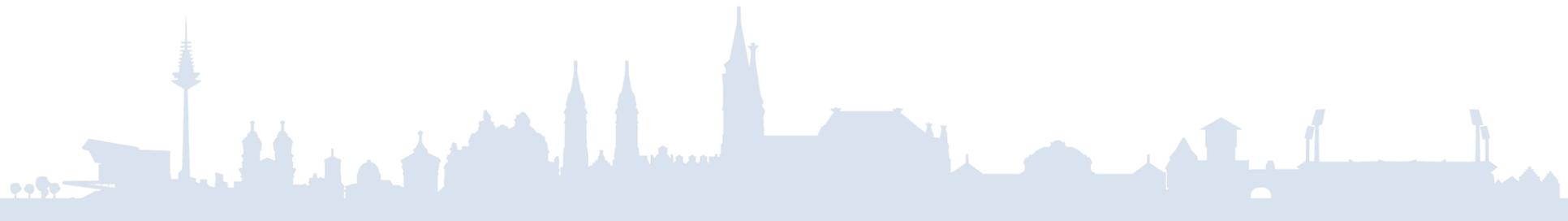




# Anwendertage 2019

News, Tipps & Tricks





- Sprachumschaltung für Mobile
- Elvis Viewer Fernzugriff mit Fritzbox
- SequenzConfigControl Mobile
- MQTT (IoT-Protokoll)
- Elvis Designer:
  - Auslagerungen von Resource-Dateien



- T\* vor jeden zu übersetzenden Text setzen  
→ z.B.: OnText="T\*Ein" OffText="T\*Aus"
- Projekt → Übersetzung Synchronisieren  
→ OnText="{Binding Application.Texts[Ein],  
FallbackValue=Ein}"
- Erzeugt ApplicationText.csv

# Sprachumschaltung



- UTF-8 codierung  
→ Bearbeitung z.B. mit OpenOffice Calc
- Im Projektordner abspeichern

id	de	en	zh
Ein	Ein	On	开
Aus	Aus	Off	熄
Heizung	Heizung	Heating	暖气
Licht	Licht	Light	灯
Alarm	Alarm	Alarm	报警

## Beispiel

Ein

Deutsch

Aus



Raum

# Elvis Viewer Fernzugriff

---



- Welche Varianten gibt es...
  - ISE Portal
  - VPN (unter Android auch automatisiert möglich mit Tasker)
  - Freigabe über Router

# Fernzugriff mit Fritzbox

---



- Myfritzfreigabe auf Elvis
- Nat loopback einrichten auf Myfritzadresse/ DynDns
- Portfreigabe auf 31082 oder gewünschten

# Elvis Viewer Fernzugriff



- Elvis Viewer

- https

- öffentliche IP ( MyFritz)

- Benutzername PW

# SequenceConfigControl



- Verfügbar für Mobile
- Projektierung gleich wie im Terminal
  - Aktuell nur Buttons möglich

Instanz1 			
	Zeit	Wert	Add
01: 00:00:02 True, 00:00:08 False 			
	00:00:02		
	00:00:08		
			
02: 00:00:04 True, 00:00:08 False 			
03: 00:00:06 True, 00:00:08 False 			



## Auslagern von Ressourcen

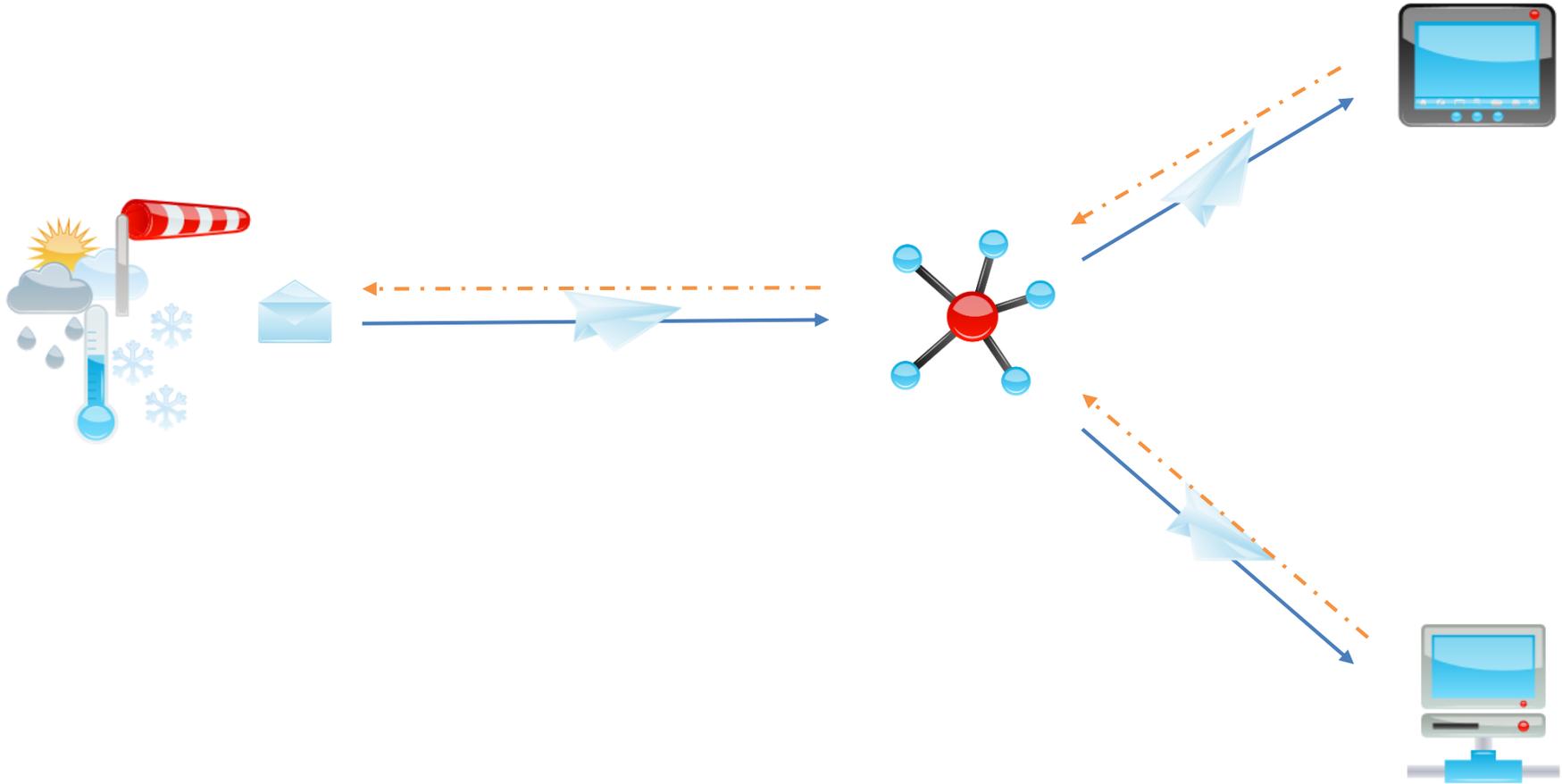
Beim Bearbeiten von Seiten aus einem sehr großen Projekt (z.B. 1800 Seiten und 3000 Ressource-Dateien) kann die Performance sehr schlecht werden.

Abhilfe:

- „Werkzeugkastenelemente automatisch ausfüllen“ deaktivieren  
→ ElvisDesigner=>Tools=>Optionen=>Text-Editor=>XAML=>Sonstiges=>Werkzeugkasten
- Ressource Dateien organisieren:  
Die Performance („XAML-Ansicht“) lässt mit der Anzahl der Dateien stark nach.  
→Ressource-Dateien in mehrere Unterordner aufteilen

- 
- IoT-Protokoll
  - Anbindung verschiedenster Geräte an Elvis
  - Wie funktioniert ... ?

# MQTT Broker (Server)





# MQTT Broker

---

- Server
- Hält je nach Konfiguration mehrere 1000 Client-Verbindungen
- Empfang, Filterung und Verteilung der Nachrichten
- Authentifizierung und Autorisierung von Clients



- Publish/Subscribe
- Topics and subscriptions
- Quality of service levels
- Retained message
- Clean sessions
- Will message



# Topics

---

- Nachrichten werden auf Topics veröffentlicht
- Aufbau Topic  
z.B.: EG/Wohnzimmer/Sensor/Temperatur
- Ein Client abonniert bestimmte Topics oder benutzt eine „Wildcard“



# Wildcards

---

- Bestimmtes Topic:  
EG/Wohnzimmer/Sensor/Temperatur
- Single-Level-Wildcard  
EG/+ /Sensor/Temperatur
- Multi-Level-Wildcard  
EG/#
- Kombination von Wildcards  
EG/+ /Sensor/#

# Quality of Service QoS

---

- QoS = 0: Nachrichten werden genau einmal versendet ohne Bestätigung
- QoS = 1: Nachricht wird mindestens einmal gesendet.
- QoS = 2: Nachricht wird exakt einmal gesendet.  

 → Zweistufige Empfangsbestätigung



# Quality of Service

- Das gesamte QoS ist immer gleich dem niedrigsten QoS des Publishers oder Subscribers

QoS Pub	QoS Sub	Gesamt
0	0, 1, 2	0
1	0	0
1	1, 2	1
2	0	0
2	1	1
2	2	2

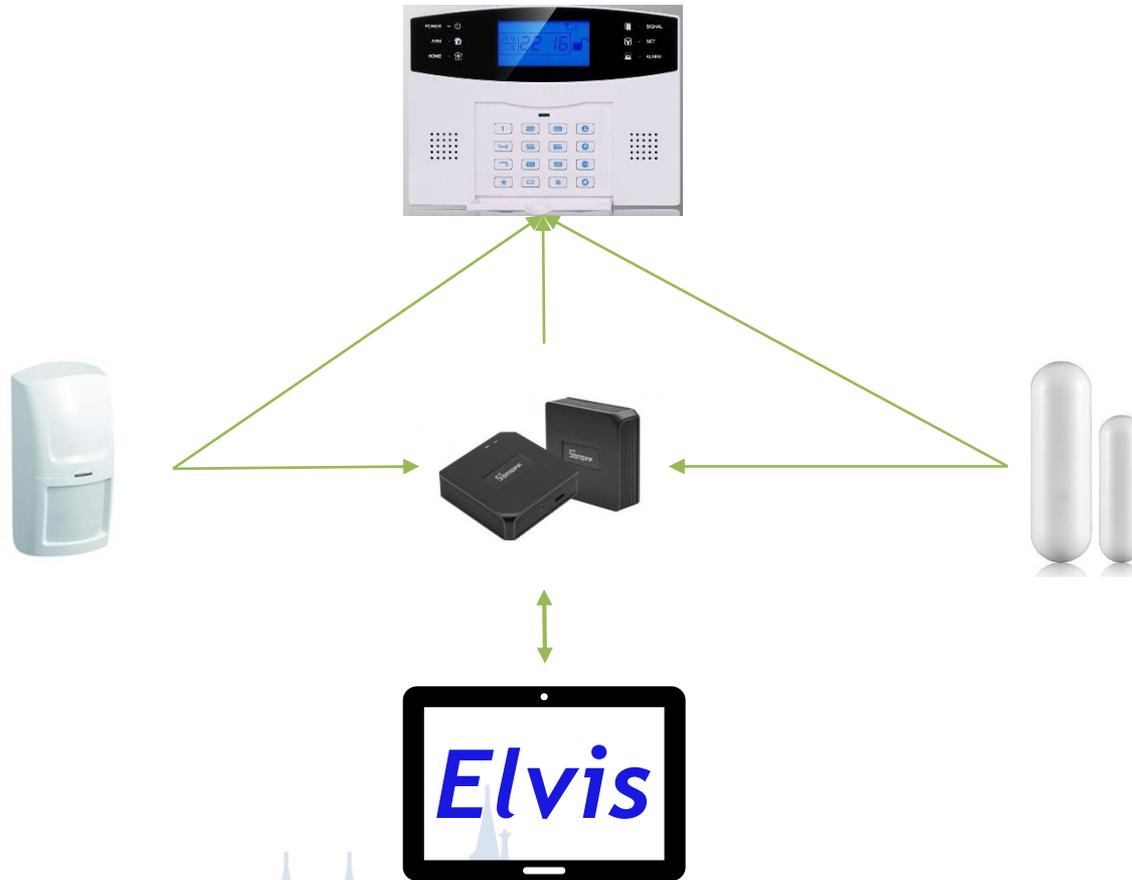


- Anbindung von Liegenschaften
  - Shelly IoT-Geräte
  - Einbindung von 433 MHz Funksensoren
- Broker und Client im Elvis Viewer direct

# Anbindung Konsumeralarmanlage



Mit MQTT:



# Projektierung MQTT



Name	Text	Treibertyp
MQTT		MQTTDriver

Prozessanschluss	
Konfiguration	
Abfrageintervall beim Hochlauf (ms)	200
Certificate Store	LocalMachine
Client Certificate Name	
Host Name	192.168.2.102
Leseabfragen beim Hochfahren p...	<input type="checkbox"/> False
Lesen-Abstand (ms)	0
Password	
Port	1883
Prefix	
Secure	<input type="checkbox"/> False
Service Configuration	bulb.xml
User Name	
Warten auf Lesebestätigung (ms)	500
Warten auf Schreibbestätigung (ms)	0
Startabfrage überspringen	<input type="checkbox"/> False
Treibertyp	MQTTDriver

Name	Text	Datenpunkttyp	Prozessanschluss
Temperatur		String	MQTT
Feuchtigkeit		String	MQTT
Licht Schalten		String	MQTT
Licht Status		String	MQTT

# MQTT mit Config



- Json String kann ausgewertet werden

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<!-- Configuration file for Elvis3  
MQTTDriverConfiguration -->
```

```
<MQTTDriverConfiguration
```

```
  PayloadPattern='{ "ison":false, "mode":"{mode}", "red":"{red}", "green":"{green}", "blue":"{blue}", "white":0, "gain":"{gain}", "temp":"{temp}", "brightness":"{brightness}", "effect":0}'
```

```
  ContentType='json'
```

```
/>
```

# MQTT: Json Beispiel



- **Original:**

```
{...“mode“:color;...}
```

- **In xml Config**

```
{...“mode“:“{mode}“, ...}
```

- **Prozessadresse:**

```
shellies/shellybulb-3CC597/color/0/set/mode  
-W---;
```

# MQTT: Json Beispiel

---



- *Prozessadresse:*  
*tele/sonoff/RESULT/RfReceived/Data/6471*  
*--/--;*

# MQTT: Json Beispiel



- Hier geht noch mehr!
- PayloadPattern=  
`'{"RfReceived":{...,"Data":"{Data}",...}}'`
- :"*{Data}*" → 6471EB / E7 / EB  
*Open / Close / Manipulate*  
→ Prozessadresse → Converter



- Performance Mobile:
  - Verwendung von jpeg
  - nur sichtbare Teile werden erzeugt
- ProcessValueConverter

# Verwendung von .jpeg



- Jpeg wird nicht mehr in png konvertiert

## Vorteil:

- Deutlich kleiner bei gleicher Auflösung (png 10 fache Dateigröße)
- Schnelleres Laden von Bedienseiten

## Nachteil:

- z.B. keine Transparenz möglich mit jpeg

# Nur sichtbares wird geladen



Problem :

- Bedienseiten mit vielen Popups  
→ Popups enthalten z.B. Parts mit mehreren Steuerelementen

→Ladezeit...? 

Abhilfe:

→ Nur Sichtbares wird geladen

Ladezeit...? 

# Nur sichtbares wird geladen



Hotelgebäude Appartmenthaus Villa de Lux ZENTRAL FUNKTIONEN

Notlichtanlagen  
Aussenbel.  
Aussenheizungen  
Dachrinnenheizung  
Wassertechnik  
Speisereste

EVG LED  
Treiber

DALI Steuergeräte ok  
Alarm

1  
2  
3  
4  
5  
6  
7  
8

Nur sichtbares wird geladen

---



# Demo





- Namespace zu ControlResources.xaml hinzufügen

```
<ResourceDictionary
```

```
  xmlns:s="clr-namespace:System;assembly=mscorlib" ...
```

...

- ConverterParameter definieren

```
<s:String x:Key="SonoffValues">0 EE-1 E7-2 EB</s:String>
```

```
<s:String x:Key="ShellyValues">0 off-1 on</s:String>
```

```
<s:String x:Key="ShellyMode">0 white-1 color</s:String>
```



Angabe des Konverters im Binding:

→ Referenz in ElvisPage notwendig

`xmlns:p=http://schemas.it-gmbh.de/ElvisPresentationExtensions`

- `<c:MultiStateControl Value="{Binding [Datenpunkte\\Sonoff\\Fensterkontakt].Value, ConverterParameter={StaticResource SonoffValues}, Converter={x:Static p:Converters.ConvertIntToSpecificString}}" .../>`

→ Einfacher: Part erstellen und dort den Konverter im Binding angeben



# Es geht einfacher ...



- Verwendung von Konvertern im Datenpunkttyp  
→ EnumeratedString



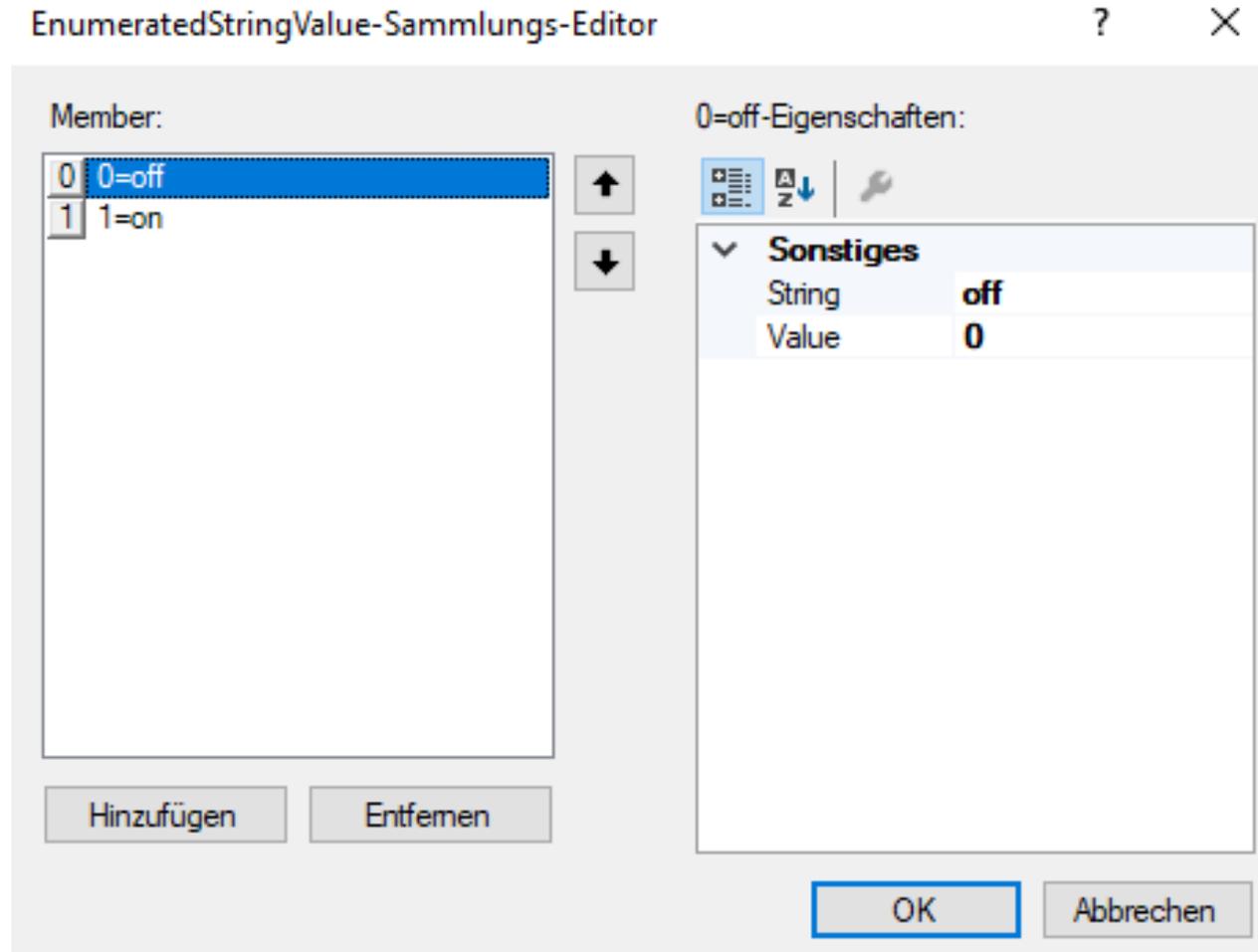
Name	Text	Datentyp	Prozess-TypInfo	Konverter
StringConvert		String		EnumeratedString

**Datenpunkttyp**

> Attribute	IsValue
Aufzählungswerte	(Sammlung)
Datentyp	String
Einheit	
Konverter	EnumeratedString
▼ Konverter-Eigenschaften	0=off; 1=on
EnumValues	(Sammlung) ...
Maximum	NaN
Minimum	NaN
Prozess-TypInfo	

- Eigenschaften Konverter

# Eigenschaften Konverter





## Vorhandene Konverter

- Enumerated String
- ScaledDouble
- ScaledInteger

# ScaledDouble



Eigenschaften festlegen:

- Factor
- Offset

• Einlaufende Richtung:

$$-ElvisWert = ProzessWert * Factor + Offset$$

• Ausgehende Richtung:

$$-ProzessWert = (ElvisWert - Offset) / Factor$$

Datenpunkttyp	
> Attribute	IsValue
Aufzählungswerte	(Sammlung)
Datentyp	Double
Einheit	
Konverter	ScaledDouble
▼ Konverter-Eigenschaften	
Factor	1
Offset	0
Maximum	NaN
Minimum	NaN
Prozess-TypInfo	



Unterscheidet sich in ausgehende Richtung:  
→ Rundung auf den nächsten ganzzahligen Wert

Beschränkung auf den **MinimumProcessValue** und/oder **MaximumProcessValue** falls vorhanden

Datenpunkttyp	
> Attribute	IsValue
Aufzählungswerte	(Sammlung)
Datentyp	Int16
Einheit	
Konverter	ScaledInteger
▼ Konverter-Eigenschaften	
Factor	1
MaximumProcessValue	
MinimumProcessValue	
Offset	0
Maximum	NaN
Minimum	NaN
Prozess-TypInfo	

<b>Datenpunkttyp</b>	
> Attribute	IsValue
Aufzählungswerte	(Sammlung)
Datentyp	Int16
Einheit	
Konverter	ScaledInteger
∨ Konverter-Eigenschaften	
Factor	1
MaximumProcessValue	
MinimumProcessValue	
Offset	0
Maximum	NaN
Minimum	NaN
Prozess-TypInfo	

