

digitalSTROM virtual device container overview

digitalSTROM

Version: origin/master-branch*

May 27, 2015

*Revision: 92f6614dc98b8d73a5e80d43763692f73a81100d

©2015 digitalSTROM AG. All rights reserved.

The digitalSTROM logo is a trademark of the digitalSTROM. Use of this logo for commercial purposes without the prior written consent of digitalSTROM may constitute trademark infringement and unfair competition in violation of international laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. digitalSTROM retains all intellectual property rights associated with the technology described in this document. This document is intended to assist developers to develop applications that use or integrate digitalSTROM technologies.

Every effort has been made to ensure that the information in this document is accurate. digitalSTROM is not responsible for typographical errors.

digitalSTROM AG
Building Technology Park Zürich
Brandstrasse 33
CH-8952 Schlieren
Switzerland

Even though digitalSTROM has reviewed this document, digitalSTROM MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT THIS DOCUMENT IS PROVIDED "AS IS", AND YOU, THE READER ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL DIGITALSTROM BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. NO DIGITALSTROM AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION, EXTENSION, OR ADDITION TO THIS WARRANTY.

Contents

1	Terms	4
2	Context	5
2.1	device types	5
2.2	device detail	6
2.3	physical device OS	6
2.4	digitalSTROM configuration UI device representation	6
2.5	Discovery	6
3	Licensing	7
4	Documentation And Sample Code	7

1 Terms

Term	Description
(logical) vDC	virtual device connector. A vDC is primarily a logical entity within the dS system and has its own dSUID. A vDC represents a type or class of external devices.
vDC host	network device offering a server socket for vdsM to connect to. One vDC host can host multiple logical vDCs, if the host supports multiple device classes.
vdSM	virtual digitalSTROM Meter. A vdSM can connect to one or several vDC hosts to connect one or several logical vDCs to the dS system.
vdSD	virtual digitalSTROM device. A vdSD represents a single device in the dS system, and behaves like a real digitalSTROM terminal block (dSD, digitalSTROM device)
vDC session	logical connection between a vdSM and a vDC host (representing one or multiple vDCs)

2 Context

digitalSTROM virtual device containers (vDC) enable the integration of IP based devices into the digitalSTROM System similar to adding digitalSTROM terminal blocks (dSD) to the system. IP based devices in the local network run a vDC to connect to the digitalSTROM System thru a virtual digitalSTROM Meter (vdSM). A vDC can host multiple devices or device classes. A vDC is even suitable to integrate a gateway to another bus- or network-technology e.g. enocean, dali asf. 3 different device types can be distinguished.

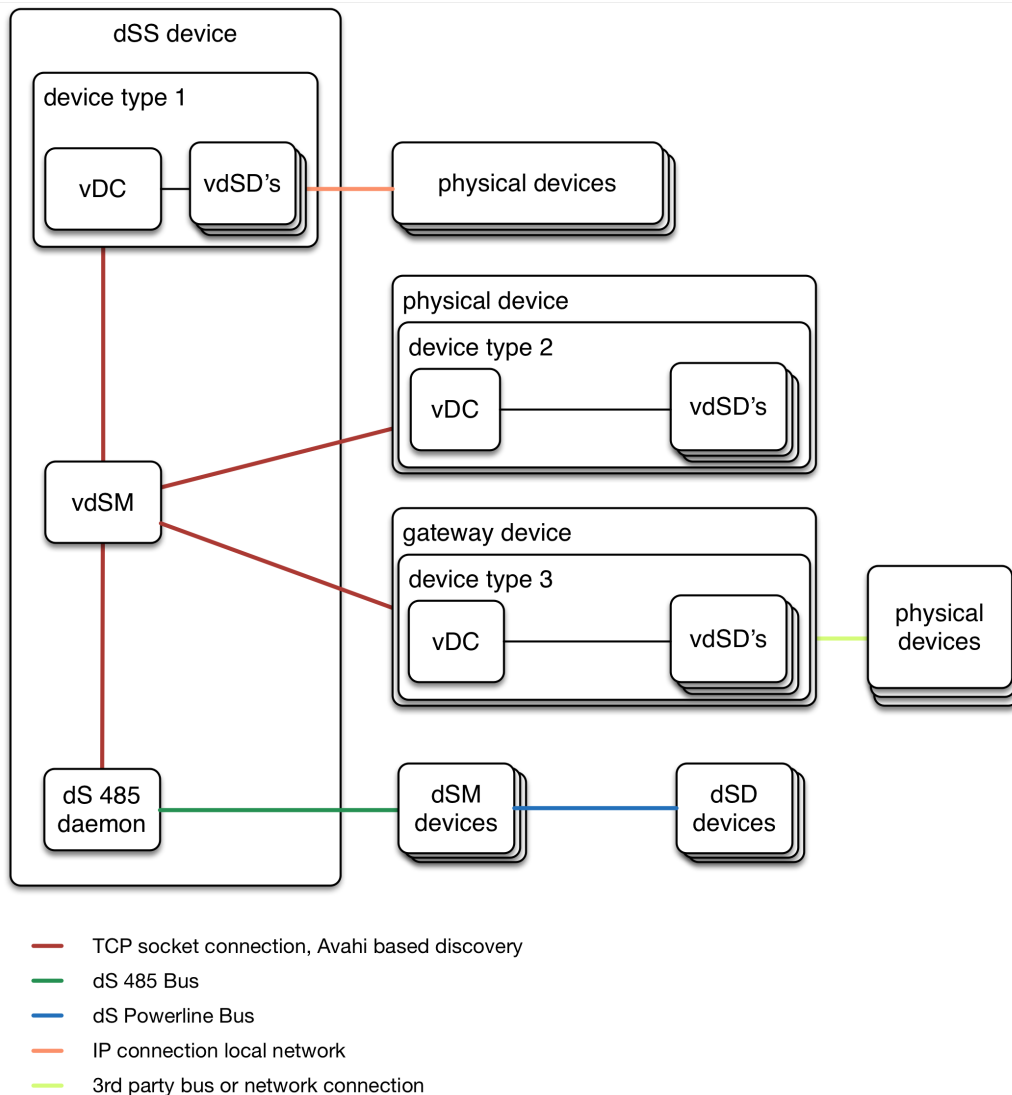


Figure 1: digitalSTROM vDC overview

2.1 device types

- device type 1 integrates devices thru the vdSM and the vDC running on the digitalSTROM Server module of the digitalSTROM installation.
- device type 2 integrates devices thru the vdSM running on the digitalSTROM Server module of the digitalSTROM installation. The vDC component is running on the physical device.
- device type 4 serves as a gateway to integrate multiple devices using bus- or network-technologies like IP, EnOcean, Zigbee, EEBUS etc.

Please get in touch with the digitalSTROM AG to define which device type is suitable for your product.

2.2 device detail

The effort to integrate devices is reduced to program the glue logic to connect the vDC API to the real devices API. Currently there are two open source projects, libdsdc and vdc, both abstracting the vDC API and offering a set of basic functionality which greatly reduces the effort to integrate devices. Depending on the type of device(s) to integrate, one or the other project might be more helpful.

- libdsdc is a lightweight C library, targeted at device vdc (device type 1+2)
- vdc is a C++ framework designed primarily for gateway devices (device type 3) integrating entire device classes with standard dS behaviour.

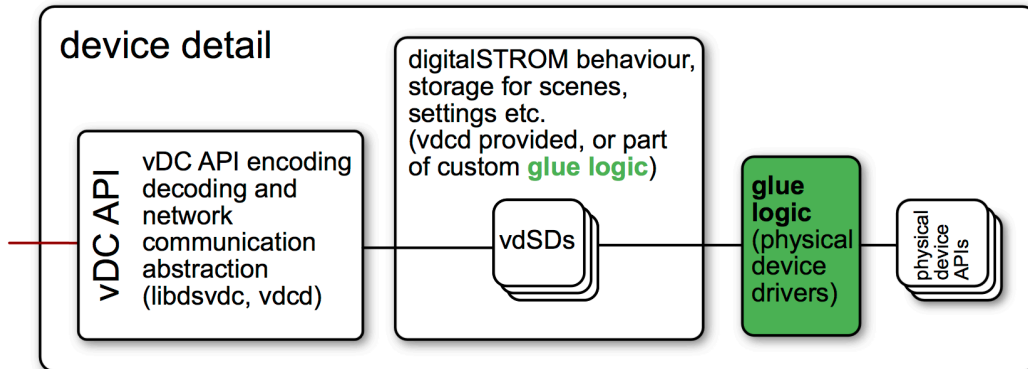


Figure 2: device detail

2.3 physical device OS

vDC is available for embedded linux. Running a vDC using another physical device OS is possible.

2.4 digitalSTROM configuration UI device representation

vDC's are displayed in the UI together with dSM devices. A vDC has a (virtual) standard room where all connected vdSD's are shown initially. vdSD's can be assigned to real rooms and behave exactly like real digitalSTROM Terminal blocks.

2.5 Discovery

vDC hosts announce their services using Avahi (gnu implementation of Apple's Bonjour)
vdSMs look for available vDC hosts in the Avahi announcements and connect to at least one of them.
vDC hosts might reject connections if they are already connected to another vdSM.
To avoid wrong connections, the vdSM which initiates a connection must be able to check possible vDC announcements received via Avahi against an optional whitelist. If the whitelist exists, only listed vDCs might be connected. The idea is that in small/simple installations connection is fully automatic, but if conflicts arise in large setups (like show rooms or developer environments) these can be solved by adding whitelists. Whitelists are located and maintained on the dss device.

3 Licensing

vDC code is available under GPL V3 . Commercial licences for vDC are available from digitalSTROM AG.

4 Documentation And Sample Code

Documentation for the vDC API and the vDC API properties as well as this document is available [here](#).

Sample code is available [in the libdsvdc library repository](#) and [in a netatmo integration project](#)