Sensio integrasjoner

1 Integrasjoner og åpenhet i plattformen

Sensio 365 er en helhetlig velferdsteknologisk plattform og kjerneløsning som de lokale løsningene kan spille sammen på. Løsningen definerer et felles rammeverk som sikrer at de ulike komponentene fungerer sammen. Integrasjoner realiseres gjennom bruk av åpne grensesnitt og standarder.

1.1 Integrasjon med andre systemer

Følgende standarder og protokoller kan benyttes for integrasjon og informasjonsflyt mellom Sensio og andre systemer:

- **REST-API**: Standard for webservices. Kan benyttes for datautveksling av bruker- og utstyrsinformasjon, samt hendelser og aggregerte data i plattformen.
- **HL7 FHIR**: Internasjonal standard for integrasjoner der helseinformasjon utveksles, i.e. EPJ og Velferdsteknologisk Knutepunkt (VKP)
- **Azure AD**: Standard protokoll for autentisering og brukerinformasjon. Brukes for å forenkle registrering og administrasjon av ansatt-brukere.
- **Open ID Connect (OIDC)**: Benyttes for autentisering av Sensios brukerflater mot Azure AD, ADFS, og andre identitetstjenester.

Sensio har også et sett med åpne API-er som er åpent tilgjengelig for 3. part som ønsker å koble seg til plattformen, der det ikke finnes en relevant nasjonal standard som kan benyttes til formålet:

- **Sensio Partner API (REST)**: Åpent REST-basert API, fritt tilgjengelig for tilkobling av 3. parts teknologi og løsninger i plattformen. Se mer informasjon nedenfor.
- **Sensio Alarm API (Web Socket)**: Sensio Alarm API er bygget på Web Sockets for utveksling av alarmer med 3. Parts system.
- **Sensio Admin API (kommer)**: Et åpent API for administrasjon av løsningen.

1.2 Integrasjon med alarm- og sensorsystemer

Sensio kan kommunisere med systemer og sensorer via anerkjente teknologier og protokoller. Sensorer kan kobles til systemet via en trådløs universalinngang eller støttede protokoller.

Nedenfor er en oppsummering av relevante protokoller som støttes:

- **SCAIP**: Internasjonal standard for integrasjon av trygghetsalarmer
- **Climax RF 869MHz**: Trådløs kommunikasjonsprotokoll som benyttes for pasientsignal, posisjonering, sensorikk m.m. både bolig og institusjon. Støtter kun Climax GX trygghetsalarmer og receivere med påkoblet utstyr.
- **ESPA444**: Seriell kommunikasjonsprotokoll for integrasjon mot brannsentral.
- **TCP/IP**: Generell protokoll for kommunikasjon med IP-basert utstyr. Benyttes i dag til f.eks. rompaneler og kamera for digitalt tilsyn.
- **SIP (Session Initiation Protocol)**: Benyttes for digitalt tilsyn og to-veis audio-visuell kommunikasjon (telefoni eller videosamtale). Bruk av standarden åpner for bruk av mange ulike løsninger så lenge SIP-protokoll støttes.

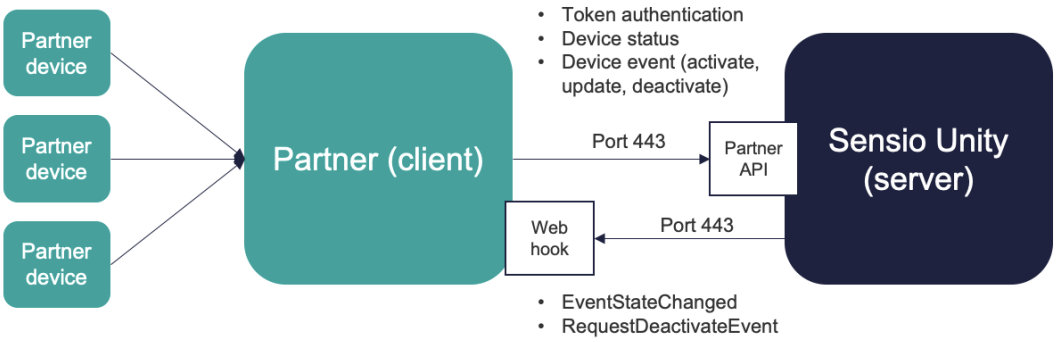
Integrasjoner kan også utvikles mot tredjepart REST API der dette er mer hensiktsmessig. Eksempler på slike integrasjoner som finnes i dag er:

- **Evondos Service API**: To-veis utveksling av hendelse- og utstyrsinformasjon.
- **Safemate API**: Skreddersydd integrasjon utviklet spesielt for samspill mellom Sensio og Safemate systemer for helhetlig og brukervennlig håndtering av mobile trygghetsalarmer, herunder administrasjon, mottak av alarmer og posisjonsdata, samt støtte for toveis tale
- **RoomMate API**: To-veis datautveksling av bruker- og utstyrsinformasjon, samt hendelser og aggregerte data i plattformen, herunder både overføring og tilpasning av alarmer og utføre proaktivt digitalt tilsyn.
- **Velferdsteknologisk Knutepunkt (VKP)**: For utveksling av informasjon med pasientjournalsystemer (EPJ).

1.3 Sensio Partner API

Sensio leverer et åpent Partner API som allerede benyttes av flere aktører, blant annet Vestfold Audio (FlexiBlink), Dignio, SensCom (DignaCare) og VitalThings (Somnofy).

Informasjon er tilgjengelig for aktører som ønsker å koble seg på, og fremgangsmåten er lagt ut Sensio sine nettsider: https://sensio.no/hvordan-integrere-med-sensios-plattform. API-et er basert på REST-API som er en anerkjent internasjonal standard for web services for integrasjon med 3. parts systemer.



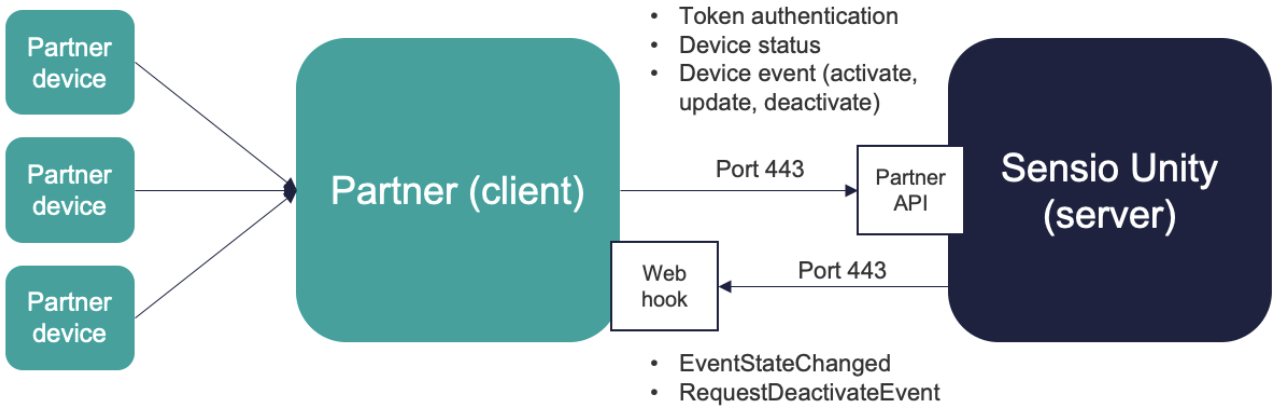
Figur 1 Sensio Partner API er åpent tilgjengelig for alle som ønsker å integrere sine løsninger i kjerneplattformen.

Sensio Partner API

Unity version	Description	Edited by	Date
6.3	Documentation of new Partner API	Mathias Lervold	10.10.2019
6.8	Added new events for supervision sensors for 6.8	Mathias Lervold	22.01.2021
6.9	Added Web hook functionality for two-way deactivation of events from Unity 6.9	Mathias Lervold	15.04.2021
6.11	• Added ExternalEventId to Webhook request • Renamed “Pill dispenser” to “Medicine dispenser” in documentation • Added new alerts to Medicine dispenser device type • Added description text as parameter in Devices POST and PUT requests	Mathias Lervold	19.10.2021
6.13.4	Code for Medicine Not Taken changed from 917 to 923	Gard Sveen	01.07.2022

Sensio Partner API allows 3rd party technology suppliers to integrate events from their devices into the Sensio Unity Platform. This is a server-server API and currently 1-way triggering of events.

From Unity 6.9 it is possible to deactivate events from Sensio to Partner via a Web hook.



API

API-documentation is available at: https://demo2.sensio.no/partner/swagger/

Most of Sensios customers have separate servers (or at least tenants), so the Partner needs to configure Sensio Unity endpoint for each customer that wants the integration (https://<lokal server>/partner/), and the customer must open up for the integration in their network from Partner server to Sensio Unity server at TCP port 443 (HTTPS).

Tokens

POST <unity partner api base>/api/v1/tokens/oauth/accesstoken is used to generate tokens for Partner authentication

- Input parameters are provided by Sensio
- The token is based onOauth2 and used in all API calls from Partner
- The token is valid for 24 hours before it must be updated

Devices

POST <unity partner api base>/api/v1/devices/{deviceIdentifier}/events/{eventType} is used to trigger events (see list below for supported devices and events)

- **deviceIdentifier** is an identifier registered in the Sensio Welfare Portal by the customer for each device - e.g. a serial number
- **triggerValue** can be used to include information with the event
- **images** - the API supports to add images to the alarm in jpeg (byte-array) format - this will be available on the event in Sensio Pocket
  - Image must be <= 50kB (50*1024*1024), JPG format
  - Max 5 images per event
  - Needs to be base64 encoded
  - Max resolution: 4096x4096 – Sensio Pocket will scale to fit
- **externalEventId** - Partner's sin own event ID - used for debugging and in the future for two-way integration
- **state** - alarm state for certain alarms that have this (see list below for supported devices and events)
- **description (Unity 6.11)** - text that can be attached to the event for presentation to the user in Sensio Pocket. Maximum 200 characters.

PUT <unity partner api base>/api/v1/devices/{deviceIdentifier}/events/{eventType} is used to update status/parameters on an active event

DELETE <unity partner api base>/api/v1/devices/{deviceIdentifier}/events/{eventType} is used to deactivate an active event

PUT <unity partner api base>/api/v1/devices/{deviceIdentifier}/status is used to update status on a device (heartbeat, battery)

PUT <unity partner api base>/api/v1/devices/status is used to update status on an array of devices (heartbeat, battery)

⚠ When passing a body be sure to not send anything unnecessary as it could create a [500] Internal server error. i.e. If imageData is not used don't pass the example body.

```
{
  "triggerValue": 0,
  "images": [
    {
      "imageData": "string", // NB! Needs to be base64 encoded
      "name": "string"
    }
  ],
  "externalEventId": "string",
  "state": 0,
  "description": "string"
}
```

Web hook (Unity 6.9)

In order for Sensio to close events from Sensio Pocket or other UIs, Partner can provide a Web hook URL and a validation token (static per integration).

Sensio may either send deactivation as a state change on the event (`EventStateChanged` - informational) or request the Partner to deactivate the event (`RequestDeactivateEvent` - command).

If the `RequestDeactivateEvent` is receieved, the Partner is asked to deactivate the alarm locally and, if successful, send DELETE message to deactivate event back to Unity server partner API.

```
DELETE <unity partner api base>/api/v1/devices/{deviceIdentifier}
/events/{eventType}
```

Partner API Web Hook POST requests

Two HTTP headers:

```
X-SensioPartner-EventName : <EventStateChanged | RequestDeactivateEvent>
X-SensioPartner-ValidationToken :  <web-hook-validation-token>
```

Web hook body content for EventStateChanged, JSON:

```
string : DeviceIdentifier
int : EventType
bool : IsActive
string : ExternalEventId (added in 6.11)
```

Web hook body content for RequestDeactivateEvent, JSON:

```
string : DeviceIdentifier
int : EventType
string : ExternalEventId (added in 6.11)
```

The Web hook needs to be configured on Sensio side with the following information needed:

- Web hook URL
- web-hook-validation-token
- Wether EventStateChanged or RequestDeactivateEvent is preferred

Web hook requst validation

Each web hook request invoked by Unity on partner back-end must be validated. The value sent in the `X-SensioPartner-ValidationToken` HTTP header must be verified agaist a value configured in partner back-end. If validation fails, the partner back-end shall return status code 403 Forbidden.

Web hook request responses

Response code	Condidition
200 Ok	Request was sucessfully processed
403 Forbidden	Authorizing validation token failed
400 Bad request	Web hooks request contains event name or event data that is not supported on invalid format
500 Internal error	Something went wrong on partner back-end while processing web hook request

Supported devices and events

The following devices and events are supported from a Partner

Device	Event	AlarmCode	State
Employee alarm	Mobile assault alarm	6	
	Fixed assault alarm	7	
	Mobile assistance	14	
Social care alarm	Social care alarm	4	
	Mobile social care alarm	11	
Motion detector	Wandering	10	
	Inactivity	901	
Door contact	Wandering	10	
	Not Closed	899	
Smoke/fire detector	Fire alarm	47	
Fall sensor	Fall	906	
Water sensor	Incontinence	907	
	Water leakage	910	
Supervision sensor	Wandering	10	
	Out of bed	12	
	Not stood up	900	

	Inactivity	901	
	Low food intake	905	
	Not in bed	908	
	Fall	906	
	Sound level alarm	903	
	Up in bed	911	
	Activity	912	
	Enter room	913	
	Out of chair	914	
	Too long gone	915	
	Absence	916	
New alarms in Unity 6.8	Respiration rate	921	
	Oxygen saturation	922	
Multi detector	Alarm	1	
	High temperature	8	
	Low temperature	9	
	Sound level alarm	903	
	Epilepsy	904	
	Stove guard alarm	909	
Medicine dispencer	Medicine Not Taken	923	
	Medicines Not Taken	918	
	Medicine Soon Empty	919	
	Medicine Empty	920	
	Medicine Taken	10014	
New alarms in Unity 6.11	Malfunction Content Container Alert	967	
	Malfunction Excess Container Alert	968	
Pull cord	Patient signal	2	AlarmCall = 50, Present = 88, Assistance = 110, EmergencyCall = 10,

The following operational alerts are supported for all devices:

Device	Event	AlarmCode
All	Offline	950
	Low battery	951
	Battery failure	952
	Power failure	954

Tamper	955
Low signal	956
Critically low battery	957
Malfunction	966
Maintenance needed	1001
Power overload	1002