Лекція 10. Загальні відомості про Azure IoT Central.

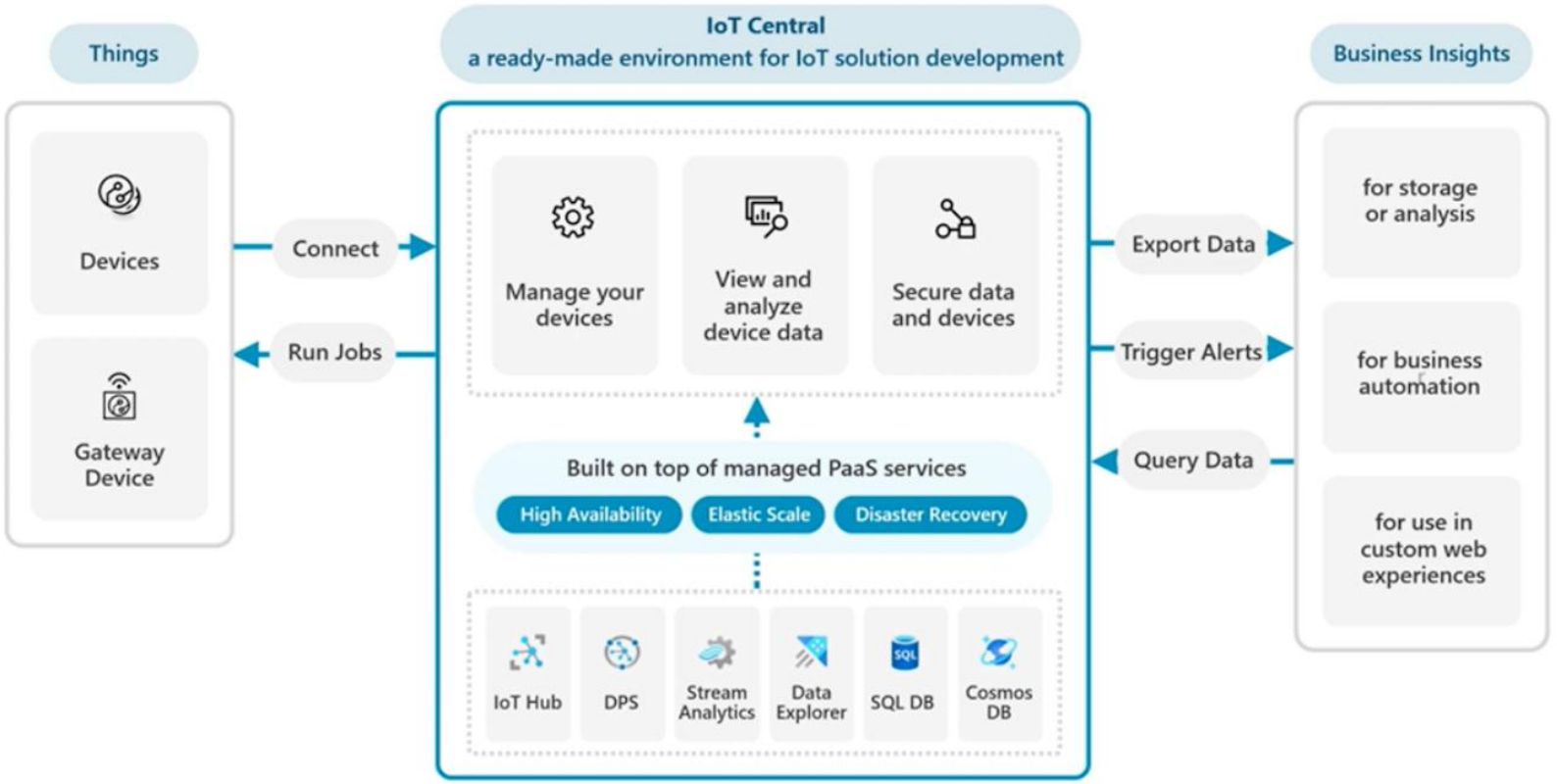
Тема 1. Структура Azure IoT Central

Тема 2. Відомості про інтерфейс Azure IoT Central

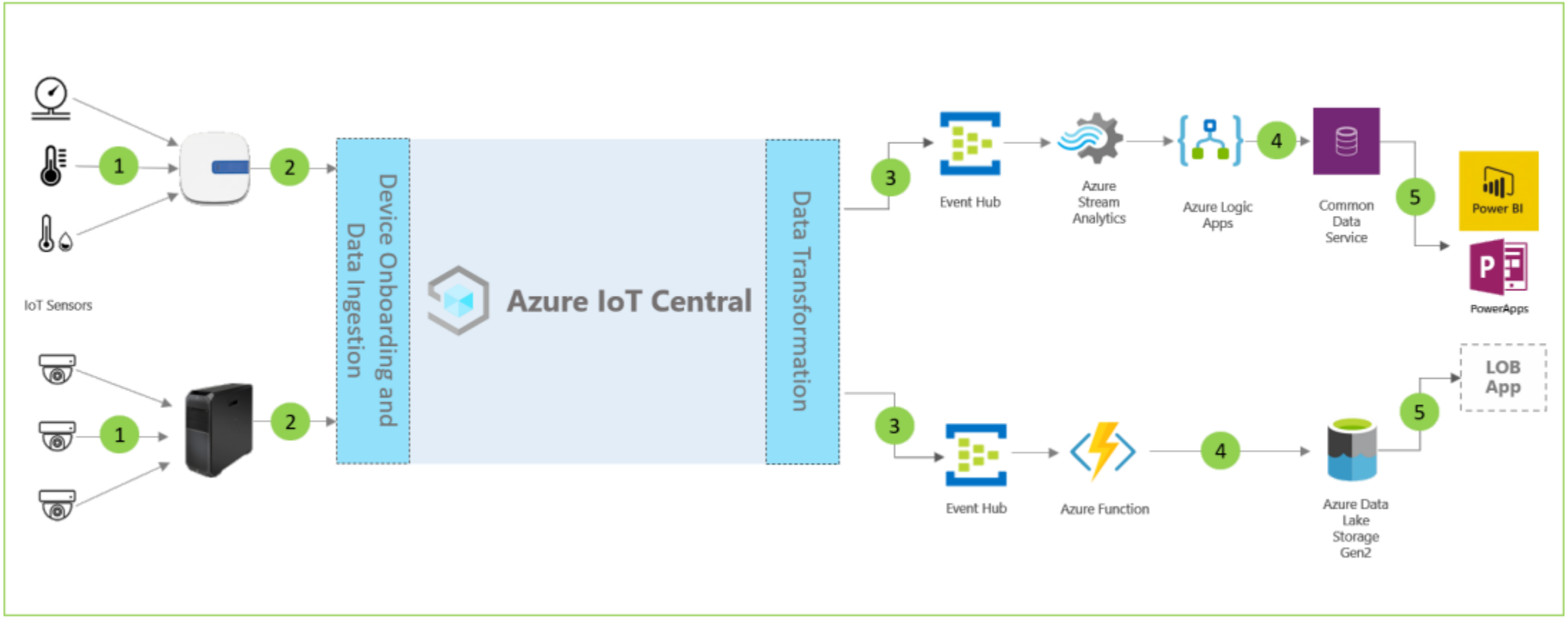
Тема 3. Архітектура Azure IoT Central

Тема 4. Керування пристроями IoT за допомогою Azure Central

Архітектура IoT Central



- Дозволяє керувати пристроями, а саме визначати шаблони для пристрою та пов'язувати пристрій з шаблоном.
- Візуалізувати потокові дані
- Перетворювати дані складних типів на структуровані
- Аналізувати дані за допомогою вбудованих інструментів аналітики
- Забезпечує безпеку даних, та пристроїв
- Функції експорту даних дозволяють отримати потоків дані для різних доданків.



(1) Підключіть різні датчики IoT до екземпляра програми IoT Central.
(2) Багато датчиків IoT можуть передавати необроблені сигнали безпосередньо в хмару або на шлюз розташовані біля них. Пристрій шлюзу збирає дані на межі перед ним надсилає підсумкові статистичні дані до програми IoT Central. Пристрій шлюзу також є відповідальний за передачу команд і контрольних операцій сенсорним пристроям коли застосовно.
(3) Створіть спеціальні правила, які використовують умови середовища для запуску оповіщення для менеджерів магазинів.
(4) Перетворення умов навколишнього середовища PaaS служби можуть виконувати додаткові функції з даними
(5) Експортуйте агреговану статистику в існуючі або нові бізнес-додатки

IoT Central Application ...

IoT central

IoT Central — це платформа додатків IoT як послуга (aPaaS), яка зменшує навантаження і витрати на розробку, управління та підтримку рішень IoT.

Basics Tags Review + create

Create an IoT Central application with an application template. IoT Central is an IoT app platform that allows you to rapidly build enterprise-grade IoT solutions on a secure, reliable and scalable infrastructure. Learn more

Project details

Select the subscription to manage the deployed IoT Central resource and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure subscription 1

Resource group * : (New) demo

Create new

Instance details

Resource name * : soloveidemo

Application URL * : soloveidemo

azureiotcentral.com

Template * : Custom application

Region * : West Europe

Pricing plan * : Standard 0

Pricing Tier	Standard Tier 0	Standard Tier 1	Standard Tier 2
Use Case	For devices sending a few messages per day	For devices sending a few messages per hour	For devices sending a message every few minutes
Price per device per month	$0.08 per Month	$0.40 per Month	$0.70 per Month
Monthly device message allocation*	400 messages	5,000 messages	30,000 messages
Included free quantities per application	2 free devices (800 included messages)	2 free devices (10,000 included messages)	2 free devices (60,000 included messages)
Overage pricing per 1K messages[1]	$0.07 per 1K messages	$0.015 per 1K messages	$0.015 per 1K messages

Управління інтерфейсом – IoT Central

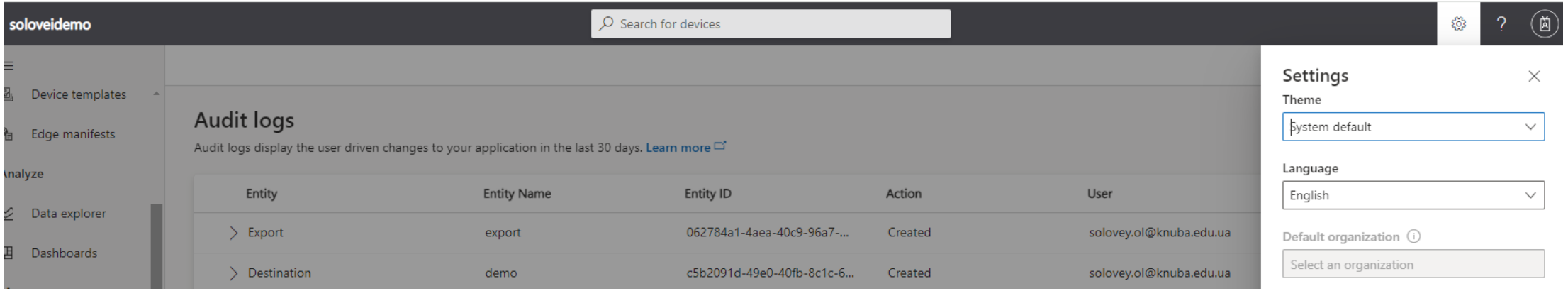
Журнал подій

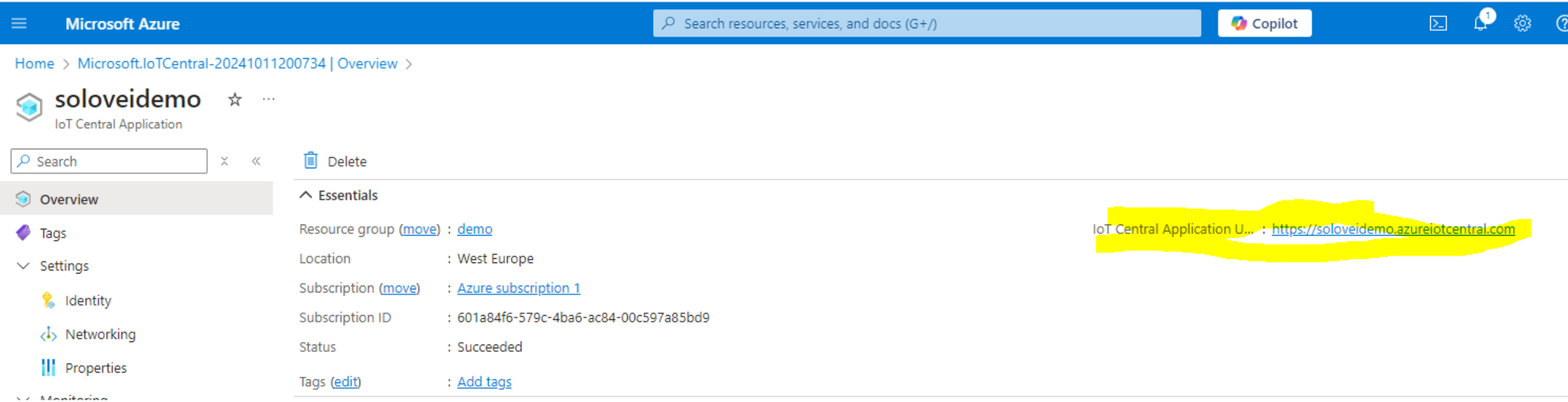
Audit logs

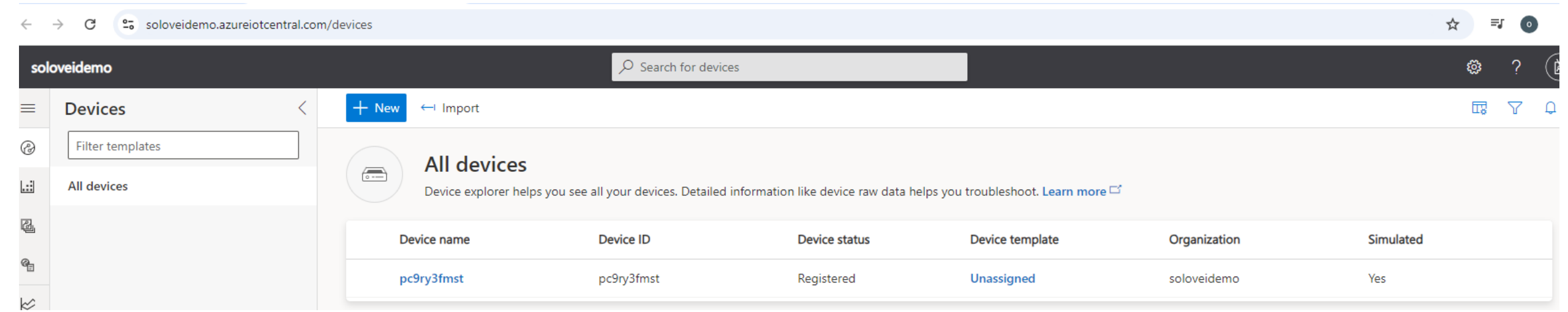
Audit logs display the user driven changes to your application in the last 30 days. Learn more

Entity	Entity Name	Entity ID	Action	User	Timestamp ↓
Export	export	062784a1-4aea-40c9-96a7-...	Created	solovey.ol@knuba.edu.ua	10/18/2024, 8:16:58 AM
Destination	demo	c5b2091d-49e0-40fb-8c1c-6...	Created	solovey.ol@knuba.edu.ua	10/18/2024, 8:16:55 AM
Device	demo2	demo2	Created	solovey.ol@knuba.edu.ua	10/18/2024, 8:15:18 AM
Device	demo	demo	Created	solovey.ol@knuba.edu.ua	10/18/2024, 8:01:27 AM
Device template	Hobo MX-100	dtmi:rigado:gm5j45ik	Created	solovey.ol@knuba.edu.ua	10/18/2024, 7:52:34 AM
Application customization	soloveidemo	1b15bc75-96ca-4772-af97-5...	Updated	solovey.ol@knuba.edu.ua	10/18/2024, 7:50:11 AM

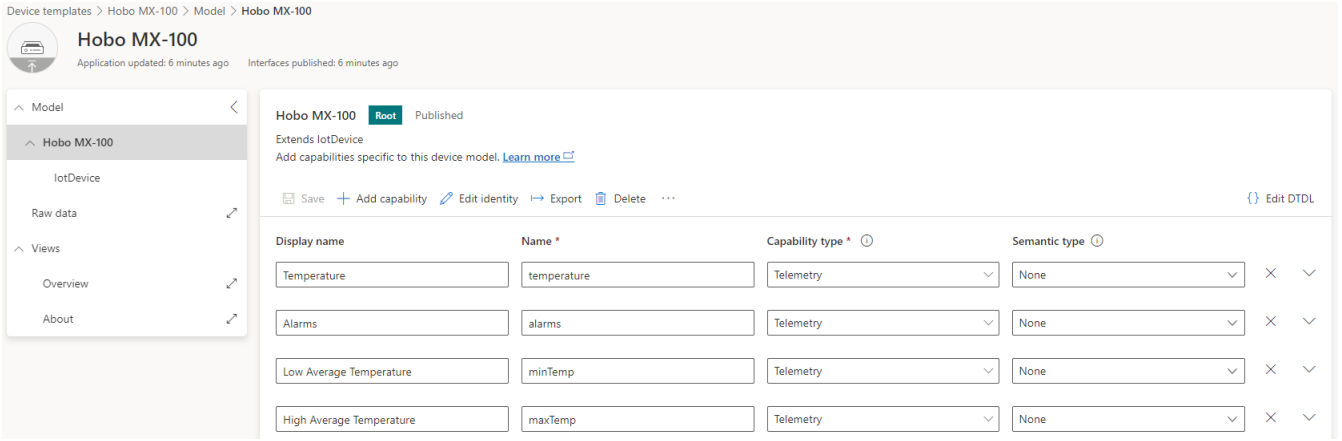
Налаштувати мову




Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Microsoft.IoTCentral-20241011200734 | Overview >
soloveidemo
IoT Central Application
Search
Overview
Tags
Settings
Identity
Networking
Properties
Delete
Essentials
Resource group (move) : demo
Location : West Europe
Subscription (move) : Azure subscription 1
Subscription ID : 601a84f6-579c-4ba6-ac84-00c597a85bd9
Status : Succeeded
Tags (edit) : Add tags
IoT Central Application U... : https://soloveidemo.azureiotcentral.com

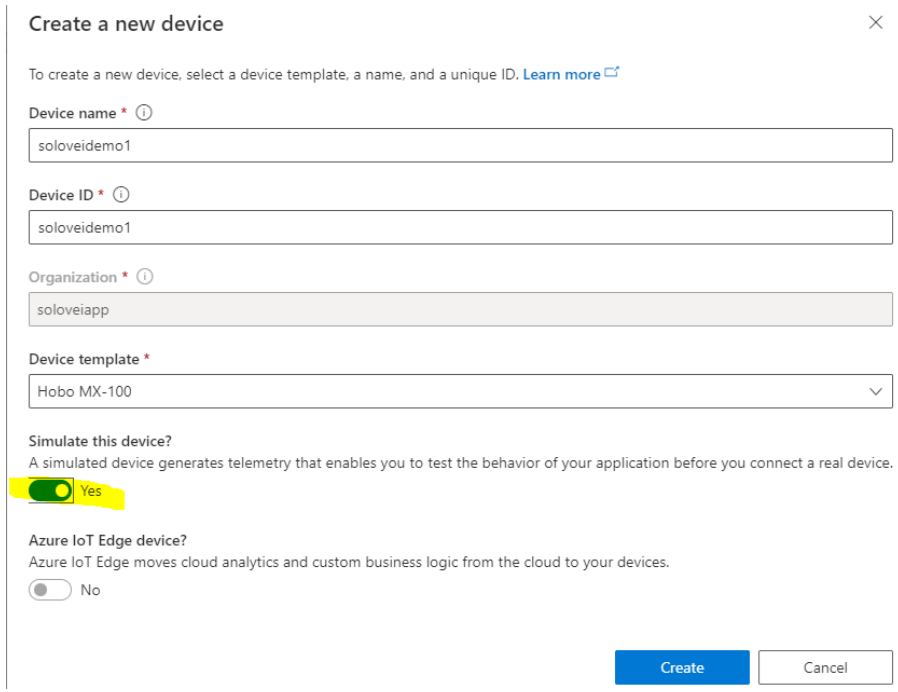

soloveidemo.azureiotcentral.com/devices
soloveidemo
Search for devices
Devices
Filter templates
All devices
New
Import
All devices
Device explorer helps you see all your devices. Detailed information like device raw data helps you troubleshoot. Learn more
Device name
Device ID
Device status
Device template
Organization
Simulated
pc9ry3fmst
pc9ry3fmst
Registered
Unassigned
soloveidemo
Yes

Шаблон пристрою



Шаблон пристрою – це проект, який визначає характеристики та поведінку типу пристрою. Приклади включають температуру та вологість. Телеметрія є потокові дані.
Бізнес-властивості, які оператор може змінити. Приклади включають клієнта адресу та дату останнього обслуговування.
Властивості пристрою, які встановлює пристрій і які доступні лише для читання в програмі.
Властивості пристрою, які встановлює оператор і які визначають поведінку пристрій. Наприклад, цільова температура для пристрою.
Команди, викликані оператором і які виконуються на пристрої. Наприклад, команда для віддаленого перезавантаження пристрою

Publish template

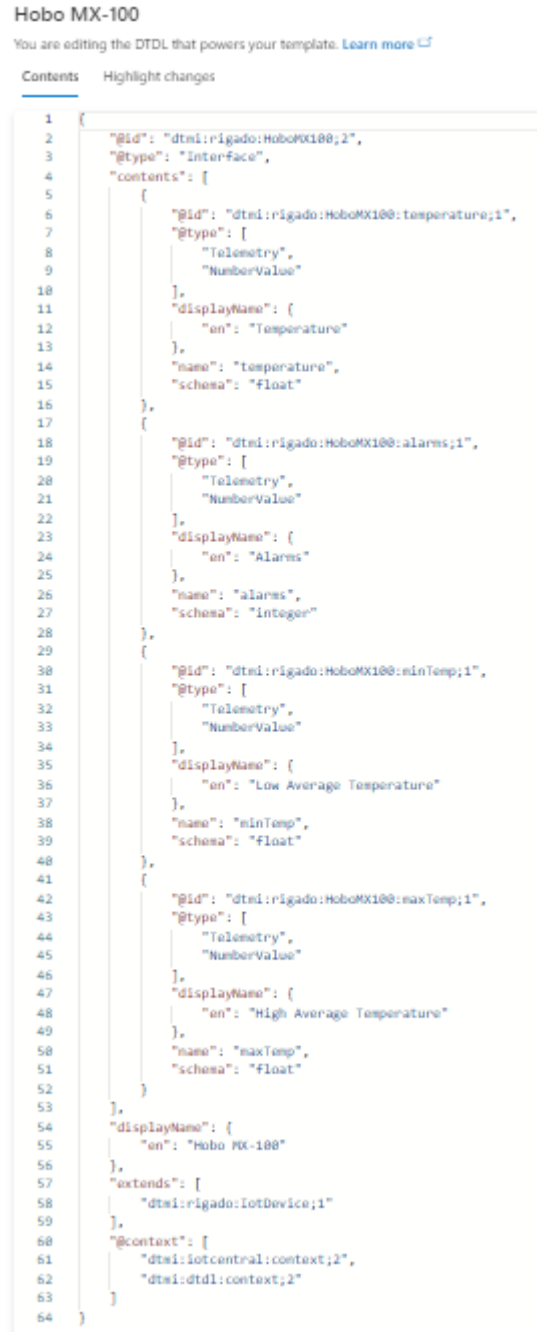


Додайте пристрій (з даними, які отримуються на основі симулятору) і зв’яжить його з шаблоном пристороою

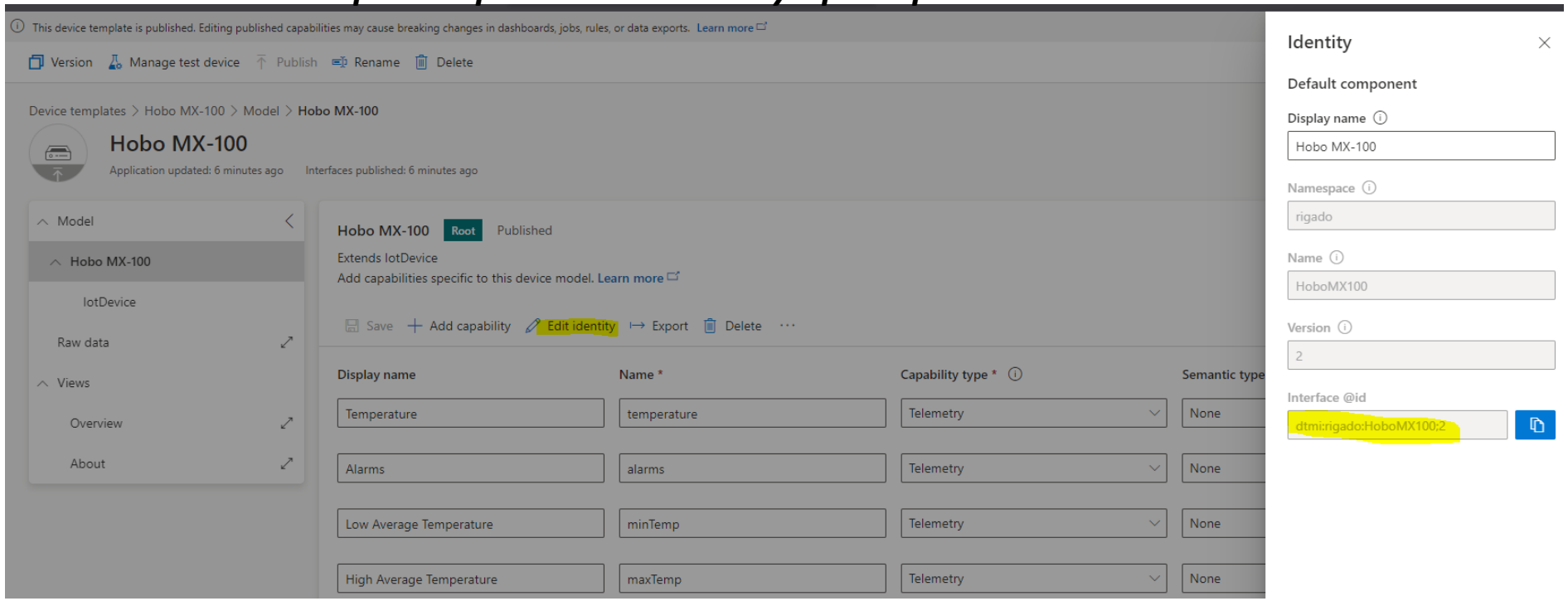
Шаблон пристрою в Azure IoT Central

DTDL (Digital Twins Definition Language) — це мовний опис цифрових двійників (Digital Twins), розроблений компанією Microsoft для роботи в екосистемі Azure Digital Twins. DTDL використовується для моделювання фізичних об'єктів та їх поведення в цифровому вигляді. З його допомогою можна описати різні аспекти сутності, включаючи їх властивості, відносини, події та команди.
Ключові аспекти DTDL:
Моделювання сутностей: DTDL дозволяє описувати об'єкти (наприклад, будівлі, машини, пристрої), які становлять частину цифрового двойника. Це робиться за допомогою класів та інтерфейсів, описуючих властивостей, методів і відносин об'єкта.
Свойства (Властивості): Свойства в DTDL — це дані, пов'язані з цифровим двійником. Наприклад, здание может иметь свойства в роде "количество поверхів" або "год постройки".
Отношения (Relationships): Це зв'язок між різними об'єктами цифрового двойника. Наприклад, зданіе може бути пов'язане з окремими приміщеннями або інженерними системами.
Команды (Команди): Команди — це дії, які можна виконувати над об'єктами. Наприклад, можна відправити команду на включення кондиціонера в зданіі.
Модель DTDL може бути безкомпонентною або багатокомпонентною:
Безкомпонентна модель: проста модель не використовує вбудовані чи каскадні компоненти. Уся телеметрія, властивості та команди визначені в одному корені компонент. Для прикладу дивіться модель Термостат (https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v2/samples/Thermostat.json)

Багатокомпонентна модель. Більш складна модель, яка включає два або більше компоненти. Ці компоненти включають один кореневий компонент і один або більше вкладених компонентів. Для прикладу - регулятор температури модель. https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDL/v2/samples/TemperatureController.json



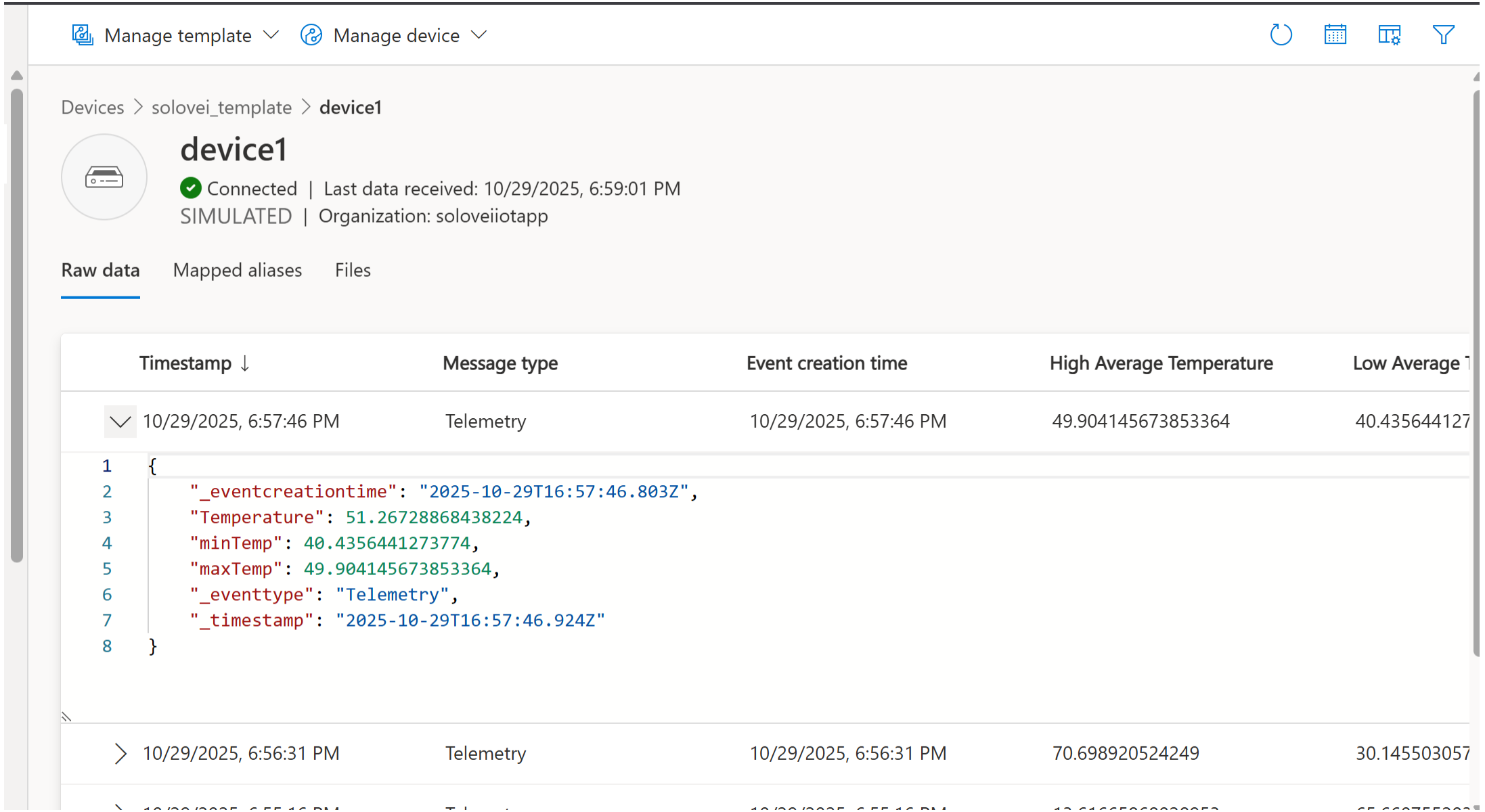
Ідентифікатор моделі шаблону пристрою в IoT Central



файл JSON визначає повну модель пристрою або окремий інтерфейс, у шаблон пристрою.
У файлі JSON, використовується мова Digital Twin Definition Language (DTDL) v2.
Моделі, створені в IoT Central, мають контекст dtmi:iotcentral:context;2, який вказує, що модель створено в IoT Central

```
"@context": [
  "dtmi:iotcentral:context;2",
  "dtmi:dtdl:context;2"
]
```

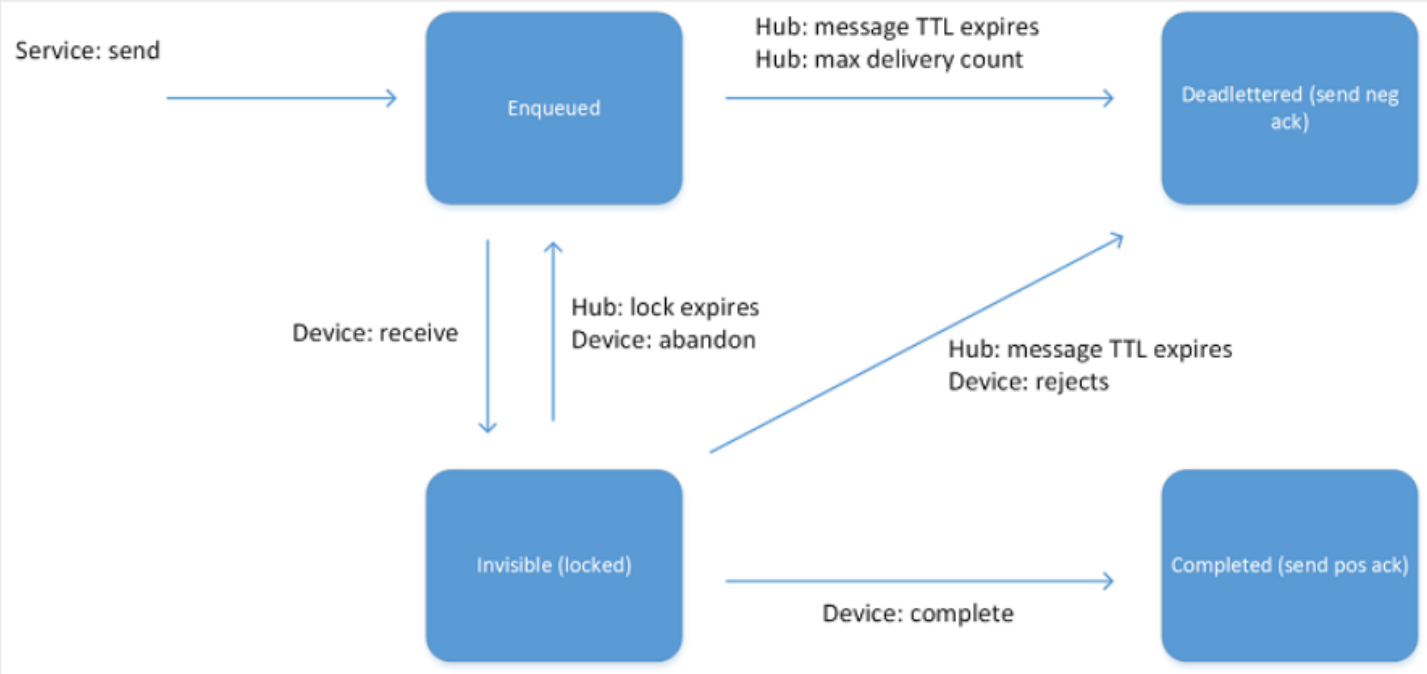
https://learn.microsoft.com/en-us/training/modules/monitor-and-manage-device-with-iot-central/3-define-coffee-machine-device-template


Manage template
Manage device
Devices > solovei_template > device1
device1
Connected | Last data received: 10/29/2025, 6:59:01 PM
SIMULATED | Organization: soloveiiotapp
Raw data
Mapped aliases
Files
Timestamp ↓
Message type
Event creation time
High Average Temperature
Low Average
10/29/2025, 6:57:46 PM
Telemetry
10/29/2025, 6:57:46 PM
49.904145673853364
40.435644127
{
"_eventcreationtime": "2025-10-29T16:57:46.803Z",
"Temperature": 51.26728868438224,
"minTemp": 40.4356441273774,
"maxTemp": 49.904145673853364,
"_eventtype": "Telemetry",
"_timestamp": "2025-10-29T16:57:46.924Z"
}
10/29/2025, 6:56:31 PM
Telemetry
10/29/2025, 6:56:31 PM
70.698920524249
30.145503057

Шаблон пристрою – Capability types

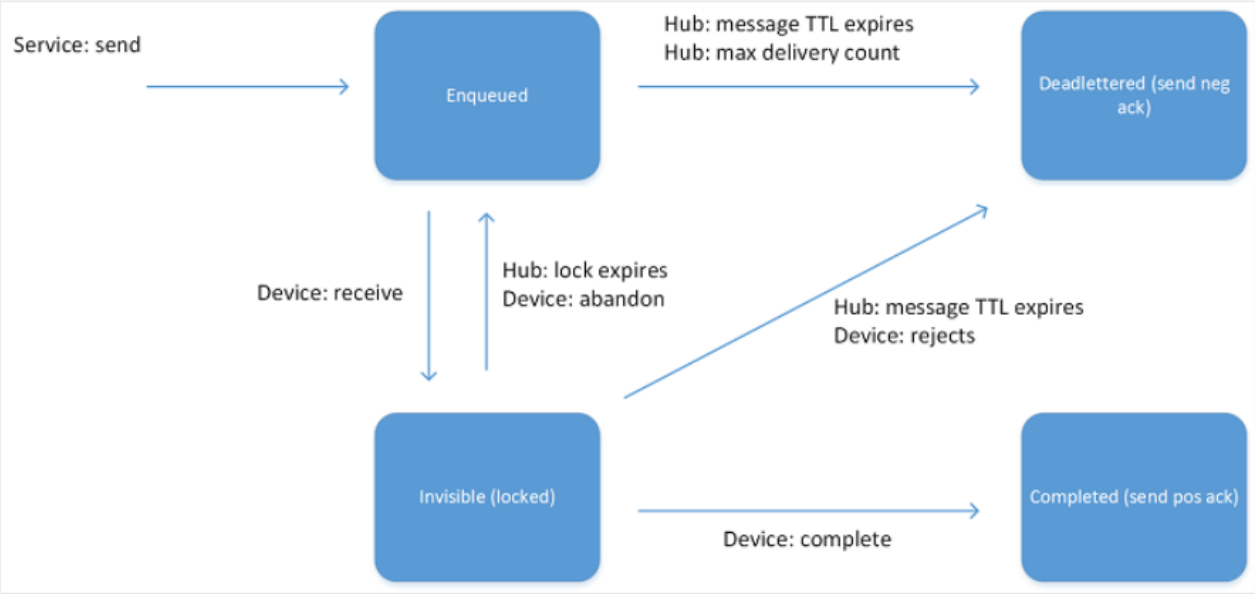
Telemetry (Телеметрія):
Телеметрія відправляє дані з пристрою в хмару. Це однонаправлений потік інформації від пристрою до хмарного сервісу.
Properties (Властивості):
Властивості можуть бути як читабельними (reported properties), так і налаштовуваними (cloud properties).
Reported properties: Це властивості, які пристрій може передавати для опису свого стану (наприклад, поточна версія прошивки).
Cloud properties: Це властивості, які можна задати на стороні хмарного сервісу для зміни конфігурації пристрою (наприклад, зміна цільової температури на термостаті).
Commands (Команди):
Команди дозволяють відправляти керуючі інструкції з хмарного сервісу на пристрій. Це двонаправлене управління.

Життєвий цикл повідомлень із хмари на пристрій



Щоб гарантувати доставку повідомлень принаймні один раз, центр Інтернету речей зберігає повідомлення з хмари на пристрій у чергах для кожного пристрою. Пристрої повинні явно підтвердити завершення повідомлення, перш ніж центр IoT видалить повідомлення з черги. Такий підхід гарантує стійкість до збоїв підключення та пристроїв.

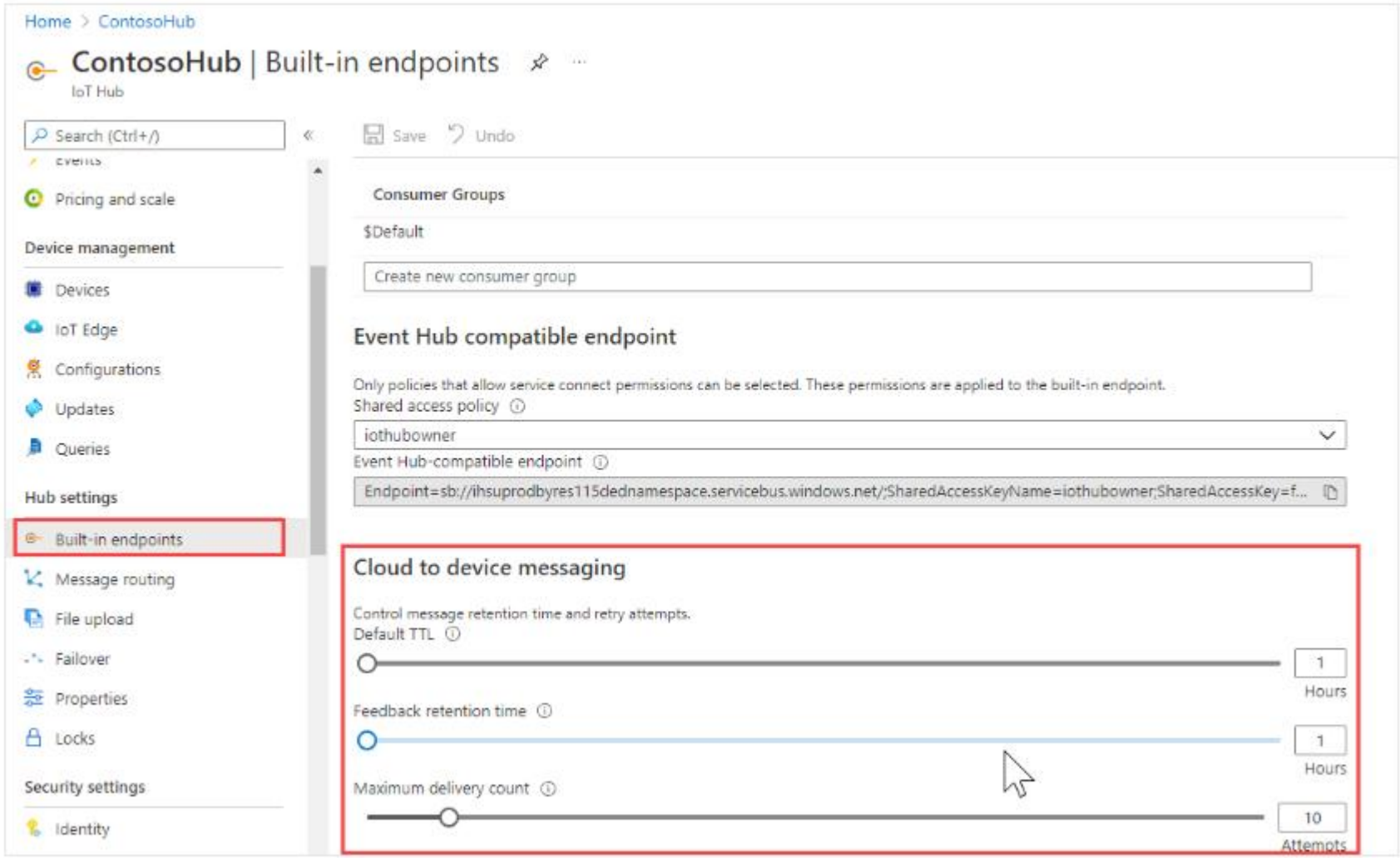
Коли служба центру Інтернету речей надсилає повідомлення на пристрій, служба встановлює стан повідомлення на Поставлено в чергу (Enqueued).
Коли потік пристрою готовий отримати повідомлення, центр Інтернету речей блокує повідомлення, установлюючи стан на Невидимий (Invible locked).
Коли потік пристрою завершує обробку повідомлення, він сповіщає центр IoT про завершення повідомлення. Потім центр IoT встановлює стан на Завершено.



Пристрій також може: Відхилити повідомлення, що призведе до того, що стан повідомлено буде переведе в «Deadlettered». Пристрої, які підключаються через протокол Message Queuing Telemetry Transport (MQTT), не можуть відхиляти повідомлення з хмари на пристрій.

Пристрій може не обробити повідомлення, яке йому надіслано. У цьому випадку повідомлення автоматично повертаються зі стану «Невидимий» назад у стан «Поставлено в чергу» після закінчення часу очікування (або часу очікування блокування). Тривалість цього тайм-ауту становить одну хвилину і не може бути змінена.

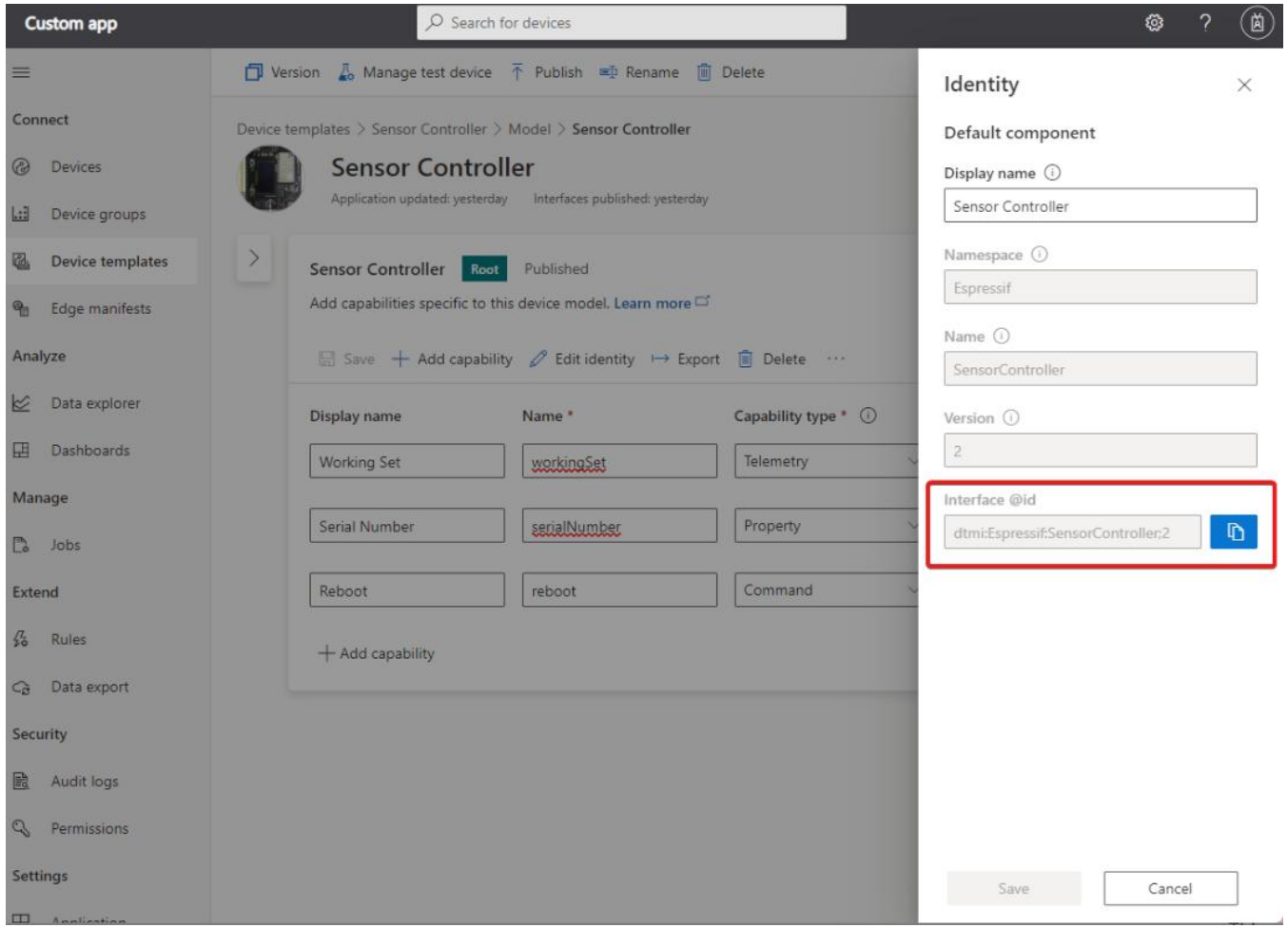
Налаштування властивостей для повідомлень типу «хмара-прилад» виконується на вкладці «кінцеві точки» (Build-in-endpoints)

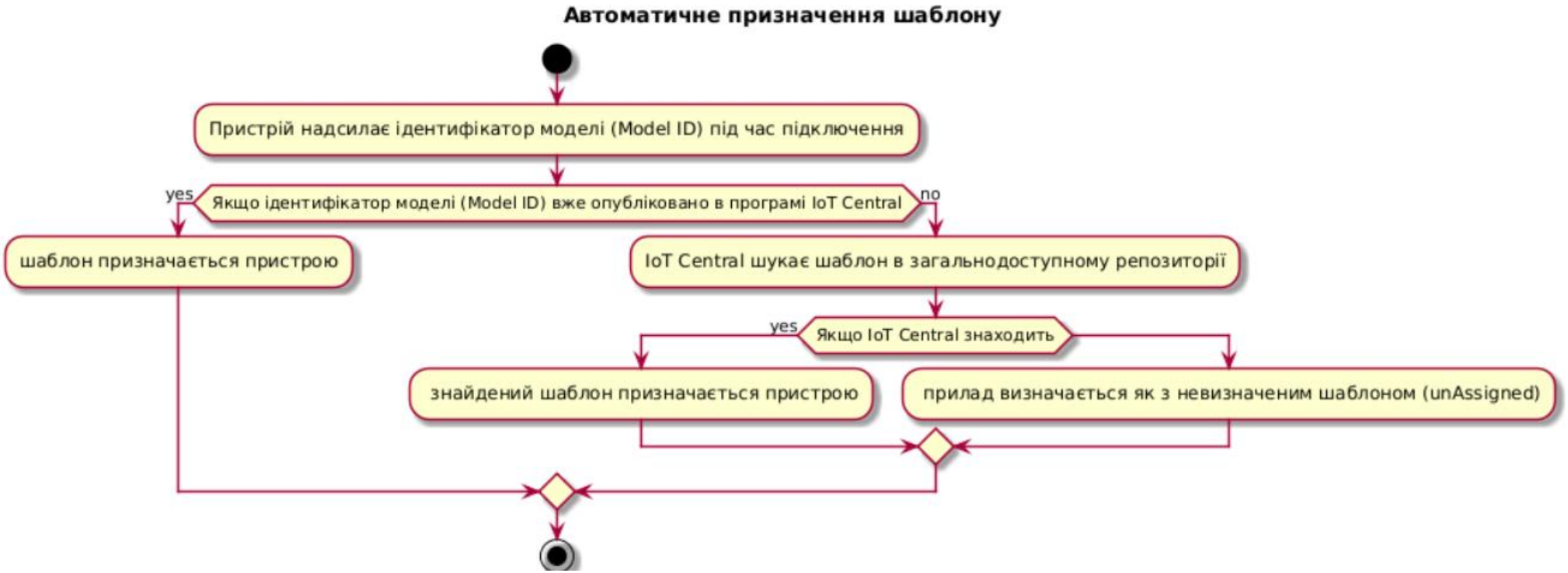


Визначення шаблону пристрою

Щоб пристрій міг взаємодіяти з IoT Central, йому потрібно призначити шаблону пристрою. Це призначення виконується одним із чотирьох способів:

- Під час реєстрації пристрою на сторінці «Пристрої»;
- Під час імпорту пристрої – можна визначити шаблон;
- Автоматично, шляхом надсилання ідентифікатора моделі під час першого підключення пристрою.

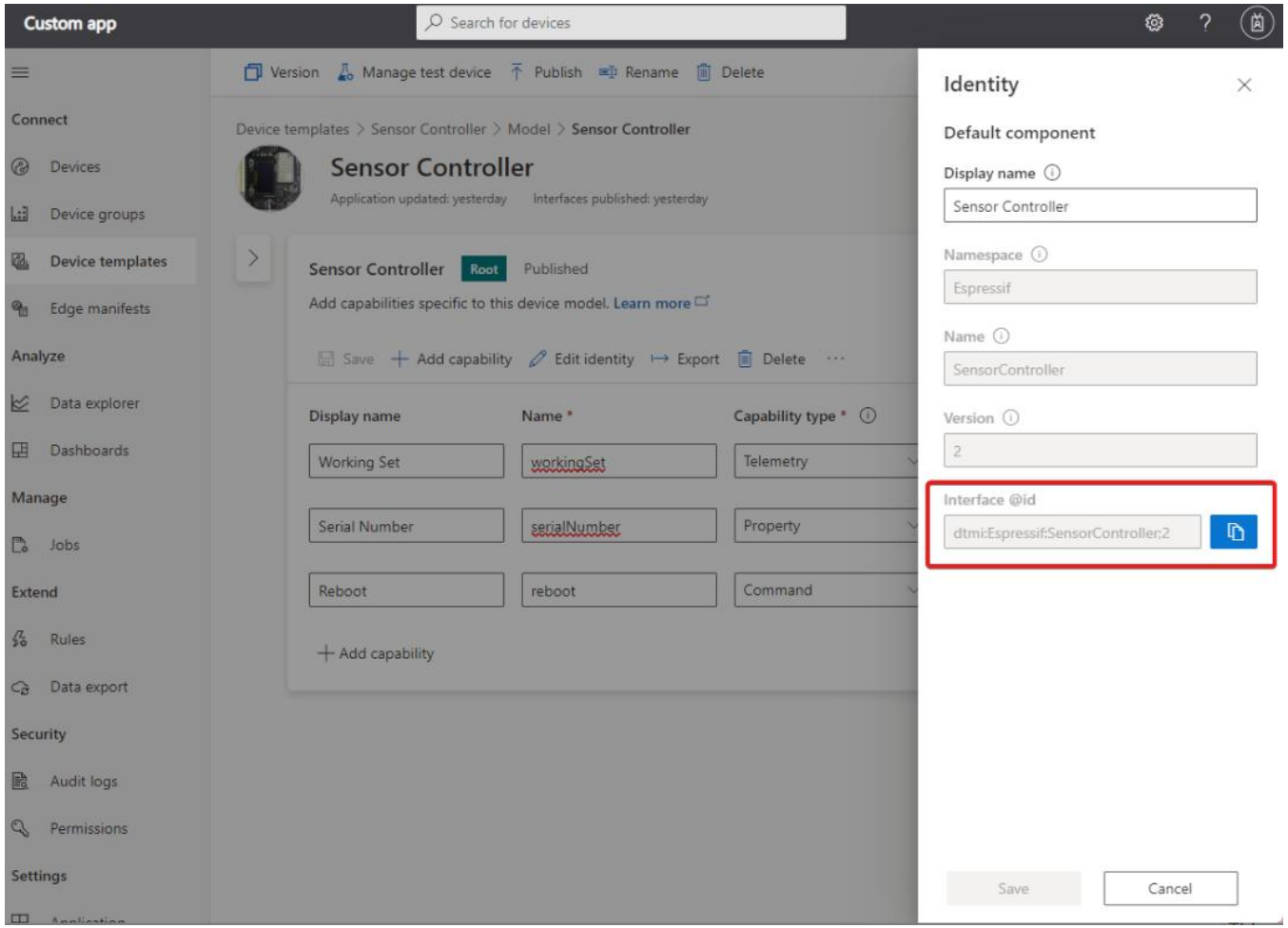




Після зв'язування пристрою з шаблоном – пристрій починає реалізовувати поведінку описану в файлі шаблону, який має формат JSON і включає опис:

підключення пристрою до IoT Central

Під час підключення пристрій має надіслати ідентифікатор моделі. IoT Central використовує ідентифікатор моделі для визначення шаблону пристрою для конкретної моделі пристрою. Процес включає:



Властивість «max delivery count» визначає максимальну кількість разів, коли повідомлення може переходити між станами «Поставлено в чергу» та «Невидимий». Після цієї кількості переходів центр IoT встановлює стан повідомлення на Dead lettered. Подібним чином концентратор IoT встановлює стан повідомлення на Dead lettered після закінчення терміну дії.

Кожне повідомлення з хмари на пристрій має термін дії. Цей час встановлюється параметром: ExpiryTimeUtc.

Для повідомлення, яке надсилається з хмари, Центр IoT може запросити доставку зворотного зв'язку, коли:

Positive –повідомлення з хмари досягає статусу «Completed»

Negative - повідомлення з хмари досягає статусу «Dead lettered»

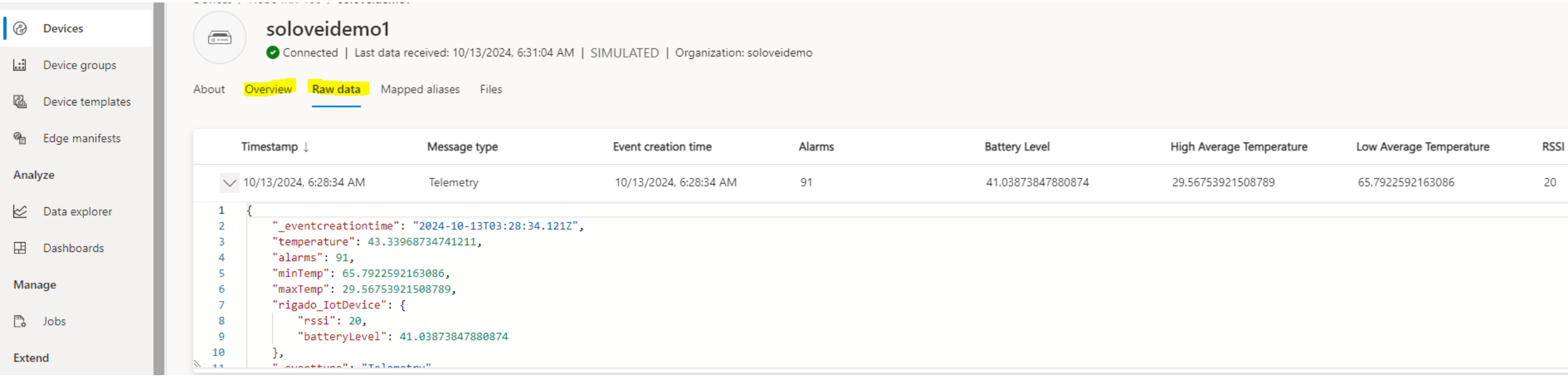
Центр IoT може запросити доставку зворотного зв'язку у всіх випадках, тоді значення Full

None – значення за замовчення, коли немає запиту на зворотній зв'язок.

```
[
  {
    "originalMessageId": "0987654321",
    "enqueuedTimeUtc": "2015-07-28T16:24:48.789Z",
    "statusCode": "Success",
    "description": "Success",
    "deviceId": "123",
    "deviceGenerationId": "abcdefghijklmnopqrstuvwxyz"
  },
  {
    ...
  },
  ...
]
```

Приклад успішно надісланого повідомлення

Створити прилад і зв'язати з шаблоном

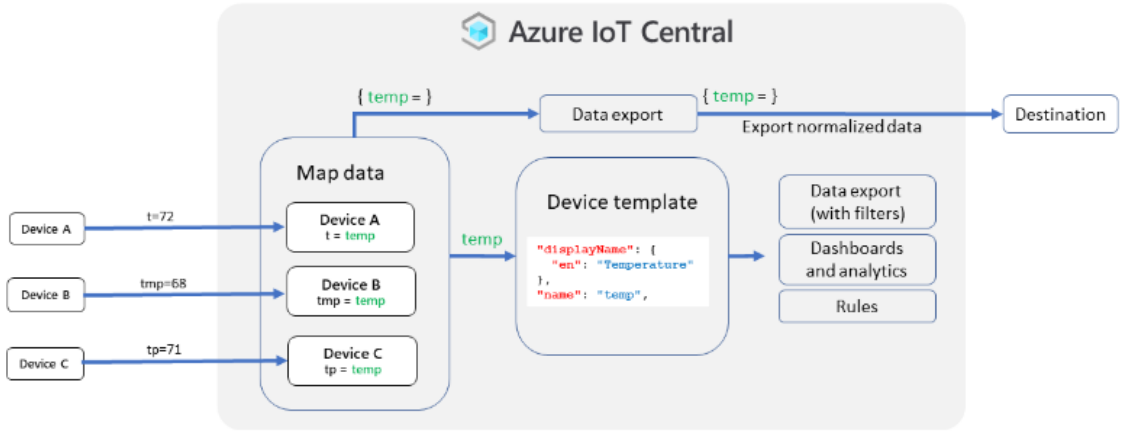


Модель. Кожна модель має унікальний ідентифікатор моделі та визначає можливості пристрою. Можливості згруповані в інтерфейси. Інтерфейси дозволяють повторно використовувати компоненти в різних моделях або використовувати успадкування для розширення набору можливостей.

Необроблені дані –необроблені дані, надіслані пристроєм. Це подання корисне під час налагодження або усунення несправностей шаблону пристрою.

Перегляди – використовуйте представлення для візуалізації даних із пристрою та форм для керування пристроєм і керування ним.

Подання даних



Подання даних дозволяє створити складні дані телеметрії пристрою в структурованих даних в IoT Central. Для кожного пристрою можна зіставити певний шлях JSON з псевдонімом. **Псевдонім** (alias)— це понятне ім'я цілого об'єкта. Спрощені дані, таким чиномможна використовувати:

- Створення шаблонів пристроїв і можливості управління пристроями в IoT Central.
- Нормалізація телеметрії з різних пристроїв
- Експорт за межами IoT Central.

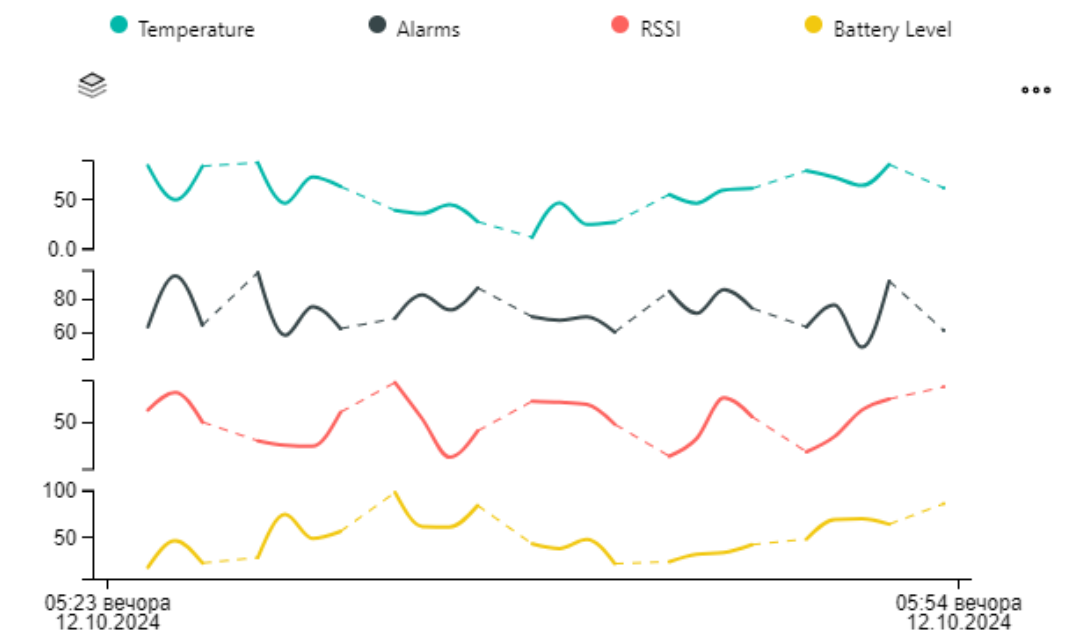
Timestamp: **Sun Oct 13 2024 06:41:03 GMT+0300 (Eastern European Summer Time)**


```
{
    "_eventcreationtime": "2024-10-13T03:41:03.387Z",
    "temperature": 22.880718231201172,
    "alarms": 19,
    "minTemp": 72.63876342773438,
    "maxTemp": 46.10273742675781,
    "rigado_IotDevice": {
        "rssi": 91,
        "batteryLevel": 65.96347418603315
    },
    "_eventtype": "Telemetry",
    "_timestamp": "2024-10-13T03:41:03.524Z"
}
```


soloveidemo

Connected | Last data received: 10/12/2024, 5:51:51 PM | SIMULATED | Organization: soloveidemo

Temperature, Alarms, RSSI, Battery Level



Temperature

51.41

Average, Past 12 hours

Alarms

73.10

Average, Past 12 hours

RSSI

50.42

Average, Past 12 hours

Battery Level

51.33

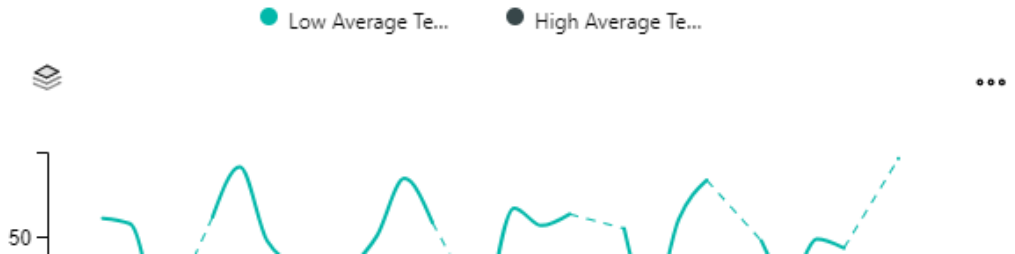
Average, Past 12 hours

Low Average Temperature

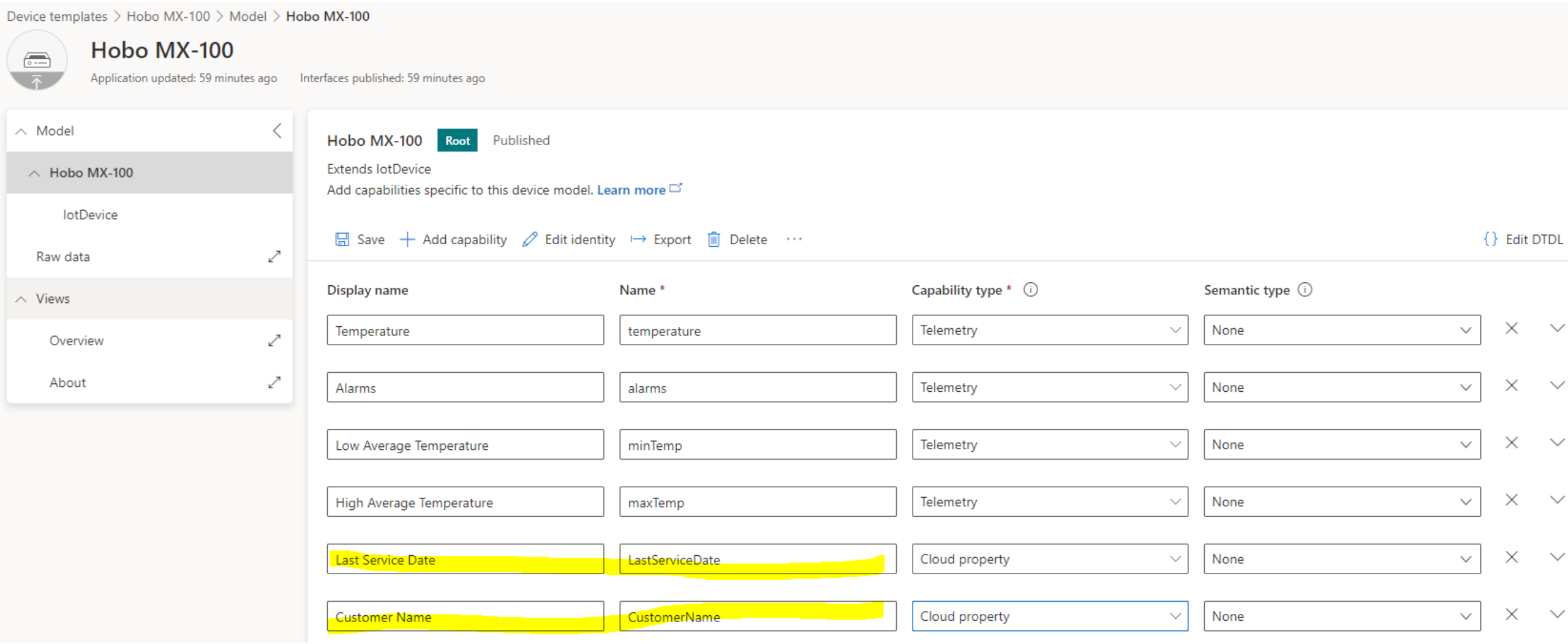
51.94

Average, Past 12 hours

Low Average Temperature, High Average Temperature



Створити нову модель приладу – на основі Hobo MX-100 (додати властивості)



Manage template | Manage device

Devices > Hobo MX-100 > soloveidevice

soloveidevice

Connected | Last data received: 10/12/2024, 4:56:41 PM | SIMULATED | Organization: soloveidemo

About | Overview | **Raw data** | Mapped aliases | Files

Timestamp ↓	Message type	Event creation time	Battery Level	RSSI	Temperature	Device ID
10/12/2024, 4:51:42 PM	Telemetry	10/12/2024, 4:51:42 PM	47.24484487420561	60	29.448490142822266	


```
{
    "_eventcreationtime": "2024-10-12T13:51:42.249Z",
    "temperature": 29.448490142822266,
    "rigado_IotDevice": {
        "rssi": 60,
        "batteryLevel": 47.24484487420561
    },
    "_eventtype": "Telemetry",
    "_timestamp": "2024-10-12T13:51:42.383Z"
}
```


10/12/2024, 4:51:36 PM Device connected

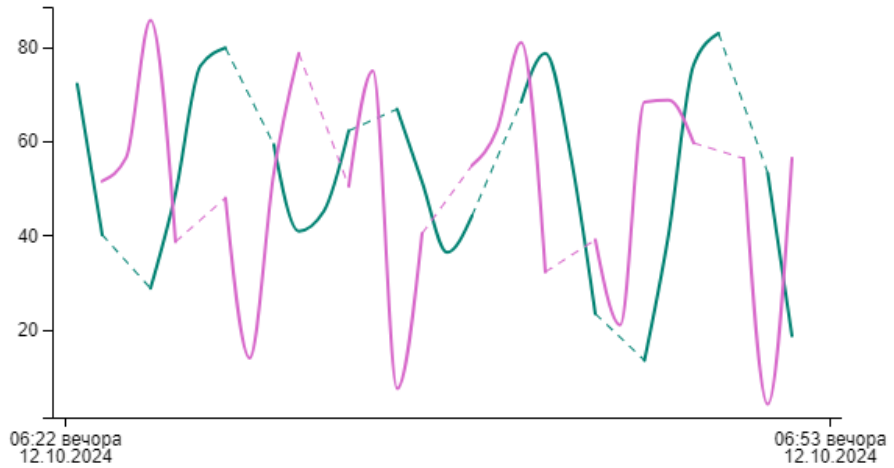

```
{
    "_eventtype": "Device connected",
    "_timestamp": "2024-10-12T13:51:36.7771323Z"
}
```

Edit Copy Delete

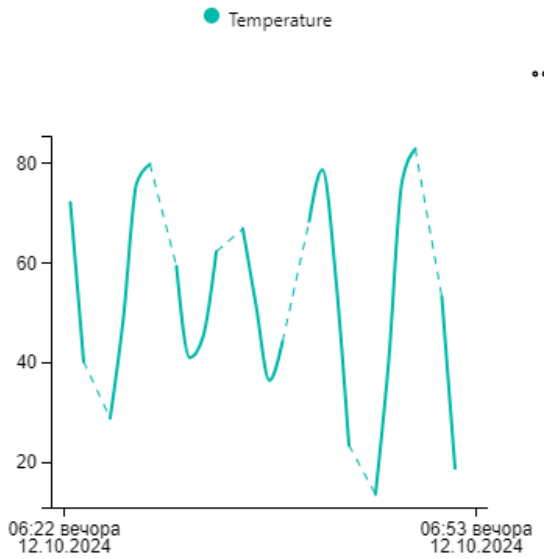
Connect
Devices
Device groups
Device templates
Edge manifests
Analyze
Data explorer
Dashboards
Manage
Jobs
Extend
Rules
Data export
Security
Audit logs
Permissions
Settings
Application
Customization

Temperature

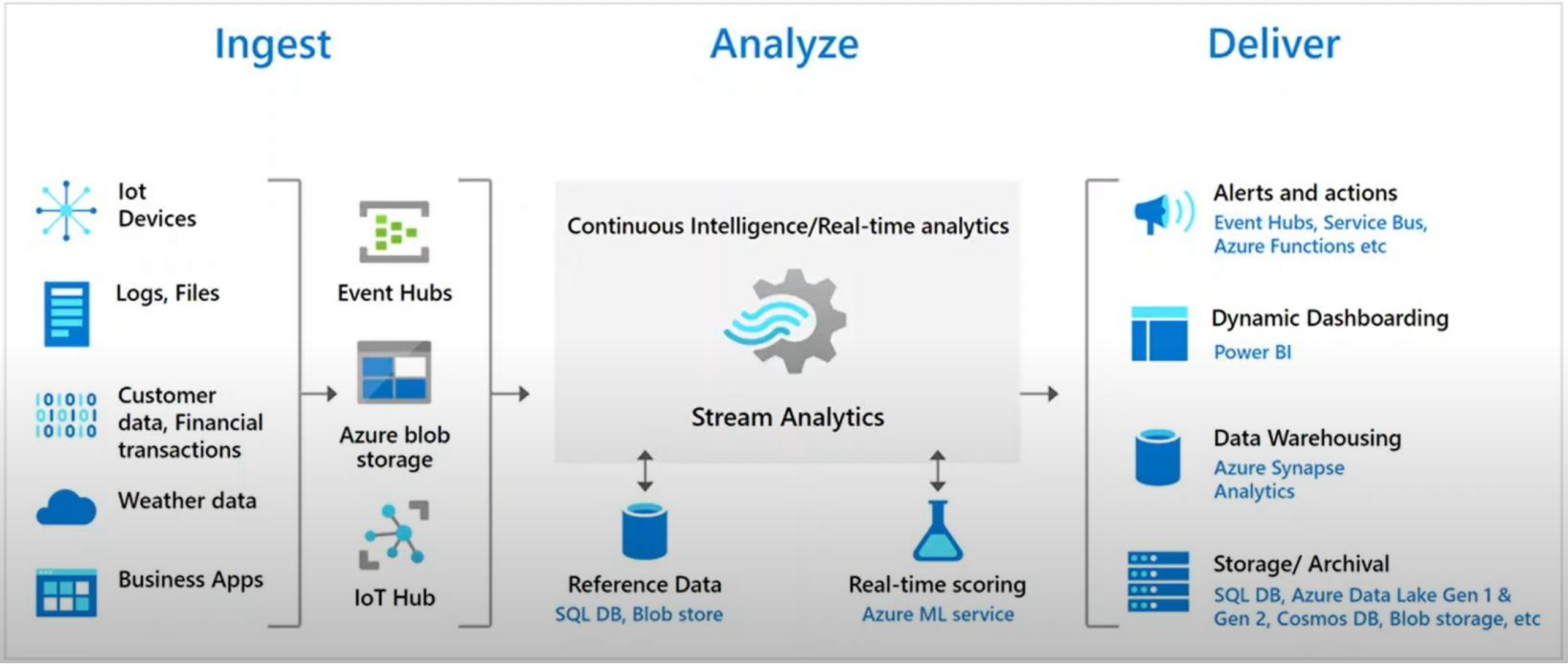
soloveidevice2 (...
Average
soloveidevice3 (...
Average



Temperature



Azure Stream Analytics



Питання

1. Що таке Azure IoT Central і які ключові компоненти входять до його структури?
2. Яку роль у структурі IoT Central відіграє Application Template?
3. Що таке Device Template та як він впливає на моделювання пристроїв?
4. Які основні розділи містить інтерфейс Azure IoT Central (Dashboard, Devices, Analytics, Administration)?
5. Для чого використовується панель Dashboard і яку інформацію там можна відобразити?
6. Які функції доступні в розділі Devices інтерфейсу IoT Central?
7. Як у IoT Central створюються та налаштовуються візуалізації телеметрії (charts, tiles)?
8. Як Azure IoT Central взаємодіє з IoT Hub у фоновому режимі?
9. Яким чином у IoT Central реалізується масштабування та ізоляція ресурсів?
10. За що відповідає рівень Data Export в архітектурі IoT Central?
11. Які основні відмінності між IoT Central та IoT Hub з точки зору архітектури та керування?
12. Які типи операцій можна виконувати над пристроєм у IoT Central (перегляд телеметрії, зміна параметрів, оновлення)?
13. Що таке Commands у керуванні пристроями і як вони працюють?
14. Як у IoT Central реалізовано відстеження стану пристрою (Device Properties, Cloud Properties, Device Status)?