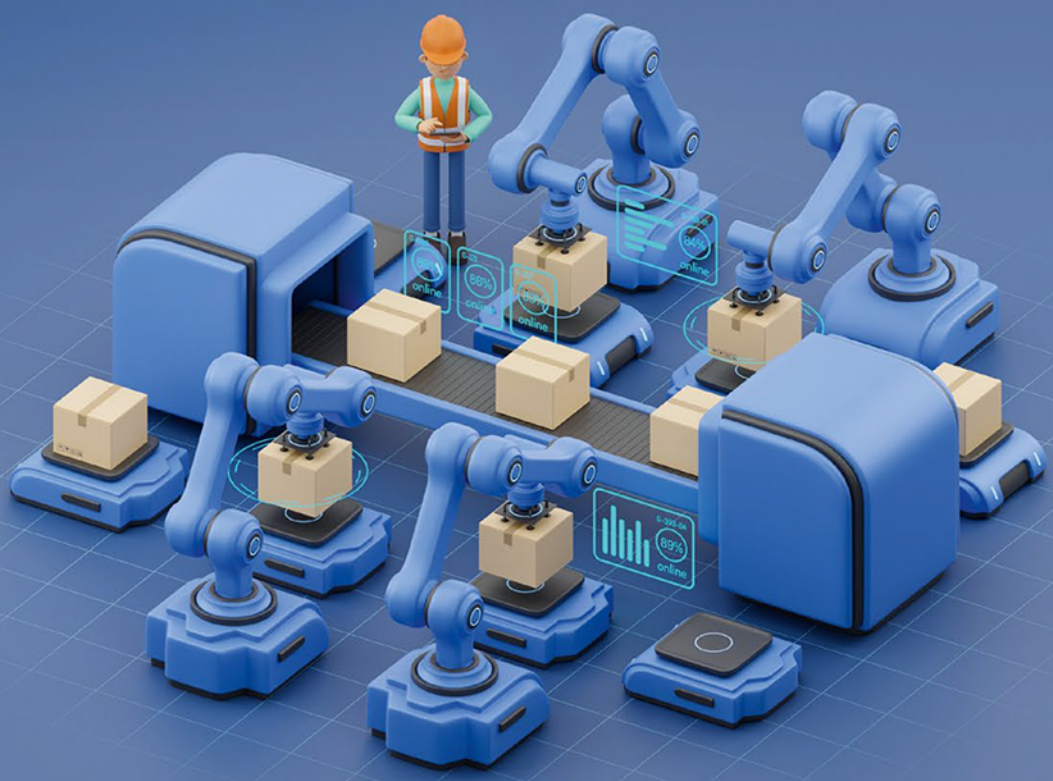
SIEMENS

Artur **Nowocień**

DIGITALIZACJA

w systemach automatyki SIMATIC

Wydanie II



Teoria Przykłady Ćwiczenia

Helion

Redaktor prowadzący: Szymon Szwajger, Małgorzata Kulik

Projekt okładki: Studio Gravite / Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki
Grafika na okładce została wykorzystana za zgodą Shutterstock.com.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *https://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
https://helion.pl/user/opinie/dib2b2
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-289-0209-1

Printed in Poland.

Spis treści

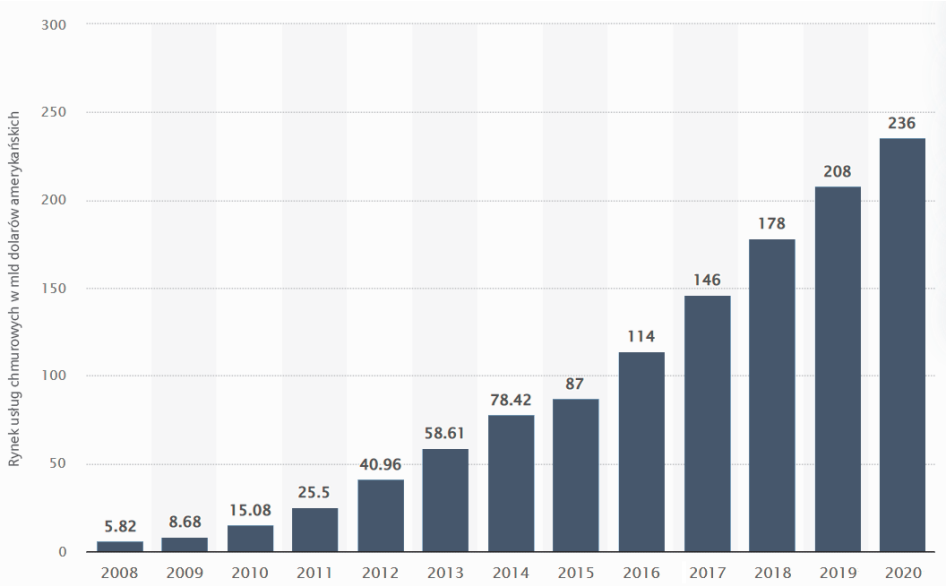
Rozdział 7.

Chmury obliczeniowe

Założę się, że jeszcze kilka lat temu każdy z nas miał w domu bogatą kolekcję płyt CD i DVD, nośników USB, kart SD i innych przenośnych magazynów danych. Cóż, wszyscy musieliśmy przechowywać gdzieś zdjęcia z wakacji, ulubioną muzykę czy dokumenty niezbędne do szkoły lub do pracy. Gdyby zastanowić się nad tym, ile z tych urządzeń wykorzystujemy dzisiaj, okazałoby się, że zapewne niewiele. Płyty już dawno odeszły do lamusa, nośniki USB traktujemy raczej jako szybki sposób na transfer pliku z miejsca A do B, karty wykorzystujemy tylko do obsługi urządzeń, które ich wymagają (np. aparatów fotograficznych), a potem i tak zgrywamy z nich pliki, żeby zwolnić miejsce przed kolejnym urlopem. Gdzie więc podziewają się te wszystkie giga- lub nawet terabajty danych, które tak sumiennie gromadzimy przez lata? Chyba nie wyparowują?

A może jednak? Podobnie jak kałuża po deszczu prędzej czy później zamieni się w parę wodną, a w efekcie w chmurę, tak nasze dane czeka podobny los — z tym, że chmura, o której mówimy, jest dużo stabilniejsza niż pierwszy lepszy cumulus. Ostatnie kilka lat przyniosło prawdziwą rewolucję, jeśli chodzi o podejście do przechowywania danych. Kamieniem milowym było upowszechnienie się smartfonów (koniec pierwszej dekady XXI w.) i ich możliwości związane z obsługą i generowaniem plików, a także łącznością z Internetem. Od kiedy zaczęliśmy nosić w kieszeni mobilny komputer, osobiste biuro, odtwarzacz muzyki i aparat fotograficzny w jednym, ilość danych, jakie produkujemy każdego dnia, bardzo szybko zaczęła przekraczać możliwości pamięci wbudowanych w urządzenia. Bardzo szybko więc zarówno czołowi gracze z branży IT, jak i użytkownicy zaczęli dostrzegać potrzebę wyniesienia magazynów danych poza poziom lokalny, gdzie pojemność dysku lub karty SD nie będzie ograniczeniem. W efekcie od ok. 15 lat możemy obserwować dynamiczny rozwój rynku usług serwerowych, oferujących przestrzeń dyskową do przechowywania naszych prywatnych plików, zwanych powszechnie **chmurami** lub usługami typu ***cloud storage*** (przechowywanie w chmurze) (rysunek 7.1).

Obecnie prawie każdy telefon komórkowy jest zsynchronizowany z przynajmniej jedną chmurą w celu gromadzenia w niej danych generowanych w urządzeniu. Nie przechowujemy już zdjęć czy plików na dyskach lokalnych, zamiast tego trzymamy je na serwerach, do których mamy dostęp o dowolnej porze z dowolnego urządzenia i miejsca na świecie. Olbrzymia popularność i wygoda usług chmurowych sprawiła, że do chmury podłączamy nie tylko telefony czy komputery, ale również sprzęty RTV, dyski sieciowe w celu



RYSUNEK 7.1. *Wzrost udziału chmur obliczeniowych w rynku w latach 2008 – 2020. Zauważalna jest wyraźna tendencja wzrostowa (źródło: Statista)*

synchronizacji, a także wiele innych urządzeń wymagających dodatkowej przestrzeni na pliki czy dane. Mechanizm ten nie jest jednostronny. Prawie wszystkie współczesne serwisy zapewniające rozrywkę, a więc serwisy VOD, muzyczne, a nawet dostawcy gier, oferują swoje usługi na poziomie chmury. Wystarczy jedynie łączność z Internetem i już możemy obejrzeć ulubiony film bez konieczności jego pobierania, posłuchać nowej płyty ulubionego artysty bez wizyty w sklepie muzycznym czy zagrać w grę na konsoli bez posiadania fizycznej kopii.

Chmura to jednak nie tylko przestrzeń na dane. Aby w pełni zrozumieć jej możliwości, musimy się zastanowić, co właściwie kryje się pod tą nazwą. Cóż, nasze pliki nie trafiają przecież w eter — do obsługi wszystkich publicznych usług chmurowych niezbędna jest odpowiednia infrastruktura. W praktyce przyjmuje ona postać centrów danych rozlokowanych na całym świecie, wyposażonych w potężne serwery i niezliczone liczby dysków twardych. Każda subskrypcja usługi chmurowej daje nam dostęp do niewielkiej części mocy obliczeniowej serwera potrzebnej do obsługi naszych danych, a także do wydzielonego fragmentu dysku lub dysków, gdzie te dane są przechowywane. Przeglądając zdjęcia na Google Photos czy otwierając dokument na OneDrive, nawiązujemy tak naprawdę połączenie z serwerem, na którym, w odpowiedniej aplikacji przeglądarkowej lub mobilnej, otwieramy wskazany plik. Możliwości obliczeniowe dostępne w większości centrów danych są jednak znacznie wyższe, niż potrzeba do obsługi plików czy danych. Dostawcy usług doszli więc do wniosku, że warto spieniężyć ten fakt, oferując użytkownikom możliwość skorzystania z tej mocy w celu realizacji operacji obliczeniowych czy obsługi aplikacji. Obok angielskiego zwrotu *cloud storage* wykształciła się zatem nowa fraza — ***cloud computing*** (obliczenia w chmurze).

7.1. Geneza chmur obliczeniowych, czyli Amazon w natarciu

Wróćmy do roku 1997. To wtedy, w maju, wystartowała internetowa księgarnia Amazon.com założona przez Amerykanina Jeffa Bezosa. Rok później platforma poszerzyła swoją działalność o sprzedaż zabawek, gier i elektroniki użytkowej, stając się wkrótce jedną z największych platform e-commerce (sprzedaży internetowej) w kraju. Poskutkowało to bardzo dużym zainteresowaniem odbiorców i znacznym obciążeniem serwerów w okresach zwiększonej sprzedaży — okolicy świąt czy akcji promocyjnych takich jak *black friday*. Zarząd firmy stanął więc przed wyzwaniem poradzenia sobie z tym kłopotem, w efekcie czego po 2000 r. poczyniono kroki dążące do przebudowania infrastruktury serwerowej. Serwery zostały zamienione na wersje wykorzystujące system operacyjny Linux, cechujące się większą elastycznością i niższą ceną. Zwiększono też ich liczbę, aby wyeliminować problem przeciążenia w gorących okresach. Wkrótce okazało się jednak, że przez pozostałą część roku używane jest zaledwie 10% posiadanej mocy obliczeniowej. Szybko zaczęto myśleć, jak wykorzystać potencjał pozostałych 90%. Odpowiedź pojawiła się już w lipcu 2002 r., kiedy to wystartowała pierwsza wersja usługi **Amazon Web Services**, udostępniającej programistom funkcje pozwalające na uruchomienie własnych aplikacji na serwerach firmy. Popularność AWS szybko przerosła oczekiwania twórców, co poskutkowało dalszym rozwojem serwisu. W 2006 r. udostępniono usługi przechowywania plików (S3), dzierżawy wirtualnych maszyn uruchamianych na serwerach AWS (E2C), a także system bazodanowy (RDS). Od tego czasu AWS jest ciągle rozwijana, a fakt bycia pionierem w dziedzinie chmur obliczeniowych wydaje się doceniany przez klientów, którzy wywindowali serwis na szczyt popularności wśród wszystkich dostawców tego typu rozwiązań.

Amazon nie jest oczywiście jedynym dostawcą usług chmurowych oferującym tak szeroki zestaw funkcjonalności i technologii. W topowej trójce znajduje się także Microsoft ze swoją platformą Azure uruchomioną w 2010 r., a także Google z Google Cloud Platform (GCP) dostępną od roku 2011. Niezależnie od wyboru dostawcy platformy chmurowe cechują się podobną architekturą i zestawem usług dla klientów. Od kilku lat stają się popularną alternatywą dla serwerów utrzymywanych i zarządzanych indywidualnie przez firmy i korporacje. Usługi dostarczane w ramach platform takich jak AWS, Azure czy GCP mogą posłużyć do budowania prostych serwisów internetowych, poprzez rozległe magazyny danych aż po zaawansowane aplikacje uruchamiane w środowiskach wirtualnych. O użyteczności i popularności chmur obliczeniowych może świadczyć także fakt, że są one wykorzystywane przez instytucje rządowe i służą do uruchamiania kluczowych dla działania państwa usług i serwisów. Przykładem może być wdrożony przez Ministerstwo Finansów system e-Pit, usprawniający proces składania corocznego zeznania podatkowego. Usługa została uruchomiona na platformie Microsoft Azure i działa nieprzerwanie od 2019 r. Microsoft na swojej stronie wymienia zresztą więcej przypadków wdrożeń chmury Azure w instytucjach publicznych, włączając w to australijską policję, libańskie ministerstwo zdrowia czy nowozelandzką straż pożarną. Również wiele usług

wykorzystywanych przez nas na co dzień jest opartych na platformach chmurowych — mówimy tu o sklepach i innych witrynach internetowych, platformach multimedialnych czy aplikacjach przeglądarkowych.

Duża popularność tych rozwiązań, a także stosunkowo wysoki poziom zaufania nie bierze się znikąd. Chmury obliczeniowe cechują się zaletami i cechami, które odpowiednio spożytkowane mogą znacznie usprawnić proces wdrażania nowych rozwiązań IT — są to m.in.:

- Dostępność licznych, gotowych usług, które mogą być użyte do budowy własnych aplikacji. Zaliczamy do nich usługi przechowywania plików na serwerach, magazyny danych SQL i NoSQL, obsługę wirtualnych maszyn czy gotowe serwisy dla poszczególnych branż itp. Olbrzymią zaletą chmur obliczeniowych jest także wysoka skalowalność mocy obliczeniowej, dzięki czemu nasza aplikacja czy usługa może działać płynnie niezależnie od obciążenia.
- Krótki czas wdrożenia. Gotowe usługi wykorzystywane do budowy aplikacji pozwalają na znaczne skrócenie czasu od koncepcji do wdrożenia gotowego prototypu. Tym samym pomysły mogą być szybko zweryfikowane i skonfrontowane z rzeczywistością, bez ponoszenia niepotrzebnych nakładów na sprzęt czy budowanie aplikacji całkowicie od zera.
- Płatność uzależniona od zużycia. Jeśli nasza aplikacja zużywa mniej zasobów w pewnych okresach, spowoduje to naliczenie mniejszej opłaty niż w pozostałych miesiącach. W porównaniu z utrzymaniem lub uruchomieniem własnej infrastruktury o podobnej mocy obliczeniowej, zazwyczaj będziemy mogli zaobserwować znaczne oszczędności.
- Wysoki poziom bezpieczeństwa. Choć wielu sceptyków obawia się, że dane przechowywane na obcych serwerach nie są do końca bezpieczne, w rzeczywistości jest zgoła inaczej. Są one zabezpieczane poprzez cykliczne kopie zapasowe, a nakłady ponoszone na cyberbezpieczeństwo przez dostawców platform są niewspółmiernie wyższe od nakładów, na jakie może sobie pozwolić indywidualny użytkownik lub firma.

Zastosowanie platform chmurowych nie ogranicza się tylko do konkretnych dziedzin związanych z informatyką. Korzyści może odnieść także branża przemysłowa. Wspomniana wyżej elastyczność, bezpieczeństwo i mnogość rozwiązań służących do magazynowania danych mogą stanowić doskonałą alternatywę dla uruchamianych lokalnie baz danych służących do archiwizacji czy sprzęgniętych z systemami *traceability*. Wizualizacja tych danych, raportowanie i wykonywanie na nich obliczeń analitycznych, w połączeniu z możliwością zdalnego dostępu z dowolnego urządzenia i miejsca na świecie, czynią z chmur doskonałe narzędzie do zarządzania produkcją dla kierowników i menedżerów. Chmura może posłużyć też do wyniesienia pewnych operacji lub funkcjonalności poza lokalne urządzenia. Doskonałym przykładem jest chociażby uczenie sieci neuronowej, które wymaga sporej mocy obliczeniowej. Realizacja takiego zadania lokalnie wiązałaby się

z potrzebą uruchomienia dedykowanego komputera z modułami GPU lub TPU (wykorzystywanymi właśnie w operacjach uczenia maszynowego), którego cena może znacznie przekroczyć koszt usługi chmurowej. Czasami łączność z chmurą wymuszana jest przez przedsiębiorstwo — zdarzają się sytuacje, w których systemy CRM czy MES dostarczane są w modelu SaaS (ang. *Software as a Service*, czyli udzielenie klientowi dostępu do usługi uruchomionej na serwerze producenta lub w chmurze publicznej — o modelach usług chmurowych dowiesz się więcej w rozdziale 7.3). Jeśli więc niezbędna jest łączność pomiędzy tymi aplikacjami a systemem automatyki, opcją jest skorzystanie z jednego z dostępnych mechanizmów IoT, np. bramki lub bezpośredniego podłączenia sterownika do Internetu. Przykłady te można mnożyć, pewne jest jednak, że stosowanie chmur obliczeniowych może przynieść korzyści w sporej liczbie dziedzin przemysłu, zwłaszcza tam, gdzie z uwagi na wymagania procesów lub branż (np. farmacja, automotive) realizowany jest przepływ olbrzymich ilości danych. Już teraz wiele firm i przedsiębiorstw korzysta z namiastki chmur obliczeniowych, przenosząc część usług i aplikacji z poziomu lokalnych komputerów na serwery. Nie zawsze jednak takie podejście ma przewagę nad korzystaniem z publicznych platform — ustalmy więc, która ścieżka jest lepsza.

7.2. Chmura publiczna vs. chmura prywatna

W poprzednim akapicie skupiłem się na omówieniu chmury publicznej. Jest to model, w którym usługi i dane są hostowane na serwerach dostawcy i dostępne publicznie dla wszystkich odbiorców. W tym wariancie jedyną odpowiedzialnością spoczywającą na kliencie jest regulowanie kosztów i zarządzanie architekturą aplikacji wdrażanej na platformie chmurowej. Kwestie związane z zachowaniem ciągłości usług, utrzymaniem serwerów czy wdrażaniem zabezpieczeń zarządzane są zawsze przez dostawcę, dzięki czemu klient może się skoncentrować na aplikacji, zamiast poświęcać zasoby i nakłady na utrzymanie infrastruktury.

Alternatywą jest przeniesienie elementów chmury publicznej w całości na serwer lub serwery uruchomione lokalnie w przedsiębiorstwie lub uruchomienie ich w specjalnie wydzielonym dla danego klienta fragmencie infrastruktury utrzymywanej przez dostawcę usług chmurowych — taki model nosi nazwę **chmury prywatnej**. W odróżnieniu od wariantu publicznego, w chmurze prywatnej to klient jest odpowiedzialny za bezpieczeństwo danych i serwerów, a także ich utrzymanie i dbanie o nieprzerwaną pracę (jeśli platforma jest uruchamiana lokalnie). W zamian otrzymuje dużo większe możliwości personalizacji usług i środowiska pod kątem indywidualnych wymagań biznesowych. Wszystkie elementy takiej platformy dostępne są wyłącznie w obrębie danego przedsiębiorstwa, co daje większą kontrolę nad danymi i nad prywatnością całego rozwiązania. Tego typu usługi wymagają jednak poniesienia dużo większych nakładów, wynikających z konieczności zakupu lub dzierżawy infrastruktury serwerowej oraz opłacania zespołu odpowiedzialnego za jej utrzymanie. Kluczowa jest zatem kwestia związana z wyborem właściwego rozwiązania w stosunku do wymagań.

W praktyce chmura prywatna wybierana jest najczęściej przez organizacje, gdzie poziom krytyczności procesów lub operacji biznesowych wymaga zwiększonej kontroli nad środowiskiem, w którym są one przeprowadzane. Możemy zaliczyć do nich agencje rządowe, instytucje finansowe (np. banki), operatorów infrastruktury krytycznej, a także niektóre, zazwyczaj większe firmy. Ponieważ usługi chmurowe są uruchamiane w obrębie infrastruktury zarządzanej i finansowanej przez klienta, możliwe jest wystąpienie sytuacji, w której przestanie być ona wystarczająca. W przypadku chmury prywatnej może to doprowadzić do spowolnienia lub całkowitego przestoju aplikacji bądź usług uruchomionych w ramach platformy. Dlatego dostawcy bardzo często oferują także trzeci model usług, zwany **chmurą hybrydową**. Łączy ona zalety chmury prywatnej i publicznej, dzięki czemu klient zachowuje pełną kontrolę nad środowiskiem, ale z możliwością wykorzystania zasobów publicznych, gdy lokalna infrastruktura przestanie być wystarczająca.

W kontekście tej książki pojęciem chmury prywatnej będę określał także niezależne serwery konfigurowane i utrzymywane przez przedsiębiorstwo. Takie rozwiązanie spotykane jest dużo częściej niż uruchamianie usług chmurowych on-premise (lokalnie), zwłaszcza w przedsiębiorstwach związanych z przemysłem. Wynika to przede wszystkim z potrzeb — zazwyczaj serwery te wykorzystywane są w głównej mierze przez oprogramowanie przemysłowe, takie jak systemy SCADA lub *traceability*, które trudno byłoby zintegrować z usługami chmurowymi lub byłoby to nieopłacalne. Obsługa aplikacji i usług odpowiedzialnych np. za wizualizację danych czy uruchamianie algorytmów pełni w nich funkcję dodatkową, więc dostosowywanie ich wyłącznie pod kątem tych zadań zwykle bywa nieuzasadnione, chyba że dana aplikacja wymaga specyficznej konfiguracji lub większej mocy obliczeniowej. Dodatkowo wdrożenie usług chmurowych on-premise jest często dużo droższe niż koszt chmury publicznej czy utrzymania niezależnego serwera, na co nie wszystkie firmy mogą sobie pozwolić, szukając tym samym alternatyw, które są w ich zasięgu.

Zestawienie cech poszczególnych modeli platform chmurowych zostało przedstawione w tabeli 7.1.

Wybór rozwiązania będącego odpowiedzią na wyzwanie, przed którym stoimy, bywa trudny. Decyzja związana z doborem właściwego modelu zawsze będzie kompromisem pomiędzy *kosztem, bezpieczeństwem i łatwością wdrożenia*. Rozważmy tę kwestię na przykładzie jednego z najbardziej standardowych systemów, jakie mogą wymagać skorzystania z chmury obliczeniowej — monitorowania produkcji w zakładzie. Taki system zawsze wymaga mechanizmu akwizycji danych pochodzących z maszyn i linii produkcyjnych, magazynu w postaci bazy danych oraz aplikacji do ich wizualizacji i raportowania.

Jeśli w naszym zakładzie dostępny jest serwer (np. z uruchomionym systemem SCADA), dysponujemy już pewną platformą, która może stanowić bazę do uruchomienia aplikacji. Co więcej, bardzo prawdopodobne jest to, że pełni on funkcję magazynu danych, być może i tych, które musielibyśmy gromadzić w celu monitorowania produkcji. Problematyczną kwestią może być zatem sam proces wdrożenia aplikacji — trzeba ją najpierw napisać lub

TABELA 7.1. *Porównanie chmur — prywatnej, publicznej i hybrydowej*

	Chmura publiczna	Chmura prywatna	Chmura hybrydowa
Koszt	Ponoszony wyłącznie za wykorzystywane usługi	Ponoszony za usługi i utrzymanie infrastruktury serwerowej	Ponoszony za usługi i utrzymanie infrastruktury serwerowej
Dostępność usług	Wszystkie oferowane przez dostawcę	Wszystkie oferowane przez dostawcę	Wszystkie oferowane przez dostawcę
Elastyczność	Możliwość skalowania zasobów w zależności od potrzeb i bieżącego zużycia	Ograniczona przez posiadaną infrastrukturę	Ograniczona przez posiadaną infrastrukturę, ale z możliwością rozszerzenia zasobów o te dostępne w chmurze publicznej
Prywatność	Współdzielenie zasobów z innymi użytkownikami	Zasoby dostępne wyłącznie dla danego przedsiębiorstwa	Zasoby dostępne wyłącznie dla danego przedsiębiorstwa, z możliwością wykorzystania zasobów współdzielonych
Bezpieczeństwo	Wysokie — wynikające z dużych nakładów na utrzymanie odpowiedniego poziomu cyberbezpieczeństwa przez dostawcę usług. Każdy użytkownik ponosi koszty części tych nakładów w ramach opłat za wykorzystanie chmury	Zależne od przedsiębiorstwa — poziom cyberbezpieczeństwa uzależniony jest od tego, jak skonfigurowana i zabezpieczona zostanie infrastruktura. Wszystkie koszty są ponoszone przez jednego użytkownika	Zależne od przedsiębiorstwa — w kontekście infrastruktury lokalnej Wysokie — w kontekście wykorzystania zasobów współdzielonych
Przeznaczenie	Wszyscy klienci wymagający szybkiego wdrożenia rozwiązania IT	Dedykowane grupy klientów, wymagające zwiększonej kontroli nad środowiskiem IT	Dedykowane grupy klientów, wymagające zwiększonej kontroli nad środowiskiem IT

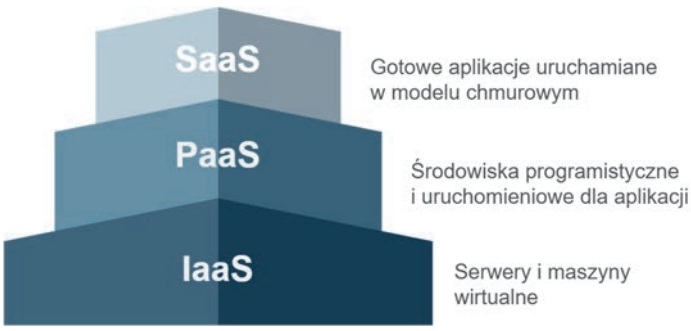
pozyskać od zewnętrznego dostawcy, uruchomić tak, aby nie kolidowała z oprogramowaniem pracującym już na serwerze, i, przede wszystkim, udostępnić wszystkim zainteresowanym osobom. Nie zawsze też dysponujemy serwerem dostępnym w przedsiębiorstwie, co wiąże się z dodatkowym kosztem jego zakupu lub dzierżawy, konfiguracji i utrzymania. Musimy także zadbać o jego odpowiednie zabezpieczenie, aby nie wprowadzić dodatkowych, niepożądanych podatności na ataki.

Alternatywną opcją jest wykorzystanie chmury publicznej i uruchomienie aplikacji w odizolowanym środowisku, dostępnym publicznie dla każdej osoby, której je udostępnimy. To podejście potrzebuje jednak nawiązania łączności pomiędzy systemem automatyki i chmurą, co często wymaga uzupełnienia systemu o dodatkowe bramki IoT lub uzyskania zgody działu IT na bezpośrednie połączenie sterowników z Internetem, a tym samym konieczności ich dodatkowego zabezpieczenia. Musimy także uwzględnić całkowity miesięczny koszt chmury, którego nie braliśmy pod uwagę w sytuacji wykorzystania posiadanego serwera. Plusem jest natomiast łatwość i szybkość w budowaniu aplikacji, która może być oparta na szeregu gotowych usług i serwisów dostarczanych przez operatora chmury. W stosunkowo krótkim czasie jesteśmy więc w stanie dostarczyć gotowe rozwiązanie, cechujące się wysokim poziomem bezpieczeństwa i spójności danych.

Jak więc widzimy, decyzja, jaką chmurę wybrać, nie jest prosta i wymaga indywidualnego podejścia do każdego projektu. Bardzo często jest ona uzależniona od sumarycznego kosztu rozwiązania, rzadziej pod uwagę brane są takie kwestie jak łatwość i szybkość wdrożenia i późniejszego utrzymania. Jeśli masz trudności z doborem właściwego rozwiązania, zawsze możesz skorzystać ze wsparcia niezależnych doradców, którzy pomogą Ci przeanalizować wszystkie plusy i minusy danego wariantu w odniesieniu do Twojego projektu. Dobrym pomysłem jest także kontakt z dostawcą systemu automatyki, z którego korzystasz, w celu przedyskutowania możliwych opcji łączności i integracji z systemami IT.

7.3. Modele usług chmurowych

Jeśli zdecydowałaś/zdecydowałeś się na zastosowanie możliwości chmury publicznej, pierwszą i najtrudniejszą decyzję masz już za sobą. Kolejnym etapem jest wybór modelu usług, z jakiego będziesz korzystać. Na platformach chmurowych możemy wyróżnić zazwyczaj trzy podstawowe modele (rysunek 7.2), definiujące poziom dostępu do elementów składowych całego systemu.



Rysunek 7.2. *Piramida trzech podstawowych typów usług chmurowych*

- **IaaS (ang. *Infrastructure as a Service*)** — bezpośredni dostęp do najniższego poziomu chmury, a więc systemu operacyjnego uruchamianego zazwyczaj w obrębie maszyn wirtualnych. Model ten oferuje najszersze możliwości związane z konfiguracją i dostosowaniem środowiska do własnych potrzeb, wymaga także największego nakładu pracy w celu wdrożenia odpowiednich zabezpieczeń, instalacji niezbędnych usług i bibliotek oraz przygotowania aplikacji całkowicie od zera. W praktyce przypomina uruchamianie własnych rozwiązań informatycznych na niezależnych komputerach lub serwerach. Przykładem usług w modelu IaaS są maszyny wirtualne dostępne w ramach publicznych platform chmurowych (np. E2C od AWS) lub serwery chmurowe oferowane przez dostawców usług hostingowych.
- **PaaS (ang. *Platform as a Service*)** — dostęp do zdefiniowanych usług i mikroserwisów, które mogą być wykorzystane do zbudowania własnego rozwiązania aplikacyjnego. Jest to najpopularniejszy model usług chmurowych, najczęściej stosowany w procesie prototypowania czy uruchamiania aplikacji produkcyjnych. W ramach modelu PaaS zyskujemy dostęp do platformy umożliwiającej konfigurację i powiązanie ze sobą poszczególnych elementów składowych aplikacji (np.: usług komunikacyjnych, bazy danych i wizualizacji) w celu zbudowania kompletnego rozwiązania realizującego niezbędne funkcjonalności. Przykładami platform udostępniających usługi w modelu PaaS są MindSphere, przemysłowa chmura obliczeniowa oferowana przez firmę Siemens, a także czołowe platformy chmurowe dostępne na rynku, takie jak Microsoft Azure, AWS czy Google Cloud Platform.
- **SaaS (ang. *Software as a Service*)** — najwyższy poziom dostępu, ograniczony wyłącznie do pojedynczej aplikacji uruchamianej w ekosystemie chmurowym. W ramach modelu SaaS nie mamy możliwości modyfikacji funkcjonalności aplikacji — możemy obsługiwać ją za pomocą udostępnionego interfejsu i jesteśmy ograniczeni funkcjonalnością narzuconą przez programistę. Ten model spotykamy bardzo często w życiu codziennym, korzystając z witryn internetowych czy mediów społecznościowych, które uruchomione są właśnie na platformach chmurowych. Przykładem tego typu usługi jest poczta Gmail dostarczana przez firmę Google czy wirtualny magazyn danych OneDrive od Microsoftu.

Zagłębiając się nieco bardziej w tematykę chmur obliczeniowych, możemy spotkać też inne modele oznaczone ogólnie jako **XaaS** (ang. *Anything as a Service*). Są to różnego rodzaju usługi uruchamiane w środowiskach chmurowych i udostępniające specyficzne funkcjonalności. Na przykład **DaaS** (ang. *Data as a Service*) może być usługą magazynowania danych w chmurze, **AIaaS** (ang. *Artificial Intelligence as a Service*) udostępnia algorytmy uczenia maszynowego, które możemy zintegrować z naszą aplikacją, a **PaaS** w rozumieniu *Payment as a Service* dostarcza mechanizmy płatności online. W odniesieniu do dostawców chmury publicznej, zwłaszcza w kontekście digitalizacji procesów przemysłowych, najczęściej

stosowanymi modelami są jednak PaaS i SaaS, z uwagi na łatwość wdrożenia i integracji z systemami automatyki. Z tych modeli będziemy korzystać w przykładach omawianych w dalszej części książki.

7.4. Integracja z chmurą obliczeniową

Chmury publiczne, jako platformy dostępne w globalnym Internecie, naturalnie wymagają dostępu do tej sieci z poziomu urządzeń klienckich. Aby więc zintegrować system automatyki z dowolną chmurą, musimy zadbać o zapewnienie stałego łącza internetowego w szafie sterowniczej. W zależności od aplikacji i możliwości urządzeń, jakimi dysponujemy, może być to łącze przewodowe, bezprzewodowe (wi-fi) czy oparte na sieci komórkowej. W przykładach do tego akapitu skupimy się głównie na integracji sterownika SIMATIC z rozmaitymi wariantami chmur obliczeniowych za pośrednictwem bramki IOT2050 (znanej Ci już z rozdziału 6.). To rozwiązanie zapewnia największą elastyczność w kontekście sposobu łączności z Internetem, a także obsługi danych w momencie utraty połączenia. Bramka udostępnia bowiem mechanizmy umożliwiające nawiązywanie połączenia za pomocą wszystkich trzech wariantów:

- **Połączenie przewodowe** — bramka IOT2050 jest wyposażona w dwie karty sieciowe z interfejsami Ethernet 1 Gbit. Możliwe jest zatem odseparowanie sieci lokalnej (z systemem automatyki) od globalnej sieci Internet i dodatkowe zabezpieczenie połączenia za pomocą zapory ogniowej konfigurowanej na poziomie systemu operacyjnego.
- **Połączenie bezprzewodowe (wi-fi)** — urządzenie jest wyposażone zarówno w porty USB umożliwiające podłączenie miniaturowego modemu wi-fi, jak i złącze miniPCI Express z dedykowanymi wyjściami antenowymi w obudowie. Możemy więc rozbudować je o dodatkowe moduły bez wpływania na zewnętrzne gabaryty.
- **Połączenie oparte na sieci komórkowej** — urządzenie ma zintegrowany slot na kartę micro SIM, który połączony jest wewnętrznie z interfejsem miniPCI Express. Możliwe jest zatem rozbudowanie go o dodatkowy modem 4G i wykorzystanie łączności z siecią nawet tam, gdzie nie da się doprowadzić stałego łącza przewodowego lub udostępnić sieci wi-fi.

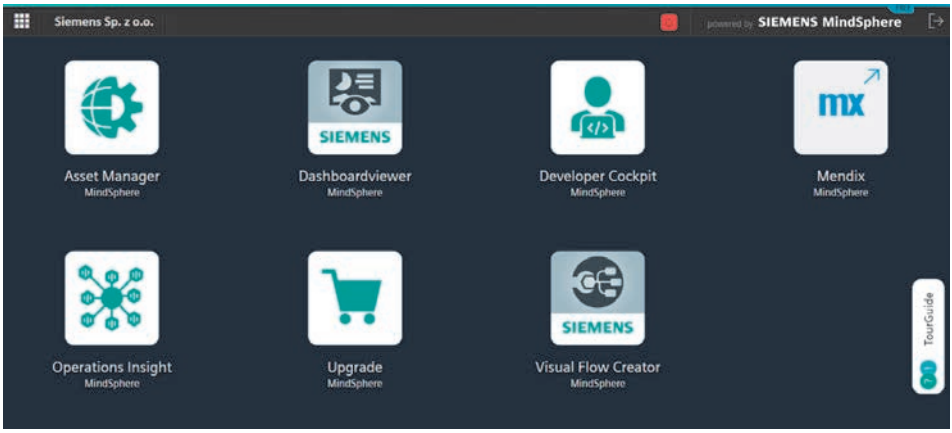
Metodyka łączenia z chmurą obliczeniową uzależniona jest w głównej mierze od platformy, uruchomionych usług czy funkcjonalności zaimplementowanych we własnej aplikacji chmurowej. Wszyscy dostawcy chmury publicznej oferują jednak usługi dedykowane dla rozwiązań IoT (np. IoT Hub na platformie Azure czy IoT Core oferowany przez AWS). Umożliwiają one nawiązywanie szyfrowanej łączności z chmurą na podstawie popularnych i lekkich protokołów komunikacyjnych, takich jak **MQTT**, **AMQP** (ang. *Advanced Message Queuing Protocol*) czy **HTTPS**. Możliwe jest oczywiście uruchomienie własnej usługi odpowiedzialnej za komunikację i implementację innych protokołów, w tym OPC UA czy Modbus TCP. Nie były one jednak projektowane z myślą o transferze dużych serii danych, zatem nie są zalecane w tego typu rozwiązaniach.

Niektóre systemy chmurowe, takie jak platforma MindSphere, poza podstawowymi protokołami IoT oferują także łączność opartą na dedykowanych urządzeniach zwanych **MindConnect**. Są to niewielkich rozmiarów mikrokomputery konfigurowane w pełni z poziomu interfejsu chmury, działające na zasadzie *plug & play*. Nie wymagają więc programowania komunikacji na poziomie urządzeń automatyki lub bramek — same przejmują tę rolę, dbając o nieprzerwany transfer danych i buforowanie w przypadku utraty połączenia internetowego.

7.4.1. MindSphere — platforma IoT dla przemysłu

Gdy w 2017 r. Siemens ogłosił start nowej platformy chmurowej dedykowanej dla przemysłu, minęło zaledwie sześć lat od targów w Hanowerze, podczas których ukuto termin Przemysłu 4.0, i zaledwie trzy lata od przedstawienia założeń do tej koncepcji przez grupę roboczą kierowaną przez Siegfrieda Daisa z firmy Bosch. I choć pierwsze rozwiązania wpisujące się w definicję internetu rzeczy zaczęły powstawać już pod koniec XX w., w roku 2017 wciąż nie były na tyle spopularyzowane, żeby platforma ta przeszła bez echa.

MindSphere jest bowiem systemem określanym jako ***IIoT as a Service*** (przemysłowy internet rzeczy w formie usługi). Dostarcza zatem gotowe rozwiązania umożliwiające łatwą akwizycję, archiwizację, wizualizację i analitykę danych na podstawie zaawansowanych algorytmów sztucznej inteligencji. Tym, co odróżnia system Siemensa od innych dostawców chmury publicznej, jest forma oferowanych usług. Zaprojektowany w modelu PaaS MindSphere to tak naprawdę platforma uruchomieniowa dla aplikacji informatycznych przeznaczonych dla przemysłu (rysunek 7.3). Część z nich dostarczana jest bezpośrednio przez producenta, część to rozwiązania firm zewnętrznych oferowane i sprzedawane w oficjalnym sklepie z aplikacjami. W kontekście architektury MindSphere przypomina więc bardziej mobilny system operacyjny niż konwencjonalną chmurę obliczeniową.



Rysunek 7.3. *Interfejs środowiska MindSphere*

Jedną z niewątpliwych zalet rozwiązania jest możliwość tworzenia kont i subkont w kontekście aplikacji i danych. Możliwa jest zatem integracja platformy z maszynami sprzedawanymi do klientów końcowych i udostępnienie im w pełni niezależnych od siebie pulpitów z opcją monitorowania tychże, przy jednoczesnym zachowaniu możliwości wglądu w stan pracy każdej z maszyn. Kolejnym plusem jest bogata biblioteka aplikacji dla różnych sektorów przemysłu i różnorodnych potrzeb klientów. Dane mogą być monitorowane, analizowane z użyciem zaawansowanych algorytmów, a maszyny diagnozowane zdalnie z wykorzystaniem mechanizmów *predictive maintenance*. Co więcej, w przypadku konieczności zbudowania własnego rozwiązania na platformę możliwe jest użycie oprogramowania low-code Mendix i bezpośredni transfer przygotowanej w ten sposób aplikacji do biblioteki MindSphere.

Rozwiązanie Siemensa uruchamiane jest na podstawie infrastruktury największych dostawców chmury publicznej. Decydując się na nie, możemy zdecydować, u jakiego dostawcy chcemy hostować naszą usługę i w jakich lokalizacjach mają być przechowywane nasze dane. Do wyboru dostępne są platformy AWS i Azure, a także Alibaba Cloud, jeśli aplikacja jest przeznaczona na rynek azjatycki. Sama platforma uruchamiana jest w przeglądarce internetowej, zatem dostęp do danych możliwy będzie z dowolnego urządzenia mobilnego lub komputera. Łączność z systemem automatyki może być nawiązywana za pomocą dedykowanych modułów MindConnect, popularnych protokołów IoT, takich jak np. MQTT, bibliotek przeznaczonych dla środowisk programistycznych, zwanych MindConnect SDK, oraz API udostępnianego przez platformę.

7.4.1.1. Przykład: integracja z chmurą MindSphere przy wykorzystaniu bramki IOT2050, środowiska Node-Red i MindConnect API

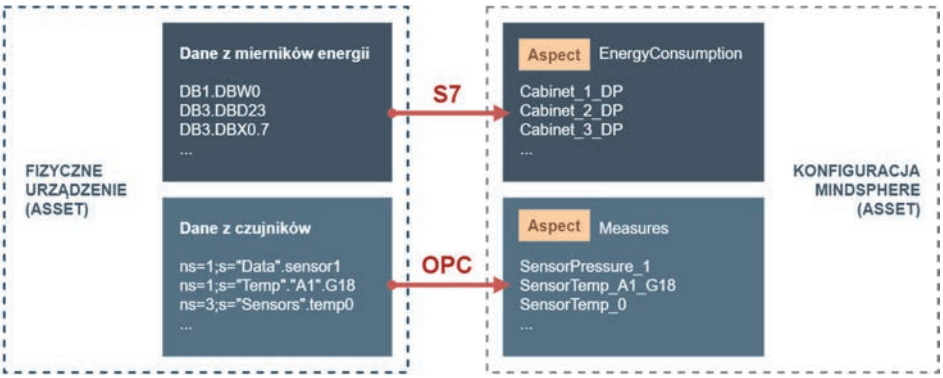
Platforma MindSphere, choć podobnie jak inne publiczne usługi chmurowe jest płatna, oferuje możliwość rejestracji darmowego konta testowego. Daje ono dostęp do kilku podstawowych usług oferowanych w ramach platformy, m.in. środowisk low-code Visual Flow Creator (odpowiednik Node-Red) i Mendix oraz aplikacji takich jak Operations Insight czy Asset Manager. W przykładzie wykorzystamy ostatnią z nich w celu konfiguracji i podłączenia nowego urządzenia IOT2050, wykorzystywanego w rozdziale 6.

Darmowe konto na platformie MindSphere możesz zarejestrować po zeskanowaniu kodu QR lub po wejściu na stronę platformy i kliknięciu przycisku *Start for free*. W przeciwieństwie do dostawców chmury publicznej, przy rejestracji nie jest wymagane podawanie danych rozliczeniowych, nie musisz się zatem martwić naliczaniem nieprzewidzianych kosztów.

Po założeniu konta uruchom platformę MindSphere w przeglądarce. Zostaniesz przekierowana/przekierowany do ekranu startowego, takiego jak na rysunku 7.3. W ramach darmowego konta dostępnych jest bardzo niewiele aplikacji, są one jednak wystarczające

do zapoznania się ze środowiskiem, a nawet prostego monitorowania danych pochodzących z urządzeń. Możesz zrealizować takie zagadnienie np. w aplikacji **Asset Manager**. Umożliwia ona zdefiniowanie punktów dostępowych do urządzeń IoT i struktur danych dostarczanych przez nie do chmury. MindSphere wykorzystuje ustrukturyzowany i uporządkowany model danych (rysunek 7.4), w którym można wyróżnić tzw.:

- **Assets** (assety, urządzenia) — odpowiedniki fizycznych komponentów używanych w systemie automatyki. W MindSphere assety to tak naprawdę logiczne modele urządzeń IoT odpowiedzialnych za akwizycję i przesyłanie danych do chmury. Mogą to być fizyczne sterowniki PLC z funkcjonalnością klienta MQTT, dedykowane komputery MindConnect lub biblioteki programistyczne wykorzystywane do łączności z poziomu własnych aplikacji.
- **Aspects** (aspekty) — logiczne modele danych archiwizowanych i przetwarzanych w kontekście konkretnego zagadnienia. Mogą to być pakiety danych informujących o przebiegu produkcji, zużyciu energii bądź innych mediów.

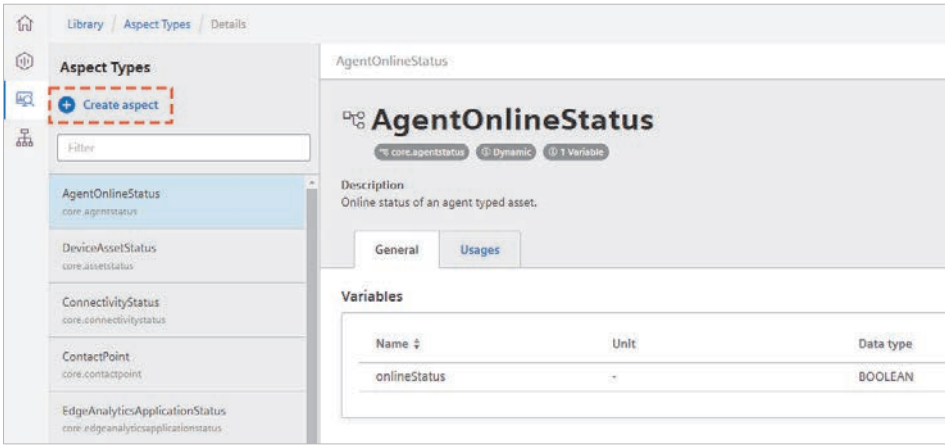


Rysunek 7.4. *Model mapowania struktur danych w środowisku MindSphere na fizyczne urządzenia i maszyny*

W naszym wypadku funkcję assetu przejmie bramka IoT gromadząca dane z PLC, a aspektem będą zagregowane, losowe dane, symulujące konkretne parametry, np. jakość bądź efektywność produkcji. Przejdźmy więc do konfiguracji naszego środowiska chmurowego.

Krok 1. Utworzenie nowego modelu danych w aplikacji Asset Manager

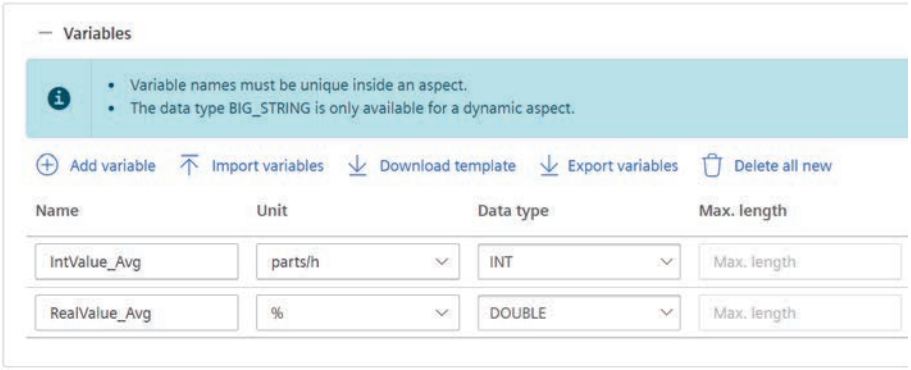
Uruchom aplikację Asset Manager. W pierwszej kolejności zdefiniujemy aspekt, który wykorzystamy następnie jako bazową strukturę danych przy konfiguracji nowego połączenia z bramką IoT. Przejdź zatem do zakładki *Aspects*, klikając widoczny na ekranie głównym przycisk *View your aspects*. Jak możesz zauważyć w kolumnie z lewej strony okna, konto testowe oferuje kilka predefiniowanych modeli. Żaden z nich nie odpowiada jednak danym generowanym przez naszą bramkę, musimy więc utworzyć nowy, własny model. W tym celu kliknij przycisk *Create aspect* (rysunek 7.5).



RYSUNEK 7.5. *Dodawanie nowego aspektu w aplikacji Asset Manager*

Na ekranie definicji nowego aspektu wpisz jego nazwę, opis, który pozwoli zorientować się w przeznaczeniu danego modelu danych, a także wybierz rodzaj aspektu. Ponieważ dane wysyłane przez bramkę IoT są danymi zmiennymi w dziedzinie czasu, zaznacz opcję *Dynamic*. Aspekty statyczne przeznaczone są dla danych, które nie ulegają zmianom, np. informacji o maszynie czy danych firmy.

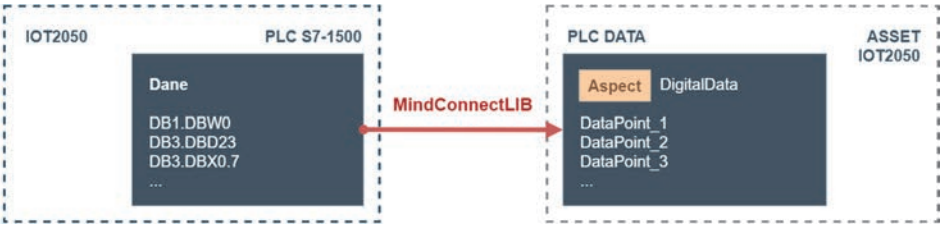
Następnie przejdź do definicji zmiennych składających się na nasz aspekt (rysunek 7.6). W przykładzie zdefiniujemy dwie zagregowane zmienne, pochodzące z funkcji Node-Red wyliczającej średnią arytmetyczną. Aby dodać je do modelu, kliknij przycisk *Add variable* i określ nazwę (np. `Digital_DATA`), typ oraz, opcjonalnie, jednostkę, w której reprezentowana będzie dana wartość.



RYSUNEK 7.6. *Definiowanie zmiennych dostępnych w ramach aspektu. Logiczna struktura tego elementu przypomina nieco zmienne UDT tworzone w środowisku TIA Portal*

Zatwierdź zmiany przyciskiem *Save*. Nowo utworzony aspekt powinien pojawić się na liście po lewej stronie okna.

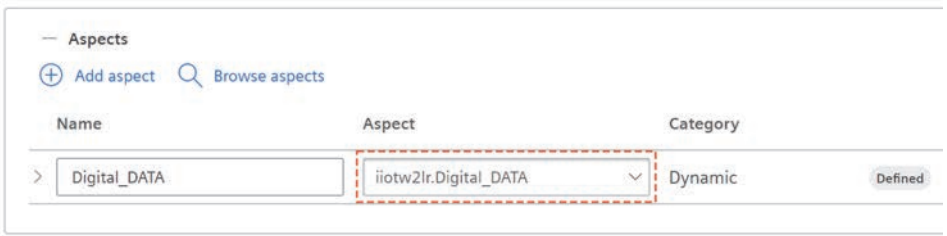
Przed utworzeniem nowego assetu przygotujemy nowy model obiektu. W przykładzie posłużymy się dwoma rodzajami assetów — jeden z nich będzie odpowiedzialny za obsługę połączenia pomiędzy bramką i chmurą, a drugi za wymianę danych między nimi. W praktyce architektura przedstawiona na rysunku 7.7 znajduje odzwierciedlenie w fizycznych urządzeniach, gdzie jeden z obiektów odpowiada bramce IoT, a drugi PLC.



Rysunek 7.7. *Mapowanie fizycznych urządzeń i zestawu danych na logiczne assety i aspekty w środowisku MindSphere*

Krok 2. Definiowanie urządzeń (assetów)

Przed przystąpieniem do konfiguracji logicznego połączenia pomiędzy bramką i chmurą zdefiniujemy nowy model obiektu reprezentującego dane pochodzące ze sterownika PLC. W tym celu przejdź do zakładki *Library/Asset types* dostępnej w nawigacji i kliknij przycisk *Create type*. Wpisz nazwę modelu i, opcjonalnie, opis. Następnie rozwiń zakładkę *Aspects* i klikając przycisk *Add aspect*, wybierz z listy utworzony wcześniej model danych (rysunek 7.8). Zatwierdź zmiany przyciskiem *Save*.

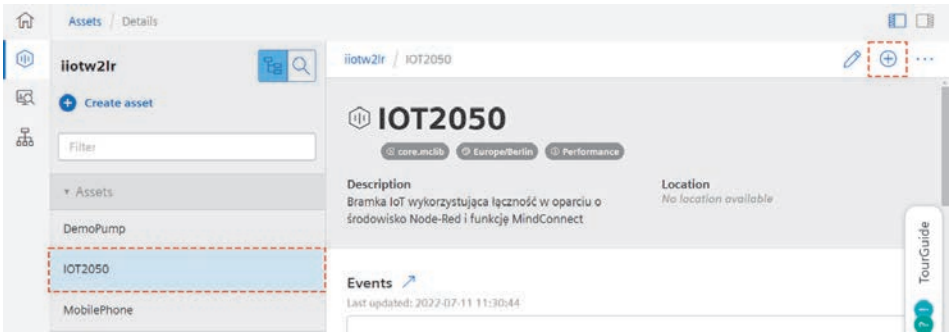


Rysunek 7.8. *Wybór aspektu w ramach nowo tworzonego assetu odpowiedzialnego za reprezentację sterownika PLC w chmurze*

Kolejnym etapem będzie zdefiniowanie nowego assetu, odpowiedzialnego za komunikację z bramką IoT. Przejdź zatem do zakładki *Assets* dostępnej w nawigacji. Klikając przycisk *Create asset*, uruchom panel konfiguracji nowego urządzenia. Z listy dostępnych mechanizmów komunikacyjnych z chmurą wybierz pozycję *MindConnectLib* — pozwoli ona na nawiązanie połączenia za pomocą funkcji dostępnej w środowisku Node-Red. Zatwierdź wybór przyciskiem *Create*. Podobnie jak przy konfiguracji dowolnego elementu w chmurze tutaj również niezbędne jest zdefiniowanie podstawowych danych informacyjnych, takich jak nazwa (np. IOT2050) czy opis. W przypadku tego rodzaju assetu nie ma możliwości określenia zmiennych, które będą wymieniane

między urządzeniem IoT i platformą — zrobimy to w następnym kroku. Zatwierdź zmiany, klikając przycisk *Save*. Nowo utworzone urządzenie pojawi się na liście po lewej stronie okna.

Po kliknięciu urządzenia wyświetlony zostanie ekran informujący o aktualnym statusie połączenia czy danych wymienianych między urządzeniem a chmurą. Aby dane faktycznie mogły być przesłane i odczytane, niezbędne jest dodanie nowego, podległego assetu, zgodnego z utworzonym wcześniej typem `PLC_Data`. Aby dodać element, kliknij ikonę plusa (*Add child asset*) widoczną w prawym górnym rogu ekranu (rysunek 7.9).



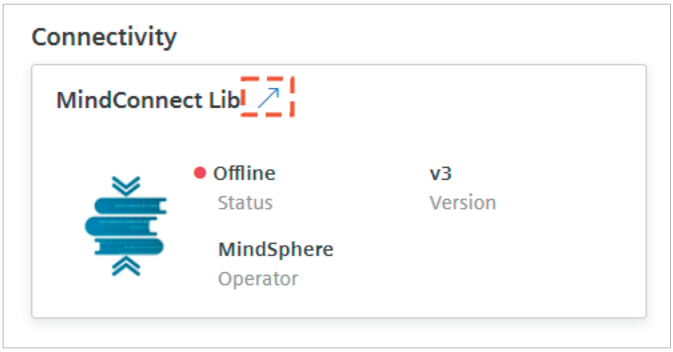
Rysunek 7.9. *Dodawanie assetu reprezentującego sterownik PLC, podległego pod główny asset odpowiedzialny za komunikację z IOT2050*

Postępując tak jak przy tworzeniu assetu IOT2050, dodaj nowy obiekt, tym razem wybierając z listy typ PLC_Data utworzony chwilę wcześniej. Zwróć uwagę, że po utworzeniu obiektu Twoje urządzenie zyskało strukturę drzewiastą, analogiczną do schematu przedstawionego na rysunku 7.7.

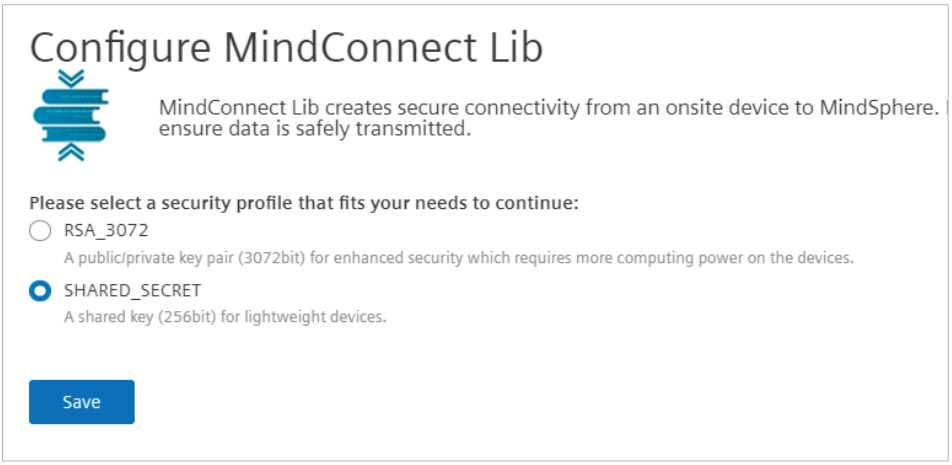
Krok 3. Łączenie bramki IoT z chmurą MindSphere

Po zdefiniowaniu struktury urządzeń i danych zgodnie z opisem w krokach 1 i 2 możesz przejść do konfiguracji połączenia pomiędzy bramką IoT i chmurą. Jest ono definiowane na poziomie nadrzędnego urządzenia, zdefiniowanego z typem *MindConnectLib* — w przykładzie nosi on nazwę IOT2050. Przejdź zatem do widoku tego assetu w platformie MindSphere i na ekranie informacyjnym znajdź pole o nazwie *Connectivity*. Klikając niebieską strzałkę, przejdź do konfiguracji połączenia (rysunek 7.10).

Pierwszym krokiem konfiguracji jest określenie sposobu zabezpieczenia transmisji danych — dostępne są mechanizmy wykorzystujące szyfrowanie asymetryczne (oparte na parze kluczy, prywatnym i publicznym), a także symetryczne, wykorzystujące jeden 256-bitowy klucz. Opcja ta nosi nazwę *SHARED_SECRET* i jest przeznaczona dla niewielkich i prostych urządzeń IoT — użyjemy jej w przypadku bramki IOT2050 (rysunek 7.11). Zaznacz opcję i przejdź dalej, klikając przycisk *Save*.



RYSUNEK 7.10. *Konfiguracja połączenia z biblioteką MindConnectLib*

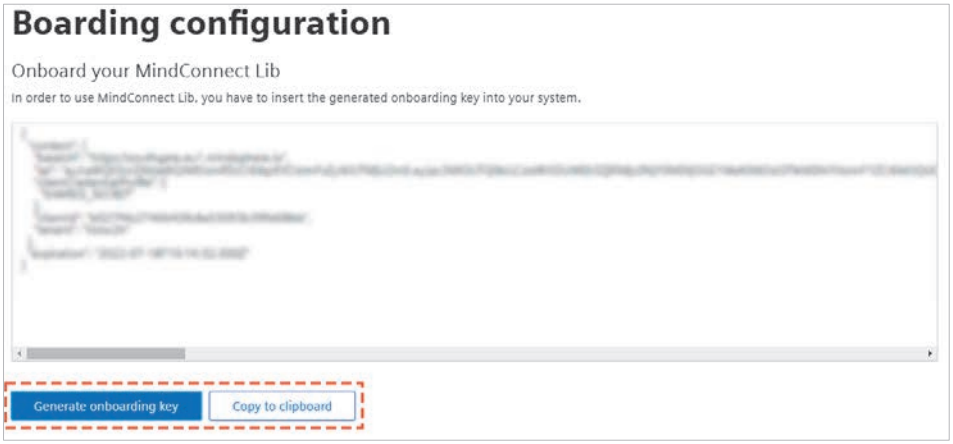


RYSUNEK 7.11. *Wybór sposobu szyfrowania — w przykładzie zdecydowano się na szyfrowanie symetryczne, oparte na jednym 256-bitowym kluczu zwanym SHARED_SECRET*

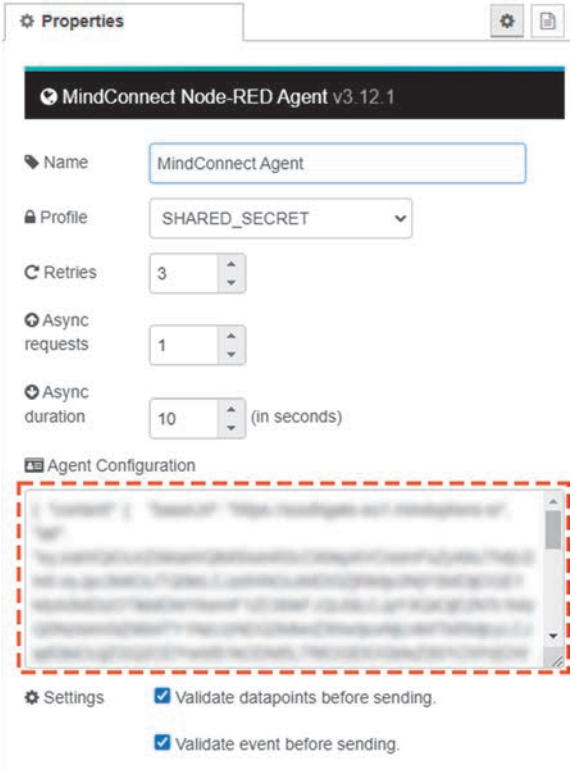
Na kolejnym ekranie MindSphere pozwoli Ci wygenerować dane konfiguracyjne w postaci obiektu JSON, które wykorzystamy następnie do szybkiej konfiguracji bloku `mindconnect` w środowisku Node-Red (rysunek 7.12). Kliknij przycisk *Generate onboarding key*, a następnie skopiuj wygenerowany obiekt do schowka.

Możesz teraz przejść do konfiguracji połączenia i wymiany danych w środowisku Node-Red uruchomionym na Twojej bramce IOT2050.

Dodaj do obszaru roboczego blok `mindconnect` dostępny w bibliotece. Klikając go dwukrotnie, przejdź do konfiguracji (rysunek 7.13) i wklej skopiowane wcześniej dane konfiguracyjne do dedykowanego pola w bloku, po czym zatwierdź zmiany przyciskiem *Done*.

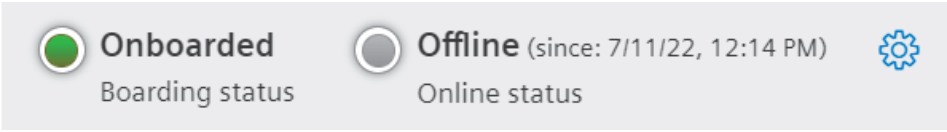


RYSUNEK 7.12. *Generowanie gotowych danych konfiguracyjnych, przeznaczonych do konfiguracji biblioteki MindConnectLib*



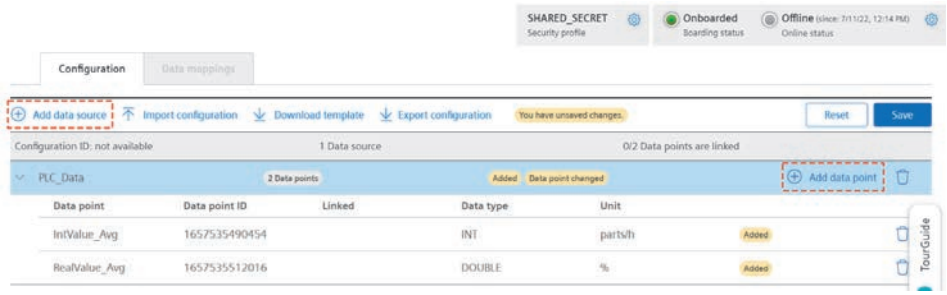
RYSUNEK 7.13. *Konfiguracja bloku mindconnect w środowisku Node-Red. Blok oparty jest na bibliotece MindConnectLib*

Uruchom zmodyfikowany projekt, klikając przycisk *Deploy*. Jeśli Twoje urządzenie IoT jest podłączone do sieci Internet, po chwili powinno nawiązać połączenie z chmurą MindSphere, co zostanie zasygnalizowane zarówno w bloku `mindconnect`, jak i w konfiguracji urządzenia w chmurze. Jeśli urządzenie zyskało status *Onboarded* (rysunek 7.14), proces łączenia przebiegł pomyślnie. Status *Offline* na tym etapie jest nieistotny — zmieni się dopiero wtedy, kiedy do chmury zaczną napływać dane.



Rysunek 7.14. *Status poprawnego połączenia urządzenia IoT z chmurą*

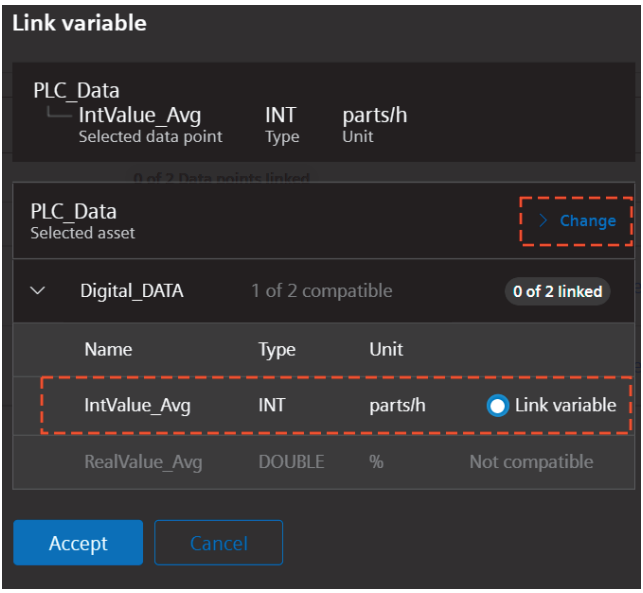
Po nawiązaniu łączności z chmurą konieczne jest dostosowanie konfiguracji assetu tak, aby dane wysyłane przez urządzenie IoT były odpowiednio mapowane na dane zdefiniowane w aspekcie. Możesz to zrobić w ustawieniach assetu nadrzędnego, przechodząc do zakładki *Configuration* (rysunek 7.15).



Rysunek 7.15. *Konfiguracja zmiennych wymienianych w ramach komunikacji*

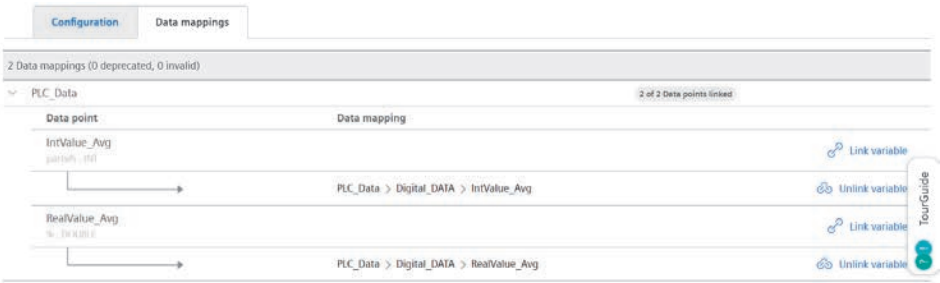
Zdefiniuj nowe źródło danych i zmienne, które będą za jego pomocą wysyłane. Jest to struktura, którą w sposób automatyczny będzie można pobrać z bloku `mindconnect` w środowisku Node-Red i na którą fizycznie trafiać będą dane z bramki. Aby móc obserwować te dane w aplikacji Asset Manager, konieczne jest ich przypisanie do odpowiednich elementów aspektu zdefiniowanego w kroku 1. Jeśli zdefiniowałaś/zdefiniowałeś poprawne dane, zatwierdź zmiany przyciskiem *Save*.

Następnie przejdź do zakładki *Data mappings*, w której możesz połączyć zdefiniowaną przed momentem strukturę z aspektem. Korzystając z opcji *Link variable*, utwórz powiązanie pomiędzy zmiennymi (rysunek 7.16). Jeśli w oknie wyboru zmiennej domyślnie wyświetlany jest nadrzędny asset, bez określonego modelu danych, zmień go, korzystając z przycisku *Change*.



RYSUNEK 7.16. *Mapowanie zmiennych skonfigurowanych w ustawieniach komunikacji na zmienne zdefiniowane w aspekcie PLC_Data*

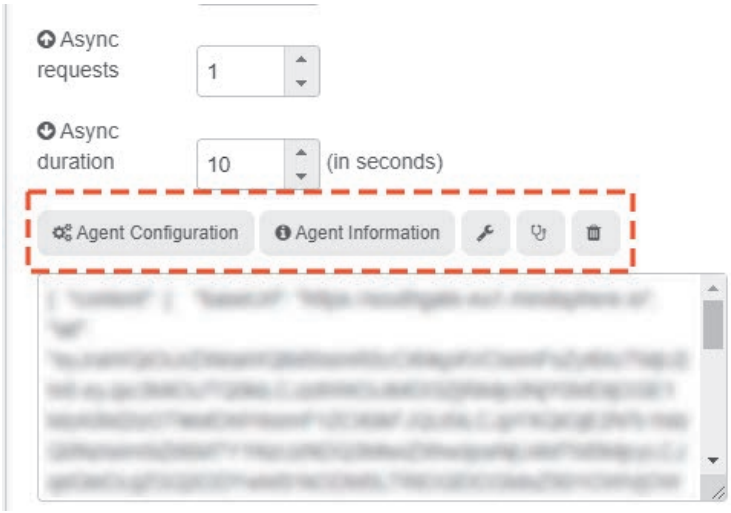
Efektem powyższych działań powinna być w pełni zdefiniowana struktura powiązanych ze sobą zmiennych (rysunek 7.17).



RYSUNEK 7.17. *Gotowa struktura zmiennych poprawnie powiązanych z modelem danych w aplikacji Asset Manager*

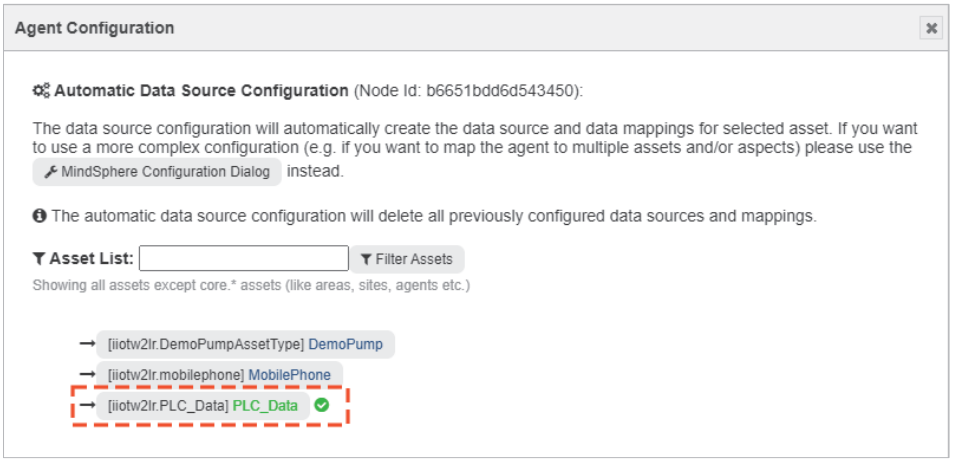
Krok 4. Wysyłanie danych do chmury i monitorowanie w aplikacji Asset Manager

Jeśli poprawnie skonfigurowałaś/skonfigurowałeś strukturę obiektów w chmurze MindSphere, wróć do środowiska Node-Red. Po wejściu w ustawienia bloku `mindconnect` możesz zauważyć, że pojawiły się w nim nowe opcje, niedostępne przed wykonaniem połączenia. Najważniejsze z nich to *Agent Configuration* i *Agent Information*, ułatwiające pracę z modelem danych MindSphere (rysunek 7.18).



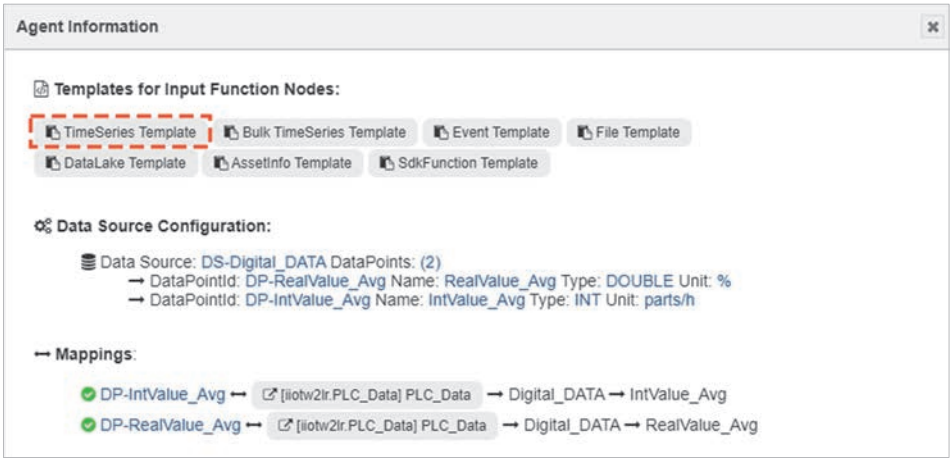
RYSUNEK 7.18. *Dodatkowe funkcje, które pojawiają się w bloku mindconnect po nawiązaniu połączenia z chmurą*

Kliknij pierwszą opcję, *Agent Configuration*, aby określić asset, do którego odwoływać się będzie blok. Z listy dostępnych obiektów wybierz ten o nazwie *PLC_Data* (rysunek 7.19).



RYSUNEK 7.19. *Wybór assetu w bloku mindconnect*

Następnie przejdź do opcji Agent Information i korzystając z przycisku *TimeSeries Template*, skopiuj gotowy szablon funkcji (rysunek 7.20), która powinna zostać podłączona do parametru wejściowego bloku `mindconnect`, aby dane były wysyłane do chmury.



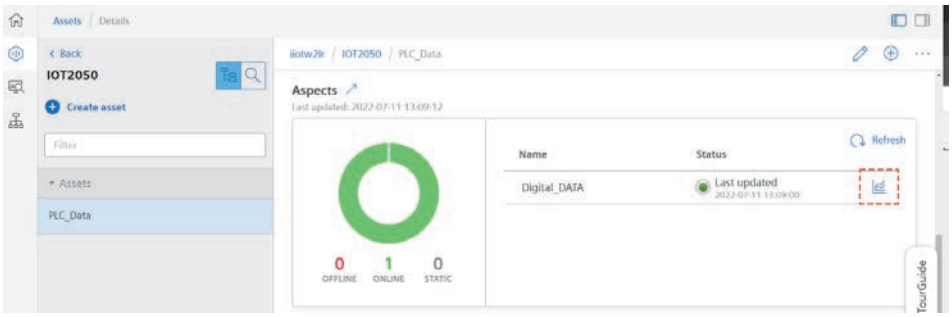
RYSUNEK 7.20. *Kopiowanie szablonu predefiniowanego obiektu, który należy wysłać do chmury, aby dane zostały poprawnie przetworzone*

Wstaw blok funkcyjny do obszaru roboczego Node-Red i wklej do niego skopiowany wcześniej szablon. Zawiera on tablicę obiektów o nazwie `values`. Każdy obiekt składa się z identyfikatora zmiennej w MindSphere (pobieranego automatycznie przez blok `mindconnect` po wskazaniu assetu) i jej wartości, do której należy przypisać rzeczywiste dane pochodzące z PLC (rysunek 7.21).

```
const values = [
  {
    "dataPointId": "DP-RealValue_Avg",
    "qualityCode": "0",
    "value": msg.payload.RealValue_0_Avg.toString()
  },
  {
    "dataPointId": "DP-IntValue_Avg",
    "qualityCode": "0",
    "value": Math.floor(msg.payload.IntValue_1_Avg).toString()
  }
]
msg._time = new Date()
msg.payload=values
return msg
```

RYSUNEK 7.21. *Modyfikacja skopiowanego szablonu poprzez przypisanie rzeczywistych wartości do kluczy value*

Dane gromadzone w ten sposób w chmurze możesz monitorować po otwarciu assetu `PLC_Data` i kliknięciu odnośnika *Open timeseries chart*, widocznego na rysunku 7.22.



Rysunek 7.22. *Uruchamianie podglądu przebiegów zmiennych wysyłanych do chmury w ramach assetu IOT2050*

MindSphere wyświetli prosty wykres przedstawiający wartości zebrane w ciągu ostatniej godziny. Archiwizowane w ten sposób dane dostępne są następnie dla innych aplikacji uruchomionych na platformie.

7.4.2. Integracja z chmurami publicznymi Azure i AWS

Rynek usług chmurowych jest obecnie bardzo obszerny i jak wynika z rysunku 7.1 przedstawionego na początku rozdziału, rok do roku rośnie w tempie wykładniczym. Nic więc dziwnego, że wybór dostawcy chmury publicznej może przyprawić o ból głowy. Jak w każdej branży, tak i tutaj istnieją rankingi określające udział danego dostawcy w rynku, a także sumaryczną ocenę oferowanych usług. Jak wynika z raportu firmy Gartner, pierwsze trzy miejsca niezmiennie są zarezerwowane dla czołowych graczy, takich jak Amazon z platformą Amazon Web Services (AWS), Microsoft z Azure i Google z chmurą Google Cloud Platform (GCP) (tabela 7.2). Poszukując rozwiązań dedykowanych dla naszej branży lub analizując studia przypadków wdrożenia chmury u innych klientów, możemy spotkać się także z takimi dostawcami jak Alibaba, Oracle, IBM czy VMware (platforma do rozwiązań wirtualizacyjnych oferowana w modelu IaaS).

Decyzja o wyborze dostawcy uzależniona jest zazwyczaj od indywidualnych preferencji, umów biznesowych pomiędzy firmami, a także usług oferowanych w ramach danej platformy. Nie bez znaczenia są oczywiście koszty, a te mogą się różnić w zależności od dostawcy, stopnia zużycia zasobów (często spotykany jest model, w którym wzrost zużycia powoduje spadek ceny jednostkowej — np. w chmurze AWS im więcej plików przechowujesz na dyskach S3, tym niższy jest koszt 1 Gb danych) oraz, przede wszystkim, sposobu konfiguracji i wykorzystania usług. W niektórych przypadkach kryterium wyboru będzie profil i renoma danego dostawcy. Jeśli chcesz skorzystać z zaawansowanych i rozwiniętych algorytmów uczenia maszynowego, najprawdopodobniej sięgniesz po usługi oferowane w ramach Google Cloud Platform, z uwagi na duży udział firmy Google w rozwoju i implementacji sieci neuronowych.

TABELA 7.2. *Zestawienie najpopularniejszych dostawców usług chmurowych według rankingu firmy Gartner*

Platforma	Oferowane modele usług	Czy zawiera usługi dedykowane dla IoT?	Udział w rynku
Amazon Web Services	IaaS, PaaS, SaaS, DaaS	Tak, IoT Core	33%
Microsoft Azure	IaaS, PaaS, SaaS, DaaS	Tak, IoT Hub	21%
Google Cloud Platform	IaaS, PaaS, SaaS, DaaS	Tak, IoT Core	10%
Alibaba Cloud	IaaS, PaaS, SaaS, DaaS	Tak, IoT Platform	6%
IBM Cloud	IaaS, PaaS, SaaS	Tak, IBM Watson IoT Platform	4%
Oracle Cloud	IaaS, PaaS, SaaS, DaaS	Tak, IoT Intelligent Applications	2%

W dalszej części rozdziału skupimy się głównie na integracji z chmurami AWS, Azure i GCP. Oferują one najszersze zestawy usług, dedykowanych nie tylko rozwiązaniom IoT, ale przeznaczonych także do magazynowania danych, budowania aplikacji, wirtualizacji czy skalowania zasobów w zależności od zużycia. Każda z tych platform oferuje też darmowe plany testowe, dzięki którym możesz się zapoznać z ich funkcjonalnością i dostępnymi usługami bez ponoszenia opłat.

Większość dostawców oferuje darmowy plan wyłącznie na ograniczony czas, zazwyczaj 12 miesięcy. Część z nich dostarcza także pewną ilość środków, które możesz przeznaczyć na korzystanie z usług (np. po założeniu darmowego konta w Azure otrzymujesz 200 dolarów do wykorzystania w ciągu 30 dni). Możesz także spotkać się z planami, w ramach których nie otrzymujesz środków, ale masz prawo korzystać z zasobów wyłącznie do pewnego poziomu zużycia (np. w chmurze AWS). Pamiętaj jednak, że przy zakładaniu darmowego konta zwykle niezbędne jest podanie danych Twojej karty kredytowej lub płatniczej w celu realizacji rozliczeń po zakończeniu okresu próbnego. Jeśli więc nie chcesz już korzystać z chmury, deaktywuj konto przed upływem tego czasu. Zawsze też monitoruj aktualne zużycie zasobów, aby nie przekroczyć limitów oferowanych w ramach Twojego planu, co mogłoby spowodować naliczanie niechcianych opłat.

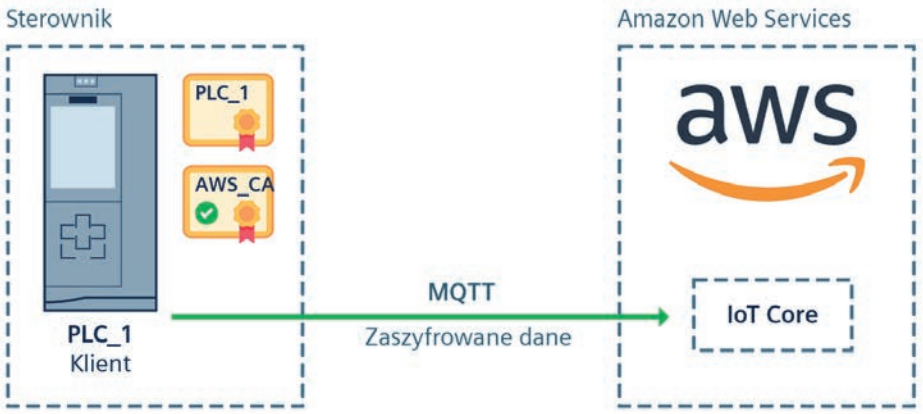
Jeśli chcesz dowiedzieć się więcej na temat darmowych planów oferowanych przez wspomnianych największych dostawców chmury publicznej, odwiedź ich strony internetowe.

7.4.2.1. Przykład: integracja z chmurą AWS i usługą IoT Core z poziomu sterownika SIMATIC

Z uwagi na stale rosnącą popularność chmury publicznej i coraz większy udział jej dostawców w rynku niektóre aplikacje informatyczne wykorzystywane w przemyśle zaczynają być powoli migrowane na tego typu platformy. Dotyczy to zarówno systemów

służących do zarządzania produkcją (MES czy ERP), jak i aplikacji uruchamianych dotychczas wyłącznie na poziomie lokalnym, takich jak SCADA. Nic dziwnego — praca zdalna i globalizacja sprawiają, że coraz częściej wgląd w parametry produkcji i kluczowe wskaźniki efektywności wymagany jest z dowolnego miejsca na świecie i o dowolnym czasie. Trend ten wymusza na programistach sterowników PLC coraz częstszą konieczność integracji konwencjonalnych urządzeń automatyki z serwerami w Internecie.

W poprzednich akapitach poznałeś/poznałaś sposób wymiany danych pomiędzy systemem automatyki a chmurą obliczeniową z wykorzystaniem bramki IoT i prostego środowiska niskokodowego Node-Red. Nie zawsze jednak zastosowanie bramki jest możliwe (z uwagi na brak miejsca w szafie) lub uzasadnione (jeśli sterownik ma dostęp do Internetu). W takiej sytuacji komunikacja pomiędzy systemem automatyki a chmurą może być nawiązana w sposób bezpośredni, przy użyciu poznanego wcześniej mechanizmu komunikacyjnego opartego na protokole MQTT (rysunek 7.23).

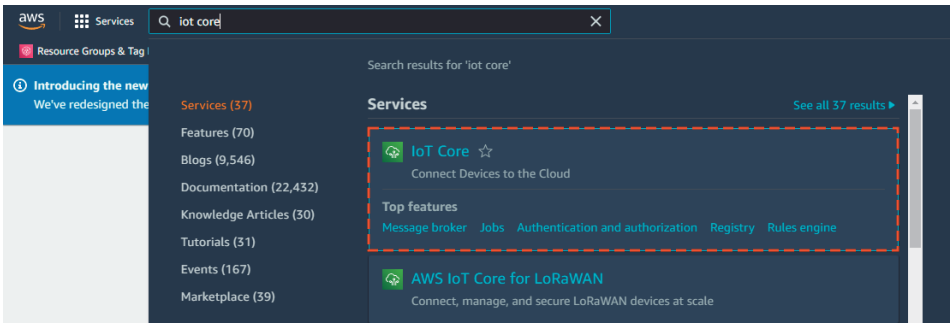


Rysunek 7.23. *Schemat omawianego przykładu aplikacji*

W przykładzie skupimy się na wysyłce danych pochodzących bezpośrednio ze sterownika PLC do usługi IoT Core dostępnej w ramach publicznej platformy chmurowej Amazon Web Services. Pełni ona funkcję brokera MQTT, gromadzącego dane z wielu różnych urządzeń IoT i udostępniającego je do innych usług uruchamianych w ramach platformy. Choć ćwiczenie oparte jest na ekosystemie AWS, architektura usług IoT jest podobna u wszystkich dostawców chmury. Podobnie też będzie wyglądał proces konfiguracji zabezpieczeń i parametryzacji funkcji na poziomie sterownika PLC.

Krok 1. Konfiguracja usługi IoT Core w chmurze AWS

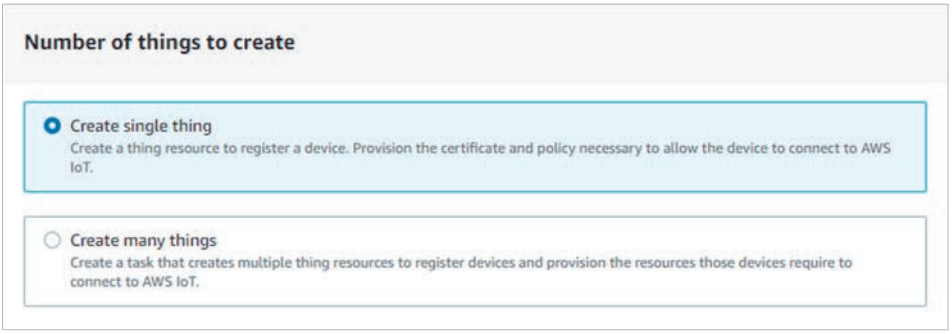
Jeśli masz już darmowe konto na platformie AWS, zaloguj się i korzystając z wyszukiwarki, przejdź do usługi IoT Core (rysunek 7.24).



RYSUNEK 7.24. *Usługa IoT Core dostępna w ramach platformy chmurowej AWS*

Umożliwia ona dodawanie nowych urządzeń IoT (pod ogólną nazwą *Things*, z definicji *Internet of Things*), definiowanie dla nich polityk dostępowych i podział na grupy. W celu zdefiniowania nowego urządzenia przejdź do zakładki *Manage/All devices/Things*, widocznej w menu nawigacyjnym z lewej strony okna, po czym kliknij przycisk *Create things*. Zostaniesz przekierowana/przekierowany do kreatora tworzenia nowego urządzenia.

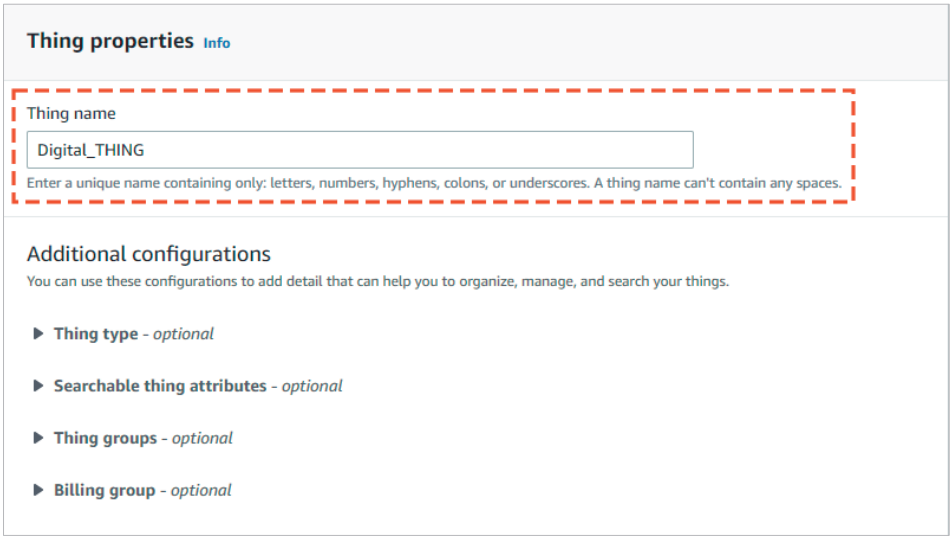
AWS umożliwia zarówno utworzenie pojedynczego elementu, jak i masowe dodawanie wielu urządzeń. Ta opcja może być przydatna przede wszystkim wtedy, kiedy potrzebujesz podłączyć do chmury większe grupy inteligentnych czujników bądź bramek IoT. W naszym przypadku połączenie będzie nawiązywane przez pojedynczy sterownik, wybierz więc opcję *Create single thing* i przejdź dalej (rysunek 7.25).



RYSUNEK 7.25. *IoT Core umożliwia dodanie zarówno pojedynczego urządzenia IoT, jak i całej serii urządzeń*

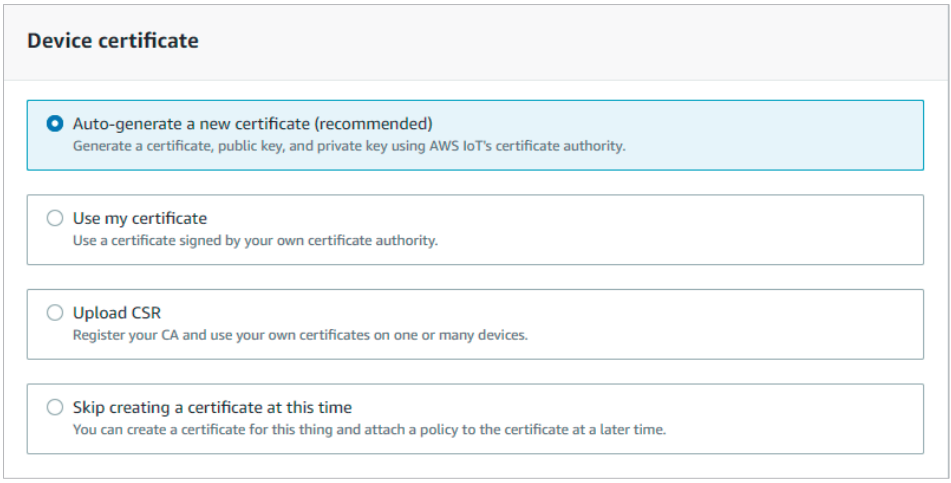
W kolejnym kroku IoT Core poprosi o zdefiniowanie nazwy urządzenia. W tym miejscu możesz także przypisać je do grupy lub określić dodatkowe atrybuty ułatwiające późniejsze wyszukiwanie i filtrowanie (rysunek 7.26). Pomiń je i przejdź dalej.

Ponieważ komunikacja z chmurą prawie zawsze realizowana jest w sposób niejawny, niezbędne jest przypisanie unikatowych certyfikatów X.509 do urządzenia. Będą one używane do uwierzytelniania naszego sterownika w trakcie nawiązywania łączności z chmurą,



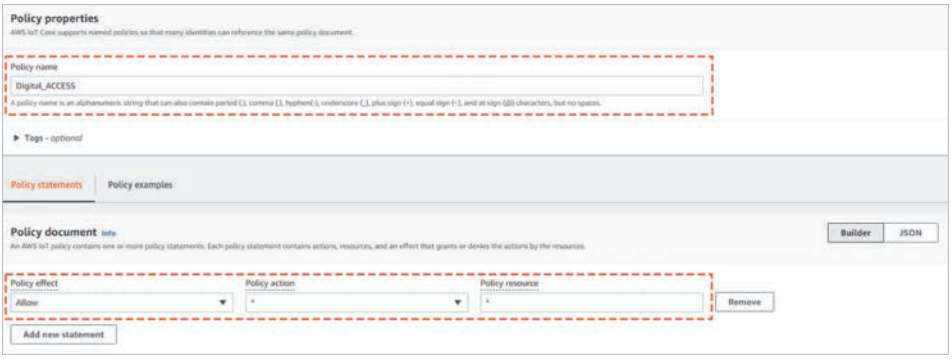
RYSUNEK 7.26. *Definiowanie podstawowych parametrów urządzenia*

a także do szyfrowania danych w procesie transmisji. AWS pozwala na zastosowanie własnych certyfikatów lub ich automatyczne wygenerowanie (rysunek 7.27). Wykorzystamy więc tę możliwość, aby nie martwić się dodatkowo pozyskiwaniem lub generowaniem dokumentów. Zaznacz opcję *Auto-generate a new certificate* i przejdź dalej.



RYSUNEK 7.27. *Generowanie nowych certyfikatów przypisanych do urządzenia zdefiniowanego w ramach usługi IoT Core*

Kolejny krok także jest związany z bezpieczeństwem i polega na określeniu polityk dostępowych do zasobów usługi IoT Core. Możesz dzięki nim zdefiniować uprawnienia dla poszczególnych urządzeń, np.: wyłącznie zapis danych, odczyt informacji statusowych, subskrypcja tematów z brokera. Przy pierwszej konfiguracji prawdopodobnie nie masz żadnej zdefiniowanej polityki, konieczne jest więc jej utworzenie. Kliknij przycisk *Create policy* — zostaniesz przekierowana/przekierowany do nowej zakładki, służącej do utworzenia zbioru zasad dostępowych. Zdefiniuj nazwę polityki i poziom uprawnień. Możesz je wybierać z listy predefiniowanych akcji dostępnych w polu *Actions*. Jeśli chcesz dodać kolejny rekord do polityki, możesz to zrobić za pomocą przycisku *Add new statement*. Na potrzeby przykładu i testów zdefiniuj pełen dostęp do zasobów IoT Core, wpisując w polach *Actions* i *Policy resource* znak * (rysunek 7.28). Pozwoli on na pełen dostęp do wszystkich zasobów usługi. Pamiętaj jednak, że podczas konfigurowania usługi w środowisku produkcyjnym udzielanie nieograniczonego dostępu nie jest dobrą praktyką, w myśl zasady najmniejszego uprzywilejowania (omawianej w rozdziale 4.). Po poprawnym zdefiniowaniu polityki potwierdź zmiany przyciskiem *Create* i wróć do poprzedniej karty w przeglądarce. Zaznacz w niej nowo utworzoną politykę i zakończ proces dodawania urządzenia, klikając przycisk *Create thing*.



RYSUNEK 7.28. *Definiowanie polityki dostępowej do usługi IoT Core z poziomu nowo utworzonego urządzenia*

Po dodaniu urządzenia AWS wyświetli okno dialogowe umożliwiające pobranie certyfikatów i kluczy wymaganych do poprawnego nawiązania połączenia. Pobierz pozycje opisane jako:

- *Device certificate* — certyfikat potwierdzający tożsamość urządzenia nawiązującego łączność z usługą IoT Core.
- *Root CA Certificates* — certyfikaty potwierdzające tożsamość serwera AWS, na którym uruchomiona jest usługa. Jest on niezbędny do realizacji połączenia opartego na funkcji `LMQTT_Client`. Pobierz certyfikat oznaczony jako CA 1.
- *Public key file* — klucz publiczny urządzenia, udostępniany usłudze do zaszyfrowania danych zwracanych do sterownika.

- *Private key file* — klucz prywatny urządzenia, wykorzystywany do odszyfrowania danych zwracanych przez serwer.

Jeśli wszystkie pliki zostały zapisane na Twoim dysku twardym, zakończ proces kliknięciem przycisku *Done*. Nowo utworzone urządzenie powinno pojawić się na liście w zakładce *Things*.

Krok 2. Generowanie certyfikatu urządzenia za pomocą narzędzia OpenSSL

W poprzednim kroku pobraliśmy zestaw certyfikatów i kluczy, które zostały wygenerowane automatycznie w usłudze IoT Core. Jednym z nich był certyfikat urządzenia — nie jest on jednak dostarczony w formacie kompatybilnym ze sterownikiem. Możliwe jest co prawda jego umieszczenie w globalnym menedżerze certyfikatów projektu, ale już lokalny menedżer urządzenia nie będzie w stanie go obsłużyć. Niezbędna jest zatem jego konwersja na postać archiwum zawierającego dodatkowo klucz prywatny, w pełni akceptowalnego przez TIA Portal.

W tym celu wykorzystamy darmowe narzędzie do tworzenia i obsługi certyfikatów SSL/TLS o nazwie OpenSSL. Narzędzie dystrybuowane jest bezpłatnie, na zasadach licencji Apache V2.0, i dostępne do pobrania z oficjalnej strony projektu.

Zbiór odnośników, z których możesz pobrać pliki binarne przeznaczone dla systemu Windows, dostępny jest na stronie OpenSSL Wiki.

Po instalacji uruchom wiersz poleceń i przejdź do lokalizacji, w której zostało zainstalowane narzędzie. Jeśli nie zmieniałaś/zmieniałeś jej w trakcie instalacji, możesz wykorzystać poniższą komendę:

```
cd C:/Program Files/OpenSSL-Win64/bin
```

Następnie uruchom polecenie:

```
openssl pkcs12 -export -out <ścieżkaZapisuPliku>\CertyfikatPLC.p12
-in <ścieżkaOdczytuPliku>\<indywidualnyIdUrządzeniaAWS>-certificate.pem.crt
-inkey <ścieżkaOdczytuPliku> \ <indywidualnyIdUrządzeniaAWS> -private.pem.key
```

gdzie:

- <*ścieżkaZapisuPliku*> — lokalizacja, w której zostanie zapisane nowo utworzone archiwum;
- <*ścieżkaOdczytuPliku*> — lokalizacja, w której zostały zapisane certyfikaty i klucze pobrane z AWS;

- *<indywidualnyIdUrządzeniaAWS>* — ciąg znaków zawarty w nazwie certyfikatu i kluczy urządzenia pobranych z AWS.

Przykładowe polecenie może przyjąć postać:

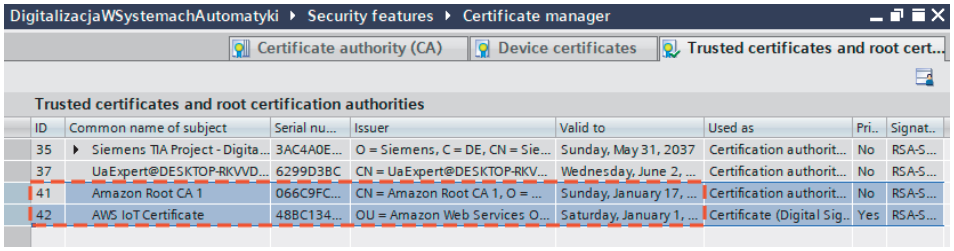
```
openssl pkcs12 -export -out C:\Users\Artur\Desktop\AWS\CertyfikatPLC.p12
-in C:\Users\Artur\Desktop\AWS\6f70d5da75-certificate.pem.crt
-inkey C:\Users\Artur\Desktop\AWS\6f70d5da75-private.pem.key
```

Po jego wykonaniu narzędzie OpenSSL poprosi o dwukrotne podanie hasła zabezpieczającego klucz prywatny zawarty w archiwum. Będzie ono niezbędne do zaimportowania pliku do środowiska TIA Portal. Jeśli operacja została wykonana poprawnie, w katalogu zdefiniowanym jako ścieżka zapisu pliku powinien się pojawić nowy plik archiwum kryptograficznego o nazwie *CertyfikatPLC.p12*.

Krok 3. Konfiguracja i parametryzacja funkcji LMQTT w sterowniku SIMATIC

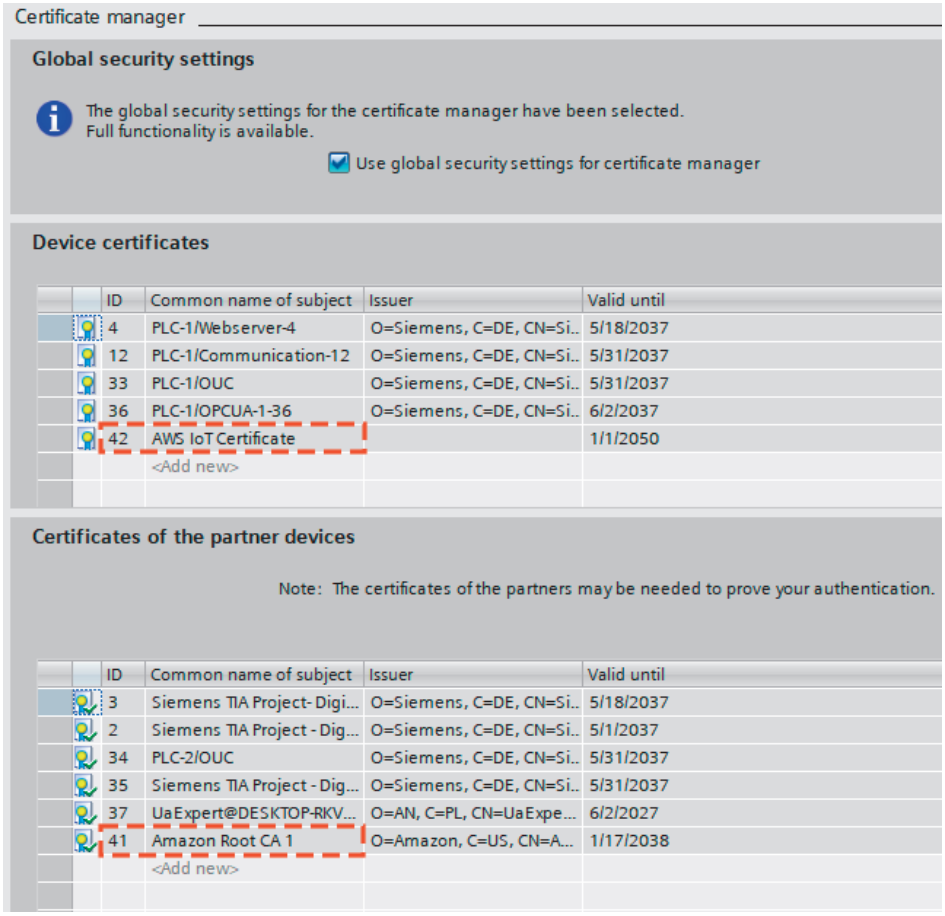
Nawiązanie bezpośredniej łączności z chmurą AWS możliwe jest w każdym sterowniku SIMATIC wspierającym obsługę certyfikatów TLS. Użyjemy do tego funkcji bibliotecznej `LMQTT`, z którą miałaś/miałeś okazję zapoznać się w rozdziale 5. Tym razem uzupełnimy konfigurację połączenia o szyfrowanie transmisji danych.

Zanim przejdziemy do rozbudowy programu PLC, konieczne jest dodanie stosownych certyfikatów do projektu TIA Portal. Przejdź więc do globalnego menedżera i w zakładce *Trusted certificates and root certificates* zaimportuj wspomniane pliki (rysunek 7.29). Aby dodać wygenerowane wcześniej archiwum, konieczna może być zmiana widocznych typów plików na PK12 w oknie importu certyfikatu. TIA poprosi Cię także o wpisanie hasła zdefiniowanego podczas tworzenia archiwum.



RYSUNEK 7.29. *Import certyfikatów do środowiska TIA Portal*

Jak zapewne pamiętasz z ćwiczenia, w którym nawiązywaliśmy szyfrowaną komunikację opartą na funkcjach OUC, certyfikaty zaimportowane w globalnym menedżerze należy jeszcze przypisać do konkretnego urządzenia w jego menedżerze lokalnym. Przejdź więc do ustawień PLC i dodaj plik *AWS IoT Certificate* do obszaru certyfikatów urządzenia, a *Amazon Root CA 1* do obszaru zaufanych partnerów komunikacyjnych (rysunek 7.30). Wgraj tak przygotowaną konfigurację sprzętową do PLC.



Rysunek 7.30. *Przypisanie certyfikatów urządzenia i chmury AWS do lokalnego menedżera w urządzeniu*

Następnie wywołaj nową instancję funkcji `LMQTT_Client`. Możesz skorzystać z programu zbudowanego w ramach ćwiczenia 5.2.4.1 lub przykładu umieszczonego w repozytorium. We wspomnianym przykładzie wykorzystywany w rozdziale 5. blok danych o nazwie `MQTT_Control` został zastąpiony nowym, analogicznym `AWS_Control`. Większość parametrów związanych z obsługą funkcji jest taka jak w przypadku komunikacji bez użycia szyfrowania. W kontekście połączenia z chmurą należy jednak dostosować kilka istotnych zmiennych zawartych w bloku danych.

- W strukturze `ConnParam` (tabela 7.3):

TABELA 7.3. *Podstawowe parametry połączenia z usługą IoT Core w chmurze AWS*

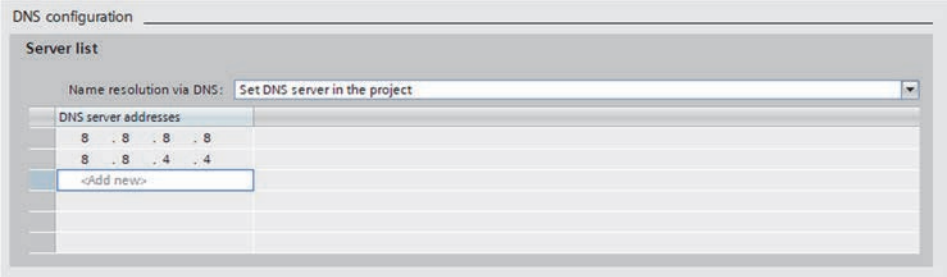
`hwId`	0
`mqttBrokerAddress.qdnAddress`	Adres serwera AWS, na którym uruchomiona jest usługa IoT Core. Znajdziesz go w menu nawigacyjnym usługi, w zakładce *Settings*, pod nazwą *Device data endpoint*. **Pamiętaj, że aby został on zinterpretowany jako kwalifikowana nazwa domenowa, musisz dodać na jego końcu kropkę!**
`mqttBrokerAddress.ipAddress`	Pozostaw pusty
`mqttBrokerAddress.port`	1883
`tls.enableTls`	TRUE
`tls.brokerCert`	Identyfikator certyfikatu serwera odczytany z menedżera certyfikatów
`tls.clientCert`	Identyfikator certyfikatu urządzenia odczytany z menedżera certyfikatów

- W strukturze `MQTTParam` (tabela 7.4):

TABELA 7.4. *Podstawowe parametry wymagane do poprawnego wysyłania wiadomości do usługi IoT Core opartej na protokole MQTT*

`ClientId`	Identyfikator urządzenia (dowolny)
`Username`	Pozostaw pusty
`Password`	Pozostaw pusty
`MqttTopic`	Temat, z którym publikowane będą wiadomości. Zdefiniuj przykładowy temat w postaci `data/IdUrządzenia`
`PublishMsgPayload`	Wiadomość publikowana do serwera. Zalecana jest serializacja wysyłanych danych do postaci formatu JSON, możesz więc użyć dokładnie tego samego łańcucha znaków, z którego korzystaliśmy w przykładzie 5.2.4.1

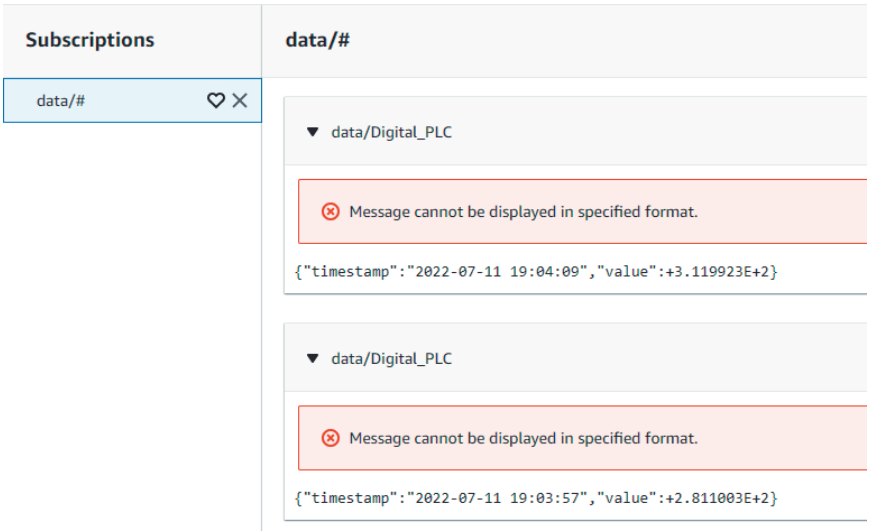
Aby połączenie mogło być nawiązane, konieczne jest oczywiście podłączenie sterownika do sieci Internet. Zalecane jest wykorzystanie urządzeń sieciowych wyposażonych w dodatkowe zabezpieczenia, np. zaporę ogniową. Pamiętaj, aby zezwolić w niej na połączenia wychodzące na porcie 1883 i komunikację z adresem domenowym serwera AWS. W ustawieniach sterownika niezbędne jest także dostosowanie adresów serwerów DNS tak, aby nazwa domenowa mogła być poprawnie rozwiązana i zrozumiała dla PLC. Opcję tę znajdziesz w zakładce *Advanced configuration/DNS configuration* (rysunek 7.31).



Rysunek 7.31. *Konfiguracja serwerów DNS w sterowniku SIMATIC. Wykorzystane zostały adresy publicznych serwerów DNS obsługiwanych przez Google'a*

Jeśli parametry połączenia zostały skonfigurowane poprawnie, a konfiguracja sprzętowa wraz z certyfikatami wgrana do sterownika, spróbuj nawiązać komunikację, ustawiając wartość wejścia *Enable* w funkcji `LMQTT_Client` na `TRUE`. Aby wysłać dane do chmury, zmień stan wejścia *Publish* na `TRUE`.

Dane, które trafiają do brokera uruchomionego na poziomie usługi IoT Core, możesz przeglądać za pomocą prostego klienta MQTT wbudowanego w usługę. W głównym menu nawigacyjnym przejdź do zakładki *Test/MQTT test client*. W polu *Subscribe to a topic* podaj nazwę tematu, z którym publikowane są dane pochodzące ze sterownika, i potwierdź akcję wciśnięciem przycisku *Subscribe*. Każda publikacja wiadomości w sterowniku spowoduje wyświetlenie jej w testowym kliencie (rysunek 7.32) — możesz w ten sposób zweryfikować poprawność połączenia i danych, które są wysyłane do chmury.



Rysunek 7.32. *Odczyt danych za pomocą testowego klienta MQTT, wbudowanego w usługę IoT Core*

Dane, które trafiają do usługi IoT Core, mogą być następnie przetwarzane przez inne serwisy dostępne w ramach ekosystemu AWS. Łącząc więc możliwości sterowników SIMATIC i chmur obliczeniowych, jesteś w stanie w stosunkowo szybki sposób zbudować nawet zaawansowane środowisko IoT, odpowiedzialne za monitorowanie, analizę i przetwarzanie danych pod kątem rozwoju biznesu i optymalizacji produkcji.

SIEMENS

DIGITALIZACJA
w systemach automatyki SIMATIC

Wydanie II

Z pamięci papieru do pamięci komputera

Współczesnym przemysłem rządzi... informatyka. Ta dziedzina stale się rozwija i zagarnia pod swoje skrzydła kolejne sektory — od produkcji, przez logistykę i księgowość, po dystrybucję i sprzedaż. Tyle teorii. W praktyce zaś często się okazuje, że podczas gdy otoczenie biznesowe i technologie pędzą naprzód, systemy stosowane w przemyśle zostają nieco z tyłu. Głównym celem, jaki przyświeca autorowi tej publikacji, skierowanej przede wszystkim do automatyków i programistów sterowników PLC, jest odczarowanie pojęcia digitalizacji i udowodnienie, że technologie, które się w nie wpisują, nie są wcale zarezerwowane dla specjalistów IT. W rzeczywistości wszyscy stosujemy je na co dzień, tylko w okrojonej formie.

W książce poruszane są takie tematy jak podstawowe założenia czwartej rewolucji przemysłowej, cyberbezpieczeństwo, mechanizmy informatyczne implementowane na poziomie konwencjonalnych urządzeń automatyki, internet rzeczy, chmury obliczeniowe, systemy brzegowe, a także technologie, które wyznaczają przyszłość automatyki przemysłowej. Każdy rozdział składa się z dwóch części: teoretycznej, zawierającej omówienie podstawowych zagadnień, które należy przyswoić, aby móc świadomie korzystać z danej technologii, i praktycznej, prezentującej jej implementację przy użyciu powszechnie stosowanych komponentów automatyki.

Helion

helion.pl

HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶

ISBN 978-83-289-0209-1

9 788328 902091

Cena: 109,00 zł