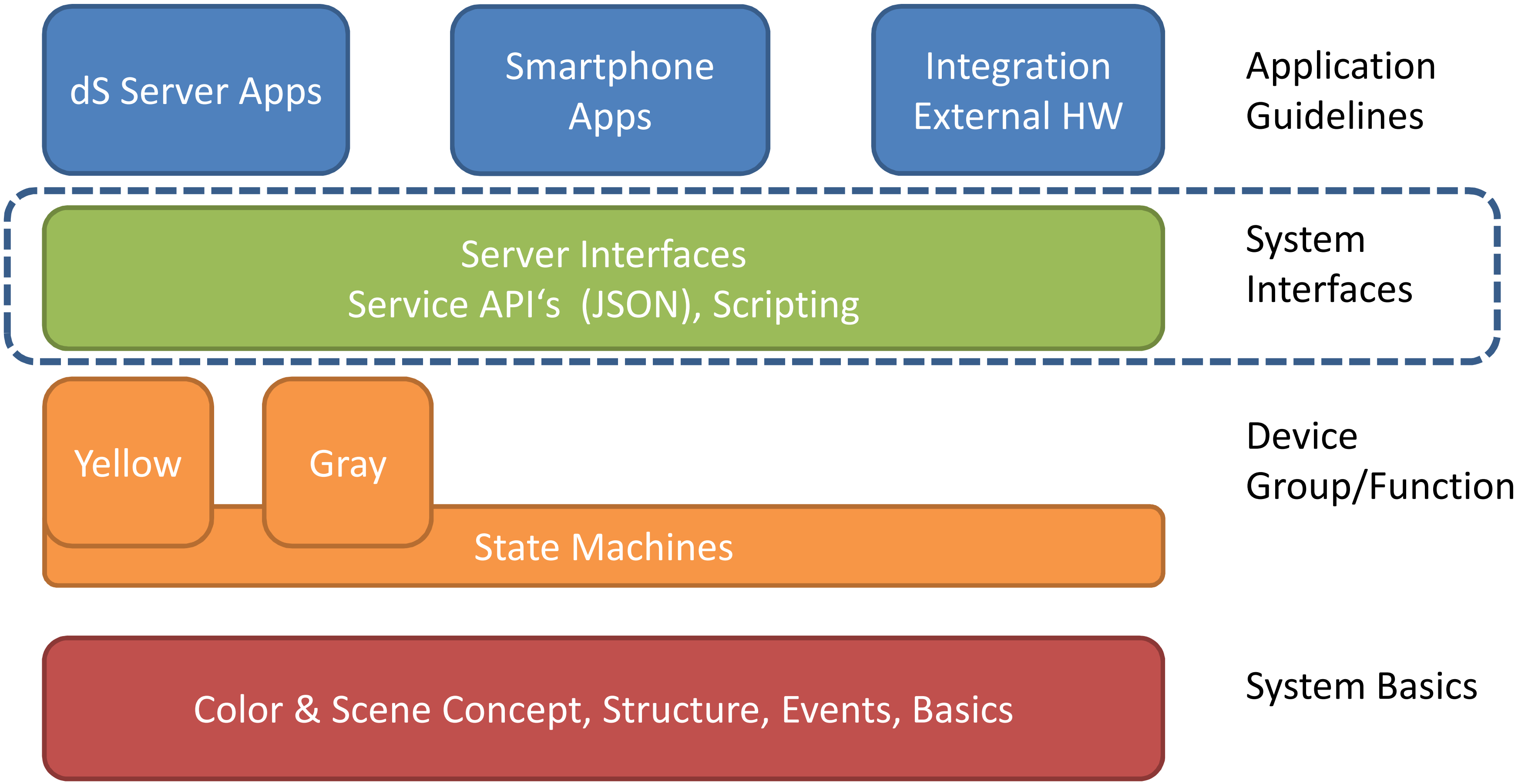
digitalSTROM

Developer Day
System Interfaces

Cornel Durrer

Sept 20th, 2012

System Documentation



dS Server Apps

Smartphone Apps

Integration External HW

Application Guidelines

Server Interfaces
Service API‘s (JSON), Scripting

System Interfaces

Yellow

Gray

State Machines

Device Group/Function

Color & Scene Concept, Structure, Events, Basics

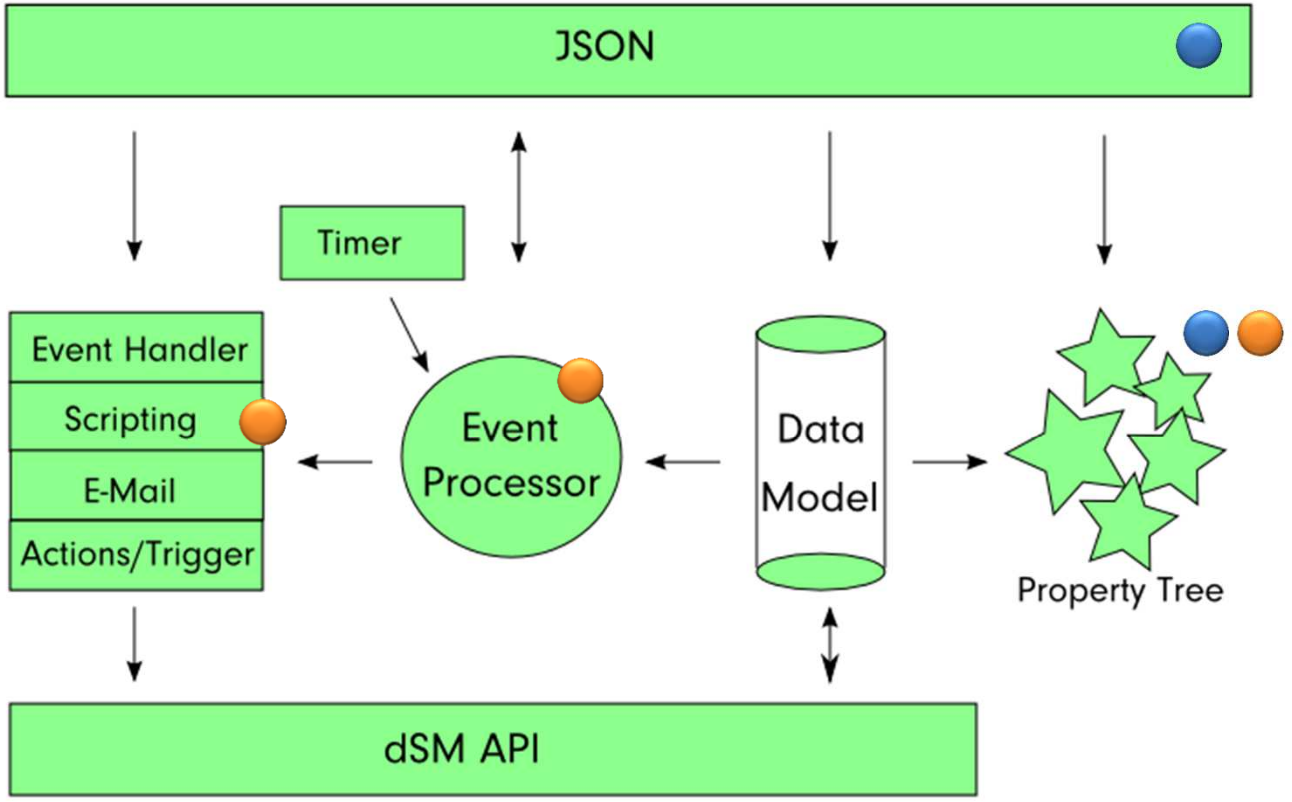
System Basics

digitalSTROM

Content

- System Interfaces
- Server App Characteristics
- Smartphone App Characteristics
- Available System Functionality
- Demo

Interfaces - Accessing the dSS



● Smartphone Apps / Web

● Server Apps

digitalSTROM

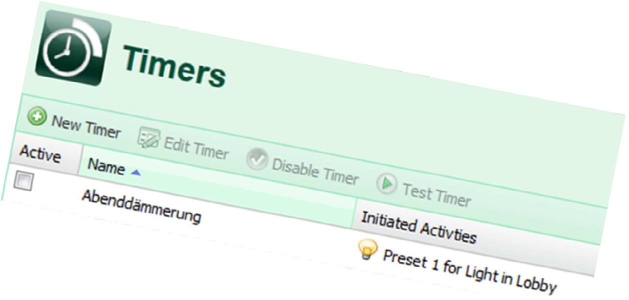
Server App Characteristics

Server Apps…

- are running locally and continuously on the dSS webserver
- have a html-based user interface stored locally on the server
- UI uses the webserver's authentication (username/password)
- can subscribe to system events and execute commands
- may read and write data from and to the property tree
- may use system functionalities like calling URLs and sending emails
- can be installed from the digitalSTROM-App-platform

Server Apps are designed to:

- Extend the digitalSTROM base functionality
- Forward local information (e.g. metering data) to cloud services



Active	Name	Initiated Activties
□	Abenddämmerung	Preset 1 for Light in Lobby

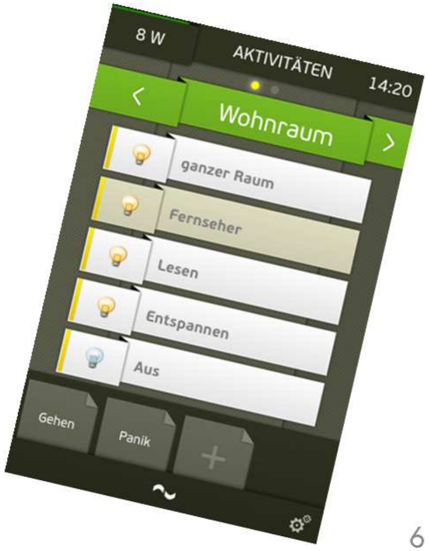
Smartphone Apps Characteristics

Smartphone Apps…

- are using the JSON interface to access the dSS
- use the personal token authentication
- can execute any commands available through JSON interface
- may read data from the property tree

Smartphone Apps are designed to:

- provide a remote user interface
  - with personalized configuration and behavior
- combine smartphone and digitalSTROM functions



Available System Functionality

Apps benefit from a broad set of functionalities provided by the dSS.

These include:

- Metering Data (power circuit and device level)
- Device Information such as configuration, EAN code, etc.
- Geo location data
- System events which can be used as a trigger (server apps only)
- Interfacing with web / cloud services
- Posting Information by E-Mail
- Logging capabilities

Metering

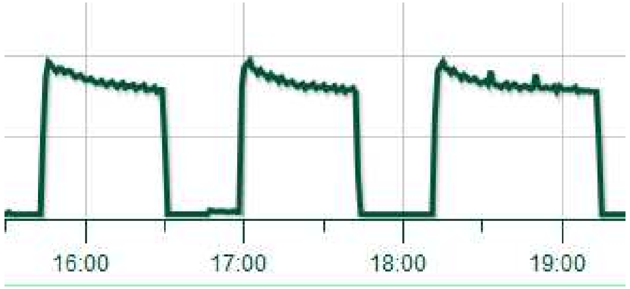
Measuring power consumption on two layers will provide full transparency to the user and flexibility for Apps:

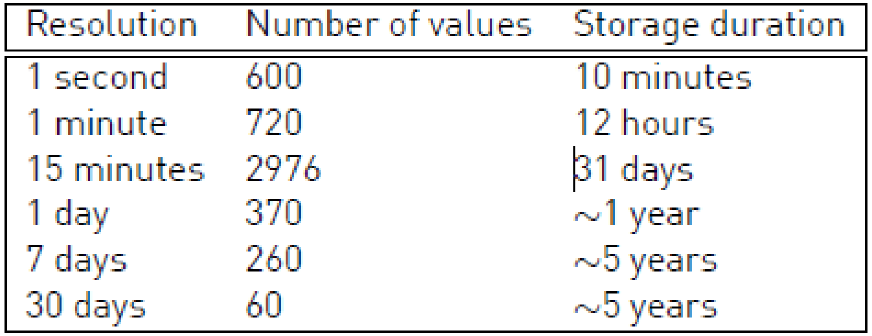
1. digitalSTROM Meter
   - Real time power consumption
   - Metering per power circuit
   - Access consumption history

2. digitalSTROM Devices
   - Get current consumption upon request
   - React to consumption events

Metering – dSM Level

- Analyze and display power-consumption and energy data in real time or by using history data.

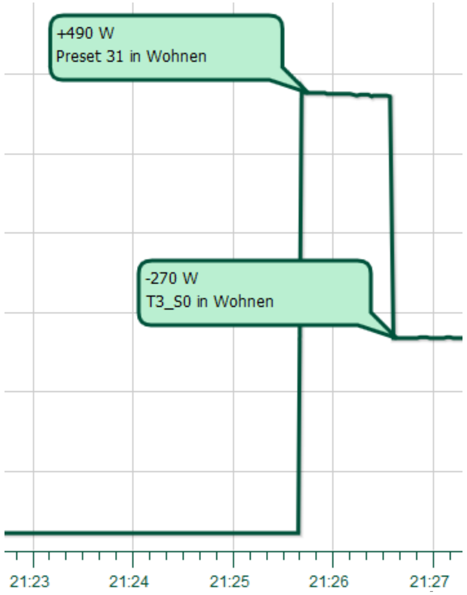




Resolution	Number of values	Storage duration
1 second	600	10 minutes
1 minute	720	12 hours
15 minutes	2976	31 days
1 day	370	~1 year
7 days	260	~5 years
30 days	60	~5 years

- Provide an even smarter analysis by combining both system events and power consumption.

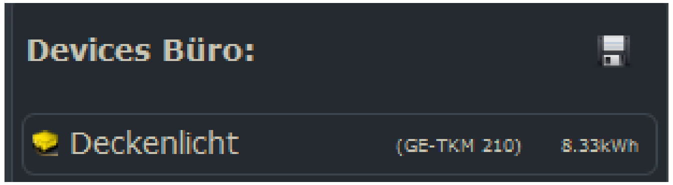
- Access metering data through property tree, JSON interface or scripting functions.



Metering – Device Level

- The current power consumption or energy counter* can be read from each device with a limited frequency.

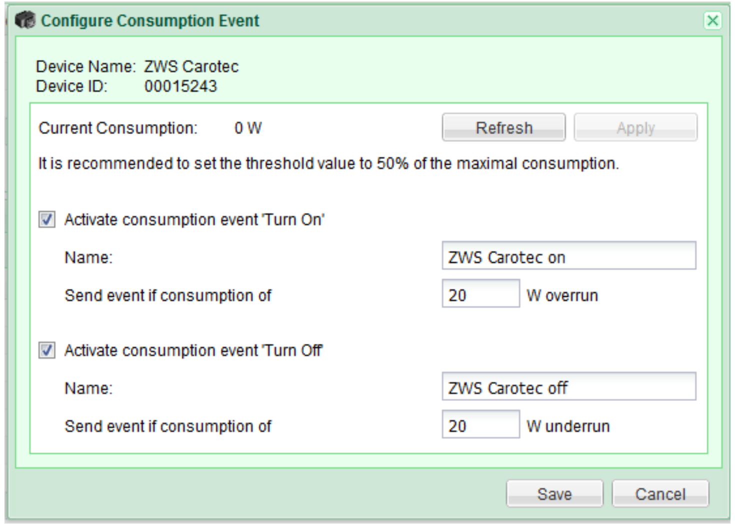
* Measured since last power cycle



Devices Büro:

Deckenlicht (GE-TKM 210) 8.33kWh

- Consumption events enable apps to executing actions or monitoring devices.
  - Two separate events will be raised when over- or under-running the threshold.
  - Available on SW-KL/ZWS



Configure Consumption Event

Device Name: ZWS Carotec
Device ID: 00015243

Current Consumption: 0 W

It is recommended to set the threshold value to 50% of the maximal consumption.

☑ Activate consumption event 'Turn On'

Name: ZWS Carotec on

Send event if consumption of 20 W overrun

☑ Activate consumption event 'Turn Off'

Name: ZWS Carotec off

Send event if consumption of 20 W underrun

digitalSTROM

Interfacing with Web/Cloud Services

- Apps can communicate with cloud services or web-devices by calling URLs
- Examples:
  - Trigger a webcam to take and post a picture
  - Enable / disable WiFi on Casacom hotspots
  - Upload data to cloud services

bidgely

Posting Information by E-Mail

- Outgoing email account is configured within dSS system settings
- Apps can easily send emails by using the *Sendmail Handler* and providing the recipient, subject and mail body only.
- Examples:
  - Confirmation messages when enabling or disabling features like presence simulation
  - Remote monitoring of devices by tracking on/off states
  - Forward emails to SMS services for notifying emergencies

digitalSTROM

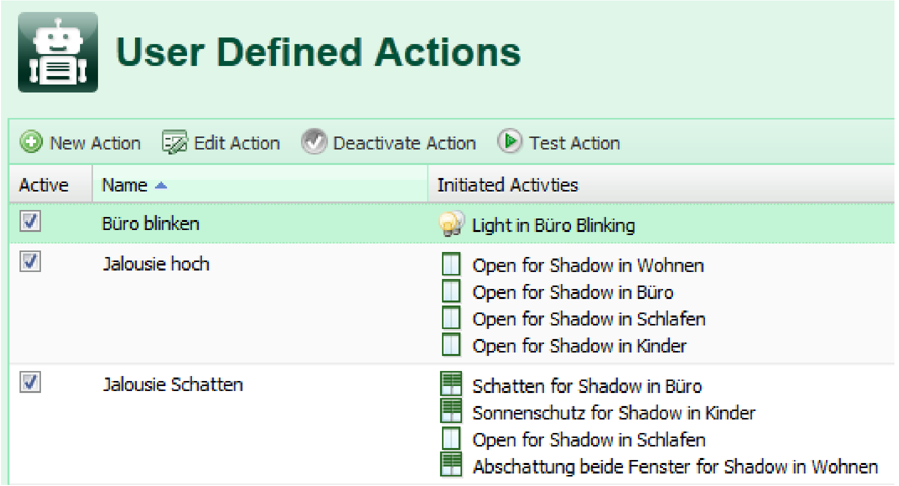
Flexibility through App-Buttons

- App-Buttons provide full flexibility without any predefined behavior.
- Customized actions can be configured for 1x – 4x clicks and the long-click
- Actions can be defined either in the Event-Responder or in your own app
- Examples:
    - Operate all shades from one central app-button
    - Call any preset in any room
    - Send an email message
    - Trigger another device by calling a URL

digitalSTROM

Automate with User Defined Actions

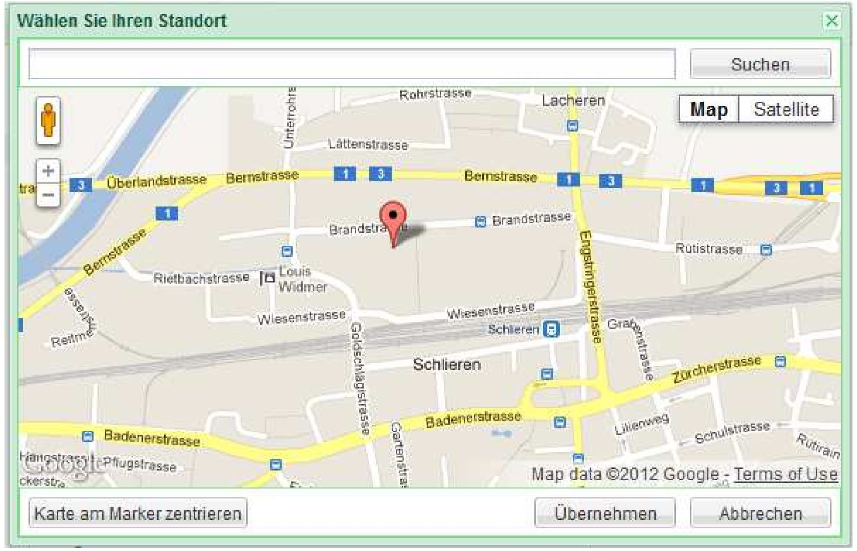
- User Defined Actions can be configured to execute repeated and commonly used activities
- They can be called from other apps, smartphones or app-buttons
- UDA's are an easy way to automate your home, e.g. to:
  - Switch on all the radio, coffee machine and light as well as open the shades when having breakfast
  - Switch off all the lights on the first floor
  - Open all shades



Active	Name	Initiated Activties
☑	Büro blinken	Light in Büro Blinking
☑	Jalousie hoch	Open for Shadow in Wohnen Open for Shadow in Büro Open for Shadow in Schlafen Open for Shadow in Kinder
☑	Jalousie Schatten	Schatten for Shadow in Büro Sonnenschutz for Shadow in Kinder Open for Shadow in Schlafen Abschattung beide Fenster for Shadow in Wohnen

Using Geo Location

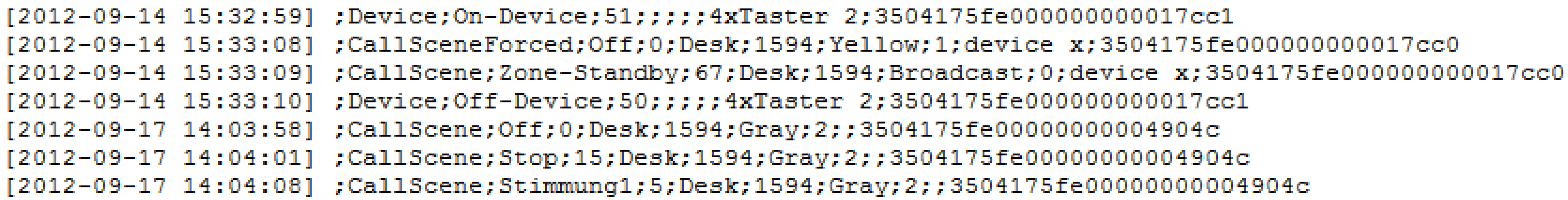
- The geographical location is known to the dSS
- Astronomical data like sunrise, sunset, dawn etc. are available from the property tree */config/geodata* and will be updated automatically
- Using this data, location based services like weather apps can be implemented



Learning from the Logfiles...

Logfiles are a great tool not only for debugging purpose, but to learn about the system behavior.

- all logfiles are located in */var/logs/* and its subfolders
- dSS-process logfiles are placed in */var/log/dss*
- Server apps can store their logfile in */var/logs/dss/js*
- System events are logged in */var/logs/dss/js/system-event.log*
  - Includes events like SceneCalls, ButtonClicks, Sensor Events etc.
  - Data allows detailed analysis of all (user-) activities within the system



```
[2012-09-14 15:32:59] ;Device;On-Device;51;;;;;4xTaster 2;3504175fe000000000017cc1
[2012-09-14 15:33:08] ;CallSceneForced;Off;0;Desk;1594;Yellow;1;device x;3504175fe000000000017cc0
[2012-09-14 15:33:09] ;CallScene;Zone-Standby;67;Desk;1594;Broadcast;0;device x;3504175fe000000000017cc0
[2012-09-14 15:33:10] ;Device;Off-Device;50;;;;;4xTaster 2;3504175fe000000000017cc1
[2012-09-17 14:03:58] ;CallScene;Off;0;Desk;1594;Gray;2;;3504175fe00000000004904c
[2012-09-17 14:04:01] ;CallScene;Stop;15;Desk;1594;Gray;2;;3504175fe00000000004904c
[2012-09-17 14:04:08] ;CallScene;Stimmung1;5;Desk;1594;Gray;2;;3504175fe00000000004904c
```

Short Demo...