项目节能量基准值管理页面（EMIC06.jsp）：录入、查看EMC项目节能量的基准值。
9. EMC项目费用类型管理页面（EMIC07.jsp）：管理EMC项目具有的费用类型。
10. EMC项目预算费用录入页面（EMIC08.jsp）：录入、查看EMC项目的预算费用。
11. EMC项目实际费用录入页面（EMIC09.jsp）：录入、查看EMC项目的实际费用。
12. 报表模板管理页面（EMIC10.jsp）：管理所有的报表模板，报表模板不依赖于具体的某个EMC项目。
13. 固定周期报表管理页面（EMIC11.jsp）：管理EMC项目中的固定周期报表。
14. 角色管理页面（EMIC12.jsp）：管理系统中具有的角色。
15. 用户管理页面（EMIC13.jsp）：管理系统中具有的用户。
16. 资源管理页面（EMIC14.jsp）：管理系统中具有的页面资源。
17. 角色成员管理页面（EMIC15.jsp）：配置角色中包含哪些用户。
18. 角色权限管理页面（EMIC16.jsp）：配置角色具有哪些资源的访问权限。
19. 角色报警管理页面（EMIC17.jsp）：配置角色关心哪些报警信息，以何种方式通知用户。
20. 操作日志查询页面（EMIC18.jsp）：按照指定的查询条件查询符合的操作日志。
21. 系统日志查询页面（EMIC19.jsp）：按照指定的查询条件查询符合的系统日志。

# GIS模块设计

## GIS模块概述

本系统通过GIS模块直观的显示EMC项目的基本信息，位置信息和工作状态。

GIS模块对应的功能需求包括：

1. 显示电子地图，及EMC项目所在的位置。
2. 电子地图基本操作。
3. 显示EMC项目的实时报警状态。
4. 检索并定位EMC项目的位置。

## GIS模块结构

GIS模块的结构如下图：

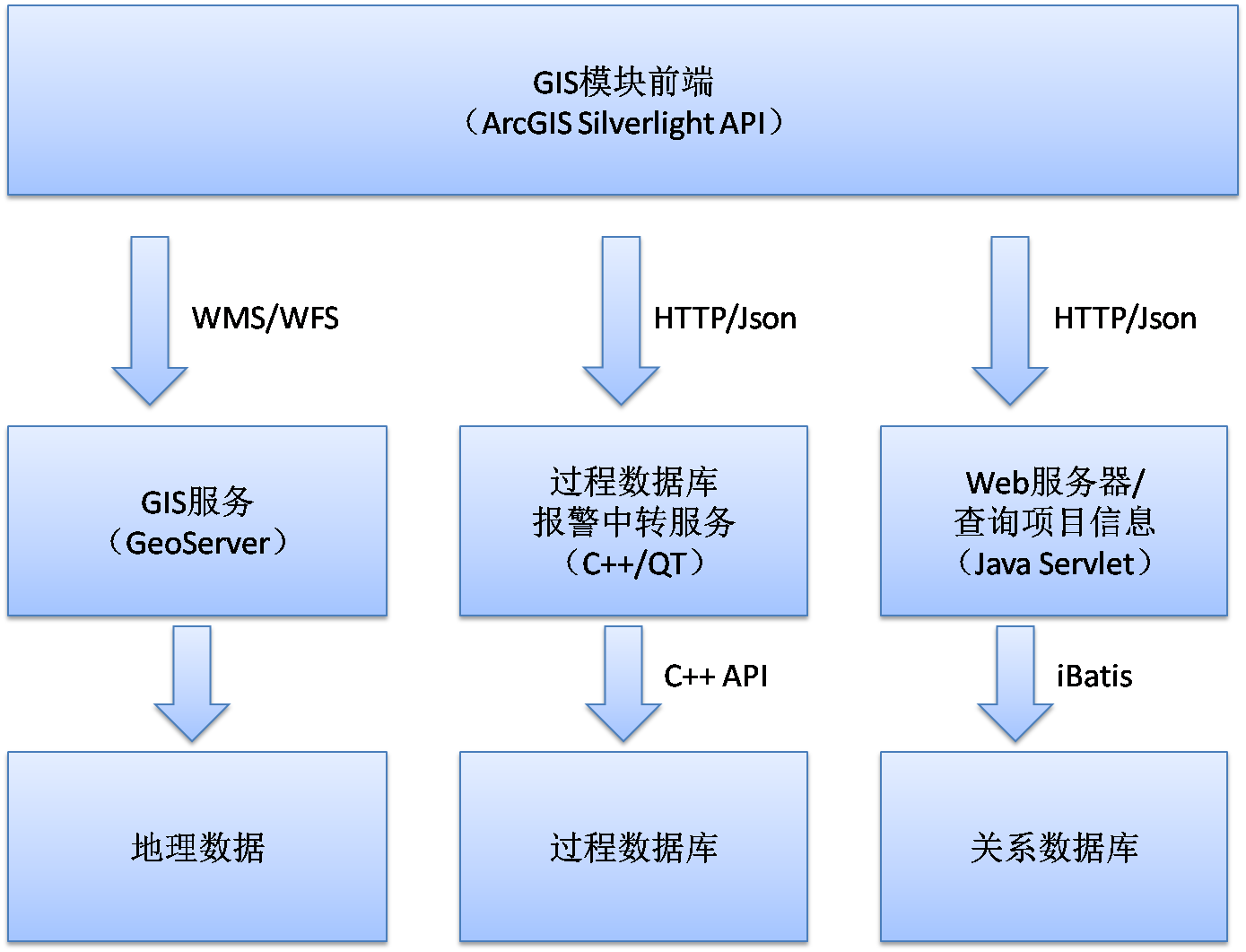


图 9‑1 GIS模块结构图

GIS模块的前端使用ArcGIS Silverlight API开发，ArcGIS Silverlight API是ESRI公司提供的免费GIS前端控件，具有丰富的功能和表现力，支持文档也比较完备。除支持ArcGIS Server和Bing Map外，ArcGIS Silverlight API也可以通过WMS/WFS标准来支持其他的GIS Server。

GIS的服务端我们选用GeoServer。GeoServer是一个社区支持的开源项目，是一个符合OpenGIS Web服务器规范的J2EE实现。GeoServer支持WMS和WFS标准，支持多种GIS数据格式，包括PostGIS、Shapefile、ArcSDE、Oracle、VPF、MySQL、MapInfo，支持上百种投影，能够将网络地图输出为jpeg、gif、png、SVG、KML 等多种图片格式。

在GIS模块中，前端和服务端使用WMS/WFS标准通讯。WMS/WFS是OGC（Open Geospatial Consortium/开放地理信息联盟）所定义的空间数据互操作的接口规范，该规范基于Http/XML来进行数据传输，并且基于XML定义了用于空间数据传输与转换的地理信息标记语言GML。WMS（Web Map Service）将地理数据定义为可视化的表现，可根据用户的命令返回不同图片格式的地图，命令通过URL参数传递，返回结果为图片或者GML描述的信息。WFS（Web Feature Service）根据用户的命令返回地理要素的信息，通过不同的命令，还可以对地理要素进行增删改，命令可以通过URL参数或GML传递，返回结果为GML描述的信息。ArcGIS Silverlight API和GeoServer都对WMS/WFS标准提供了完善的支持。

GIS模块前端从过程数据库的报警中转服务中获取实时报警信息，并展示在地图上。请求实时报警信息的命令，和返回的实时报警信息都通过Http协议传递，内容使用Json描述，报警中转服务通过过程数据库提供的API注册并获取实时报警信息。为保证实时性，报警中转服务使用C++开发，使用QT提供的网络模块库。

QT是一个开源的跨平台C++应用程序开发框架，经过多年的发展，并且近期还在不断的改进更新。QT可以支持多种平台和操作系统，提供了丰富的API和类供用户使用，模块化程度非常好，为用户的开发工作提供了方便。

除地图数据和报警信息外，GIS前端还需要展示一些EMC项目的基本信息，如项目合同起止日期等。GIS前端向在Web服务器中运行的Java Servlet服务发送http请求，Java Servlet从数据库中查询这些信息。前端发送的请求格式为Json，服务查询数据库使用iBatis组件。

# 组态模块设计

## 组态模块概述

组态模块可以使实施人员以灵活多变的组态方式来开发监控画面，并且开发过程方便简易。

组态模块可分为组态编辑模块和组态运行模块两部分。实施人员在组态编辑模块下编写开发自己需要的组态画面，并将画面上的元素属性与实际现场设备状态相对应的Tag点名称相关联。切换到组态运行模块时，用户可以实时观测到现场Tag点数据的变化情况，并对报警等异常状态进行处理。

## 组态画面的描述

组态画面采用XML语言描述，与具体的开发平台和语言无关。采用标准语言和标准规则描述保证了组态画面的向后兼容性，可以在任何平台上开发相应的组态运行系统打开同一组态画面。

组态画面描述语言的结构见下图：

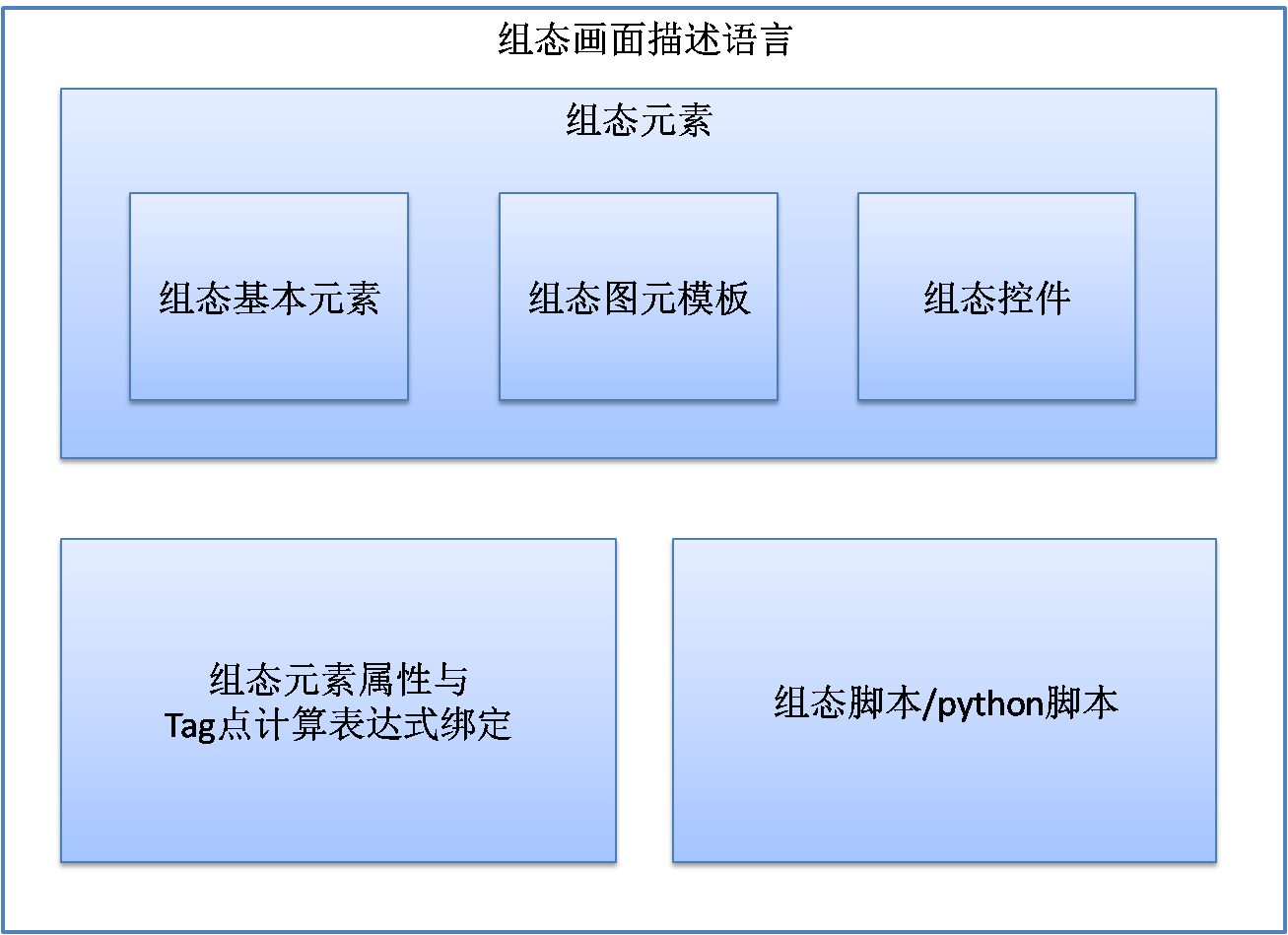


图 10‑1 组态画面描述语言结构图

如图描述，组态画面的描述可分为三部分：组态元素、组态元素属性与Tag点计算表达式绑定和组态脚本。

组态元素部分又可分为三部分：组态基本元素，如矩形、椭圆、线等基本图形和按钮、文本框、单选框等基本控件；组态图元模板，如仪表盘、水泵、阀门等对应于设备的图元；组态控件，如实时报警控件、趋势图控件、直方图控件等。

一个示例组态画面的文件内容如下，内容为XML格式：

<Pane>

<BasicControl:Canvas Name="Canvas" Width="1024" Height="768" BackColor="#FFFFFFFF" >

<BasicControl:Canvas.WildcardList>

</BasicControl:Canvas.WildcardList>

<BasicShape:Rectangle Name="Rectangle1" Left="167" Top="86" Width="151" Height="60" LineColor="#FF000000" ></BasicShape:Rectangle>

<BasicControl:TextBox Name="TextBox1" Left="409" Top="96" Width="96" Height="78" ForeColor="#FF000000" Text="TextBox" ></BasicControl:TextBox>

<BasicControl:Button Name="Button1" Left="562" Top="76" Width="112" Height="64" ForeColor="#FF000000" Text="Button" ></BasicControl:Button>

</BasicControl:Canvas>

<Tags>

<TagBinding ElementName="TextBox1" PropertyName="Text" IsEnable="True" Expression="TestDataServer.TestTag2.PV">

</TagBinding>

<TagBinding ElementName="Rectangle1" PropertyName="Width" IsEnable="True" Expression="TestDataServer.TestTag4.PV">

</TagBinding>

</Tags>

<Script>

<Element Name="Button1" >

<Event Name="Click" Content=" TextBox1.Text = TagMethods.GetTagValue(&quot;TestTag1&quot;)" />

</Element>

</Script>

</Pane>

其中“BasicControl:Canvas”节点部分描述的是组态画面中的组态元素，“Tags”节点部分描述的是组态元素属性与Tag点计算表达式绑定列表，“Script”节点部门描述的是组态画面中的脚本，组态画面中的脚本用Python语言编写。

## 组态模块结构

组态编辑模块的结构如下图：

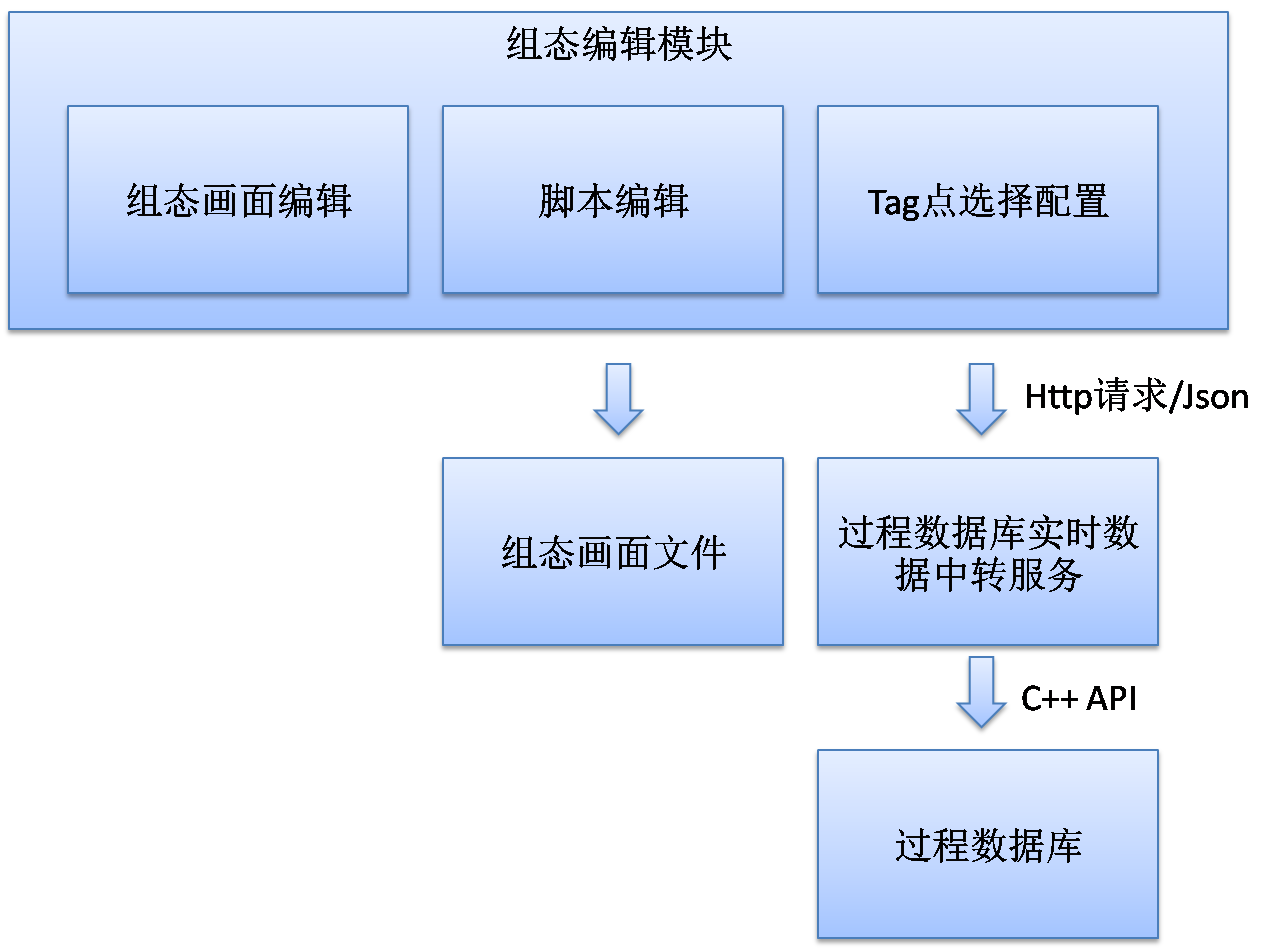


图 10‑2 组态编辑模块结构图

组态编辑模块由三部分组成：组态画面编辑，脚本编辑和Tag点选择配置。实施人员制作组态画面时，也需要经过这三个步骤。组态编辑模块使用Silverlight开发，脚本使用Python语言，在脚本编写中，可以使用Silverlight自带的所有类库，以及系统内置的系统函数。实施人员编写好组态画面后，将画面保存为XML格式的组态画面文件。

实施人员在为组态画面中的元素配置Tag点表达式时，需要从过程数据库中获取当前所有Tag点的列表。组态编辑模块的Tag点选择配置部分向过程数据库实时数据中转服务中发送格式为Json的Http请求，实时数据中转服务调用过程数据库的C++ API获得所需的Tag点列表，并返回给组态编辑模块。

组态运行模块的结构如下图：

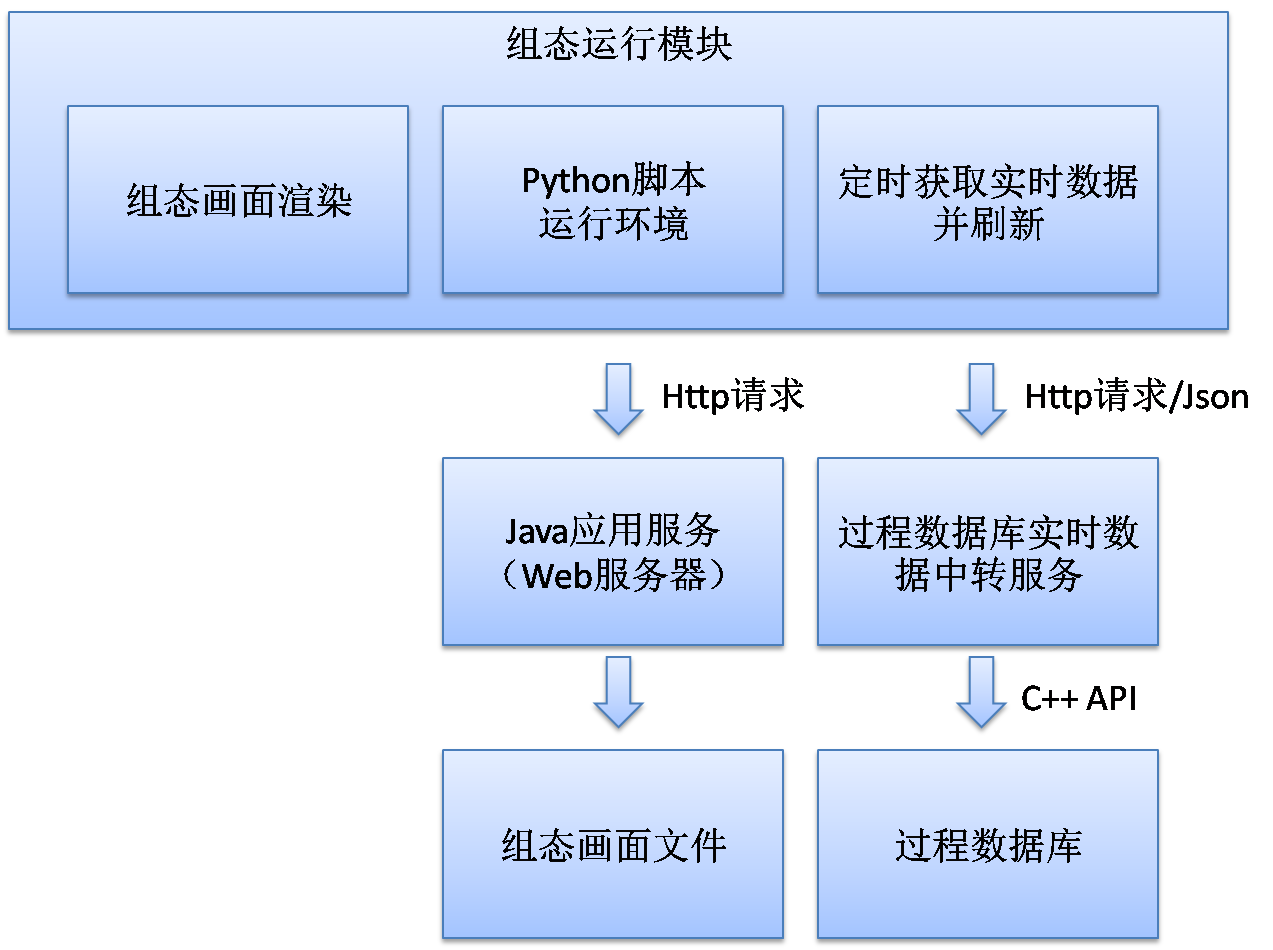


图 10‑3 组态编辑模块结构图

组态运行模块由三部分组成：组态画面渲染、Python脚本运行环境，定时获取实时数据并刷新。组态运行模块使用Silverlight开发。

组态运行模块加载组态画面时，向运行在Web服务器中的Java应用服务发送Http请求，Java应用服务从服务器本地读取所需的组态画面文件内容，返回给组态运行模块。

组态运行模块加载好组态画面后，首先向过程数据库实时数据中转服务中发送格式为Json的Http请求进行Tag点注册，实时数据中转服务调用过程数据库的C++ API注册所需的Tag点列表，注册成功后过程数据库定周期将最新的实时数据推送到实时数据中转服务中。组态运行模块定周期刷新组态画面时，向实时数据中转服务发送格式为Json的Http请求，实时数据中转服务将最新的数据返回给组态运行模块。

# 实时报警控件设计

## 实时报警控件概述

实时报警控件可以集成入组态模块内，为用户提供实时报警的显示和声音提醒。

当报警发生时，在实时报警控件中显示一条新报警信息，并且发出报警声音；当报警恢复后，该条报警在实时报警控件中的状态变为已恢复，并且报警声音消失；用户可以通过实时报警控件来确认一条或多条报警信息。

## 实时报警控件结构

实时报警控件的层次结构如下图：

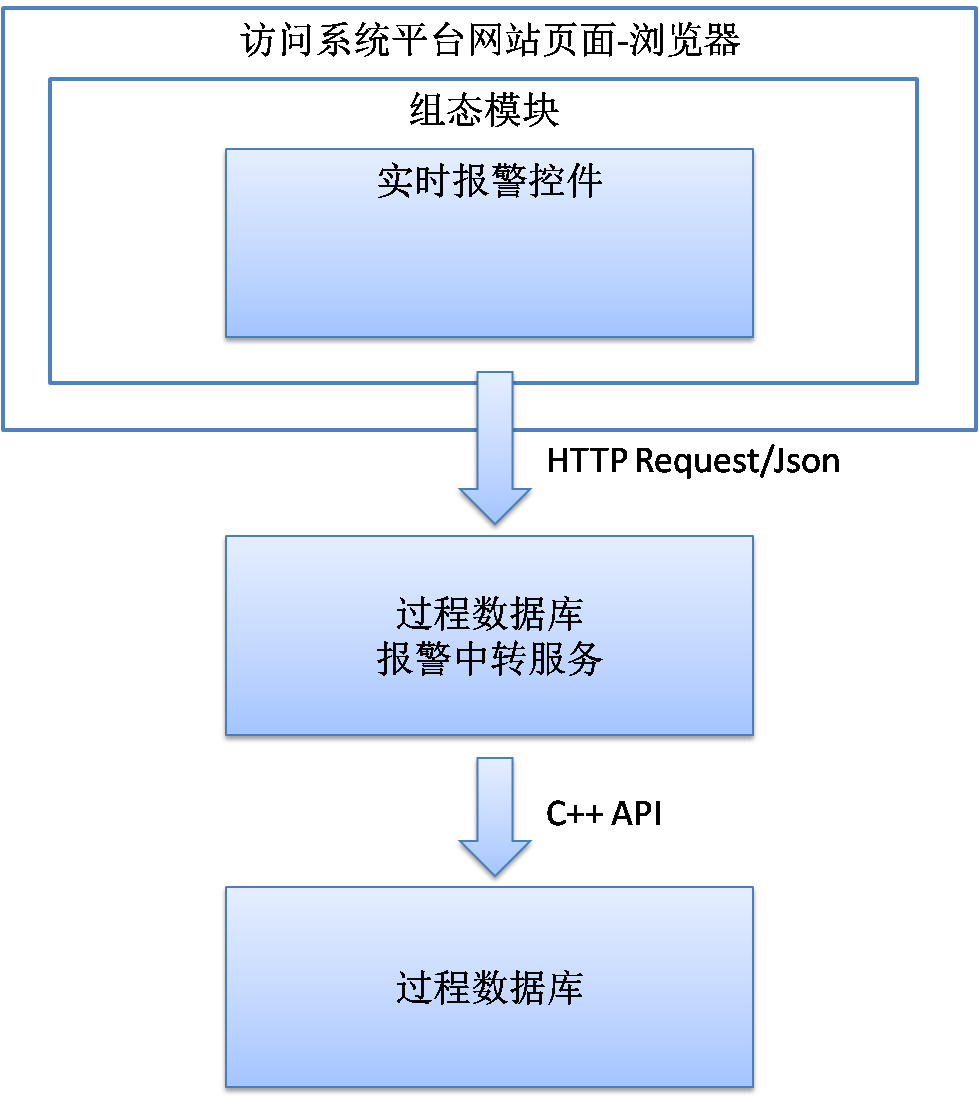


图 11‑1 实时报警控件结构图

实时报警控件作为组态控件的一种，可以嵌入在组态运行模块中运行。组态运行模块可以作为一种object元素，嵌入在系统平台网站的网页中运行。

实时报警控件通过向过程数据库报警中转服务发送Http请求来获取最新的报警数据，并且完成报警确认。Http请求和返回结果的数据格式使用Json格式。

为了保证实时报警控件获取报警信息的实时性，发送获取实时报警信息的Http请求时，如果报警中转服务中没有新报警信息，等待一段时间再返回没有新报警信息的结果，如果这等待期间有新报警信息产生，立刻返回包含新报警信息的结果，如果报警中转服务中已有新报警信息，立刻返回已有报警信息。实时报警控件在前一次请求结束后（可能为成功、失败或超时），再进行下一次请求。

过程数据库报警中转服务使用过程数据库的C++ API向过程数据库进行报警注册，当有新报警产生时，过程数据库立刻将报警信息推送至报警中转服务。

# 趋势图控件设计

## 趋势图控件概述

趋势图控件可以集成入组态模块内，用户可以通过趋势图控件查看过程变量或统计变量的实时趋势曲线和历史趋势曲线。

## 趋势图控件结构

趋势图控件的层次结构如下图：

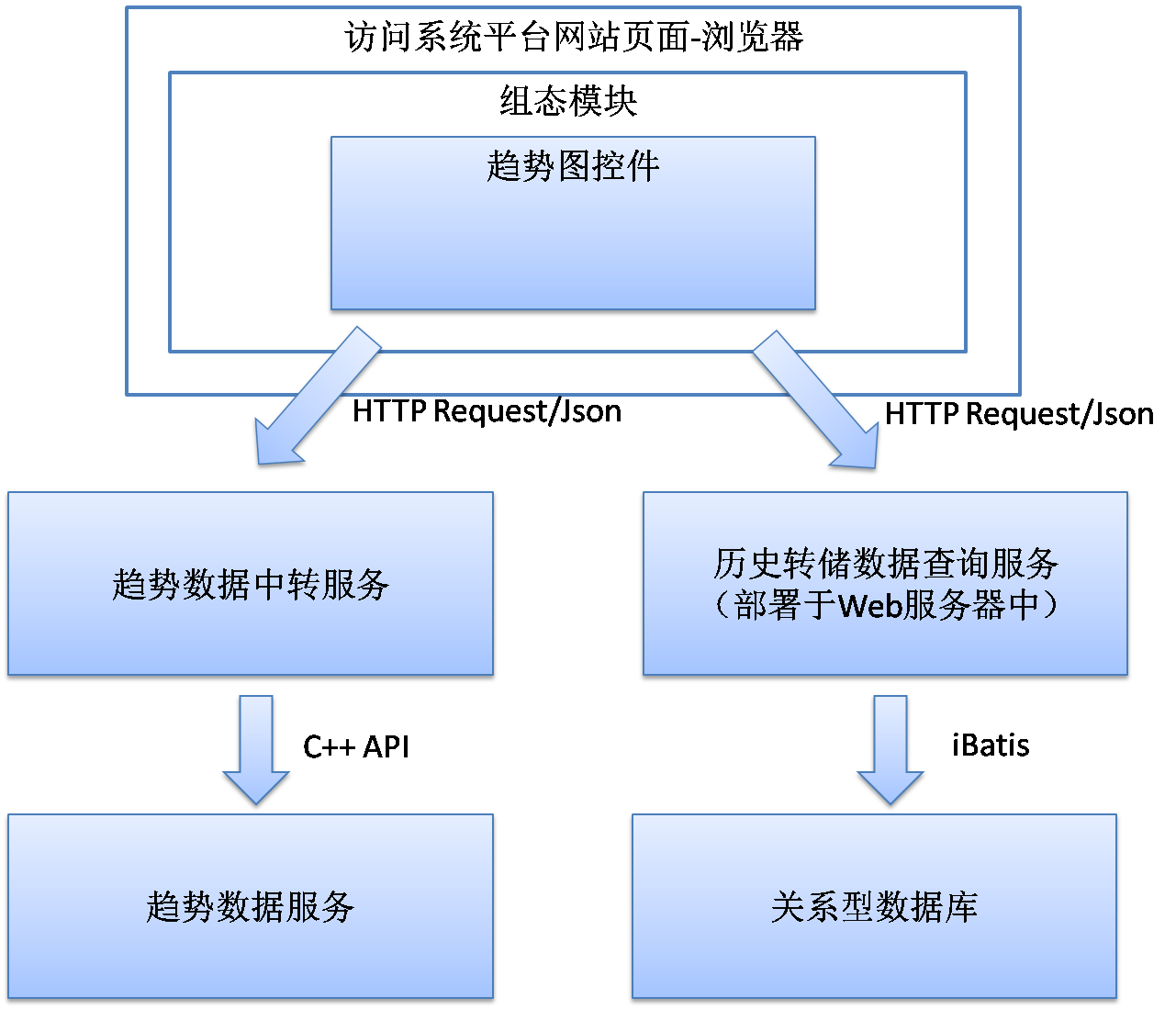


图 12‑1 趋势图控件结构图

趋势图控件作为组态控件的一种，可以嵌入在组态运行模块中运行。组态运行模块可以作为一种object元素，嵌入在系统平台网站的网页中运行。

趋势图控件优先向趋势数据中转服务请求数据。当请求的数据时间较老，趋势数据服务中没有保存时，趋势图控件向历史转储数据查询服务请求数据。

趋势图控件通过向趋势数据中转服务和历史转储数据查询服务发送Http请求来获取需要的趋势数据。Http请求和返回结果的数据格式使用Json格式。

趋势数据中转服务使用C++开发，通过趋势数据服务的C++ API获取趋势数据。历史存储数据查询服务使用Java开发，部署于Web服务器中，通过iBatis向关系型数据库查询数据。

# 报表模块设计

## 报表模块概述

用户可以通过报表模块对系统运行产生的设备过程数据、统计数据和报警信息等进行查询统计分析。

报表模块可分为报表设计工具和报表查询页面两部分。报表设计工具可以设计固定格式报表，也可以按用户自身需求对报表模板进行修改，设计灵活的自定义格式报表；报表查询页面除从数据库中查询转储数据和录入数据，然后生成报表之外，也可以访问按周期生成预先配置好的固定周期报表，访问固定周期报表不需要访问数据库。

## 报表模块结构

报表模块的结构如下图：

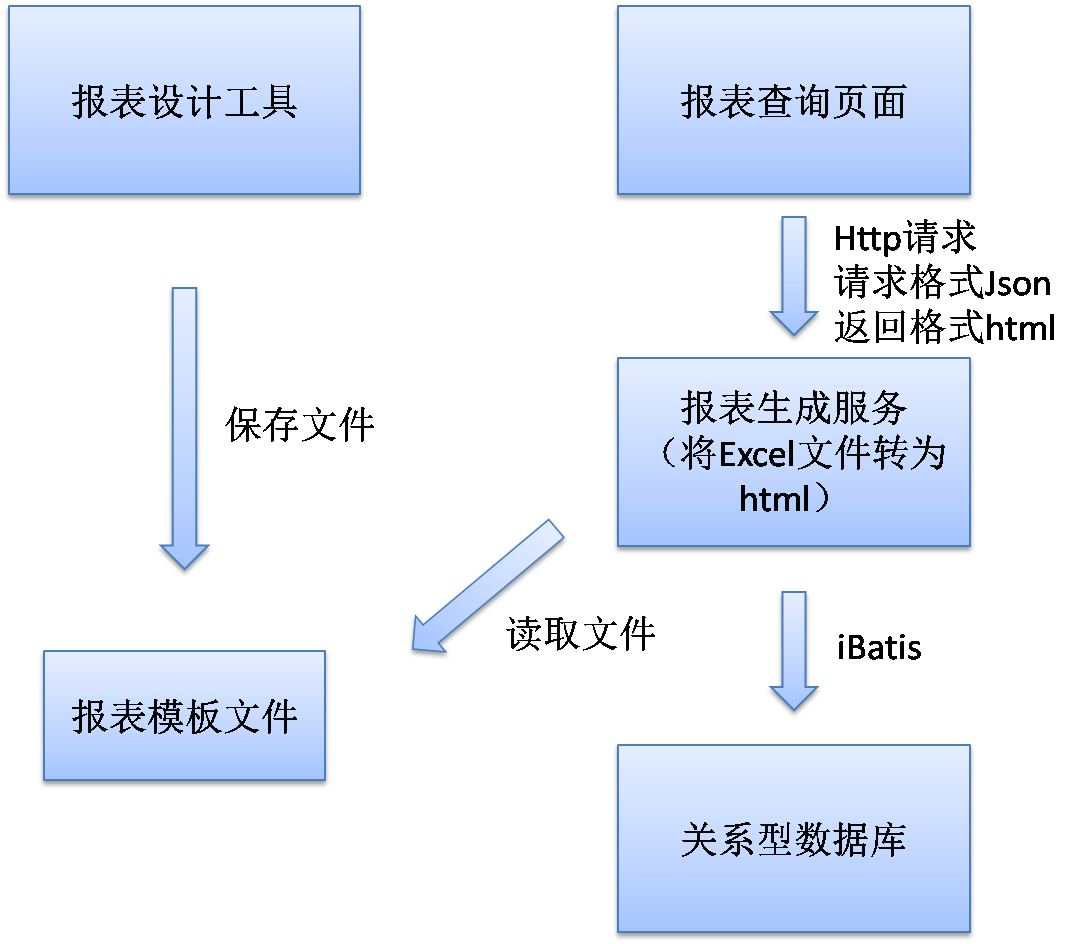


图 13‑1 报表模块结构图

实施人员或者用户使用报表设计工具（Excel）编辑报表模板文件，将编辑好的文件保存在服务器端的指定位置下。

报表查询页面运行在系统平台网站下，用户在报表查询页面中访问报表时，首先向报表生成服务发送Http请求，请求内容包括请求的报表模板，请求的数据查询条件，格式为Json。报表生成服务从关系型数据库中查询到需要的数据，包括转储的设备过程数据和统计数据，以及用户录入的数据，根据这些数据与报表模板文件生成Excel报表文件，报表生成服务再把Excel报表文件转为html报表文件，并返回给请求的客户端。客户端将返回的html报表文件嵌入至网页显示。

这个方案的主要优势在于，不需要在每个客户端都安装Excel，只需要在服务端安装一次Excel即可，体现了B/S方案部署的灵活性。

固定周期报表生成与查询的结构如下图：

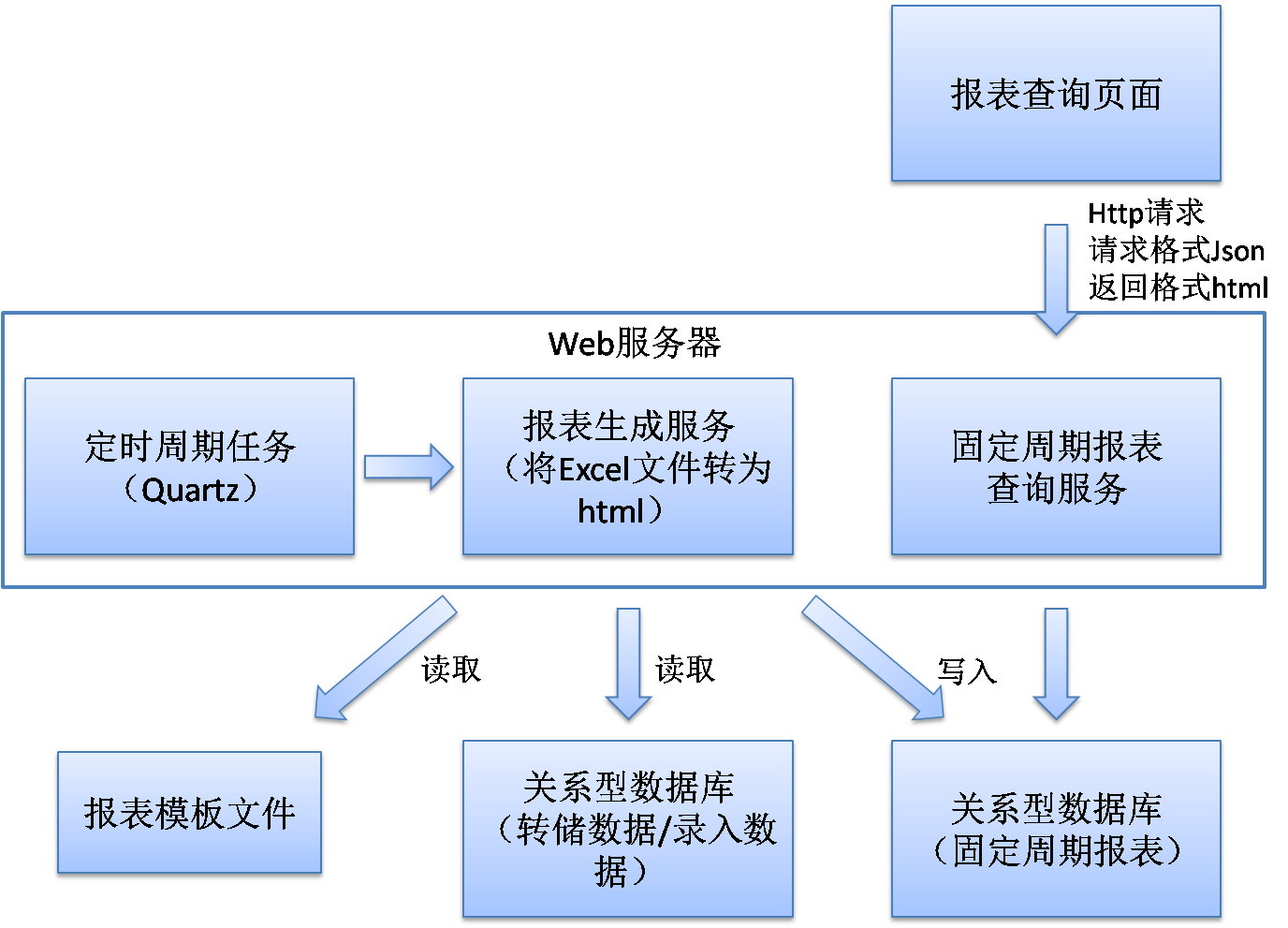


图 13‑2 固定周期报表生成与查询结构图

固定周期报表配置完成后，系统会设置一个定时周期任务，在指定的时刻生成固定周期报表。

定时周期任务与报表生成服务都运行在Web服务器中。定时周期任务利用Quartz框架来执行，Quartz是一个完全由Java编写的开源作业调度框架，可以并发管理大量的定时任务，一旦用户配置好执行时间表和任务类，Quartz会在执行时间表到达时执行任务类中的Execute方法，在执行过程中，如果关键事件被触发，Quartz会执行用户配置好的回调函数来通知用户。

报表生成服务被调用后，和普通的查询报表流程类似，首先从关系型数据库中读取需要的转储数据和录入数据，并按照报表模板文件生成Excel报表，然后报表生成服务将Excel报表转化为html报表文件，并存储在关系型数据库中。

用户查询固定周期报表时，不需要再次读取数据或报表模板文件，直接发送格式为Json的http请求到固定周期报表查询服务中，服务从数据库中读取固定周期报表html文件，然后返回到前端页面中。固定周期报表查询服务使用Java开发，运行于Web服务器中。

# 身份认证与权限模块设计

## 身份认证与权限模块概述

身份认证与权限模块为系统提供安全性保障。系统采用基于角色和资源的权限管理机制，控制每个角色可以访问的资源。

## 身份认证与权限模块模型

身份认证与权限的模型如下图：

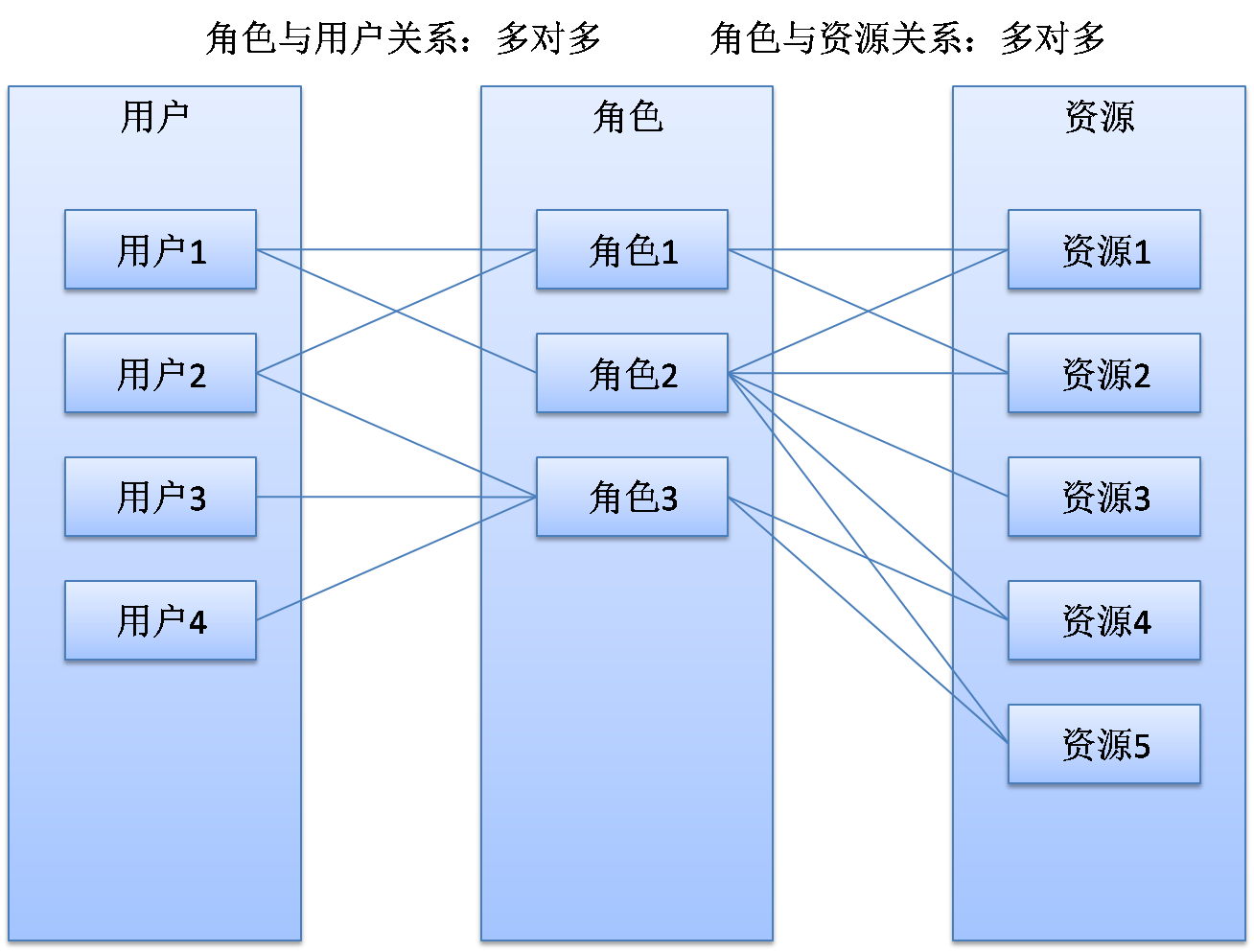


图 14‑1 身份认证与权限模块模型图

在身份认证与权限的模型中，主要的概念有三个：角色、用户和资源。主要的关系有两个：角色与用户的关系，角色与资源的关系。

角色的概念相当于用户的集合，与用户之间是多对多的关系，一个角色中可以包含多个用户，一个用户也可以分配多个角色。资源的概念相当于系统中可以被授权的元素，在本系统中主要是页面，角色和资源之间也是多对多的关系，一个角色可以分配给多个资源的权限，一个资源也可以授权给多个角色。

当用户访问系统中的某个资源时，如果用户所属的任意一个角色具有对该资源的权限，则该用户可以访问该资源，反之，则不可访问。

## 身份认证与权限模块结构

身份认证与权限模块的结构如下图：

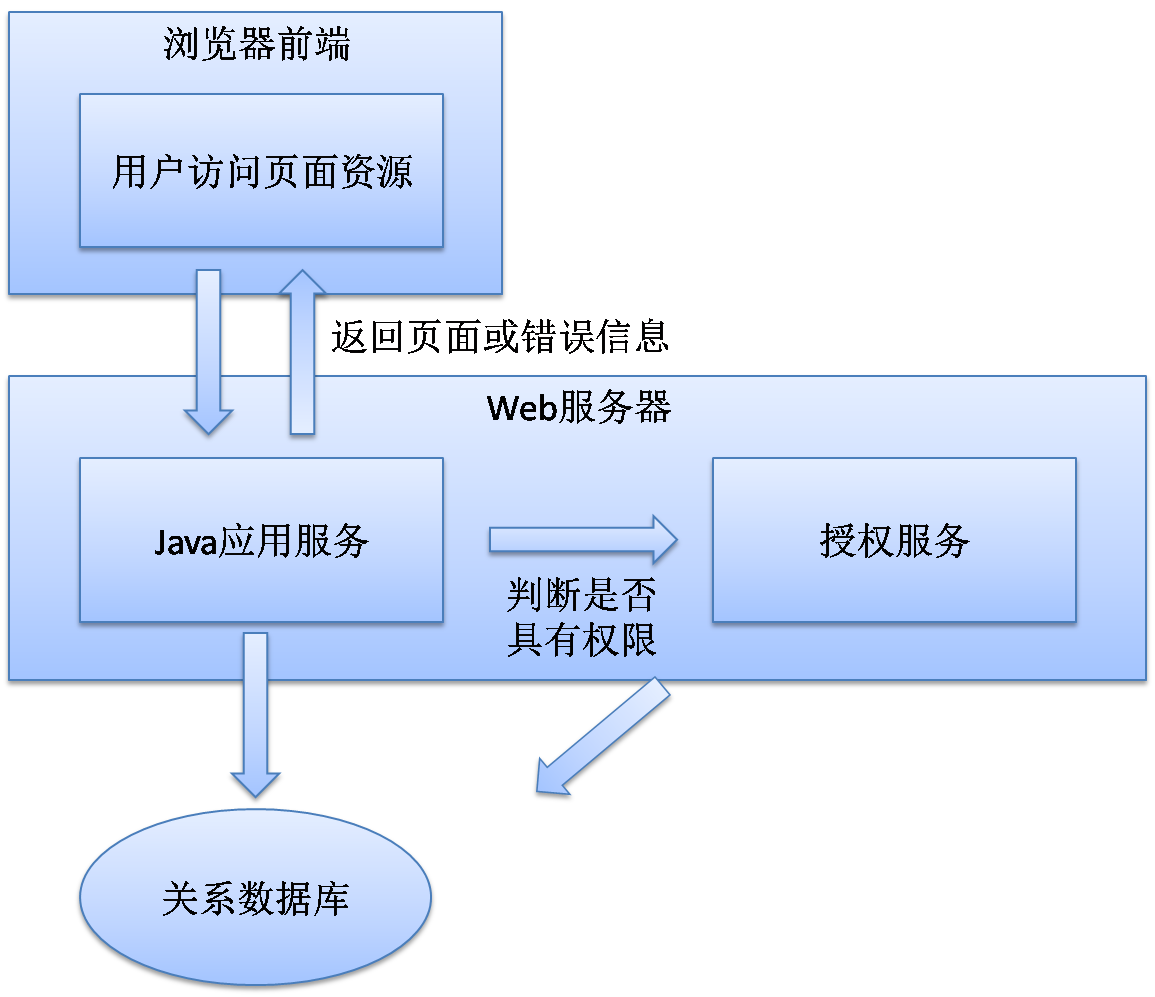


图 14‑2 身份认证与权限模块结构图

当用户在浏览器中访问系统平台网站中的一个页面资源时，向Java应用服务发送Http请求，Java应用服务首先调用授权服务判断当前用户是否对该页面资源具有权限，如果具有权限，Java应用服务继续读取关系数据库，生成页面并返回给前端，如果不具有权限，Java应用服务返回给前端错误信息。

# 系统业务数据库设计

## E-R图

系统业务数据库E-R图如下：

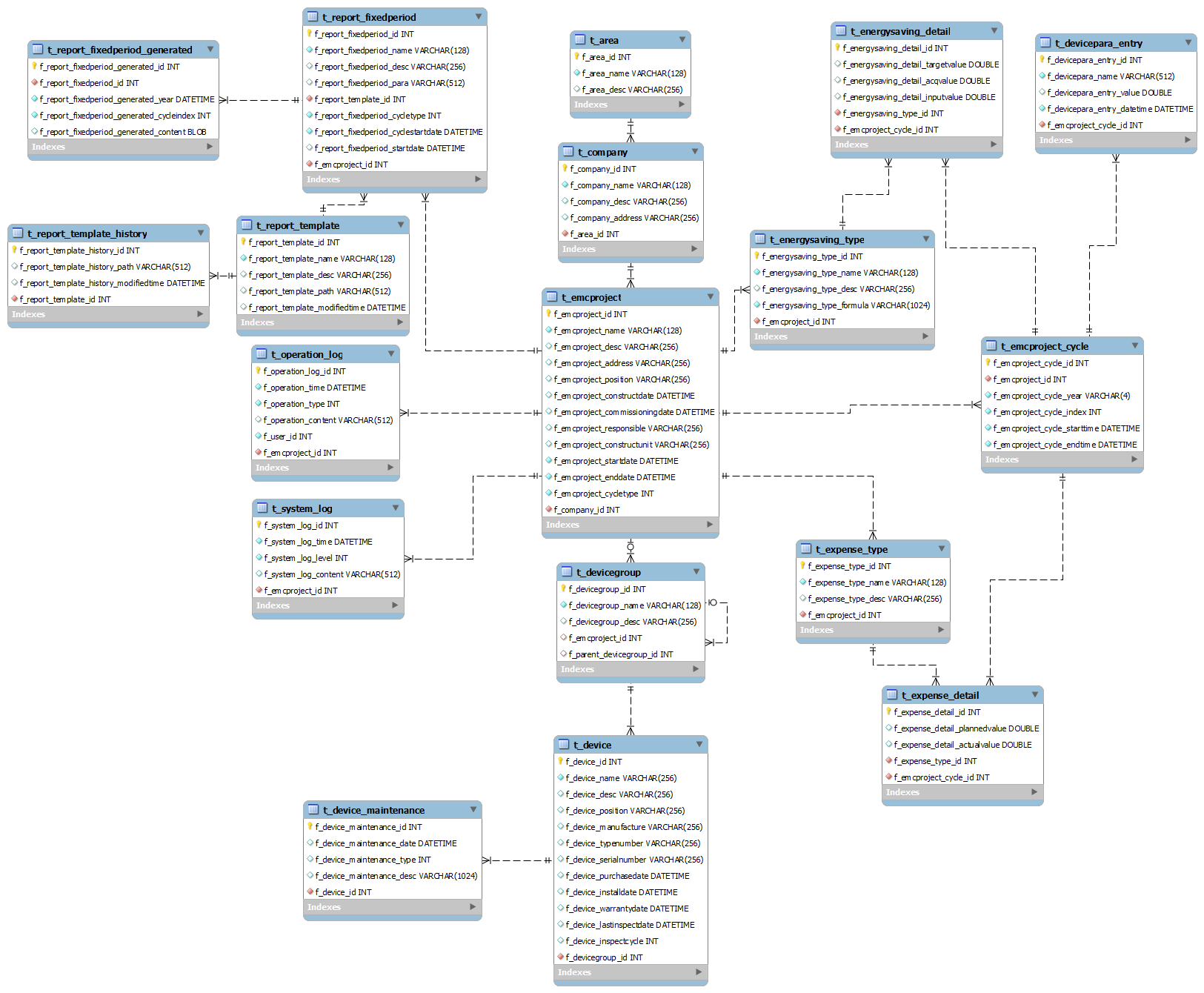


图 15‑1 系统业务数据库E-R图

## 数据库结构说明

系统业务数据库的各个数据表的用途说明如下：

表 ‑1 系统业务数据库数据表用途说明

|  |  |
| --- | --- |
| 表名称 | 表描述 |
| t\_area | 区域信息编码表 |
| t\_company | 公司信息编码表 |
| t\_emcproject | 项目信息编码表 |
| t\_devicegroup | 设备组信息编码表 |
| t\_device | 设备信息编码表 |
| t\_devicemaintenance | 设备维护信息表 |
| t\_emcproject\_cycle | 项目周期信息表 |
| t\_energysaving\_type | 节能量类型表 |
| t\_energysaving\_detail | 节能量数据表 |
| t\_devicepara\_entry | 设备参数录入表 |
| t\_expense\_type | 费用类型表 |
| t\_expense\_detail | 费用数据表 |
| t\_operation\_log | 操作日志表 |
| t\_system\_log | 系统日志表 |
| t\_report\_template | 报表模板表 |
| t\_report\_template\_history | 历史报表模板表 |
| t\_report\_fixedperiod | 固定周期报表表 |
| t\_report\_fixedperiod\_generated | 已生成固定周期报表表 |

系统业务数据库的各个数据表的结构详细说明如下：

表 ‑2 t\_area结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_area\_id | 区域id | INT | Y | Y |
| f\_area\_name | 区域名称 | VARCHAR(128) |  | Y |
| f\_area\_desc | 区域描述 | VARCHAR(256) |  |  |

表 ‑3 t\_company结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_company\_id | 公司id | INT | Y | Y |
| f\_company\_name | 公司名称 | VARCHAR(128) |  | Y |
| f\_company\_desc | 公司描述 | VARCHAR(256) |  |  |
| f\_company\_address | 公司地址 | VARCHAR(256) |  |  |
| f\_area\_id | 所属区域id | INT |  | Y |

表 ‑4 t\_emcproject结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_emcproject\_id | 项目id | INT | Y | Y |
| f\_emcproject\_name | 项目名称 | VARCHAR(128) |  | Y |
| f\_emcproject\_desc | 项目描述 | VARCHAR(256) |  |  |
| f\_emcproject\_address | 项目地址 | VARCHAR(256) |  |  |
| f\_emcproject\_position | 项目位置 | VARCHAR(256) |  | Y |
| f\_emcproject\_constructdate | 建设时间 | DATETIME |  |  |
| f\_emcproject\_commissiondate | 完工时间 | DATETIME |  |  |
| f\_emcproject\_responsible | 负责人 | VARCHAR(256) |  |  |
| f\_emcproject\_constructunit | 建设单位 | VARCHAR(256) |  |  |
| f\_emcproject\_startdate | 开始时间 | DATETIME |  | Y |
| f\_emcproject\_enddate | 结束时间 | DATETIME |  | Y |
| f\_emcproject\_cycletype | 周期类型 | INT |  | Y |
| f\_company\_id | 所属公司id | INT |  | Y |

表 ‑5 t\_devicegroup结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_devicegroup\_id | 设备组id | INT | Y | Y |
| f\_devicegroup\_name | 设备组名称 | VARCHAR(128) |  | Y |
| f\_devicegroup\_desc | 设备组描述 | VARCHAR(256) |  |  |
| f\_emcproject\_id | 所属项目id | INT |  |  |
| f\_parent\_devicegroup\_id | 所属父设备组id | INT |  |  |

表 ‑6 t\_device结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_device\_id | 设备id | INT | Y | Y |
| f\_device\_name | 设备名称 | VARCHAR(128) |  | Y |
| f\_device\_desc | 设备描述 | VARCHAR(256) |  |  |
| f\_device\_position | 设备位置 | VARCHAR(256) |  |  |
| f\_device\_manufacture | 制造商 | VARCHAR(256) |  |  |
| f\_device\_typenumber | 设备型号 | VARCHAR(256) |  |  |
| f\_device\_serialnumber | 设备序列号 | VARCHAR(256) |  |  |
| f\_device\_purchasedate | 购买日期 | DATETIME |  |  |
| f\_device\_installdate | 安装日期 | DATETIME |  |  |
| f\_device\_warrantydate | 保修截止日期 | DATETIME |  |  |
| f\_device\_lastinspectdate | 最后检查日期 | DATETIME |  |  |
| f\_device\_inspectcycle | 检查周期 | INT |  | Y |
| f\_devicegroup\_id | 所属设备组id | INT |  | Y |

表 ‑7 t\_devicemaintenance结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_device\_maintenance\_id | 设备维护信息id | INT | Y | Y |
| f\_device\_maintenance\_date | 设备维护日期 | DATETIME |  | Y |
| f\_device\_maintenance\_type | 设备维护类型 | INT |  | Y |
| f\_device\_maintenance\_desc | 设备维护描述 | VARCHAR(1024) |  |  |
| f\_device\_id | 所属设备id | INT |  | Y |

表 ‑8 t\_emcproject\_cycle结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_emcproject\_cycle\_id | 项目周期id | INT | Y | Y |
| f\_emcproject\_id | 所属项目id | INT |  | Y |
| f\_emcproject\_cycle\_year | 周期年份 | VARCHAR(4) |  | Y |
| f\_emcproject\_cycle\_index | 周期序号 | INT |  | Y |
| f\_emcproject\_cycle\_starttime | 开始时间 | DATETIME |  | Y |
| f\_emcproject\_cycle\_endtime | 结束时间 | DATETIME |  | Y |

表 ‑9 t\_energysaving\_type结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_energysaving\_type\_id | 节能量类型id | INT | Y | Y |
| f\_energysaving\_type\_name | 节能量类型名称 | VARCHAR(128) |  | Y |
| f\_energysaving\_type\_desc | 节能量类型描述 | VARCHAR(256) |  |  |
| f\_energysaving\_type\_formula | 节能量类型计算公式 | VARCHAR(1024) |  | Y |
| f\_emcproject\_id | 所属项目id | INT |  | Y |

表 ‑10 t\_energysaving\_detail结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_energysaving\_detail\_id | 节能量数据id | INT | Y | Y |
| f\_energysaving\_detail\_targetvalue | 节能量数据目标值 | DOUBLE |  |  |
| f\_energysaving\_detail\_acqvalue | 节能量数据采集值 | DOUBLE |  |  |
| f\_energysaving\_detail\_inputvalue | 节能量数据录入值 | DOUBLE |  |  |
| f\_energysaving\_type\_id | 所属节能量类型id | INT |  | Y |
| f\_emcproject\_cycle\_id | 所属项目周期id | INT |  | Y |

表 ‑11 t\_devicepara\_entry结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_devicepara\_entry\_id | 设备参数录入id | INT | Y | Y |
| f\_devicepara\_name | 设备参数名称 | VARCHAR(512) |  | Y |
| f\_devicepara\_entry\_value | 录入值 | DOUBLE |  | Y |
| f\_devicepara\_entry\_datetime | 录入时间 | DATETIME |  | Y |
| f\_emcproject\_cycle\_id | 所属项目周期id | INT |  | Y |

表 ‑12 t\_expense\_type结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_expense\_type\_id | 费用类型id | INT | Y | Y |
| f\_expense\_type\_name | 费用类型名称 | VARCHAR(128) |  | Y |
| f\_expense\_type\_desc | 费用类型描述 | VARCHAR(256) |  |  |
| f\_emcproject\_id | 所属项目id | INT |  | Y |

表 ‑13 t\_expense\_detail结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_expense\_detail\_id | 费用数据id | INT | Y | Y |
| f\_expense\_detail\_plannedvalue | 费用数据计划值 | DOUBLE |  |  |
| f\_expense\_detail\_actualvalue | 费用数据实际值 | DOUBLE |  |  |
| f\_expense\_type\_id | 所属费用类型id | INT |  | Y |
| f\_emcproject\_cycle\_id | 所属项目周期id | INT |  | Y |

表 ‑14 t\_operation\_log结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_operation\_log\_id | 操作日志id | INT | Y | Y |
| f\_operation\_time | 操作时间 | DATETIME |  | Y |
| f\_operation\_type | 操作类型 | INT |  | Y |
| f\_operation\_content | 操作日志内容 | VARCHAR(512) |  |  |
| f\_user\_id | 操作用户id | INT |  | Y |
| f\_emcproject\_id | 所属项目id | INT |  | Y |

表 ‑15 t\_system\_log结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_system\_log\_id | 系统日志id | INT | Y | Y |
| f\_system\_log\_time | 日志时间 | DATETIME |  | Y |
| f\_system\_log\_level | 日志级别 | INT |  | Y |
| f\_system\_log\_content | 日志内容 | VARCHAR(512) |  |  |
| f\_emcproject\_id | 所属项目id | INT |  | Y |

表 ‑16 t\_report\_template结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_report\_template\_id | 报表模板id | INT | Y | Y |
| f\_report\_template\_name | 报表模板名称 | VARCHAR(128) |  | Y |
| f\_report\_template\_desc | 报表模板描述 | VARCHAR(256) |  |  |
| f\_report\_template\_path | 报表模板路径 | VARCHAR(512) |  |  |
| f\_report\_template\_modifiedtime | 最后更改时间 | DATETIME |  |  |

表 ‑17 t\_report\_template\_history结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_report\_template\_history\_id | 历史报表模板id | INT | Y | Y |
| f\_report\_template\_history\_path | 历史报表模板路径 | VARCHAR(512) |  |  |
| f\_report\_template\_history\_modifiedtime | 最后更改时间 | DATETIME |  | Y |
| f\_report\_template\_id | 所属报表模板id | INT |  | Y |

表 ‑18 t\_report\_fixedperiod结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_report\_fixperiod\_id | 固定周期报表id | INT | Y | Y |
| f\_report\_fixperiod\_name | 固定周期报表名称 | VARCHAR(128) |  | Y |
| f\_report\_fixperiod\_desc | 固定周期报表描述 | VARCHAR(256) |  |  |
| f\_report\_fixperiod\_para | 固定周期报表参数 | VARCHAR(512) |  |  |
| f\_report\_template\_id | 报表模板id | INT |  | Y |
| f\_emcproject\_id | 所属项目id | INT |  | Y |

表 ‑19 t\_report\_fixedperiod\_generated结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_report\_fixperiod\_generated\_id | 已生成固定周期报表id | INT | Y | Y |
| f\_report\_fixperiod\_id | 固定周期报表id | INT |  | Y |
| f\_report\_fixperiod\_generated\_year | 生成年份 | VARCHAR(4) |  | Y |
| f\_report\_fixperiod\_generated\_cycleindex | 生成周期序号 | INT |  | Y |
| f\_report\_fixperiod\_generated\_content | 报表内容 | BLOB |  |  |

# 系统转储数据库设计

## E-R图

系统转储数据库E-R图如下：

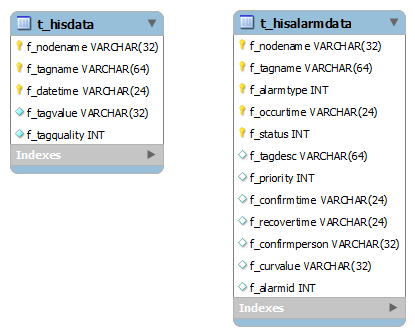


图 16‑1 系统转储数据库E-R图

该图仅用于说明单个EMC项目历史数据存储的结构，对于每个EMC项目，存储该项目的历史过程数据和历史报警数据的表是独立的，在创建EMC项目时，自动会创建这两张表，表名称为t\_hisdata\_”projectname”和t\_hisalarmdata\_”projectname”。

## 数据库结构说明

系统转储数据库的各个数据表的用途说明如下：

表 ‑1 系统转储数据库数据表用途说明

|  |  |
| --- | --- |
| 表名称 | 表描述 |
| t\_hisdata | 历史过程数据转储表 |
| t\_hisalarmdata | 历史报警数据转储表 |

系统转储数据库的各个数据表的结构详细说明如下：

表 ‑2 t\_hisdata结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_nodename | 节点名 | VARCHAR(32) | Y | Y |
| f\_tagname | 变量名 | VARCHAR(64) | Y | Y |
| f\_datetime | 时间戳 | VARCHAR(24) | Y | Y |
| f\_tagvalue | 数据值 | VARCHAR(32) |  | Y |
| f\_tagquality | 数据质量 | INT |  | Y |

表 ‑3 t\_hisalarmdata结构说明表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 字段描述 | 数据类型 | PK | NOT NULL |
| f\_nodename | 节点名 | VARCHAR(32) | Y | Y |
| f\_tagname | 变量名 | VARCHAR(64) | Y | Y |
| f\_alarmtype | 报警类型 | INT | Y | Y |
| f\_occurtime | 报警开始时刻 | VARCHAR(24) | Y | Y |
| f\_status | 报警状态 | INT | Y | Y |
| f\_tagdesc | 变量描述 | VARCHAR(64) |  |  |
| f\_priority | 报警优先级 | INT |  | Y |
| f\_confirmtime | 报警确认时刻 | VARCHAR(24) |  |  |
| f\_recovertime | 报警恢复时刻 | VARCHAR(24) |  |  |
| f\_confirmperson | 确认用户 | VARCHAR(32) |  |  |
| f\_curvalue | 当前值 | VARCHAR(32) |  |  |
| f\_alarmid | 报警ID | INT |  | Y |