# 可视化云平台开发前期技术准备和后期初步规划

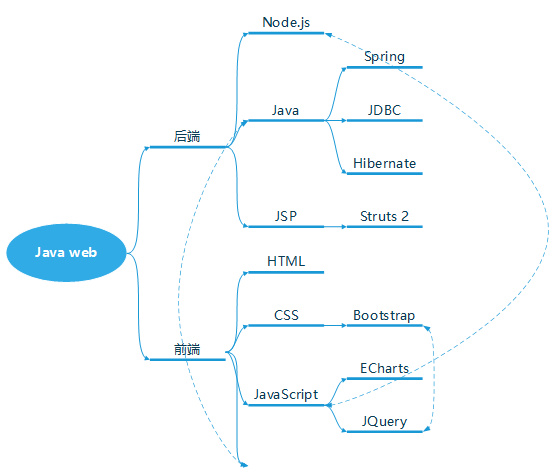
2016.10.27

通过近几年的开发和现场经验，以新能源为例，我发现该领域的监控系统品牌非常多，除了传统电力系统集成商，如南瑞科技、南瑞继保、长园深瑞、许继电气等，还包括华为、淘科、木联能等新能源领域专业集成商。   
传统电力系统监控厂家一般在变电站自动化监控系统或或调度自动化系统的基础上衍生新能源监控系统。这些系统都是较为传统的C/S模式，用前置机或通信管理机采集各单元的信息，并以一次接线图为主要方式在监控画面上反映这些信息。监控画面一般直接沿用监控系统的HMI，图形的绘制也与传统的厂站或调度一次接线图为主。   
新能源领域专业集成商一般采用B/S模式，以网页作为HMI，搭建轻量级的监控系统。得益于近几年网页前端技术的飞速发展，这些监控系统画面精美、视觉冲击力强。除了本地监控以外，这些监控系统也融入了大量的运维方面的功能，还可以根据用户的级别，选配云数据平台、手机App等组件，进一步提高用户体验。   
两种类型的监控系统侧重点不一，前者强调高效地反映厂站的真实情况，后者强调对数据的筛选、分析。发电系统或电力系统外的用户更加青睐与运维结合紧密、有较好用户体验的系统，对数据的统计、分析有助于提高系统的价值，开发具有互联网风格的监控系统有其必要性。同时公司的D6000主站系统作为传统的电力系统监控系统，在采集、监控领域有很深的积累。为了能够发挥这项电力系统外集成商不可比拟的优势，传统的电力监控系统将继续作为数据平台存在。   
互联网风格的监控系统与公司其它监控系统产品有所不同，它主要面向是发电系统或系统外用户，这也是我们公司的主要客户。这些用户对系统布局、风格没有统一的企业标准约束，因此很难用一套标准的产品来满足所有用户的需要，需要较多的UI开发，同时数据统计分析的功能也需要在现场逐渐提炼。所以当前较为紧迫的是首先做相关技术储备，接着再开发一套B/S模式的可视化基础云平台，连接监控系统基础平台和UI，为系统开发和工程开发提供基础设施，降低应用开发的成本。同时这也是对公司产品线的极大补充，和公司以及集团的其它已有产品形成相互促进的作用。

下面开始具体阐述开发前的技术储备和后期的初步规划：

**BS部分**

一般来说，对于一个以前没有涉及过的知识领域，开发前的技术准备通常都是一个先熟悉的过程，首先对技术体系有个大致的了解，具体的技术细节和技术运用还需要在实际开发中不断摸索和掌握。下面先看一下Java web开发的技术体系：



需要重点梳理和了解的知识有：

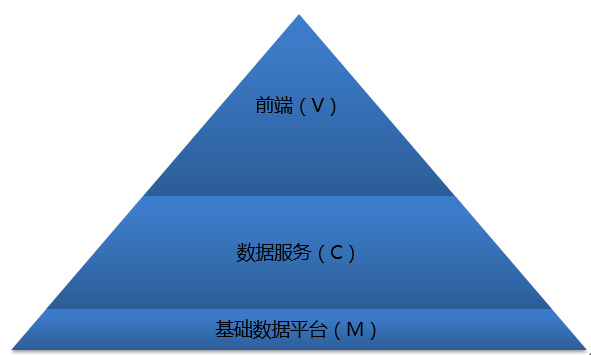
* Java语言
* jsp相关知识
* HTML/js脚本语言/CSS
* ECharts框架

可以集合教学视频和airacorba工程学习以上几个知识点。对于BS部分的前期学习，达到以下几点即可达到预定目标：

1. 对于相关知识系统有了初步的理解和掌握；
2. Java web工程的开发环境的搭建；
3. 可以利用IDEA开发工具自己动手创建一个测试用例，调试js脚本语言、CSS样式表以及ECharts相关图表的静态页面文件等；
4. 可以读懂一个Java web工程的前端和后端代码（如airacorba），可以理解工程配置文件，并对工程的前端展示、后端服务以及前后端交互脉络有一个整体的认识； BS部分的学习时间初步定为两周，到本月23号，可根据实际情况调整。

**CS部分**

对于云平台开发，BS部分负责业务逻辑的实现，而CS部分则是数据池，是整个云平台的基础。下图为云平台整体架构示意图：



其中：

* 前端（V）表示前台的可视化展示部分，开发框架为HTML/CSS/JavaScript...；
* 数据服务（C）表示数据挖掘、统计分析、运维管理等功能的后端实现，开发框架是以Java语言为基础的Spring+Hibernate框架；
* 基础数据平台（M）是系统组网、数据采集，处理，存盘等基础功能支撑平台，是实时数据和历史数据的提供者，开发框架是以C++语言为基础的底层框架。

对于CS部分，等BS部分熟悉之后，再搭建D6000后台环境，后台部分大家基础都比较好，熟悉起来会比较块。目前我们的开发和运行环境大概说明如下：

1. 操作系统为Win10 64bit/RedHat 5.10/RedHat6.6；
2. 数据库为Oracle 11g 64bit/达梦/MySQL；
3. QT版本为qt484 64bit；
4. 开发工具为VS2008；
5. 其他第三方库，如qwt/ace/snmp等；

一般我们自己的开发机器环境为：Win10+VS2008+Oracle11g+qt484 (均为64位)，供参考。开发环境后期会视情况逐步升级。

**BS和CS的交互**

BS和CS的交互分为实时数据交互和历史数据交互:

* 实时数据交互，采用小快灵的web service方式，既适合电力系统实时数据的特点，又有效的屏蔽了C++开发框架和Java开发框架的异构特性。
* 历史数据交互。CS部分的存盘采用D6000的DBi，BS部分利用JDBC自己封装数据库接口。

此部分等D6000系统后台熟悉后，视公司和部门情况开始着手准备部分C++接口的修改以及JDBC接口的封装和开发。