**Aug 31**

14

**UCB OpenBAS**

Introduction to the UCB OpenBAS Prelease

UCB OpenBAS team

08

**Fall**

This document provides a overview of the UCB OpenBAS 8/21/14 Pre-release, describes some of the underlying concepts, and illustrates a typical usage case. This in not the same as installing a consumer product, as this seeks to illustrate how the current OpenBAS system is created from its Open Source code base. Much of what is presented here would be handled within typical commercial project packaging. It is not a “users manual” or “product installation guide”, but rather an introduction to the design and implementation of a powerful and flexible open source system. Our understanding is that DOE is keen to see what is “beneath the hood”.

Figure 1 shows the internal architecture of the OpenBAS system (<https://github.com/SoftwareDefinedBuildings/openbas> ) is shown It is accessed from any display device, such as a computer browser, tablet, phone, or embedded display. We discuss the OpenBAS presentation later. It is a simple, open source web application that can be extended, customized, or modified, built using the meteor framework (<https://www.meteor.com/>) over the OpenBAS api (application programming interface). The core of OpenBAS is constructed upon the open source sMAP (simple Monitoring and Actuation Protocol) infrastructure (<https://github.com/SoftwareDefinedBuildings/smap> ). This comprises four kinds of RESTful web services:

* sMAP drivers: continuously running, monitored processes that connect to one of a wide variety of devices used for HVAC, Lighting, and General controllers and present it as a restful web service. Instances of sMAP drivers are described by a configuration (.ini) that specifies metadata relevant to the device and registers the data streams from these devices to the Archiver. A large collection of such drivers is available at <https://github.com/SoftwareDefinedBuildings/smap/tree/master/python/smap/drivers>.

* sMAP services: higher level, continuously running, monitored process that provider higher level OpenBAS functionality, such as discovery services (recognizing devices, loading their driver and constructing .ini files for the particular sMAP source), scheduling services (e.g., executing an integrated master schedule by publishing setpoints or commands to tstats, lighting controllers, or other devices.)
* sMAP archiver: a robust and efficient broker that stores Metadata associated with collections and streams, supports publication and subscription to streams, and implements a powerful query language over such richly tagged streams. It pushes updates to zero or more timeseries data stores.
* sMAP timeseries store: an optimized time series store for data streams that efficiently implements sMAP queries.

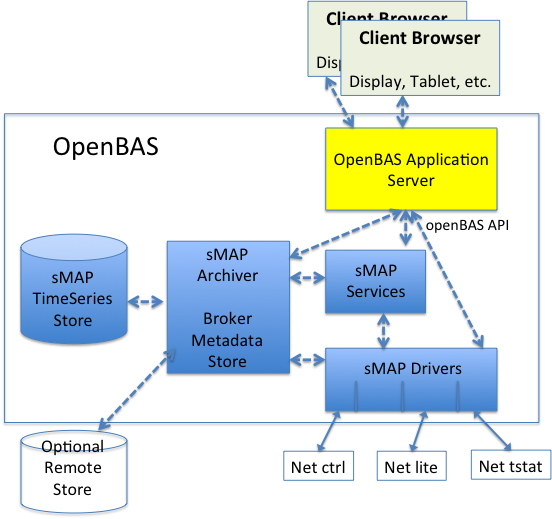


Figure OpenBas RESTfull WebService Architcture

In the DOE 822 OpenBAS implementation for small to medium scale commercial buildings, all five of the components execute within the small, embedded linux box, typically on premises. However, the system is designed to scale up and down. Each of the five components can potentially run on distinct computer systems connected via the internet. For example, only the drivers might run on premises on a very small embedded device, while the rest might be on a facilities server. Or parts, such as the Timeseries store or the app server might run in the crowd.

A typical installation might be as in Figure 2 where the OpenBAS server is simply another device on the private LAN/WLAN along with the various devices it controls, such as WiFi thermostats, Ethernet thermostats, networked lighting controllers, gateways to various other networks, and so on. All of the open source code described here runs on the embedded linux box, as does the OpenBAS application.



Figure OpenBAS running on premises within an existing private LAN/WLAN

Another common deployment strategy would be to utilize the OpenBAS server as rooting the network of HVAC, lighting and general controller units within the building, optionally connected to other networks, as Shown in Figure 3.



Figure OpenBAS forming a private Building Area Network

Github.com/SoftwareDefinedBuildings/OpenBAS

User: OpenBASGuest

Pass: openbas822project

PlexiBAS: <http://128.32.37.44:3000/>

CIEE: <http://169.229.141.11/>