



MergenHub

DOKÜMANTASYON

Kullanım Kılavuzu

Kurulum, yayınlama, Windows servisleri ve loglar, lisanslama, tüm modüller, MergenHub Mobil, cihaz şablonları ve uygulama içi tam kılavuz.

Envest Enerji ve Su Teknolojileri Ltd. Şti.

mergenhub.envest.com.tr/tr/docs • info@envest.com.tr

Ekim 2026 • Çevrimiçi dokümantasyon her zaman en güncel sürümdür.

İçindekiler

BAŞLANGIÇ

- 1 MergenHub nedir
- 2 Kurulum
- 3 Hızlı başlangıç: ilk canlı veri
- 4 Domainde yayınlama
- 5 Windows Güvenlik Duvarı ve portlar
- 6 Windows servisleri ve tray uygulaması
- 7 Loglar ve teşhis verileri
- 8 Lisanslama ve deneme modu
- 9 Yedekleme, geri yükleme, yükseltme, kaldırma

MODÜLLER

- 10 Kanallar, cihazlar ve tag'ler
- 11 Protokol sürücüler
- 12 OPC UA sunucusu
- 13 REST API
- 14 Data Bridge (IoT gateway)
- 15 Advanced (virtual) tag'ler
- 16 Alarmlar ve bildirimler
- 17 Historian (veri kaydı)
- 18 SCADA / HMI
- 19 Kullanıcılar, gruplar ve güvenlik
- 20 Yapay zekâ asistanı (MCP sunucusu)
- 21 Otomasyon (iş akışları)
- 22 Uygulama-içi yapay zekâ asistanı
- 23 Sorun giderme

MOBİL VE CİHAZ ŞABLONLARI

- 24 MergenHub Mobil (iOS ve Android)
- 25 Cihaz şablonu (paket) oluşturma

KULLANICI KILAVUZU (UYGULAMA İÇİ, TAM)

26 Genel Bakış

27 Bağlanabilirlik

28 Protokoller (Sürücüler)

29 OPC UA Server

30 OPC UA İstemci

31 Advanced Tags

32 Veri Köprüsü (Data Bridge)

33 Historian

34 Alarmlar (A&E)

35 Aliases

36 Proje Taşıma (Dışa/İçe Aktarma)

37 SCADA / HMI

38 Kori1000 Ekran Şablonları

39 REST API

40 JSON Property Referansı

41 Güvenlik & Yetkilendirme

42 Kurulum & Dağıtım

43 Sorun Giderme

MergenHub nedir

Mimari, modüller, servisler, veri akışı ve diskte neyin nerede durduğu.

MergenHub, Windows üzerinde çalışan yazılım tabanlı bir **endüstriyel veri toplama platformu ve OPC UA aggregation sunucusudur**. Saha cihazlarını 11 endüstriyel protokolle okur, her değeri tek bir canlı tag deposunda tutar ve bu veriyi kuzey yönünde **OPC UA sunucusu, REST API, web SCADA/HMI** ve dışa veri iten **Data Bridge** ajanları (MQTT / REST) olarak sunar. Alarm, historian, virtual tag, kullanıcı yönetimi ve denetim yerleşiktir. Tek kurucu, üç Windows servisi, istemci tarafında kurulum yok.

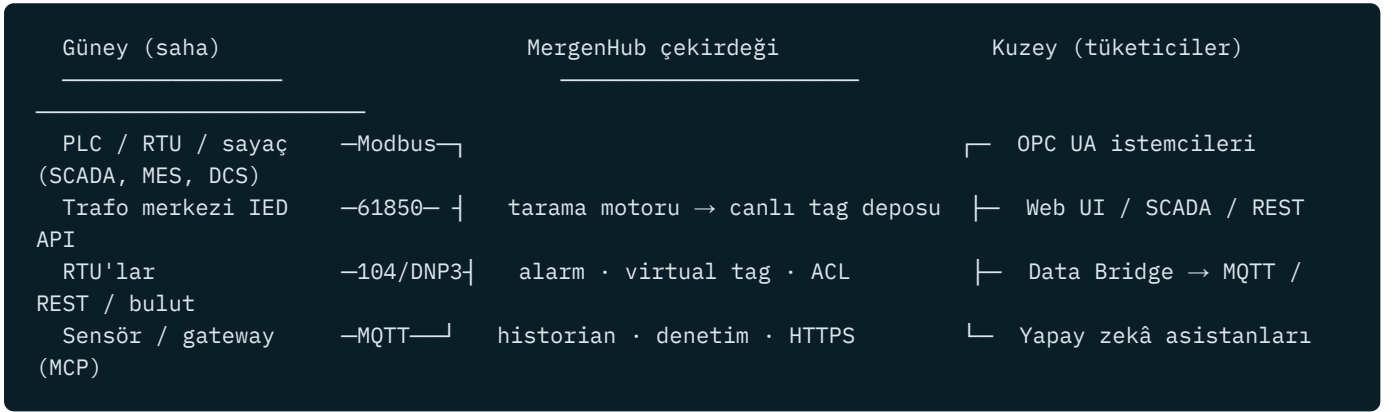
Kurulumdan hemen sonra MergenHub panosu – simülatör kanalıyla bağlantı görünümü

Tek kurulumda ne var

ALAN	NE YAPAR
Sürücüler	Modbus TCP/RTU/ASCII, Siemens S7, OMRON FINS ve NJ/NX CIP, DNP3, IEC 60870-5-104, IEC 61850 MMS, MQTT, OPC UA istemci, simülatör
OPC UA sunucusu	Tüm kanallar tek adres uzayında; çoklu endpoint; güvenlik ilkeleri; sertifikalar; kullanıcı bazlı erişim; geçmiş erişimi; alarmlar UA koşulu olarak
REST API	Oturum ya da token doğrulamalı HTTPS/JSON üzerinden yapılandırma ve canlı değerler; yerleşik Swagger
Web SCADA / HMI	Tarayıcı editörü: sembol kütüphaneleri, borular, akış animasyonu, faceplate, şablonlar; masaüstü ve mobil operatör görünümü

ALAN	NE YAPAR
Alarmlar	ISA-18.2 yaşam döngüsü, deadband'li koşullar, e-posta ve SMS, kullanıcı bazlı bildirim tercihi, geçmiş
Historian	SQLite ya da ODBC hedefler, tetikleyicili log grupları, store-and-forward, trend ve CSV/Excel dışı aktarım
Data Bridge	REST sunucu (Kepware uyumlu), REST istemci ve MQTT istemci ajanlarıyla buluta / üçüncü parti sistemlere veri
Virtual tag'ler	İfade motoru: ortalama, min/maks, totalizer, türev, link, birikimli; fonksiyon kütüphanesi
Güvenlik	Kullanıcı, grup, rol, kaynak seviyesinde ACL, HTTPS, oturum politikası, denetim günlüğü
Yapay zekâ asistanı	MCP sunucusu: yapay zekâ asistanları tesisi sizin yetkilerinizle sorgular

Mimari



Her sürücü aynı bellek içi **tag deposuna** yazar. OPC UA sunucusu, REST API, alarm motoru, historian, SCADA ekranları ve Data Bridge aynı anda aynı canlı değeri okur. Tarama **talep güdümlüdür**: bir cihaz yalnızca bir tüketici (OPC UA aboneliği, açık ekran, log grubu, alarm) tag'lerini gerçekten istediği sürece sorgulanır.

PROJE HİYERARŞİSİ

Kanal → Cihaz → Tag (ve tag grupları). Kanal, bir sürücü + taşıma ayarlarıdır (TCP portu, seri port, broker). Cihaz, o kanal altındaki bir PLC/RTU/IED'dir. Tag'ler okuduğunuz ya da yazdığınız adreslerdir. Bkz. [Kanallar, cihazlar ve tag'ler](#).

Windows servisleri

SERVİS (SCM ADI)	GÖRÜNEN AD	GÖREV
MergenHub	MergenHub Runtime (OPC UA + Config API)	Çekirdek: tarama motoru, OPC UA sunucusu, REST API, web arayüzü
MergenHubDataBridge	MergenHub Data Bridge	MQTT/REST push-pull ajanları (ayrı süreç)
MergenHubHistorian	MergenHub Historian	Tag geçmişini SQLite/ODBC'ye yazar (ayrı süreç)

İki yardımcı servis çekirdeğe bağımlıdır ve dururlarsa Windows tarafından otomatik başlatılır. Oturum açılışında başlayan **system tray** uygulaması (`MergenHub.Tray.exe`) hızlı durum, yeniden başlatma ve kısayollar sunar. Ayrıntı: [Windows servisleri ve tray uygulaması](#).

Diskte ne nerede

YOL	İÇERİK
C:\Program Files\MergenHub\	Program dosyaları (yükseltmede değişir)
C:\ProgramData\MergenHub\	Tüm veri: proje, kullanıcılar, ayarlar, lisans, sertifikalar, loglar (yükseltme ve kaldırmada korunur)
...\MergenHub\settings.json	Sunucu ayarları (portlar, HTTPS, OPC UA, lisanslama, saklama)
...\MergenHub\projects\	Proje (kanallar, cihazlar, tag'ler, alarmlar, log grupları)
...\MergenHub\scada.json	SCADA ekranları ve bölgeler
...\MergenHub\mergenhub.db	Kullanıcılar, gruplar, roller, oturumlar
...\MergenHub\events.db , audit.db , logs.db	Olay günlüğü, denetim günlüğü, historian (bkz. Loglar)
...\MergenHub\pki\ , webpki\	OPC UA sertifikaları, HTTPS sertifikası
...\MergenHub\license\	Lisans belgesi ve makine anahtarı

Not C:\ProgramData\MergenHub klasörünü yedeklediğinizde her şeyi yedeklemiş olursunuz. Bkz. [Yedekleme, geri yükleme, yükseltme, kaldırma](#).

Varsayılan portlar

AMAÇ	VARSAYILAN
Web arayüzü / REST API	8090 (HTTP) — HTTPS açıkken HTTPS portu (varsayılan 443)
OPC UA sunucusu	48010 — <code>opc.tcp://<host>:48010/MergenHub</code>

İkisi de kurulumda ya da sonradan *Ayarlar > OPC UA Sunucu / Web & REST API* bölümünden değiştirilir. Bkz. [Windows Güvenlik Duvarı ve portlar](#).

Paketler ve kısıtlar

Tüm özellikler her pakette açıktır; paketler yalnızca **tag sayısı**yla ayrılır. Lisanssız kurulum **deneme modunda** çalışır (500 tag, her başlatmada 2 saatlik çalışma penceresi). Bkz. [Lisanslama ve deneme modu](#).

Kurulum

Gereksinimler, kurulum sihirbazı adım adım, kurucunun yaptıkları, yükseltme ve temiz kurulum.

MergenHub tek bir Windows kurulum paketi olarak gelir: `MergenHub-Setup-<sürüm>.exe`. Paket **kendi kendine yeterlidir**: hedef makineye .NET çalışma zamanı ya da veritabanı sunucusu kurmanız gerekmez.

Gereksinimler

	EN AZ	ÖNERİLEN
İşletim sistemi	Windows 10 / 11, Windows Server 2016+ (64-bit)	Windows Server 2019+
İşlemci	2 çekirdek	50K+ tag için 4 çekirdek
RAM	4 GB	100K+ tag için 8 GB
Disk	2 GB boş	SSD; historian büyümesi log gruplarına bağlıdır
Yetki	Kurulum için yerel yönetici	—
Ağ	Cihazlara TCP erişimi; istemciler için açık portlar (bkz. Güvenlik duvarı)	—

Tarayıcı: web arayüzü için güncel Chrome, Edge, Firefox ya da Safari. Operatör bilgisayarlarına hiçbir şey kurulmaz.

Kurulum sihirbazı

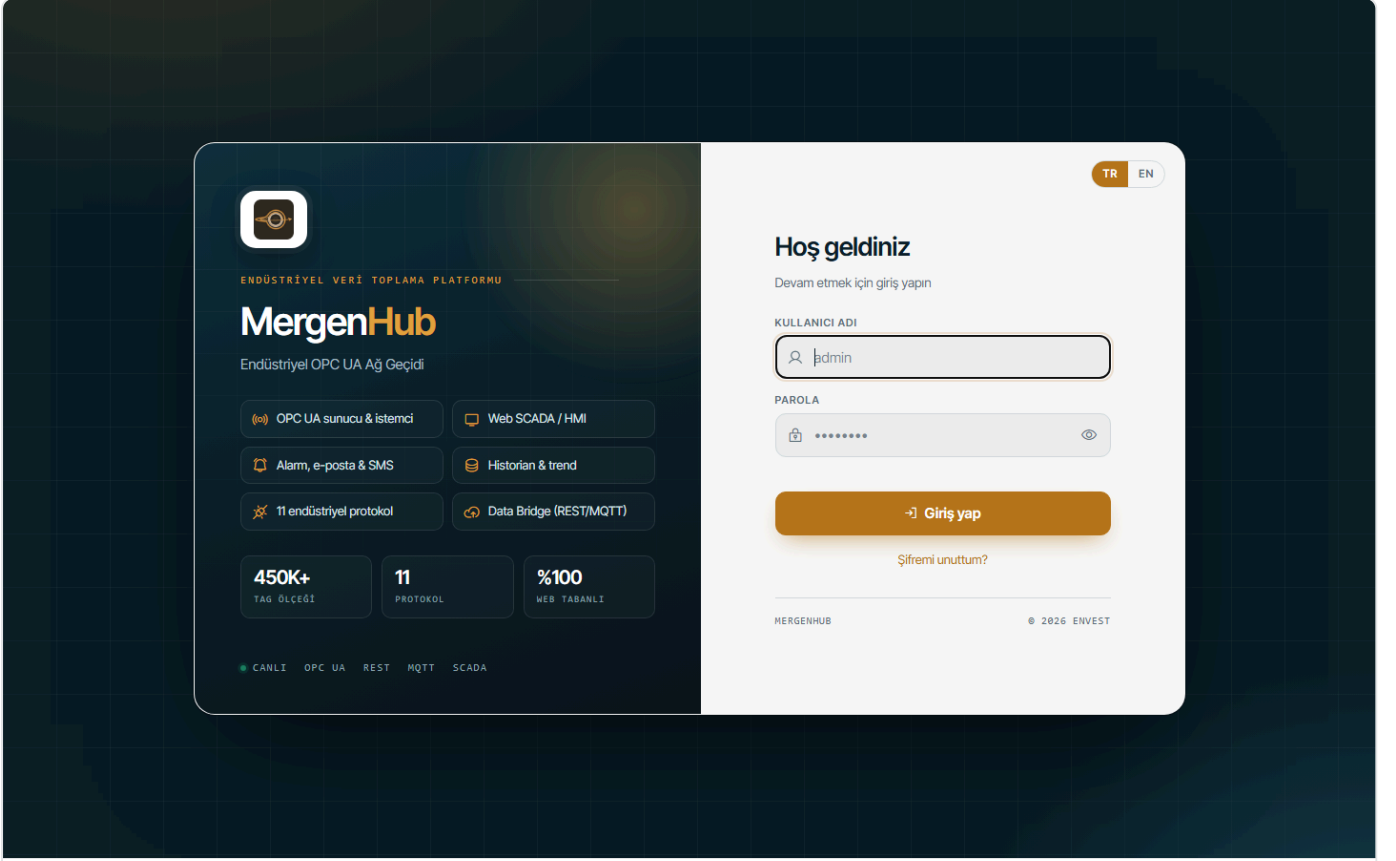
Kurucuyu yönetici olarak çalıştırın. Her sayfada makul varsayılanlar vardır; **İleri** diyerek geçebilirsiniz.

- Lisans sözleşmesi ve deneme bilgilendirmesi** — kurucu deneme modunu (500 tag, 2 saatlik çalışma penceresi) ve sonradan lisans anahtarı girildiğinde kısıtların yeniden kurulum gerekmeden kalktığını anlatır.
- Sunucu adresi ve portlar**
 - Host / Domain** — yalnız yerel kullanım için `localhost` ; tüm ağ arayüzlerinde dinlemek (LAN erişimi) için `0.0.0.0` ; internette yayınlarlarken makine adı ya da gerçek bir domain.
 - Web UI / REST API portu** — varsayılan `8090` .
 - OPC UA portu** — varsayılan `48010` .
- Yönetici hesabı** — kullanıcı adı sabittir: `admin` .
 - Şimdi bir parola girin ya da boş bırakıp `admin / admin` ile başlayın; ilk girişte değiştirmeniz istenir.
 - Güçlü parola oluşturun** 16 karakterlik bir parola üretir.
 - Host `localhost` dışında bir değerse güçlü parola **zorunludur** (10+ karakter; büyük/küçük harf, rakam, sembol sınıflarından üçü). Kurucu zayıf parolayla sunucuyu ağa açmaz.

4. **Temiz kurulum** (varsayılan kapalı) — işaretlenirse `C:\ProgramData\MergenHub` altındaki her şey (proje, kullanıcılar, ayarlar, loglar) **silinir** ve boş projeyle başlarsınız. Yerinde yükseltme için kapalı bırakın.
5. **OPC UA sunucu güvenliği** — ilk bağlantı kolay olsun diye ikisi de açık gelir: *Anonim erişime izin ver* ve *Güvenilmeyen istemci sertifikalarını otomatik kabul et*. Üretime geçmeden *Ayarlar > OPC UA Sunucu*'dan kısıtlayın.
6. **HTTPS / web güvenliği**
 - Yalnız HTTP (varsayılan; yerel ve test için uygun)
 - Kurulumda üretilen self-signed sertifikayla HTTPS (tarayıcı bir kez uyarır)
 - Kendi PFX sertifikanızla HTTPS (yol ve parola sorulur)
 - HTTPS seçilince HTTPS portu (varsayılan `443`) sorulur.
7. **Domain ve DNS kontrolü** — yalnız gerçek bir domain girildiğinde görünür; domaini çözer ve makinenin genel IP'siyle karşılaştırır, A kaydının buraya işaret edip etmediğini söyler.
8. **Servis başlatma davranışı** — açılışta başlat (varsayılan) ve ayarlanabilir gecikmeyle otomatik çökme kurtarma.

Sonunda kurucu:

- `settings.json` ve `appsettings.Production.json` (veri kökü) dosyalarını yazar, `C:\ProgramData\MergenHub` klasörünü oluşturur,
- OPC UA sunucu sertifikasını ve seçildiyse HTTPS sertifikasını üretir,
- web, OPC UA ve (açıksa) HTTPS portları için **Windows Güvenlik Duvarı** kuralları ekler,
- üç Windows servisini kaydedip başlatır,
- tüm kullanıcılar için system tray yöneticisini kurar ve web arayüzünü açar.



Kurulumdan sonraki ilk ekran: giriş sayfası

İlk giriş

`http://localhost:8090` (ya da yapılandırıdığınız adres) adresini açın. `admin` ve seçtiğiniz parolayla girin. Parolayı boş bıraktıysanız `admin / admin` ile girer, hemen yeni parola belirlersiniz.

Devamı: [Hızlı başlangıç: ilk canlı veri.](#)

Mevcut kurulumu yükseltme

Yeni kurucuyu eskisinin üzerine çalıştırın. Mevcut kurulumu algılar, servisleri durdurur, program dosyalarını değiştirir ve `C:\ProgramData\MergenHub` altındaki her şeyi korur: proje, kullanıcılar, ayarlar, sertifikalar, lisans ve geçmiş. Eski sürümlerdeki servis adları (`MergenHubIoTGateway` , `MergenHubDataLogger`) otomatik olarak kaldırılıp yeni adlarla yeniden kaydedilir.

İpucu Lisanslı kurulumlar yeni sürümden uygulama içinde haberdar edilir (*Ayarlar > Lisans > Güncellemeleri denetle* ve günde bir bildirim). Yükseltmeler lisans ömrü boyunca ücretsizdir.

Temiz kurulum

Test için kullanılmış bir makinede eski projeyi atmak istiyorsanız sihirbazda **Temiz kurulum**'u işaretleyin. Geri alınamaz; değerli bir şey varsa önce projeyi dışa aktarın (*Bağlantılar > Dışa aktar*).

Sessiz / gözetimsiz kurulum

Kurucu Inno Setup ile yapılmıştır ve standart anahtarları kabul eder:

```
MergenHub-Setup-3.1.19.exe /VERYSILENT /SUPPRESSMSGBOXES /NORESTART /LOG="C:\Temp\mh-setup.log"
```

Sihirbaz değerleri varsayılanlara döner (localhost, 8090, 48010, ilk girişte değiştirilecek `admin/admin`). Betikli dağıtım için `C:\ProgramData\MergenHub\settings.json` dosyasını önceden hazırlayın; kurucu var olan dosyayı korur.

Kurulum logu

`%TEMP%\MergenHub-Setup.log` servis kaydı ve güvenlik duvarı komutları dahil her adımı içerir. Kurulum başarısız olursa destek talebine ekleyin.

Hızlı başlangıç: ilk canlı veri

Giriş, kanal-cihaz-tag ekleme, canlı değerleri görme, OPC UA istemcisi bağlama.

Bu yürüyüş sizi sıfır kurulumdan on dakikada bir OPC UA istemcisinde canlı değerlere götürür. Ekran görüntülerinde yerleşik **simülatör** kullanılır; donanım olmadan da izleyebilirsiniz.

1. Giriş yapın

Web arayüzünü açın (varsayılan `http://localhost:8090`) ve `admin` ile girin. Soldaki ağaç **projenizdir**: *Bağlantılar* kanal, cihaz ve tag'leri; *Eklentiler* virtual tag, alias, alarm, historian, Data Bridge ve MCP'yi tutar.

The screenshot shows the MergenHub OPC UA Gateway interface. On the left, there is a project tree with the following structure:

- PROJE
 - Bağlantılar
 - Simulation Examples (24)
 - Functions (24)
 - Eklentiler
 - Virtual Tag'ler
 - Tag Kısayolları
 - Alarmlar
 - Veri Logları
 - Data Bridge
 - MCP (AI)

At the bottom left, the user is logged in as `admin` (Root (super-admin)).

The main area displays the 'Simulation Examples' table with the following columns: Ad, Model, ID, and Tag. The table contains one row:

Ad	Model	ID	Tag
Functions	—	4	24

At the bottom of the interface, there is a text box that reads: "Bağlantılar görünümü – proje ağacı ve kanal listesi".

2. Kanal ekleyin

+ **Kanal ekle**'ye tıklayın, sürücü seçin ve ad verin. Gerçek bir cihaz için protokolü seçin (ör. *Modbus TCP/IP Ethernet*); bu tur için **Simulator** seçin. Her sürücünün kendi kanal ayarları vardır (ağ adaptörü, seri port, zaman aşımı, yeniden deneme); varsayılanlar güvenlidir.

İpucu Kanal, bir fiziksel ya da mantıksal bağlantıdır. Aynı Modbus TCP anahtarındaki cihazlar tek kanalı paylaşabilir; farklı seri portlardaki cihazlar için port başına bir kanal gerekir.

3. Cihaz ekleyin

Kanali seçip + **Cihaz ekle**'ye tıklayın. Cihaz adresini girin (Modbus TCP için 10.0.0.5:502 + unit ID, S7 için IP + rack/slot, 61850 için IED adresi). **Tarama hızını** (cihazın kaç ms'de bir sorgulanacağı) ve gerekiyorsa zaman aşımı/yeniden denemeyi ayarlayın.

4. Tag ekleyin

Cihazı seçip + **Tag ekle** ya da **Tag üret** kullanın:

- **Tag ekle** — ad, adres, veri tipi, okuma/yazma, isteğe bağlı ölçekleme ve açıklama.
- **Tag üret** — adres aralığından ya da cihazın kendisinden (OPC UA, 61850, DNP3, IEC 104, OMRON NJ sembolleri, MQTT konuları için çevrimiçi tarama) toplu tag üretir.
- **Adres biçimleri** — adres alanının yanındaki ? sürücünün anladığı adres biçimlerini örneklerle listeler.

PROJE

+

Bağlantılar

Simulation Examples 24

Functions 24

Ekentiler

Virtual Tag'ler

Tag Kısıyolları

Alarmlar

Veri Logları

Data Bridge

MCP (All)

admin

Root (super-admin)

TR TR

Simulation Examples.Functions

Sadece Konfig

Canlı Veri

24/24 aktif tag

Deneme 24/500

Functions tag'leri

Özellikler

Diagnostics

Yeniden Adlandır

Kopyala

Sil

+ Tag Ekle

Grup Ekle

Ad veya adres ile ara...

24 tag

Sistem tag'leri

Canlı veri

Ad	Adres	Tip	Tarama	Değer	Kalite	Zaman
Ramp1	RAMP (120, 35, 100, 4)	Long	100	37.63	Good	18:48:42.167
Ramp2	RAMP (300, 150.750000, 200.500000, -0.250000)	Float	100	198.81	Good	18:48:42.167
Ramp3	RAMP (250, 0, 1000, 1)	Word	100	8.12	Good	18:48:42.167
Ramp4	RAMP (2000, -1000, 1000, -5)	Long	100	994.93	Good	18:48:42.167
Ramp5	RAMP (250, -1000000, 1000000, -500)	Date	100	995942.00	Good	18:48:42.167
Ramp6	RAMP (1000, 0, 1000000000, 1250)	LLong	100	2536.25	Good	18:48:42.167
Ramp7	RAMP (1000, -1000000000, 1000000000, -5555)	Date	100	999988728.90	Good	18:48:42.167
Ramp8	RAMP (1000, 150.750000, 200.500000, -0.250000)	Double	100	199.99	Good	18:48:42.167
Random1	RANDOM (30, -20, 75)	Short	100	-10.13	Good	18:48:42.167
Random2	RANDOM (100, 0, 1000)	Word	100	381.20	Good	18:48:42.167
Random3	RANDOM (100, -1000, 0)	Long	100	-510.52	Good	18:48:42.167
Random4	RANDOM (1000, -999, 999)	Long	100	-630.22	Good	18:48:42.167
Random5	RANDOM (100, -1000000000, 1000000000)	Long	100	431500399.97	Good	18:48:42.167
Random6	RANDOM (1000, -1000000, 1000000)	Date	100	451194.95	Good	18:48:42.167
Random7	RANDOM (1000, 0, 1000000000)	LLong	100	979424945.12	Good	18:48:42.167
Random8	RANDOM (100, 0, 1000000)	LLong	100	584115.86	Good	18:48:42.167
Sine1	SINE (10, -40.000000, 40.000000, 0.050000, 0)	Float	100	23.81	Good	18:48:42.167

Cihazın tag tablosu – değer, kalite ve zaman damgası canlı güncellenir

5. Canlı değerleri izleyin

Üst çubuğu **Canlı veri**'ye alın. Her tag mevcut değerini, **kalitesini** (Good / Bad / Uncertain) ve zaman damgasını gösterir. Kırmızı kalite cihazın okunamadığı anlamına gelir; cihazdaki *Teşhis* düğmesi ham istek/yanıt çerçevelerini ve son hatayı gösterir.

6. OPC UA istemcisi bağlayın

Herhangi bir OPC UA istemcisini (UaExpert, Kepware, Ignition, bir PLC) şu adrese yönlendirin:

```
opc.tcp://<host>:48010/MergenHub
```

Objects › *<kanal>* › *<cihaz>* altında tag'lerinizi bulun. Kurucu varsayılanlarıyla anonim erişim ve güvenilmeyen sertifikalar kabul edilir; bunu sonra *Ayarlar* › *OPC UA Sunucu*'dan sertleştirin (güvenlik ilkesi, kullanıcı doğrulama, güvenilen istemci listesi). Ayrıntı: [OPC UA sunucusu](#).

7. Aynı değerleri REST ile okuyun

```
curl -c c.txt -X POST http://localhost:8090/auth/v1/login -H "Content-Type: application/json" -d "{\n  \"username\": \"admin\", \"password\": \"<parola>\"\n}"
curl -b c.txt "http://localhost:8090/runtime/v1/values?tags=Simulation%20Examples.Functions.Ramp1"
```

API'nin tamamı uygulamada ... › *REST API* (Swagger) altında ve [REST API](#) sayfasında.

8. İlk ekranı çizin

SCADA'yı açın, ekran oluşturun, sembol kütüphanesinden tank ve pompa sürükleyin, özelliklerini tag'lere bağlayın ve *Çalıştır*'a basın. Bkz. [SCADA / HMI](#).

Sonra ne yapılır

- **Alarmlar** — kritik tag'lerde eşik, e-posta/SMS bildirimi: [Alarmlar](#)
- **Historian** — trend istediğiniz tag'ler için log grupları: [Historian](#)
- **Kullanıcılar** — sınırlı yetkili operatör hesapları: [Kullanıcılar, gruplar ve güvenlik](#)
- **Yayınlama** — başka yerlerden ya da internetten erişim: [Domainde yayınlama](#)
- **Lisans** — deneme kısıtlarını kaldırmak için anahtarı girin: [Lisanslama](#)

Domainde yayınlama

DNS, port yönlendirme, HTTPS sertifikası, reverse proxy (IIS / nginx / Caddy), uzaktan OPC UA.

MergenHub Windows servisi olarak çalışır; makinede kimse oturum açmasa da çalışmaya devam eder. Bir domain altında yayınlamak üç şey demektir: makineye işaret eden bir DNS kaydı, yönlendirici ve güvenlik duvarlarından geçen doğru portlar ve parolaların açık gitmemesi için HTTPS. Bu sayfa üçünü ve reverse proxy seçeneklerini anlatır.

Sunucuya nasıl ulaşılacağına karar verin

SENARYO	YAPILACAK
Yalnız bu bilgisayar	Kurucu varsayılanlarını koruyun (localhost).
Ofis / tesis ağı (LAN)	Host'u 0.0.0.0 yapın (tüm arayüzler); istemciler makinenin IP'sini ya da adını kullanır; kurucunun eklediği güvenlik duvarı kuralları yeter.
İnternet, kendi genel IP'niz	DNS A kaydı → genel IP; yönlendiricide port yönlendirme; HTTPS açın; kullanıcıları ve OPC UA'yı sertleştirin.
İnternet, reverse proxy arkasında	DNS → proxy; proxy HTTPS'i sonlandırıp 127.0.0.1:8090 'a iletir; OPC UA portu ayrıca yönlendirilir (TCP, HTTP değil).
Yalnız VPN	Dışarıdan OPC UA ve yazma erişimi için önerilir; port yönlendirme gerekmez.

Uyarı Web arayüzünü internete asla düz HTTP ile açmayın. HTTPS'i (aşağıda) açın ve güçlü parolalar kullanın; kurucu zayıf admin parolasıyla sunucuyu yerel dışı bir host'a bağlamayı reddeder.

1. DNS

İstediğiniz ad için (ör. `scada.example.com`) internet bağlantınızın genel IP'sine (ya da reverse proxy / bulut sunucunuza) işaret eden bir **A kaydı** oluşturun. ISS adresiniz değişiyorsa dinamik DNS servisi kullanın.

Kurucunun **Domain ve DNS kontrolü** sayfası domaini girdiğinizde adın bu makinenin genel adresine çözülüp çözülmediğini söyler.

2. Portlar ve yönlendirme

TRAFİK	PORT (VARSAYILAN)	PROTOKOL	GEREKLİ OLDUĞU YER
Web arayüzü + REST API	8090, HTTPS'te 443	TCP	Tarayıcılar, REST istemcileri, Data Bridge REST sunucusu
OPC UA	48010	TCP	OPC UA istemcileri (SCADA, MES, historian)

Yönlendiricide dış portu MergenHub makinesinin LAN IP'sine yönlendirin. Makinenin kendisinde kurucu Windows Güvenlik Duvarı kurallarını zaten oluşturdu; bkz. [Windows Güvenlik Duvarı ve portlar](#). Portları *Ayarlar* > *Web & REST API* ve *Ayarlar* > *OPC UA Sunucu* dan değiştirebilirsiniz.

Ayarlar > OPC UA Sunucu — endpoint'ler, güvenlik ilkeleri ve sunucunun dinlediği port

3. HTTPS

Üç seçenek var.

A) SELF-SIGNED SERTİFİKA (EN HIZLI)

Kurucuda *HTTPS – self-signed* seçin ya da sonradan yönetici olarak çalıştırın:

```
"C:\Program Files\MergenHub\MergenHub.exe" gencert "scada.example.com"
```

C:\ProgramData\MergenHub\webpki\web.pfx yazılır. Sertifika genel bir otorite tarafından imzalı olmadığından tarayıcılar bir kez uyarır; iç kullanım için yeterlidir.

B) KENDİ SERTİFİKANIZ (PFX)

PFX'i C:\ProgramData\MergenHub\webpki\web.pfx olarak, parolasını web.pfx.pass (düz metin) olarak kopyalayın ya da kurucuda dosyayı seçin. MergenHub servisini yeniden başlatın.

C) LET'S ENCRYPT İLE ÜCRETSİZ GENEL SERTİFİKA

- win-acme** MergenHub makinesinde: `wacs.exe --target manual --host scada.example.com --store pfxfile --pfxfilepath C:\ProgramData\MergenHub\webpki --pfxfilename web.pfx --pfxpassword <pw>` + yenileme görevi; parolayı web.pfx.pass dosyasına yazın. Doğrulama sırasında 80 portu erişilebilir olmalı (ya da DNS doğrulaması).
- Caddy** reverse proxy sertifikayı kendisi alır ve yeniler — aşağıda. Üretim için en basit yol budur.

HTTPS'i ve portu *Ayarlar > Web & REST API*'de (ya da kurucuda) açın. Eski yer imleri çalışsın diye HTTP yönlendirmesini açık tutun.

4. Reverse proxy (isteğe bağlı ama önerilir)

Aynı makinede ya da bir geçit sunucusunda reverse proxy TLS'i sonlandırır, iç portu gizler ve tek IP arkasında birden çok uygulama barındırmanızı sağlar.

CADDY (OTOMATİK HTTPS)

```
scada.example.com {  
  reverse_proxy 127.0.0.1:8090  
}
```

Caddy sertifikayı kendisi alır ve yeniler. MergenHub arkada HTTP `127.0.0.1:8090` olarak kalır.

NGINX

```
server {  
  listen 443 ssl http2;  
  server_name scada.example.com;  
  ssl_certificate      /etc/ssl/scada.crt;  
  ssl_certificate_key  /etc/ssl/scada.key;  
  client_max_body_size 64m;  
  location / {  
    proxy_pass http://10.0.0.20:8090;  
    proxy_http_version 1.1;  
    proxy_set_header Host $host;  
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
    proxy_set_header X-Forwarded-Proto https;  
    proxy_read_timeout 300s;  
  }  
}  
  
server { listen 80; server_name scada.example.com; return 301 https://$host$request_uri; }
```

IIS (URL REWRITE + APPLICATION REQUEST ROUTING)

1. *URL Rewrite* ve *ARR* kurun, *Server Proxy Settings > Enable proxy* açın.
2. `scada.example.com:443` bağlı, sertifikalı bir site oluşturun.
3. Gelen kural ekleyin: `(.) eşleşmesi, eylem *Rewrite → http://127.0.0.1:8090/{R:1}`, *Append query string* işaretli.
4. Büyük SVG sembol kütüphaneleri yüklüyorsanız *Request Filtering*'de izin verilen içerik uzunluğunu artırın.

Not OPC UA ikili bir TCP protokolüdür, HTTP değildir — HTTP reverse proxy'den geçmez. 48010 portunu doğrudan yönlendirin (ya da VPN kullanın).

5. İnternette OPC UA

- *Ayarlar > OPC UA Sunucu*'da host'u genel DNS adı olan bir **endpoint** ekleyin (`opc.tcp://scada.example.com:48010/MergenHub`). İstemciler sunucu sertifikasını bağlandıkları ada göre doğrular.

- Güvenlik ilkesi **Basic256Sha256 – Sign & Encrypt** kullanın, *Anonim*'i kapatın; yalnız gereken haklara sahip bir OPC UA kullanıcısı oluşturun.
- *Güvenilmeyen sertifikaları otomatik kabul*ü kapatın; istemcileri *Ayarlar > Güvenilen istemciler*'de açıkça güvenilir yapın.
- Host adını değiştirdikten sonra sunucu sertifikasını yeniden üretin (*Ayarlar > Sunucu sertifikası*).

6. Canlıya geçmeden kontrol listesi

- [] Güçlü admin parolası; sınırlı rollü operatör hesapları ([Kullanıcılar ve güvenlik](#))
- [] HTTPS açık; HTTP yönlendiriliyor
- [] Yalnız 443 ve (gerekiyorsa) 48010 yönlendirilmiş; diğerleri kapalı
- [] OPC UA anonim erişim kapalı, güvenilen istemci listesi bakımlı
- [] Oturum politikası ve denetim saklama süresi gözden geçirilmiş ([Loglar](#))
- [] C:\ProgramData\MergenHub yedeği zamanlanmış ([Yedekleme](#))
- [] Lisans etkinleştirilmiş; deneme penceresi taramayı durdurmasın ([Lisanslama](#))

Windows Güvenlik Duvarı ve portlar

MergenHub hangi portları kullanır, kurucu hangi kuralları ekler, nasıl açılır ya da kısıtlanır.

MergenHub varsayılan olarak iki TCP portunda dinler ve kurucu bunları Windows Güvenlik Duvarı'nda açar. Bu sayfa tüm portları, kurucunun oluşturduğu kuralları ve elle nasıl açılıp kısıtlanacağını listeler.

MergenHub'ın kullandığı portlar

PORT	YÖN	AMAÇ	NEREDEN DEĞİŞİR
8090 TCP	gelen	Web arayüzü, REST API, Data Bridge REST sunucusu, MCP ucu	Kurucu › <i>Sunucu adresi ve portlar</i> ; Ayarlar › <i>Web & REST API</i>
443 TCP	gelen	Aynı servisler için HTTPS (yalnız HTTPS açıkken)	Kurucu › <i>HTTPS</i> ; Ayarlar › <i>Web & REST API</i>
48010 TCP	gelen	OPC UA sunucusu (<code>opc.tcp://host:48010/MergenHub</code>)	Ayarlar › <i>OPC UA Sunucu</i> › <i>Endpoint'ler</i>
cihaz portları	giden	Modbus 502, S7 102, OMRON 9600 (FINS) / 44818 (CIP), DNP3 20000, IEC 104 2404, IEC 61850 102, MQTT 1883/8883, uzak sunuculara OPC UA istemci	kanal/cihaz başına
25 / 465 / 587	giden	E-posta bildirimleri için SMTP	Ayarlar › <i>E-posta</i>
443	giden	SMS sağlayıcı API'si, lisans portalı (etkinleştirme, tazeleme, güncelleme denetimi)	—

Giden bağlantılara Windows Güvenlik Duvarı normalde izin verir; yalnız gelen portlar için kural gerekir.

Kurucunun oluşturduğu kurallar

Kurucu `netsh advfirewall` ile şu gelen *allow* kurallarını ekler (tüm profiller, TCP):

KURAL ADI	PORT
MergenHub Web	seçtiğiniz web portu (8090)
MergenHub OPC UA	OPC UA portu (48010)
MergenHub HTTPS	HTTPS portu (yalnız HTTPS açıksa)

Kaldırıcı bunları siler. *Windows Defender Güvenlik Duvarı (Gelişmiş Güvenlik)* › *Gelen Kuralları* altında görürsünüz.

Portları elle açma ya da değiştirme

Yönetici komut isteminde:

```
netsh advfirewall firewall add rule name="MergenHub Web" dir=in action=allow protocol=TCP localport=8090
netsh advfirewall firewall add rule name="MergenHub OPC UA" dir=in action=allow protocol=TCP localport=48010
netsh advfirewall firewall add rule name="MergenHub HTTPS" dir=in action=allow protocol=TCP localport=443
```

PowerShell karşılığı:

```
New-NetFirewallRule -DisplayName "MergenHub Web" -Direction Inbound -Protocol TCP -LocalPort 8090 -Action Allow
```

Ayarlar'da portu değiştirdiyseniz yeni port için kural ekleyip eskisini silin:

```
netsh advfirewall firewall delete rule name="MergenHub Web"
```

Kimin bağlanabileceğini kısıtlama

Kuralı tüm dünya yerine belirli ağlarla sınırlayın:

```
netsh advfirewall firewall set rule name="MergenHub OPC UA" new remoteip=10.0.0.0/24,192.168.10.0/24
netsh advfirewall firewall set rule name="MergenHub Web" new profile=domain,private
```

İnternete bakan sunucularda 443'te reverse proxy tercih edin ve 8090'ı `127.0.0.1` 'e bağlı tutun (*Ayarlar* > *Web & REST API* de host'u `127.0.0.1` yapın). Bkz. [Domainde yayınlama](#).

Üçüncü parti güvenlik duvarı ve antivirüs

- Uç nokta güvenlik paketlerinin (Kaspersky, ESET, Sophos ...) kendi güvenlik duvarı vardır; aynı TCP portlarını orada da ekleyin.
- Bazı ürünler imzasız kurucuyu işaretler. Dosyaya sağ tık > *Özellikler* > *Engellemeyi kaldır* ya da `C:\Program Files\MergenHub` için istisna ekleyin.
- AppLocker / WDAC'li kurumsal bilgisayarlarda kurucu tamamen engellenebilir; yöneticiden yayıncıya ya da yola izin vermesini isteyin.

Doğrulama

- Başka bir makineden: `Test-NetConnection <host> -Port 8090` (PowerShell) — `TcpTestSucceeded: True` port açık demektir.
- Sunucuda: `netstat -ano | findstr :8090` dinleyen soketi ve `MergenHub` servisinin süreç kimliğini gösterir.
- Olay günlüğü* (... > *Olay Günlüğü*) servisler ayaktaiken `Web API listening on ...` ve `OPC UA server started` satırlarını gösterir.

Sorun giderme

BELİRTİ	OLASI NEDEN
Web arayüzü sunucuda çalışıyor, başka PC'lerden açılmıyor	Host <code>localhost</code> — <i>Ayarlar</i> > <i>Web & REST API</i> 'de <code>0.0.0.0</code> yapıp servisi yeniden başlatın; ya da güvenlik duvarı kuralı eksik
OPC UA istemcisi endpoint'i görüyor ama bağlanamıyor	Endpoint host adı istemcinin kullandığından farklı; sertifika güvenilmiyor; güvenlik ilkesi uyuşmuyor
Tarayıcı HTTPS'te "bağlantı sıfırlandı" diyor	HTTPS açık ama <code>webpki\</code> altında geçerli <code>web.pfx</code> yok; olay günlüğüne bakın
Port zaten kullanımda	Başka bir program (çoğu kez başka bir OPC sunucusu) 48010 ya da 8090'ı kullanıyor; <i>Ayarlar</i> 'dan portu değiştirin

Windows servisleri ve tray uygulaması

Üç servis, açılışta başlatma ve çökme kurtarma, system tray yöneticisi, kullanıcı oturumu olmadan çalışma.

MergenHub üç Windows servisi ve küçük bir system tray yöneticisi olarak kurulur. Servisler arka planda *Local System* hesabıyla çalışır, açılışta başlar ve kimse oturum açmasa da çalışmaya devam eder — operatör bilgisayarı, panodaki sunucu ya da sanal makine aynı davranır.

Üç servis

SCM ADI	GÖRÜNEN AD	SÜREÇ	DÜŞTÜĞÜNDE NE DURUR
MergenHub	MergenHub Runtime (OPC UA + Config API)	MergenHub.exe	Her şey: cihaz tarama, OPC UA, web arayüzü, REST API, alarm, SCADA
MergenHubData Bridge	MergenHub Data Bridge	MergenHub.IoTGateway.exe	Dışa MQTT/REST ajanları ve Data Bridge REST sunucusu
MergenHubHistorian	MergenHub Historian	MergenHub.Historian.exe	Tag geçmişi yazımı (değerler canlıdır; kayıt olmaz)

İki yardımcı servis çekirdeğe **bağımlıdır** (`depend= MergenHub`): Windows önce çekirdeği başlatır, çekirdek iç token'ı yazar, diğerleri yerel REST API üzerinden çekirdeğe bağlanır. Aynı süreçler sayesinde bir bulut bağlantısındaki ya da veritabanı sürücüsündeki hata OPC UA sunucusunu asla düşürmez.

MergenHub
OPC UA Gateway - v3.119

Ayarlar

24/24 aktif tag Deneme 24/500

PROJE +

- Bağlantılar
 - Simulation Examples 24
 - Functions 24
- Eklentiler
 - Virtual Tag'ler
 - Tag Kısayolları
 - Alarmlar
 - Veri Logları
 - Data Bridge
 - MCP (AI)

OPC UA SUNUCU

- OPC UA Sunucu
- OPC UA Kullanıcıları
- Güvenilen Client'lar
- Sunucu Sertifikası

WEB & BAĞLANTI

- Web & REST API
- Data Bridge

BİLDİRİMLER

- E-posta
- SMS
- İletişim Merkezi

ÖRÜNÜM

- Marka

SİSTEM

- Lisans
- Denetim
- Saklama

admin
Root (super-admin)

TR TR

OPC UA SUNUCU

OPC UA Sunucu

OPC UA Kullanıcıları

Güvenilen Client'lar

Sunucu Sertifikası

WEB & BAĞLANTI

Web & REST API

Data Bridge

BİLDİRİMLER

E-posta

SMS

İletişim Merkezi

ÖRÜNÜM

Marka

SİSTEM

Lisans

Denetim

Saklama

Servisler

ERİŞİM

Kullanıcılar

Çekirdek servislerin durumu + sunucu kaynakları (MergenHub süreci ve makinenin toplamı AYRI gösterilir; uyarı eşikleri buradan ayarlanır). Ayrıca Windows servis başlatma davranışı: açılışta otomatik başlasın mı, çökünce kendini toparlasın mı. [Detaylı yardım](#)

Sunucu Kaynakları

MERGENHUB (BU SERVİS)

CPU 0% RAM 218 MB (%0.7) Thread 28 thread - 842 handle

MAKİNE (TOPLAM SİSTEM)

CPU 12% - 32 çekirdek RAM 22941 / 32530 MB (%70.5) Disk 144.8 GB boş / 930.6 GB

UYARI EŞİKLERİ CPU ≥ 85 % RAM ≥ 85 % Disk uyarı < 5 GB Disk kritik < 1 GB Kaydet

Eşik açılıncı ekranın üstünde uyarı çıkar.

24 tag - 24 taranan - 0 OPC - 0 izlenen - GC-gen2 4

Servisler & Alt Sistemler

Her modül bağımsız bir arka plan servisi. Kontrol edilebilirler başlat/durdur/yeniden başlat satır- içindedir.

Servis	Durum	Metrik	Çalışma süresi	Oto-restart	İşlem
OpcUaServer	Çalışıyor	00:01:27 0 yeniden başlatma	☒	☐ Durdur Yeniden başlat	
ScanEngine	Çalışıyor	00:01:27 0 yeniden başlatma	☒	☐ Durdur Yeniden başlat	
Historian (Veri Logları)	Boşta	0 grup - 0 etkin - 0 yazıcı çalışmıyor (etkin grup yok)	—	—	
Alarmlar	Çalışıyor	0 onaysız	—	—	

Ayarlar > Servisler – her runtime servisinin durumu, çalışma süresi, yeniden başlatma sayacı ve kontrolleri

Başlatma ve çökme kurtarma

- Üç servis de **Otomatik** başlar. Kurucunun *Servis başlatma davranışı* sayfasında *Otomatik (gecikmeli)* ya da elle seçebilirsiniz.
- Her servis için **kurtarma** tanımlıdır: süreç çökerse Windows seçtiğiniz gecikmeden sonra yeniden başlatır (varsayılan birkaç saniye). Çekirdeğin içinde her sürücü kanalı da bir gözetmen altında çalışır; ölen kanal işçisi servis yeniden başlatılmadan saniyeler içinde yeniden kurulur.
- Çekirdekte ayrıca tarama motorunu süreç içinde yeniden başlatan **runtime restart** vardır (deneme penceresi ve *Ayarlar > Servisler* kullanır). Windows servisine dokunmaz.

Servisleri yönetme

Web arayüzünde *Ayarlar > Servisler* her runtime servisini başlat/durdur/yeniden başlat ve otomatik yeniden başlatma anahtarıyla gösterir. Yönetici komut isteminden:

```
sc query MergenHub
sc stop MergenHub
sc start MergenHub
sc qc MergenHubHistorian
```

ya da PowerShell: `Get-Service MergenHub*` , `Restart-Service MergenHub` .

Not Çekirdeği durdurmak bağımlı servisleri de durdurur; çekirdeği başlatmak onları kendi başlat komutları çalışana kadar otomatik başlatmaz (açılışta Windows halleder).

System tray yöneticisi

`MergenHub.Tray.exe` tüm kullanıcılar için oturum açılışında başlar ve saatin yanında *MergenHub Administration* olarak durur.

- Servis durumunu (yeşil/sarı/kırmızı) ve lisanssızken deneme geri sayımını gösterir.
- **Runtime'ı yeniden başlat** — tarama motorunu süreç içinde yeniden başlatır (yeni deneme penceresi, sürücüler yeniden yüklenir); Windows servisini durdurmaz.
- **Servisi başlat / durdur / yeniden başlat** — Windows servisleri (yükseltme gerekir).
- **Web arayüzünü aç, Veri klasörünü aç, Olay günlüğü, sürüm ve lisans özeti.**

Tray uygulaması isteğe bağlıdır; servisler onsuz da çalışır.

Oturum açmış kullanıcı olmadan çalışma

Masaüstü oturumu gerektiren hiçbir şey yoktur. Yeniden başlatmadan sonra servisler kalkar ve kimse giriş yapmadan cihazlar taranır. Yalnız tray simgesi kullanıcı oturumu ister.

Kaynak kullanımı

Bellek tag sayısıyla büyür: birkaç bin tag için kabaca 250–350 MB, çok sayıda OPC UA aboneliyle 450K tag'de 2–3 GB. İşlemciyi sürücülerin tarama hızı belirler. *Ayarlar* > *Servisler* ve pano makinenin CPU, RAM ve diskini gösterir; RAM daralınca kapasite bandı uyarır.

Servis hesapları ve izinler

Servisler varsayılan olarak *Local System* ile çalışır ve `C:\ProgramData\MergenHub` üzerinde yazma izni ister. Özel bir hesaba geçirirseniz (örneğin ODBC kaydı için ağ paylaşımına ulaşmak amacıyla) o hesaba veri klasöründe **Değiştir** izni ve *Hizmet olarak oturum aç* hakkı verin.

Windows Olay Görüntüleyicisi

Servis başlatma/durdurma ve yakalanmamış çökmeler *Windows Logs* > *Application* altına *MergenHub* kaynağıyla da yazılır. MergenHub'ın kendi ayrıntılı logları [Loglar ve teşhis verileri](#) sayfasında.

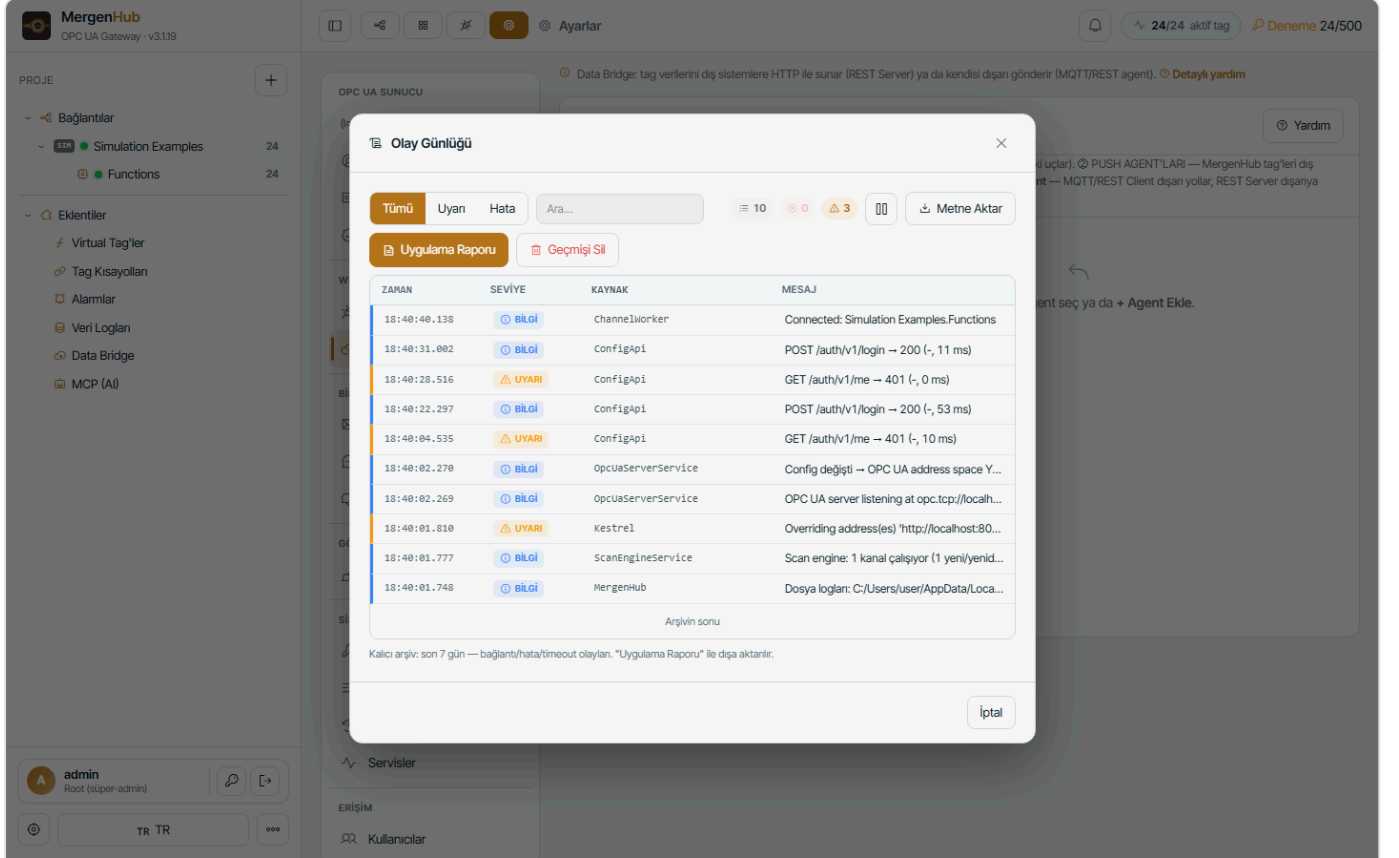
Loglar ve teşhis verileri

MergenHub'ın yazdığı her log: dosya logları, olay günlüğü, denetim günlüğü, historian veritabanı, kurulum logu; saklama süreleri ve yerleri.

MergenHub her biri farklı amaç, yer ve saklama süresine sahip birkaç tür log tutar. Hepsisi `C:\ProgramData\MergenHub` veri klasörü altındadır.

Özet

LOG	NEREDE	İÇİNDE NE VAR	SAKLAMA	NEREDEN BAKILIR
Uygulama dosya logu	<code>logs\mergenhub-YYYYAAGG.log</code>	Çekirdek sürecin logladığı her şey: açılış, sürücü bağlantıları, hatalar, yığın izleri, API istekleri	Günde bir dosya; eskiler otomatik silinir	Metin editörü; destek talebine eklenir
Olay günlüğü	<code>events.db</code> (SQLite)	Operatöre yönelik olaylar: cihaz bağlandı/koptu, zaman aşımı, servis başladı/durdu, OPC UA oturumları, alarmlar	Varsayılan 7 gün (<i>Ayarlar > Saklama</i>)	Web arayüzünde ... > <i>Olay Günlüğü</i> ; filtre, arama, <i>Metne aktar</i>
Denetim günlüğü	<code>audit.db</code>	Kim neyi değiştirdi: her yapılandırma yazımı (POST/PUT/DELETE), OPC UA ve REST tag yazmaları, girişler, MCP araç çağrıları, lisans işlemleri	Varsayılan 90 gün	<i>Ayarlar > Denetim</i> (kategori, kullanıcı, zaman çipleri)
Alarm geçmişi	proje deposunda	Her alarm geçişi: oluştu, onaylandı, silindi, kim tarafından	Varsayılan 90 gün	<i>Alarmlar > Geçmiş</i>
Historian	<code>logs.db</code> (SQLite) ya da ODBC veritabanınız	Log gruplarının kaydettiği tag değerleri	Siz silene ya da seyreltene kadar	<i>Historian > Trend</i> , CSV/Excel dışı aktarım, OPC UA HistoryRead
Cihaz teşhisi	bellekte	Cihaz başına ham istek/yanıt çerçeveleri (hex), son hata, zamanlama	Dönen tampon	<i>Cihaz > Teşhis</i> ; Wireshark tarzı iz dışı aktarımı
Kurulum logu	<code>%TEMP%\MergenHub-Setup.log</code>	Kurucunun her adımı, servis ve güvenlik duvarı komutları	Siz silene kadar	Metin editörü
Windows Olay Görüntüleyicisi	Application, kaynak <i>MergenHub</i>	Servis yaşam döngüsü, yakalanmamış çökmeler	Windows politikası	<code>eventvwr.msc</code>



Olay Günlüğü penceresi – seviye filtresi, arama, dışa aktarım ve Uygulama Raporu düğmesi

Uygulama dosya logu

Çekirdek `C:\ProgramData\MergenHub\logs\` altına dönen bir metin logu yazar. Her satırda zaman damgası, seviye (`INF` , `WRN` , `ERR`), kaynak bileşen ve mesaj vardır. Tipik satırlar:

```
2026-09-05 10:14:02.117 INF Web API listening on http://0.0.0.0:8090
2026-09-05 10:14:03.402 INF OPC UA server started: opc.tcp://0.0.0.0:48010/MergenHub
2026-09-05 10:14:05.930 INF ChannelWorker Connected: Terfi_Merkezi.PLC_1
2026-09-05 10:20:41.008 WRN ModbusTcp Read timeout 10.0.0.5:502 unit 1 (block 40001..40064),
retry 1/3
2026-09-05 10:20:44.012 ERR ChannelWorker Device Terfi_Merkezi.PLC_1 marked Bad: 3 consecutive
failures
```

Kaç gün saklanacağı *Ayarlar* > *Saklama*'dadır. Destek için sorunun yaşandığı günün dosyasını gönderin.

Olay günlüğü

Olaylar, operatörün log dosyası okumadan görmesi gerekenlerdir: bağlantı durumu değişimleri, zaman aşımaları, servis yeniden başlatmaları, OPC UA istemci oturumları, yapılandırma yeniden yüklemeleri, lisans/deneme iletileri. Sağ üstteki ... menüsünden **Olay Günlüğü** ile açılır.

- **Tümü / Uyarı / Hata** filtresi, serbest metin arama, otomatik yenilemeyi duraklatma.
- **Metne aktar** mevcut görünümü kaydeder.
- **Uygulama Raporu** sürüm, ayar özeti, kanal/cihaz sağlığı ve son olayları tek metin dosyasında toplar — destek talebine eklenecek en hızlı şey.
- Olaylar ayarlanan gün sayısı kadar (varsayılan 7) `events.db` 'de kalır ve yeniden başlatmalarda kaybolmaz.

Denetim günlüğü

Web arayüzü, REST API, OPC UA yazmaları ya da MCP aracıyla yapılan her değişiklik kullanıcı, IP, zaman, eylem ve hedefle kaydedilir. *Ayarlar > Denetim*'den açın. Örnekler:

- `channel.create` , `device.update` , `tag.delete` , `project.import`
- OPC UA ya da REST'ten `tag.write` — eski ve yeni değerle
- `user.login` , `user.login_failed` , `user.password_change`
- `license.activate` , `license.release` , `settings.update`

Saklama varsayılan 90 gündür (*Ayarlar > Saklama*). Denetim kayıtları arayüzden düzenlenemez.

MergenHub
OPC UA Gateway - v3.11.9

Ayarlar

Denetim günlüğü: kim, ne zaman, hangi işlemi yaptı — tip filtreleriyle incelenir, geçmiş temizlenebilir. [Detaylı yardım](#)

Denetim

Tümü Kanal Cihaz Tag Grup Yazma Servis Proje Güvenlik Ayarlar Alarm Denetim MCP

Diğer

İşlem/hedef ara... kullanıcı gg.aa.yyyy --:-- --:-- Tümü ✓ ✕

1000 Yenile CSV Geçmiş Temizle

Zaman	Kullanıcı	Tip	İşlem	Hedef	IP	Sonuç
05.09.2026 18:40:31	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✓
05.09.2026 18:40:22	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✓
05.09.2026 18:39:03	admin	Diğer	İşlem güncellendi	config/v1/settings	1	✓
05.09.2026 18:38:42	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✓
05.09.2026 18:02:27	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✓
05.09.2026 18:00:45	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✓
05.09.2026 17:55:34	-	Güvenlik	Kimlik/şifre eklendi/çalıştırıldı	auth/v1/login	1	✕

Her işlem otomatik kaydedilir (kim/ne zaman/IP/sonuç). Tip çipleriyle söz. - 7 kayıt - başlığa tıkla = sırala - satıra tıkla = tam detay

Ayarlar > Denetim — kategori, kullanıcı ve zaman çipleri; her yapılandırma değişikliği listelenir

Alarm geçmişi

Her alarm geçişi (oluştur, onaylandı, silindi) zaman damgası, değer, kullanıcı ve yorumla alarm geçmişi saklama süresi (varsayılan 90 gün) kadar tutulur ve *Alarmlar > Geçmiş*'te gösterilir. OPC UA Alarms & Conditions istemcilerine de aşıktır.

Historian verisi

Log grupları varsayılan olarak `logs.db` 'ye (SQLite) ya da herhangi bir ODBC veritabanına (SQL Server, MySQL/MariaDB, PostgreSQL, Oracle) yazar. Hedef ulaşılamazsa kayıtlar diske tamponlanır (**store-and-forward**) ve sonra yazılır. Trend, seyreltme ve CSV/Excel/PNG dışa aktarım *Historian* modülündedir; bkz. [Historian](#).

Bir cihaz seçip **Teşhis**'i açın: hex olarak son istek/yanıt çerçeveleri, bayt sayaçları, zamanlama ve son hata metni. Önbellek tabanlı protokollerde (DNP3, IEC 104, IEC 61850) ham TX/RX musluğu gerçek hat trafiğini gösterir. İz analiz için kaydedilebilir. Bkz. [Sorun giderme](#).

Ayarlar › Saklama:

AYAR	VARSAYILAN
Olay günlüğü (gün)	7
Denetim günlüğü (gün)	90
Alarm geçmişi (gün)	90
Dosya logu (gün)	dönen, otomatik

MergenHub

OPC UA Gateway - v3.1.19

PROJE

+

Bağlantılar

Simulation Examples

24

Functions

24

Eklentiler

Virtual Tag'ler

Tag Kısayolları

Alarmlar

Veri Logları

Data Bridge

MCP (AI)

admin

Root (super-admin)

TR TR

Ayarlar

Detaylı yardım

Saklama

Loglar/geçmiş verilen günden eski olunca sistem OTOMATİK siler (saatlik). 0 = süresiz sakla.

Denetim Günlüğü · 7 kayıt

Kim ne yaptı (audit.db) — güvenli izi

90

≈ 3 ay

Olay Günlüğü · 10 kayıt

Sistem/iletişim olayları (events.db)

7

7 gün

Alarm Geçmişi · 0 kayıt

Geçmiş alarm olayları (raise/clear/ack)

90

≈ 3 ay

Veri Logları (historian) kendi saklama süresini her log grubunda ayrı tutar.

KaydetŞimdi Temizle

0 gidersen o günlük hiç silinmez.

Görünüm

Marka

Sistem

Lisans

Denetim

Saklama

Servisler

Erişim

Kullanıcılar

Ayarlar > Saklama – olay, denetim ve alarm geçmişinin ne kadar tutulacağı

1. ... › *Olay Günlüğü* › *Uygulama Raporu* — dosyayı kaydedin.
2. O günün `logs\mergenhub-YYYYAAGG.log` dosyasını ekleyin.
3. Cihaz sorunuysa o cihazın *Teşhis* izini ekleyin.
4. Müşteri portalından (*Destek*) ya da e-postayla gönderin.

Lisanslama ve deneme modu

Deneme kısıtları, online ve çevrimdışı etkinleştirme, lisansı serbest bırakma ve taşıma, abonelik ve bakım.

MergenHub'ın her özelliği her pakette açıktır; paketler yalnız projenin içerebileceği **tag sayısı**yla ayrılır. Lisanslar kalıcı ya da yıllık abonelik, [fiyat sayfasından](#) ya da bayiden alınır ve uygulamada online ya da çevrimdışı etkinleştirilir.

Paketler

PAKET	TAG TAVANI
Başlangıç	1.000
Standart	5.000
Profesyonel	25.000
Kurumsal	50.000
Enterprise	sınırsız

Tag sayımı projedeki toplam statik tag'dir (kanal > cihaz > tag, gruplar dahil). Virtual tag'ler, sistem tag'leri, alias'lar ve SCADA ekranları sayılmaz.

Deneme modu

Lisanssız kurulum **deneme modunda** çalışır:

- tüm özellikler açık,
- en çok **500 tag**,
- runtime (tarama motoru) **her başlatmadan sonra 2 saat** çalışır; pencere dolunca tarama durur ve tag kalitesi *Bad* olur; yeni pencere için üst çubuktaki düğme, tray > *Runtime'ı yeniden başlat* ya da servisi yeniden başlatmak yeter.

Üst çubukta geri sayımlı *DENEME* rozeti görünür. Lisans anahtarı girildiğinde iki kısıt da anında kalkar; yeniden kurulum ya da yeniden başlatma gerekmez.

Online etkinleştirme

- Lisans satın alın; anahtar **MH-XXXX-XXXX-XXXX-XXXX** biçimindedir, müşteri portalında görünür ve e-postayla gelir.
- MergenHub'da *Ayarlar > Lisans*'ı açın, anahtarı yapıştırın, **Etkinleştir**'e tıklayın.
- Uygulama lisans portalına bağlanır, anahtarı bu makinenin parmak izine bağlar ve imzalı lisans belgesini yerel diske yazar.

Belge birkaç günde bir otomatik tazelenir (abonelik bitişi, bakım durumu, iptal). Portal ulaşılamazsa lisans çalışmaya devam eder; 14 gün sonra uyarı gösterilir, hiçbir şey engellenmez.

Çevrimdışı etkinleştirme (sunucuda internet yok)

1. *Ayarlar > Lisans > Çevrimdışı etkinleştirme* — anahtarı girin, **İstek dosyası oluştur** (`.mhreq`).
2. İnternetli herhangi bir bilgisayardan müşteri portalına girin, lisansı açın, isteği **Çevrimdışı etkinleştirme** altında yükleyin; lisans belgesini (`.mhlic`) indirin.
3. MergenHub'da **Belge içe aktar** ile `.mhlic` dosyasını seçin.

Belge, isteği oluşturan makineye bağlıdır.

Lisansı serbest bırakma ve taşıma

Bir lisans aynı anda tek makinede geçerlidir. Taşımak için:

- **Online:** *Ayarlar > Lisans > Serbest bırak* — portal anahtarı hemen boşa çıkarır; yeni makinede etkinleştirin.
- **Çevrimdışı:** *Serbest bırak (çevrimdışı)* imzalı kanıt dosyası (`.mhrel`) üretir; portalda lisansın altına yükleyin, sonra başka yerde etkinleştirin.
- **Makine çöktü / ulaşılamıyor:** müşteri portalında lisansı açıp *Zorla serbest bırak* kullanın (ya da bize yazın); eski aktivasyon kara listeye alınır.

Donanım değişikliği: parmak izi bir bileşenin değişmesine (ör. disk) tolerans gösterir. Büyük bir yenilemeden sonra serbest bırakıp yeniden etkinleştirin.

Abonelik ve bakım

- **Abonelik** lisansının bitiş tarihi vardır; portaldan yenileyin, uygulama yeni tarihi bir sonraki tazelemede alır (çevrimdışıysa yeni belgeyi içe aktarın). Bitişten sonra 30 günlük salt-okunur hoşgörü vardır (veri akar, yapılandırma kilitli), sonra deneme modu.
- İlk yıl **Bakım & Destek** dahildir. Bakım anlaşması olmadan da tüm yükseltmeler ücretsizdir; yalnız destek talepleri yanıtlanmaz.

Güncellemeler

Lisanslı kurulumlar portalı günde bir denetler ve indirme + sürüm notu bağlantılı *Yeni sürüm var* balonu gösterir (sürüm bazında *bir daha gösterme*). *Ayarlar > Lisans > Güncellemeleri denetle* denetimi hemen çalıştırır.

Sık sorulanlar

SORU	YANIT
Tag tavanına ulaştım	API <i>402 Payment Required</i> ile tavanı ve mevcut sayıyı döner; portaldan paketi yükseltin, yeni tavan bir sonraki tazelemede gelir.
Birden çok kurulum çalıştırabilir miyim?	Bir lisans = bir makine. Sunucu başına bir lisans alın (bayilere toplu iskonto).
Bakım bitince lisans sona erer mi?	Hayır. Kalıcı lisanslar hiç bitmez; bakım yalnız desteği etkiler.
Lisans nerede saklanıyor?	<code>C:\ProgramData\MergenHub\license\</code> — makineye bağlıdır, başka yere kopyalamak işe yaramaz.

Yedekleme, geri yükleme, yükseltme, kaldırma

Neyi yedeklemeli, proje dışı/içe aktarma, yerinde yükseltme, kaldırma.

MergenHub'ın sahip olduğu her şey tek klasörde durur; bu da yedeklemeyi, taşımayı ve yükseltmeyi basitleştirir.

Ne yedeklenmeli

ÖĞE	YER	NOT
Tüm veri klasörü	C:\ProgramData\MergenHub\	Proje, SCADA, kullanıcılar, ayarlar, lisans, sertifikalar, loglar, historian
Yalnız proje	Bağlantılar › Dışa aktar → JSON	Kanallar, cihazlar, tag'ler, alarmlar, log grupları, virtual tag'ler, alias'lar
Yalnız SCADA	SCADA › Projeyi dışa aktar	Ekranlar, bölgeler, sembol kütüphanesi, bağlamalar
Dış veritabanındaki historian	ODBC veritabanınız	Veritabanının kendi araçlarıyla yedekleyin

İpucu C:\ProgramData\MergenHub klasörünün gecelik kopyası (servisler çalışırken de olur; SQLite dosyaları tutarlıdır) sunucuyu sıfırdan kurmaya yeter.

Örnek zamanlanmış görev (robocopy, 14 günlük kopya tutar):

```
robocopy "C:\ProgramData\MergenHub" "D:\Yedek\MergenHub\%date:~-4%%date:~3,2%%date:~0,2%" /MIR /R:1 /W:1 /XD logs
forfiles /P "D:\Yedek\MergenHub" /D -14 /C "cmd /c rmdir /s /q @path"
```

Geri yükleme

1. Hedef makineye aynı ya da daha yeni MergenHub sürümünü kurun.
2. Üç servisi durdurun (`sc stop MergenHub`).
3. C:\ProgramData\MergenHub klasörünü yedekle değiştirin (host adı değiştiyse yeni webpki\ klasörünü koruyun).
4. Servisleri başlatın. Kullanıcılar, proje, ekranlar ve geçmiş geri gelir.
5. Lisansı yeni donanımda yeniden etkinleştirin (önce eski makinede serbest bırakın ya da portalda *Zorla serbest bırak*).

Başka sunucuya taşıma

Geri yüklemeyle aynıdır. Hatırlanacaklar:

- OPC UA istemcileri **sunucu sertifikasına** güvenir; pki\own kopyalanmadıysa yeni makinede yeni sertifika oluştur ve istemcilerin yeniden güvenmesi gerekebilir.
- Web HTTPS sertifikası host adına bağlıdır; ad değiştiyse MergenHub.exe gencert "<ad>" ile yeniden üretin.

- Güvenlik duvarı kuralları makineye özeldir — kurucu oluşturur; bkz. [Güvenlik duvarı](#).

Yükseltme

Yeni kurucuyu çalıştırın. Kurucu:

- servisleri ve tray uygulamasını durdurur,
- program dosyalarını tamamen değiştirir (bayat DLL kalmaz),
- `C:\ProgramData\MergenHub` klasörüne dokunmaz,
- gerekiyorsa eski servis adlarını taşır,
- her şeyi yeniden başlatır.

Kesinti tipik olarak bir dakikanın altındadır. Sürüm notları [sürümler sayfasında](#); lisanslı kurulumlar uygulama içinde bilgilendirilir. Yükseltmeler lisans ömrü boyunca ücretsizdir.

Not Sürüm düşürme desteklenmez. Geri dönüş istiyorsanız önceki kurucuyu ve bir yedeği saklayın.

Proje dışı / içe aktarma

Bağlantılar › *Dışa aktar* projeyi JSON olarak yazar; *İçe aktar* MergenHub ya da Kepware JSON dışı aktarımlarını okur. İçe aktarma projeyi **değiştirebilir** ya da içine **birleştirebilir** (aynı adlı kanallar değiştirilir). Cihaz ayarlarındaki parolalar proje anahtarıyla şifreli aktarılır; başka sunucuda içe aktarırken anahtar sorulur.

Kaldırma

Ayarlar › *Uygulamalar* ya da program klasöründeki kaldırıcı. Üç servisi durdurup siler, güvenlik duvarı kurallarını ve tray otomatik başlatmasını kaldırır, program dosyalarını siler. **Veri klasörü korunur**; sonraki kurulum her şeyi yerinde bulur. Makineyi tamamen temizlemek istiyorsanız `C:\ProgramData\MergenHub` klasörünü elle silin.

Kanallar, cihazlar ve tag'ler

Proje hiyerarşisi, tarama hızı ve modları, ölçekleme, tag grupları, adres yardımcıları, tag üretimi.

Proje üç seviyeli bir ağaştır: **kanal** → **cihaz** → **tag**. Diğer her şey (alarmlar, log grupları, virtual tag'ler, ekranlar) tag'lere tam yollarıyla `Kanal.Cihaz.Tag` biçiminde başvurur.

The screenshot shows the MergenHub interface. On the left, the project hierarchy is displayed under 'PROJE'. It includes 'Bağlantılar' (Connections) with 'Simulation Examples' (24) and 'Functions' (24). Under 'Eklentiler' (Add-ons), there are 'Virtual Tag'ler, 'Tag Kısıyolları' (Tag Restrictions), 'Alarmlar' (Alarms), 'Veri Logları' (Data Logs), 'Data Bridge', and 'MCP (AI)'. The main area shows 'Simulation Examples cihazları' (Simulation Examples devices) with a table listing devices. The table has columns: Ad (Name), Model, ID, and Tag. One device is listed: 'Functions' with ID 4 and Tag 24. The bottom status bar shows 'Bağlantılar - kanal listesi, cihaz ağacı ve canlı tag tablosu'.

Kanallar

Kanal, taşıma ayarlarıyla birlikte bir sürücü örneğidir. Fiziksel bağlantı başına bir kanal açın:

SÜRÜCÜ AİLESİ	KANAL BAŞINA...
Modbus TCP, S7, OMRON, DNP3, IEC 104, IEC 61850, OPC UA istemci	ağ segmenti ya da izole tutmak istediğiniz cihaz grubu
Modbus RTU/ASCII	seri port (COM1, COM2 ...) ya da Ethernet-encapsulation geçidi
MQTT	broker bağlantısı
Simülatör	istediğiniz kadar

Kanal ayarları: ağ adaptörü, bağlantı/istek zaman aşımı, yeniden deneme sayısı, istekler arası bekleme ve sürücüyü özel seçenekler (Modbus blok boyutu, S7 PDU boyutu vb.). Kanallar **izoledir**: yanıt vermeyen bir cihaz başka kanaldaki cihazları asla yavaşlatmaz.

Cihazlar

Cihaz; bir PLC, RTU, IED, sayaç ya da broker konu kümesidir. Önemli ayarlar:

- **ID / adres** — IP:port ve unit ID (Modbus), IP + rack/slot (S7), IED adresi (61850), istasyon adresleri (DNP3/104), konu kökü (MQTT).
- **Tarama hızı** — cihazın kaç ms'de bir sorgulanacağı. Tag'ler değiştirmezse bunu devralır.
- **Tarama modu** — *istemci hızına saygı* (OPC UA aboneleri daha hızlı/yavaş isteyebilir), *sabit* ya da *istemci güdümlü*.
- **Zaman aşımı ve yeniden deneme** — ayarlanan başarısızlıktan sonra cihaz *Bad* işaretlenir ve artan aralıklarla yeniden denenir; kendiliğinden toparlanır.
- **Talep güdümlü tarama** — cihaz yalnız bir tüketici tag'lerini istediği sürece sorgulanır; boştaki cihazlara dokunulmaz (GSM/telsiz hatlarında bant genişliği kazandırır).
- **Teşhis** — hex istek/yanıt izi, sayaçlar, son hata.
- **Kopyala** — cihazı tüm tag'leriyle çoğaltır (adresler kaydırılabilir).

Ayarlar > Servisler her kanal işçisini ve sağlığını tek bakışta gösterir

Tag'ler

Tag, cihazdaki bir adresi tipli bir değere eşler:

ALAN	ANLAMI
Ad	cihaz içinde benzersiz; OPC UA node id'lerinde ve REST yollarında kullanılır
Adres	sürücüye özel (40001 , DB10.DBD24 , D100 , ns=2;s=... , AI:3 , IOA 1001 , LD0/MMXU1.TotW.mag.f , tesis/hat1/sicaklik) — ? biçimleri gösterir
Veri tipi	Boolean, Byte, Short, Word, Long, DWord, LLong, QWord, Float, Double, String, Date, BCD/LBCD, sürücü destekliyorsa diziler
Okuma/Yazma	salt okunur ya da okuma/yazma (yazmalar hemen cihaza gider ve denetime düşer)
Ölçekleme	ham/ölçekli aralıklarla doğrusal ya da karekök ölçekleme, sınırlama; yazmalar ters ölçeklenir
Tarama hızı	cihaz hızının isteğe bağlı üzerine yazımı
Açıklama / birim	OPC UA'da (Description, EngineeringUnits) ve arayüzde görünür

Tag grupları büyük cihazları klasörlere böler ve OPC UA'da klasör olarak görünür.

Toplu tag oluşturma

- **Tag üret** — adres aralığından (başlangıç, adet, adım, adlandırma deseni).
- **Çevrimiçi tarama** — tag listesini cihazdan okur: OPC UA sunucuları, IEC 61850 (SCL/ICD içe aktarma ya da canlı GVA yürüyüşü), DNP3 integrity poll, IEC 104 genel sorgulama, OMRON NJ/NX sembolleri, JSON işaretçili MQTT konu keşfi.
- **İçe aktar** — Kepware ya da MergentHub JSON dışa aktarımları; tag tablosunun içe aktarma düğmesiyle CSV.

Kalite ve zaman damgası

Her değer bir OPC kalitesi taşır: **Good**, **Bad** (cihaz ulaşılamaz, hata ya da deneme penceresi dolmuş), **Uncertain** (bayat/ön bellek). Hem *kaynak zaman damgası* (varsa cihaz saati) hem *sunucu zaman damgası* tutulur ve OPC UA'da sunulur.

Alias'lar

Eklentiler › *Alias'lar* bir tag'i ikinci, daha anlaşılır bir adla yayınlar (ör. Modbus_1.RTU_3.AI_40011 yerine Tesis.Giris.Debi). Alias'lar OPC UA ve REST'te normal tag gibi görünür.

Sistem tag'leri

Her cihazın haberleşme sağlığını bildiren `_System` tag'leri vardır (`_Error` , `_NoError` , `_LastRead` , `_SuccessfulReads` , `_Enabled` ...); loglanabilir ve alarmlanabilir. `_Enabled` yazılarak cihaz duraklatılabilir.

Protokol sürücüleri

Modbus, Siemens S7, OMRON, DNP3, IEC 104, IEC 61850, MQTT, OPC UA istemci, simülatör — her biri neyi destekler.

Tüm sürücüler her kurulumda gelir; satın alınacak ya da açılacak bir şey yoktur. Her sürücünün tag editöründe *Adres biçimleri* paneli ve uygulama içinde yardım sayfası vardır.

SÜRÜCÜ	TAŞIMA	ÖNE ÇIKANLAR	AYRINTI
Modbus	TCP, UDP, RTU/ASCII seri, Ethernet encapsulation	FC01-04 okuma, FC05/06/15/16/22 yazma, sıfır/bir tabanlı adresleme, Modicon 5/6 hane, word/bit sırası, bölge başına maksimum blok, kanal başına tek soket	Modbus
Siemens S7	ISO-on-TCP (102)	S7-300/400/1200/1500, DB/M/I/Q/T/C alanları, optimize PDU paketleme, 64-bit tipler	Siemens S7
OMRON	FINS/TCP, EtherNet/IP CIP	CS/CJ/CP FINS bellek alanları, NJ/NX sembolik tag'ler (CIP), bit ve word erişimi	OMRON
DNP3	TCP	Integrity/event poll, unsolicited, SBO/Direct Operate, Secure Authentication, zaman senkronu, dosya kontrolü, yedeklilik	DNP3
IEC 60870-5-104	TCP	Genel sorgulama, tek/çift komut, setpoint, zaman senkronu, yedeklilik, RTU uyumluluk kipleri	IEC 104
IEC 61850	MMS	SCL/ICD içe aktarma, LN seviyesi okuma, kontrol modeli (SBO/direct), raporlar, VMD adresleri	IEC 61850
MQTT	TCP/TLS	Kanal başına broker, konu cihazları, JSON/XML/CSV/anahtar-değer ayrıştırma, keşif, varış-watchdog canlılığı	MQTT
OPC UA istemci	opc.tcp	Başka UA sunucularına bağlanır (PLC, geçit, başka MergenHub); kanal = oturum, cihaz = abonelik, tarayarak tag üretimi	—
Simülatör	—	Ekran, alarm ve kayıt testleri için rampa, sinüs, rastgele, sabit fonksiyonlar	—

Kanal ve cihaz sınırları

Her sürücü için açılacak en fazla kanal ve bir kanaldaki en fazla cihaz sayısı sınırlıdır. Değerler endüstri standardı OPC sunucusunun sürücü kılavuzlarındaki sınırlarla aynıdır; kanal tavanı ayrıca 1024'tür. Sınır **sürücü başı**na, proje toplamı değildir — farklı sürücülerle toplamda daha fazla kanal açılabilir.

SÜRÜCÜ	EN FAZLA KANAL	KANAL BAŞINA EN FAZLA CİHAZ
Modbus TCP/IP Ethernet	1024	8192
Modbus RTU Serial	1024	255
Modbus ASCII Serial	100	247
DNP3 Client Ethernet	1024	1024
IEC 60870-5-104 Client	1024	1024

SÜRÜCÜ	EN FAZLA KANAL	KANAL BAŞINA EN FAZLA CİHAZ
IEC 61850 MMS Client	256	256
MQTT Client	60	1024
OMRON FINS Ethernet	512	32
OMRON NJ Ethernet	256	1024
Siemens S7 TCP/IP	1024	1024
OPC UA istemci	256	256
Simülatör	1024	999

- Sınır yalnız **ekleme** sırasında denetlenir (arayüz, REST API, yapay zekâ asistanı, kopyalama, içe aktarma). Doluysa işlem reddedilir ve proje değişmez; içe aktarma bütünüyle reddedilir, kısmi içe aktarma yapılmaz.
- Kayıtlı mevcut proje yüklenirken hiçbir şey reddedilmez ya da silinmez.
- Her değer, çalışan bir MergenHub'a karşı protokol simülatörleriyle sınırın kendisinde denendi; tüm sürücüler **aynı anda** sınırındayken de (7584 kanal) tüm tag'ler Good kaldı ve değerler kesintisiz güncellendi.
- Bir kanal cihazlarını **sırayla** tarar: sınıra yakın kalabalık bir kanalda tarama turu uzar (testte tek kanalda 8192 Modbus TCP cihazı ~40 sn'de bir tazelendi). Hızlı güncelleme gereken cihazları birden çok kanala bölün — kanallar paralel çalışır.

Zaman aşımı ve tarama hızı seçimi

- Sürücü varsayılanları ve cihazın belgelenen yanıt süresiyle başlayın. Cihazın gidiş-dönüş süresinden kısa tarama hızı yalnız kuyruk biriktirir.
- Seri ve GSM/telsiz hatlar: daha uzun istek zaman aşımı (2–5 sn), daha az yeniden deneme ve boş cihazların hattı meşgul etmemesi için talep güdümlü tarama.
- Büyük Modbus cihazları: *maksimum blok*u cihazın desteklediğinin altında tutun (çoğu RTU 64 register üstünde resetlenir).

Yazmalar

Yazmalar tarama kuyruğunun önünde **hemen** yürütülür ve doğrulanır: sürücü cihazın yazmayı kabul ettiğini kontrol eder (Modbus istisna kodları, S7 dönüş kodları, DNP3/104 komut yanıtları). Başarısız yazma çağırana bildirilir (OPC UA StatusCode / REST hatası) ve denetim günlüğüne kullanıcı ve değerle yazılır.

Yedeklilik ve toparlanma

Her kanal bir gözetmen altında çalışır. Kopan soketler yeniden açılır, ölen işçiler yeniden kurulur, karantina/geri çekilme çırpınan bir cihazın logu boğmasını önler. DNP3 ve IEC 104 çift adres yedekliliğini destekler.

Teşhis

Cihaz › **Teşhis** ham çerçeveleri (TX/RX hex), bayt sayaçlarını ve son hatayı gösterir. Önbellek tabanlı protokollerde (DNP3, 104, 61850) paket musluğu önbellek değerlerini değil gerçek hat trafiğini gösterir. Bkz. [Sorun giderme](#).

OPC UA sunucusu

Endpoint'ler, güvenlik ilkeleri, sertifikalar, adres uzayı, sistem tag'leri, geçmiş erişimi, UA üzerinden alarmlar.

MergenHub her kanal, cihaz ve tag'i tek adres uzayında sunan bir **OPC UA sunucusudur** (Data Access, Historical Access, Alarms & Conditions, Methods). Herhangi bir UA istemcisi — SCADA, MES, historian, PLC, başka bir MergenHub — bağlanır.

```
opc.tcp://<host>:48010/MergenHub
```

The screenshot shows the MergenHub OPC UA Gateway interface. The sidebar on the left contains navigation options: PROJE, Bağlantılar, Ekentiler, and Ayarlar. The main panel displays the OPC UA Sunucu configuration. The 'Endpoint yolu' (Endpoint path) is set to 'MergenHub'. The 'Güvenlik' (Security) section shows the selected security policy 'Basic256Sha256 - Sign' and the 'Güvenilmeyen sertifikaları otomatik kabul' (Accept untrusted certificates automatically) checkbox. The 'OPC UA PROJE ÖZELLİKLERİ' (OPC UA Project Properties) section includes options for 'Log diagnostics' and 'Alarmları OPC UA'dan yayınla' (Publish alarms from OPC UA).

Ayarlar > OPC UA Sunucu – endpoint'ler, güvenlik ilkeleri, sertifikalar ve proje özellikleri

Adres uzayı

```
Objects
├── <Kanal>
│   ├── <Cihaz>
│   │   ├── <Tag> (Variable: değer, kalite, kaynak/sunucu zaman damgası, EngineeringUnits,
│   │   │   │   │   Description)
│   │   │   ├── <TagGrubu>/...
│   │   │   └── _System (_Error, _NoError, _LastRead, _SuccessfulReads, _Enabled ...)
│   ├── _Aliases
│   ├── _AdvancedTags
│   ├── _Alarms ("alarmları OPC UA'dan yayınla" açıkken)
│   └── _System (sunucu geneli istatistikler)
```

Node id'ler dizge tabanlıdır (`ns=2;s=Kanal.Cihaz.Tag`); yeniden başlatma ve dışa aktarmalarda sabit kalır.

Endpoint'ler

Sunucu aynı anda birden çok endpoint'te dinleyebilir (*Ayarlar › OPC UA Sunucu › Endpoint'ler*); örneğin yerel istemci için güvenli `localhost` ve LAN/genel host için *Basic256Sha256 – Sign & Encrypt*. Her endpoint'in kendi host, port ve izinli güvenlik ilkeleri vardır. Değişiklik runtime yeniden başlatılınca geçerli olur.

Güvenlik

AYAR	ÖNERİ
Güvenlik ilkeleri	None yalnız localhost'ta; başka her yerde Basic256Sha256 Sign & Encrypt
Anonim erişim	üretimde kapalı
Kullanıcı doğrulama	<i>Ayarlar › OPC UA Kullanıcılar</i> : rollü kullanıcı/parola hesapları; sertifika desteklenir
Güvenilmeyen sertifikaları otomatik kabul	üretimde kapalı; istemcileri <i>Ayarlar › Güvenilen istemciler</i> 'de güvenilir yapın
Sunucu sertifikası	kurulumda üretilir; host adı değişince yeniden üretin (<i>Ayarlar › Sunucu sertifikası</i>)

Kullanıcı bazlı **kaynak ACL** OPC UA'da da geçerlidir: arayüzde bir kanalı göremeyen kullanıcı adres uzayında da görmez; yazma için yazma izni gerekir. Bkz. [Kullanıcılar ve güvenlik](#).

Oturum ve abonelik sınırları

Varsayılanlar: 128 oturum, oturum zaman aşımı 20 dk, 450K izlenen öge kanıtlanmış. Yayın aralığı 100 ms'ye kadar; sunucu istemcinin istediği örnekleme hızına uyar (tarama modu *istemci hızına saygı*) — aboneler taramayı sürükleyebilir.

Geçmiş erişimi

Bir historian log grubuna ait tag'ler **HistoryRead** isteklerini (ham ve işlenmiş toplamalar) doğrudan historian veritabanından yanıtlar; UA istemcileri ayrı historian olmadan trend çizebilir. Bkz. [Historian](#).

Alarms & Conditions

Alarmları OPC UA'dan yayınla açıkken her alarm tanımı `_Alarms` altında bir `AlarmConditionType` olur; istemciler Server nesnesine olay aboneliği açar, `ConditionRefresh` ve `Acknowledge` kullanır. Durum eşlemesi ISA-18.2'yi izler (aktif/onaylı/silinmiş). Bkz. [Alarmlar](#).

Metotlar

Runtime kontrolü için (kanal yeniden başlat, okumaya zorla) sunucu tarafı metotlar yalnız yetkili kullanıcılara açıktır.

Yaygın istemcileri bağlama

- **UaExpert** — Add server › endpoint URL'sini girin › güvenlik seçin › sunucu sertifikasına güvenin; *otomatik kabul* kapalıysa UaExpert sertifikasını *Ayarlar › Güvenilen istemciler*'de güvenilir yapın.
- **Kepware / Ignition / Siemens WinCC** — aynı endpoint; imzalı oturumlar için MergenHub sertifikasını (`pki\own`) istemcinin güven deposuna alın.
- **Başka bir MergenHub** — bu sunucuyu kanal olarak *OPC UA istemci* sürücüsüyle kullanın.

Sorun giderme

BELİRTİ	ÇÖZÜM
BadCertificateHostNameInvalid	İstemci sertifikada olmayan bir adla bağlanıyor — o host'la endpoint ekleyip sunucu sertifikasını yeniden üretin
BadSecurityChecksFailed / BadCertificateUntrusted	İstemci sertifikasına güvenin (<i>Ayarlar › Güvenilen istemciler</i>) ya da geçici otomatik kabul açın
BadIdentityTokenRejected	Anonim kapalı ve kullanıcı/parola yanlış
İstemci Bad kalite görüyor	Cihaz çevrimdışı ya da deneme penceresi dolmuş — <i>Bağlantılar</i> ve Olay günlüğü
Arayüzde sunucu "çalışmıyor"	Portu başka bir OPC sunucusu kullanıyor; <i>Endpoint'ler</i> 'den portu değiştirin

REST API

Kimlik doğrulama, yapılandırma API'si, canlı değerler, yazma, Swagger, curl örnekleri.

Web arayüzünün yaptığı her şey aynı REST API'den geçer; tıkladığınız her şeyi betikleyebilirsiniz. API, web portunda (varsayılan 8090) HTTP(S) üzerinden JSON'dur ve uygulama içinde canlı belgelenmiştir (... > *REST API*, Swagger UI).

Ayarlar > Web & REST API – port, HTTPS, CORS ve API erişim seçenekleri

Kimlik doğrulama

İki yol:

- Oturum çerezi** — `POST /auth/v1/login` `{"username": "...", "password": "..."}` bir çerez döner; her istekle gönderin. Oturumlar güvenlik politikasına uyar (boşta zaman aşımı, tek/çoklu giriş).
- API token** — *Ayarlar > API token'ları*ndan bir kullanıcı için token üretin ve `Authorization: Bearer <token>` gönderin. Token kullanıcının rolünü ve kaynak ACL'sini taşır; istediğiniz an iptal edin.

```
curl -c c.txt -X POST https://scada.example.com/auth/v1/login \
-H "Content-Type: application/json" -d '{"username":"apiuser","password":"...."}'
```

Ana uç grupları

GRUP	ÖRNEK	AMAÇ
/auth/v1/*	login , me , logout , forgot	kimlik doğrulama
/config/v1/project/*	channels , channels/{ch}/devices , .../tags (GET/POST/PUT/DELETE)	proje yapılandırması, JSON dışı aktarımla aynı model
/config/v1/settings	GET/PUT	sunucu ayarları
/config/v1/license/*	activate, refresh, release, update-check	lisanslama
/config/v1/scada/*	screens, library, symbols	SCADA ekranları
/config/v1/advanced , /aliases , /alarms , /logger	CRUD	eklentiler
/runtime/v1/values? tags=A.B.C,...	GET	kalite ve zaman damgalı canlı değerler
/runtime/v1/write	POST {"tag": "A.B.C", "value": 42}	cihaza yazma (denetimli)
/runtime/v1/status , /metrics , /services/*	GET/POST	sağlık, CPU/RAM, servis kontrolü
/runtime/v1/events , /audit	GET	olay ve denetim günlükleri
/logger/v1/history?tag=... &from=...&to=...	GET	historian verisi
/iotgateway/*	GET/POST	Kepware uyumlu Data Bridge REST sunucusu (bkz. Data Bridge)

Yapılandırma yazmaları sürüm çıkışmasında **409**, lisans tag tavanı aşılabaksa **402**, kaynak ACL reddederse **403** döner.

Örnekler

Üç tag oku:

```
curl -b c.txt "http://localhost:8090/runtime/v1/values?  
tags=Tesis.PLC1.Debi,Tesis.PLC1.Basinc,Tesis.PLC1.Pompa1_Calisiyor"
```

```
[{"tag": "Tesis.PLC1.Debi", "value": 412.6, "quality": "Good", "ts": "2026-09-05T10:20:41.008Z"}, ...]
```

Setpoint yaz:

```
curl -b c.txt -X POST http://localhost:8090/runtime/v1/write \  
-H "Content-Type: application/json" -d '{"tag": "Tesis.PLC1.Setpoint", "value": 55}'
```

Tek çağrıyla tag'li cihaz oluşturun:

```
curl -b c.txt -X POST http://localhost:8090/config/v1/project/channels/Tesis/devices \
-H "Content-Type: application/json" \
-d '{ "name": "PLC2", "idString": "10.0.0.6:502", "unitId": 1, "scanRateMs": 1000,
      "tags": [ { "name": "Debi", "address": "40001", "dataType": "Float" },
        { "name": "Calisiyor", "address": "00001", "dataType": "Boolean", "readWrite": "RW" } ] }'
```

Projeyi dışa / içe aktar:

```
curl -b c.txt http://localhost:8090/config/v1/project/export -o proje.json
curl -b c.txt -X POST http://localhost:8090/config/v1/project/import?mode=merge -H "Content-Type:
application/json" --data-binary @proje.json
```

Oran sınırı ve güvenlik

- Giriş denemeleri sınırlıdır (tekrarlanan başarısızlıkta 429).
- Her değiştiren çağrı kullanıcı, IP ve içerik özetleriyle denetim günlüğüne yazılır.
- Canlı değer okumaları ucuzdur (bellek içi); cihaz tarama hızından hızlı sorgulamayın — yüksek hızlı veri için OPC UA ya da MQTT'ye abone olun.

Swagger / OpenAPI

Uygulamada ... › *REST API* etkileşimli OpenAPI belgesini açar (giriş gerekir). Her rotayı şemalarıyla listeler ve oturumunuzla denemenizi sağlar.

Data Bridge (IoT gateway)

REST sunucu, REST istemci ve MQTT istemci ajanları; payload biçimleri; Kepware uyumlu uçlar.

Data Bridge (eski adıyla IoT Gateway) veriyi MergenHub'dan **dışarı** — bulutlara, broker'lara ve üçüncü parti sistemlere — taşır ve basit HTTP istemcilerinin OPC UA olmadan tag okuyup yazmasını sağlar. Kendi Windows servisi olarak çalışır (`MergenHubDataBridge`); yavaş bir bulut ucu taramayı hiçbir zaman etkilemez.

The screenshot shows the MergenHub Data Bridge interface. The left sidebar contains a 'PROJE' (Project) section with a list of items: 'Bağlantılar' (Connections), 'Simulation Examples' (24), 'Functions' (24), 'Ekentiler' (Extensions), 'Virtual Tag'lar', 'Tag Kısayolları' (Tag Shortcuts), 'Alarmlar' (Alarms), 'Veri Logları' (Data Logs), 'Data Bridge' (selected), and 'MCP (AI)'. The main area is titled 'Tag Kısayolları' and shows a message 'Henüz kısayol yok. + ile ekle.' (No shortcuts yet. + to add). The bottom status bar indicates 'Eklentiler > Data Bridge – ajan listesi ve ajan editörü'.

Ajanlar

AJAN	YÖN	NE YAPAR
REST sunucu	gelen	Kepware uyumlu HTTP uçları (<code>/iotgateway/read</code> , <code>/iotgateway/write</code> , <code>/iotgateway/browse</code>); Kepware IoT Gateway ile aynı JSON biçimleri, mevcut entegrasyonlar değişmeden çalışır
REST istemci	giden	Seçili tag'leri değişimde ya da zamanlı bir URL'ye iter (POST/PUT); JSON/XML/CSV/anahtar-değer; özel başlık ve kimlik
MQTT istemci	giden	Seçili tag'leri broker'a yayımlar (QoS, retain, TLS, kullanıcı/parola ya da sertifika); JSON payload, tag başına konu ya da gruplu; isteğe bağlı Sparkplug tarzı şablonlar

Her ajanın kendi tag listesi, yayın hızı, *değişimde / deadband* filtreleri ve sayaçlarla son hatayı gösteren durum sayfası vardır.

REST sunucu hızlı başlangıç

1. *REST sunucu* ajanı ekleyin, tag'leri seçin, gerekiyorsa yazmaya izin verin.
2. Giriş yapın (oturum çerezi ya da API token) ve okuyun:

```
curl -b c.txt "http://localhost:8090/iotgateway/read?ids=Tesis.PLC1.Debi&ids=Tesis.PLC1.Basinc"
```

```
{ "readResults": [ { "id": "Tesis.PLC1.Debi", "s": true, "r": "", "v": 412.6, "t": 1757067641008 }, ... ] }
```

1. Yazma (*yazmaya izin açıkken*):

```
curl -b c.txt -X POST http://localhost:8090/iotgateway/write -H "Content-Type: application/json" \
-d '[{"id": "Tesis.PLC1.Setpoint", "v": 55}]'
```

MQTT istemci hızlı başlangıç

- Broker `mqtt://broker.example.com:1883` (ya da TLS ile `mqtt://`), istemci id, kimlik.
- `tesis/{channel}/{device}/{tag}` gibi konu şablonu ya da JSON dizisiyle tek konu.
- Yayın modu: *değişimde* (deadband'li) ya da *her N ms*.
- Payload örneği:

```
{ "id": "Tesis.PLC1.Debi", "v": 412.6, "q": "Good", "t": "2026-09-05T10:20:41.008Z" }
```

Dayanıklılık

- Hedef düşükken ajanlar tamponlar, dönünce boşaltır (sınırlı kuyruklu bellek içi store-and-forward).
- Ajan durumu *bağlı / yeniden deniyor* ve son hata metnini gösterir; hatalar [Olay günlüğünde](#) de vardır.
- Köprü ayrı süreç olduğundan yeniden başlatmak (*Ayarlar > Servisler*) OPC UA'yı ve taramayı kesmez.

Güvenlik notları

- REST sunucu ana API ile aynı kullanıcı, token ve kaynak ACL'sini kullanır; yalnız kullanıcının okuyabildiği tag'ler döner.
- Dış sistemler için HTTPS ve API token tercih edin; web portunu TLS'siz asla dışarı açmayın.
- MQTT: TLS ve broker'da cihaz başına kimlik; test için kapatmadıkça MergenHub broker sertifikasını doğrular.

Advanced (virtual) tag'ler

İfade motoru, ortalama/min/maks, totalizer, türev, link, Oracle birikimli tag; fonksiyon referansı.

Advanced (virtual) tag'ler mevcut tag'lerden yeni değerler hesaplar. *Eklentiler* > *Virtual tag'ler* altında yaşar, OPC UA'da `_AdvancedTags` altında görünür ve diğer tag'ler gibi loglanır, alarmlanır, ekranda gösterilir.

Eklenitiler > Virtual tag'ler – liste ve ifade editörü

Türler

TÜR	AMAÇ	NOT
İfade (Derived)	giriş tag'leri üzerinde herhangi bir formül	değişimde ya da sabit periyotta hesaplanır; matematik, mantık, karşılaştırma, dizge ve zaman fonksiyonları
Ortalama	kayan ya da pencere ortalaması	run-kapılı: yalnız <i>run</i> tag'i true iken biriktirir
Minimum / Maksimum	pencere içinde min/maks	sıfırlanabilir, run-kapılı
Totalizer	bir debiyi zamanla integre eder (m ³ /sa → m ³)	saat/dakika/saniye başına birim, yeniden başlatmalarda kalıcı, sıfırlama tag'i
Birikimli (Oracle tarzı)	fark tabanlı sayaçlar	taşmayı ele alır

TÜR	AMAÇ	NOT
Link	bir tag'i başka tag'e kopyalar (geri yazma)	cihazlar ya da protokoller arası eşleme
Tetikleyici	koşullarda boolean darbe üretir	ekranlar ve Data Bridge için

İfade dili

```
IF(Tesis.PLC1.Debi > 100 AND Tesis.PLC1.Pompa1_Calisiyor, Tesis.PLC1.Debi * 1.05, 0)
ROUND(AVG(Depo1.Seviye, Depo2.Seviye), 1)
CLAMP((Ham - 4) / 16 * 100, 0, 100)           // 4-20 mA → yüzde
LERP(0, 100, Ham / 65535)                     // doğrusal aradeğerleme
POLY(Ham, 0.0012, -0.3, 12.5)                 // polinom
MAP(Ham, 0, 1023, 0, 10)                      // aralık eşleme
```

Fonksiyonlar: aritmetik, ABS ROUND FLOOR CEIL MIN MAX AVG SUM , trigonometri, IF AND OR NOT , karşılaştırmalar, dizge fonksiyonları, NOW() HOUR() MINUTE() , PREV(tag) (önceki değer), AGE(tag) (son güncellemeden bu yana saniye), QUALITY(tag) , RANDOM() . Editördeki *Doğrula* söz dizimini kontrol eder ve canlı değerlendirme gösterir.

Kalite yayılımı

Girişlerden biri *Bad* ise, ifade QUALITY() ile açıkça ele almadıkça sonuç *Bad* olur. Eksik girişler tag'in durum sütununda ve olay günlüğünde bildirilir.

Kalıcılık

Totalizer ve birikimli tag'ler durumlarını her 30 sn'de ve kapanışta diske yazar; yeniden başlatma ya da elektrik kesintisi biriken hacmi sıfırlamaz.

Başarım

Virtual tag'ler çekirdek süreçte kendi programlarıyla hesaplanır; binlercesi sorun değildir. Yavaş değişen tag'lere bağlı ifadelerde *değişimde* hesaplamayı tercih edin.

Alarmlar ve bildirimler

ISA-18.2 yaşam döngüsü, koşullar, deadband, e-posta ve SMS, kullanıcı tercihleri, alarm geçmişi.

Alarm modülü **ISA-18.2** yaşam döngüsünü (aktif → onaylı → silinmiş) uygular, canlı tag'lerde koşulları değerlendirir, e-posta ve SMS ile bildirir, geçmiş tutar ve alarmları OPC UA istemcilerine yayınlar.

Alarmlar – aktif liste, tanımlar ve geçmiş sekmeleri

Alarm tanımlama

Eklentiler > Alarmlar > + Alarm:

ALAN	ANLAMI
Kaynak tag	değerlendirilecek tag (sayısal ya da boolean)
Koşul	bir sınıra göre > >= < <= == != , aralık içinde/dışında, boolean true/false, değişim hızı, başka tag'den sapma, kötü kalite
Deadband	sınır çevresinde dolaşan değer çırpınmasın diye histerezis
Açılış / kapanış gecikmesi	koşulun oluşması / silinmesi için sürmesi gereken saniye
Şiddet / öncelik	Düşük, Orta, Yüksek, Kritik (OPC UA için 1–1000)

ALAN	ANLAMI
Alan > Kaynak	WinCC benzeri organizasyon (tesis > ünite > ekipman) filtreleme için
Mesaj	{value} {limit} {tag} {unit} yer tutuculu metin
Otomatik onay	operatör onayı olmadan silinsin
Bildirimler	hangi grup/kullanıcı e-posta ve/veya SMS alır; isteğe bağlı yükseltme gecikmesi

Operatör görünümü

Aktif sekmesi oluşan alarmları şiddete göre renkli, zaman, değer ve **Onayla** düğmesiyle (yorum isteğe bağlı) listeler. Üst çubuktaki zil uygulamanın her yerinde aktif sayısını gösterir; SCADA ekranları bağlı nesnelerde alarm rozeti ve alarm bandı widget'ı gösterir.

Bildirimler

- **E-posta** — SMTP ayarları *Ayarlar > E-posta*'da; şiddet başına şablon.
- **SMS** — takılabilir sağlayıcılar (NetGSM ve jenerik HTTP) *Ayarlar > SMS*'te.
- **Kullanıcı tercihleri** — her kullanıcı hangi şiddet ve alanları e-posta/SMS ile istediğini ve sessiz saatleri seçer.
- **İletişim merkezi** — test mesajı, şablonlar, davet ve toplu mesaj (*Ayarlar > İletişim*).

OPC UA Gateway - v3.1.19

PROJE

Bağlantılar

Simulation Examples

24

Functions

24

Ekentiler

Virtual Tag'ler

Tag Kısıyolları

Alarmlar

Veri Logları

Data Bridge

MCP (AI)

admin

Root (super-admin)

TR TR

OPC UA SUNUCU

OPC UA Sunucu

OPC UA Kullanıcıları

Güvenilen Client'lar

Sunucu Sertifikası

WEB & BAĞLANTI

Web & REST API

Data Bridge

BİLDİRİMLER

E-posta

SMS

İletişim Merkezi

GÖRÜNÜM

Marka

SİSTEM

Lisans

Denetim

Saklama

Servisler

ERİŞİM

Kullanıcılar

Alarm bildirimlerini e-posta ile göndermek için SMTP ayarları. [Detaylı yardım](#)

E-posta

Alarm bildirimleri için SMTP gönderici ayarları.

E-posta gönderimi etkin

SMTP Sunucu

smtp.gmail.com

Port

587

SSL/TLS kullan

Kullanıcı adı

Parola

Gönderen adres

alarm@firma.com

Gönderen ad

MergenHub

TEST

test alıcı e-posta

Test Gönder

Not: port/bind/güvenlik değişiklikleri yeniden başlatmada uygulanır.

Kaydet

Ayarlar > E-posta – SMTP sunucusu, gönderen ve test mesajı

Geçmiş

Her geçiş (oluştur, onaylandı, silindi, bildirim gönderildi) zaman damgası, değer, kullanıcı ve yorumla ayarlı saklama süresi (varsayılan 90 gün) kadar tutulur ve dışa aktarılabilir. Bkz. [Loglar](#).

OPC UA Alarms & Conditions

Ayarlar › OPC UA Sunucu'da Alarmları OPC UA'dan yayınla'yı açın. Her tanım `_Alarms` altında

`AlarmConditionType` olur; UA istemcileri olay alır, koşulları yeniler ve onaylar. UA'dan gelen onaylar UA kullanıcı adıyla denetime düşer.

İpuçları

- Haberleşme kaybı bildirimi için cihaz başına `_System._NoError` alarmı kullanın.
- Kısa GSM kopmalarında uyarı gelmesin diye *kötü kalite* koşulunu kapanış gecikmesiyle birleştirin.
- KPI alarmları için önce virtual tag oluşturup ona alarm koyun.

Historian (veri kaydı)

Log grupları, tetikleyiciler, SQLite ya da ODBC hedefler, store-and-forward, trend ve dışa aktarım.

Historian tag değerlerini veritabanına kaydeder; trend, dışa aktarım ve OPC UA HistoryRead sunar. Kendi servisi olarak çalışır (`MergenHubHistorian`) ve canlı değerleri yerel API üzerinden çekirdekten okur.

Eklentiler > Historian – log grupları ve trend görüntüleyici

Hedefler

HEDEF	NE ZAMAN
SQLite (<code>logs.db</code> , varsayılan)	tek sunucu, birkaç yüz milyon satıra kadar; sıfır kurulum
ODBC	SQL Server, MySQL/MariaDB, PostgreSQL, Oracle — kurumsal veritabanları, raporlama araçları, uzun saklama

Bağlantı *Ayarlar* > *Historian*'da (ya da log grubunda); *Gözet* düğmesi ağdaki SQL Server örneklerini bulur, veritabanlarını listeler ve tabloyu sizin için oluşturabilir.

Log grupları

Log grubu, tek bir saklama kuralına sahip tag kümesidir:

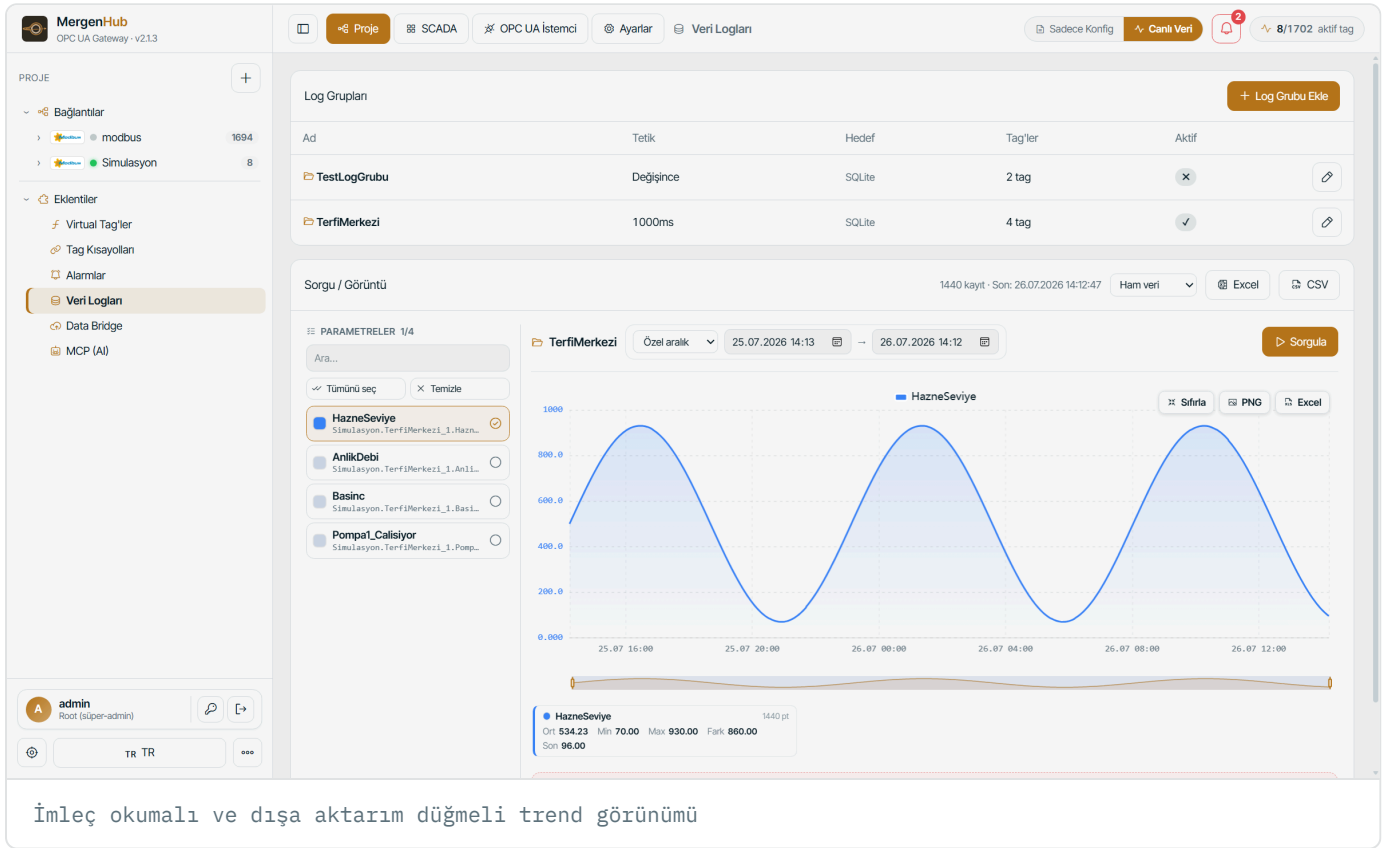
- **Tetikleyici** — periyodik (her N ms), değişimde (deadband'li), tetik tag'i kenarında ya da zaman penceresi (ör. 06:00–22:00). Gelişmiş kilitli tetikleyiciler Kepware davranışıyla birebirdir.
- **Tablo biçimi** — *dar* (tag örneği başına satır) ya da *geniş* (tag başına sütun), seçilebilir sütun eşlemesiyle.
- **Tablo modu** — tek tablo, gün/ay başına tablo ya da otomatik oluşturulan sabit tablo.
- **Toplu boyut / boşaltma aralığı** — veritabanına göre ayarlayın.

Store-and-forward

Hedef ulaşılamazsa historian örnekleri diskte tamponlar, bağlantı dönünce yazar; veritabanı bakımında ya da ağ kesintisinde hiçbir şey kaybolmaz.

Trend ve dışa aktarım

Historian > Trend: tag ve zaman aralığı seçin, yakınlaştırın, kaydırın, karşılaştırın; **CSV, Excel ya da PNG** olarak dışa aktarın. Seyreltme (kova başına ortalama/min/maks) uzun aralıkları hızlı tutar.



OPC UA HistoryRead

Log grubundaki tag'ler HistoryRead isteklerini (ham değerler ve ortalama/min/maks/aradeğerleme gibi işlenmiş toplamalar) yanıtlar; UA istemcileri veritabanına doğrudan erişmeden geçmiş çizer.

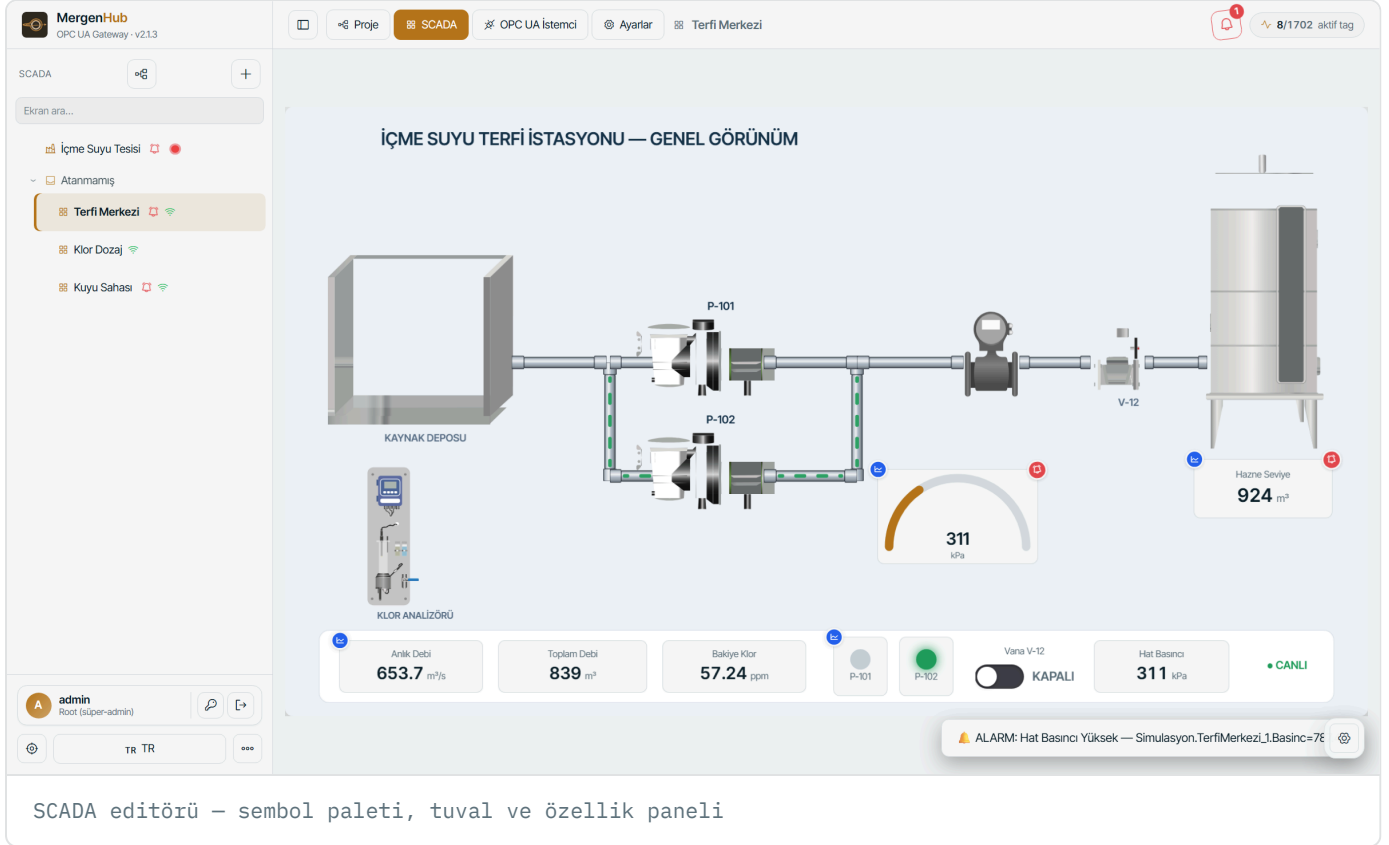
Boyutlandırma

Dar SQLite depolama örnek başına kabaca 40–60 bayt tutar. 1.000 tag her 10 sn $\approx$ günde 8,6 M örnek $\approx$ günde 0,4–0,5 GB. Yavaş değişen değerler için deadband'li değişimde kayıt kullanın; çok yıllık saklama için ODBC veritabanı seçin.

SCADA/HMI

Editör, sembol kütüphaneleri, borular ve akış animasyonu, bağlamalar, faceplate, operatör görünümü, şablonlar.

SCADA/HMI modülü tarayıcı tabanlı bir ekran editörü ve çalışma zamanıdır. Ekranlar sunucuda saklanır (`scada.json`), tarayıcıda düzenlenir ve operatörler herhangi bir cihazdan izler — istemci kurulumu yoktur.



Editör temelleri

- **Ekranlar ve bölgeler** — ekranlar bölge ağacında (tesis > alan > ünite) düzenlenir; bölgeler gezinmeyi ve erişim kontrolünü de belirler.
- **Widget'lar** — tanklar, pompalar, vanalar, motorlar, göstergeler, trendler, canlı grafikler, tablolar, düğmeler, metin, girişler, görseller, haritalar (GPS konumlu nesnelerle Web Mercator), borular ve çoklu-çizgi borular, sembol kütüphanesi nesneleri (yerleşik endüstriyel kütüphane, WinCC tarzı dinamik semboller, içe aktarılan SVG).
- **Bağlamalar** — her özellik (doluluk, renk, görünürlük, dönme, metin, konum) bir tag'e ya da ifadeye bağlanabilir.
- **Animasyonlar** — nesne başına kurallar: koşulda renk, yanıp sönme, gizle/göster, döndür, taşı, **akış** (borular boyunca ve sembol şekli içinde yönlü, hızlı animasyonlu şeritler), ölçekle.
- **Gruplar ve izolasyon** — nesneleri gruplayın, grubun içini düzenlemek için çift tıklayın (Illustrator tarzı), hizala/dağıt, geri al/yinele, ekranlar arası kopyala/yapıştır.
- **Alt ekranlar ve faceplate'ler** — tıklama eylemleri başka ekran açar, cihaza bağlı bir açılır faceplate, bir trend ya da (onaylı) değer yazar.

- **Şablonlar** — hazır su/enerji ekranları (kuyu sahası, terfi merkezi, depo, klor dozaj...) bir IP adresinden kanal, cihaz, tag ve çalışan ekran üretir.

Çalışma zamanı (operatör görünümü)

Çalıştır'a basın ya da ekranı operatör menüsünden açın. Değerler canlı güncellenir, bağlı nesnelerde alarm rozetleri görünür, girişler ve düğmeler kullanıcının yetkisine göre tag'lere yazar; görünüm tablet ve telefonda dokunmatik kaydır/yakınlaştır ile çalışır.



Sembol kütüphaneleri

- Yerleşik kütüphane: pompalar, vanalar, tanklar, motorlar, borular, elektrik tek-hat öğeleri, sensörler, kaplar, 3B görünümlü nesneler.
- **SVG içe aktar** — kendi çizimlerinizi getirin; id'ler otomatik ad-uzaylanır, birden çok örnek asla çakışmaz.
- **Nesnelerim** — herhangi bir grubu yeniden kullanılabilir nesne olarak kaydedin; kütüphane sunucu tarafında ve hesaba bağlıdır.
- WinCC `.svg`mi dinamik sembolleri bağlamalarıyla içe aktarılabilir.

Yetkiler

Bölgeler ve ekranlar kaynak ACL'sine uyar: operatörler yalnız okuyabildikleri bölgelerin ekranlarını görür, yazmalar tag'de yazma izni ister. Ekran düzenlemek SCADA düzenleme yetkisi gerektirir.

Dışa / içe aktarma

SCADA > Projeyi dışa aktar ekranları, bölgeleri ve kullanıcı kütüphanesini tek JSON dosyasına yazar; içe aktar başka sunucuda geri yükler (bağlamalar tag yollarını korur).

İpuçları

- Boru widget'larında *Akış animasyonu*'nu açıp pompa-çalışıyor tag'ine bağlayın.
- Kontrol odaları için ekran zeminini açık tutun; karanlık odalar için tema anahtarını kullanın.
- Haberleşme kaybı ekranda görünsün diye cihaz başına `_System._NoError` tag'ini bir durum lambasına bağlayın.

Kullanıcılar, gruplar ve güvenlik

Roller ve yetkiler, kaynak ACL, HTTPS, oturum politikası, denetim, sertleştirme kontrol listesi.

MergenHub'da web arayüzü, REST API, OPC UA ve Data Bridge için tek kullanıcı modeli vardır: **kullanıcılar gruplara** üyedir, gruplar ve kullanıcılar **rol/yetki** taşır, **kaynak ACL** her birinin hangi kanal, cihaz ve bölgeyi görüp yazabileceğine karar verir.

Ayarlar > Kullanıcılar - hesaplar, gruplar ve yetkiler

Roller ve yetkiler

YETENEK	TİPİK ROL
Yapılandırma okuma	projeyi açan herkes
Yapılandırma yazma (kanal, cihaz, tag, alarm, kayıt)	mühendis
Tag yazma (setpoint, komut)	operatör
SCADA düzenleme	mühendis
Kullanıcılar, ayarlar, lisans	yönetici
Root (süper yönetici)	yerleşik <code>admin</code> ; her zaman her şeye sahiptir

Yetkiler her istekte canlı değerlendirilir — bir kullanıcının haklarını değiştirmek yeniden giriş gerektirmeden hemen geçerli olur.

Kaynak ACL

Ayarlar > Kullanıcılar > Erişim kanal/cihaz/bölgeleri kullanıcı ya da gruplara *okuma* ya da *okuma+yazma* olarak atar. Root olmayanlar için **varsayılan reddir**: yeni kullanıcı bir şey verilene kadar hiçbir şey görmez. Oluşturulan nesnelere sahiplik yazılır; oluşturan erişimini korur. ACL arayüz, REST, OPC UA (adres uzayı filtrelenir), Data Bridge ve MCP'de geçerlidir.

Parolalar ve oturumlar

- Parola politikası ve ilk girişte zorunlu değişim (`admin/admin` varsayılanı).
- Giriş oran sınırı (tekrarlanan başarısızlıkta 429), başarılı ve başarısız girişlerin denetimi.
- Ayarlar > Web & REST API*'de oturum politikası: boşta zaman aşımı, kullanıcı başına tek/çoklu oturum, güvenli çerezler (HTTPS).
- Şifremi unuttum*: iletişim bilgisi tanımlı kullanıcıya e-posta ya da SMS ile 6 haneli kod; yöneticiler komut satırından sıfırlar:

```
"C:\Program Files\MergenHub\MergenHub.exe" reset-admin "<yeni-parola>" keep
```

HTTPS

Sertifikalar ve reverse proxy için [Domainde yayınlama](#). HTTPS açıkken HTTP yönlendirilir ve çerezler güvenli işaretlenir.

OPC UA güvenliği

Sign & Encrypt endpoint'ler, anonim kapalı, güvenilen istemci listesi, kullanıcı bazlı erişim — bkz. [OPC UA sunucusu](#).

API token'ları

Ayarlar > API token'ları entegrasyonlar için bearer token üretir; her token kullanıcının yetkilerini devralır ve iptal edilebilir. Token yalnız bir kez gösterilir.

Denetim

Her yazma, giriş ve yapılandırma değişikliği kullanıcı, IP ve zamanla kaydedilir (*Ayarlar > Denetim*), varsayılan 90 gün tutulur. Bkz. [Loglar](#).

Sertleştirme kontrol listesi

- ☐ `admin` parolasını değiştirin; kişilere adlı hesap açın, ortak giriş yok
- ☐ Operatörlere yalnız gereken yerde tag yazma; mühendislere kullanıcı yönetimi yok
- ☐ HTTPS açık; HTTP yönlendiriliyor; güçlü oturum politikası
- ☐ OPC UA: anonim kapalı, Sign & Encrypt, yalnız güvenilen istemciler
- ☐ Güvenlik duvarı: yalnız gerekli portlar, kısıtlı kaynak ağlar ([Güvenlik duvarı](#))
- ☐ Denetim günlüğünü düzenli gözden geçirin; yedek alın ([Yedekleme](#))

Yapay zekâ asistanı (MCP sunucusu)

MCP ile yapay zekâ asistanı bağlama, OAuth ve token, araç seti, neler yapabilir neler yapamaz.

MergenHub bir **MCP sunucusu** (Model Context Protocol) içerir; yapay zekâ asistanları — Claude, ChatGPT uyumlu ajanlar, kurum içi yardımcılar — tesisi doğal dille, yetkilendiren kullanıcının izinleriyle sorgulayıp yapılandırabilir.

MergenHub
OPC UA Gateway - v3.119

Eklientiler > MCP (AI)

Sadece Konfig **Canlı Veri** 24/24 aktif tag Deneme 24/500

MCP (Model Context Protocol): AI asistanlarının (Claude, Cursor...) sisteme GÜVENLİ bağlanması için token — yalnız okuma + SCADA ekran yapma + doküman; cihaza/tag'e yazma, config ekle/sil YOK. Token yalnız SENİN yetkin dahilindeki araçları alır.

Dış Erişim — AI sağlayıcıya verilecek adres

http://localhost:8095/mcp Kopyala Keşif testini aç

HTTPS: YOK — web tabanlı AI istemcileri yalnız **https://** adres kabul eder. Ayarlar — Web & REST API'den HTTPS'i aç (sertifika + domain).

Erişilebilirlik: bu adres YEREL görünüyor — web AI istemcileri buraya İNTERNETTEN ulaşamaz. Sabit IP ya da domain + port yönlendirme gerekir. (Masaüstü istemciler aynı ağdaysa yerel adresle çalışır.)

Genel adres (public URL): Kaydet

Boş = tarayıcının eriştiği adres kullanılır. Domain yayında MUTLAKA doldur — OAuth keşif metadata'sı (issuer) bu adresle ilan eder.

DOMAİNE ÇIKARMA ADIMLARI (BİR KEZ)

Sabit IP edin (ISS'den) YA DA dinamik DNS kullan; domain'in **A kaydını** bu makinenin genel IP'sine yönlendir.

Modern/güvenlik duvarında **443** — bu **makine** port yönlendirmesi yap (MergenHub HTTPS portu).

Sertifika: **Ücretsiz Let's Encrypt** önerilir — `win-acme` (`wacs.exe`) ile domain'e sertifika al (HTTP-01 doğrulaması için 80 portu geçici olarak bu makineye yönlendirilmeli), PFX olarak dışa aktar ve Ayarlar — REST API — Sertifika'dan yükle; sonra HTTPS'i aç. (Alternatif: self-signed üret — tarayıcı uyarı verir, claude.ai kabul etmeyecebilir.) Let's Encrypt 90 günlük; win-acme görev zamanlayıcıyla yenileme kurar — yenilenen PFX'i tekrar yüklemeyi unutma.

Buradaki Genel adres alanına `https://domain` yaz ve kaydet.

"Keşif testini aç" ile metadata'nın DİŞ ağdan da açıldığını doğrula (telefonda mobil veriyile dene).

Yeni MCP Token

Ad:

Süre:

İZİN VERİLEN ARAÇLAR — GRUBUN TAMAMINI VER YA DA "ARAÇLAR"DAN TEK TEK SEÇ

☐ **Tam Erişim — canlı tag değeri yazma HARIÇ her şey**
Tek tıkla: okuma + config/tag TANIMI + alarm + veri kaydı + üretmiş etiket + SCADA tasarımı/önizleme. **Canlı tag DEĞERİ yazma HARIÇ** (böyle bir araç yok). Her araç yine SENİN yetkinle sınırlı.

☒ **Etiketler (oku)** 3 araç ▼ araçlar

☒ **Bağlantı (kanal / cihaz)** 4 araç ▼ araçlar

☒ **Alarmlar** 3 araç ▼ araçlar

admin
Root (süper-admin)

TR TR

Eklientiler > MCP (AI) — sunucu durumu, token'lar ve bağlı istemciler

Asistan neler yapabilir

- **Okuma:** kanal/cihaz/tag listeleme, tag arama, canlı değer ve kalite, cihaz durumu ve teşhis, aktif ve geçmiş alarmlar, historian sorguları ve grafikler, metrikler ve olaylar, proje özeti.
- **Yapılandırma:** kanal, cihaz, tag, tag grubu, virtual tag (ifade doğrulamalı), alias, alarm, log grubu, Data Bridge ajanı oluştur/düzenle/sil; proje dışa/içe aktar.
- **SCADA:** ekran oluştur, gerçek sembol kütüphanesinden widget ekle, boru çiz, animasyon ver, önizleme üret.
- **Dokümantasyon:** yerleşik kılavuzu oku (`list_docs` / `read_doc`).

Asla yapamadığı: cihaza canlı değer yazmak (setpoint/komut aracı yoktur) — bu arayüzde insanlara kalır.

Uygulamanın İÇİNDEN konuşmak için ayrıca [uygulama-içi yapay zekâ asistanı](#) vardır (ChatGPT, Claude, OpenRouter, Ollama...); aynı araçlar, aynı yetki denetimi.

Bağlama

1. *Eklentiler > MCP (AI)* > sunucuyu açın. Uç: `https://<host>:<port>/mcp` (HTTP üzerinden JSON-RPC).
2. Kimlik doğrulama: destekleyen asistanlar için **PKCE ve dinamik istemci kaydıyla OAuth 2.0**, ya da aynı sayfada üretilen **bearer token**.
3. İki katmanlı yetki: token kapsamı × kullanıcının yetkileri/ACL'si. Salt-okunur token, kullanıcı yapabilse bile yapılandıramaz.
4. Her araç çağırısı kullanıcı ve token adıyla denetim günlüğüne yazılır.

Claude Desktop / Claude Code örneği (`mcp.json`):

```
{ "mcpServers": { "mergenhub": { "url": "https://scada.example.com/mcp", "headers": { "Authorization": "Bearer <token>" } } }
```

Örnek istekler

- "Son bir saatte hangi cihazlar haberleşmeyi kaybetti?"
- "10.0.0.9'da Modbus TCP cihazı oluştur; 40001–40020 holding register'larını Debi1...Debi10 adlı float tag'ler yap."
- "Tesis.PLC1.Basinc 6 bar üstünde 0,2 deadband'li yüksek alarm ekle, operatörler grubuna e-posta gitsin."
- "Depo1.Seviye son 24 saati çiz, minimumu söyle."
- "İki tank ve borularla bağlı bir pompa olan ekran kur; pompa çalışınca akış animasyonu olsun."

Güvenlik notları

- MCP sunucusunu yalnız gerekince açın; asistan başına token üretin ve aynı sayfadan iptal edin.
- Sunucuyu HTTPS arkasına alın; uç oran sınırlı ve denetimlidir.
- Asistan yalnız token'ın kullanıcısının görebildiğini görür (kaynak ACL).

Otomasyon (iş akışları)

Sürükle-bırak akış editörü: zaman/tag/alarm/webhook tetikleyicileri, koşul ve hesaplar, tag yazma, e-posta, SMS, HTTP aksiyonları; canlı koşu izleme; asistanla kurma.

Otomasyon (iş akışları)

Eklentiler › **Otomasyon** modülü, "şu olunca şunu yap" kurallarını sürükle-bırak bir akış şemasıyla kurmanızı sağlar: **tetikleyici** → **koşul / hesap** → **aksiyon**. Akışlar sunucuda canlı çalışır (saniyede bir değerlendirilir), her koşu adım adım kaydedilir ve editörde anlık izlenir. Yapay zekâ asistanı da sizin için akış kurabilir, doğrulayabilir ve çalıştırabilir.

Tetikleyiciler

TETİK	NE ZAMAN	ÖNEMLİ AYARLAR
Elle / API	Çalıştır düğmesi, API ya da asistan	— (test için ideal)
Zamanlayıcı	Sabit aralık ya da cron (<code>dk sa gün ay hafta</code>)	<code>0 8 * * 1-5</code> hafta içi 08:00 · her 300 sn
Tag değeri	Değer değişince ya da eşiği geçince	koşul (change / above / below / equals / quality), eşik, ölü bant, koşul süresi (dwell), ateşleme (bir kez / tekrar), soğuma
Alarm	Alarm oluşunca / temizlenince / onaylanınca	kaynak filtresi (<code>Kuyu*</code>), en düşük öncelik
Webhook	Dış sistem <code>POST</code> <code>/hooks/{anahtar}</code> çağırınca	anahtar (otomatik üretilir), gizli başlık <code>X-Hook-Secret</code> ; gövde <code>\$payload</code> olur

Tag tetikleyicide ilk okuma tetiklemez; ölü bant gürültüyü keser; *koşul süresi* geçici sapmaları eler; *soğuma* iki tetik arasına en az süre koyar. Kapalı çevrim kontrol (pompa aç/kapa) kurarken bu üçünü birlikte kullanın.

Mantık

- **Koşul** — ifade doğruysa *evet*, değilse *hayır* portundan devam eder.
- **VE** — bağlı tüm girişler aynı koşuda gelince devam eder; **VEYA** — ilk gelen giriş geçirir.
- **Hesapla** — ifadeyi hesaplar, sonucu `$var.ad` değişkenine yazar; sonraki koşul ve aksiyonlarda kullanılır.
- **Değişken ata**, **Bekle** (saniye).

İFADE SÖZDİZİMİ

`{Kanal.Cihaz.Tag}` canlı değer · `$value` / `$prev` tetikleyen tag'in yeni/önceki değeri · `$var.ad` · `$payload.alan` (webhook) · `$hour $minute $weekday $now` · `AND OR NOT` · `< <= > >= == !=` · `abs min max round sqrt pow if(koşul,a,b)` . E-posta konusu, mesaj, URL gibi metinlerde aynı referanslar `{...}` içinde yazılır: `Seviye {Kuyu.P1.Seviye} m, saat {$now}` .

Aksiyonlar

Tag'e yaz (setpoint/komut; değer ifade olabilir) · **E-posta** · **SMS** · **HTTP isteği** (JSON, yanıt `$var.http`) · **Olay günlüğü** · **Alarm onayla** · **Başka akışı çalıştır** · **Not** (tuvale açıklama).

Yetki ve güvenlik

- Akış okumak için *Yapılandırma okuma*, kaydetmek için *Yapılandırma yazma* yetkisi gerekir.
- Tag'e yazan** bir akışı yalnız *Tag yazma + servis kontrol* yetkisi olan kullanıcı kaydedebilir; elle çalıştırma da bu yetkiyi ister.
- Varsayılan olarak akış çalışırken gelen ikinci tetik **atlanır** (eşzamanlı koşu kapalı); zaman aşımı 60 sn (akış ayarlarından değiştirilir).
- Her oluşturma/düzenleme/silme/aç-kapa denetim günlüğüne yazılır; asistanın kurduğu akışlar "AI aracı" satırıyla görünür.
- MCP/asistan araçları hiçbir zaman doğrudan canlı tag yazmaz; yalnız akış tanımlar — yazma, koşu anında motor tarafından yapılır.

Editör

Sol palet (arama; sürükley ya da çift tık) · tuval (boş alanı sürükley = kaydır, tekerlek = yakınlaştırmak, port noktasından port'a sürükley = bağla, Delete = sil, Ctrl+D = çoğalt, Ctrl+S = kaydet) · sağ özellik paneli (tag alanında Tag Tarayıcı, ifadeye `{tag}` ekleme, webhook adresini kopyalama, canlı değer) · alt koşu paneli (koşular, adımlar, hata metni). Çalışan koşuda düğümler anlık renklenir (mavi: çalışıyor, yeşil: tamam, kırmızı: hata, sarı: VE bekliyor). Liste sayfasında klasörler, etkin anahtarı, tetik rozetleri, son koşu durumu ve JSON içe/dışa aktarma bulunur.

Asistanla kurma

"Kuyu P1 seviyesi 8 m altına inince 30 sn dolunca operatöre e-posta at, gece ise günlüğe yaz" gibi bir cümle yeterlidir. Asistan tag yollarını doğrular, akışı **kapalı** oluşturur, doğrulama sonucunu gösterir, elle tetikle test eder ve editörü açar; etkinleştirmeyi size sorar.

API

GET/POST `/config/v1/workflows`, GET/PUT/DELETE `/config/v1/workflows/{id}`, POST `.../validate`, PATCH `.../{id}/enabled`, POST `/runtime/v1/workflows/{id}/run`, GET `/runtime/v1/workflows/runs`, GET `/runtime/v1/workflows/events` (SSE), POST `/hooks/{anahtar}` (anonim webhook).

Formül editörü ve canlı izleme

Koşul/hesap alanlarında tıkla-ekle sembol çipleri ($\leq$, $\geq$, $\neq$, VE/VEYA/DEĞİL, $\times$, $\div$, tag seçici, `$value / $prev`, akıştaki değişkenler, `$payload`, `$hour / $minute / $weekday / $now`, fonksiyonlar) vardır; yazdıkça söz dizimi, tag'in var olup olmadığı, anlık değeri ve deneme sonucu görünür. Bir koşu tamamlanamazsa olay günlüğüne hata düşer, hangi sayfada olursanız olun uyarı çıkar, Otomasyon menüsünde rozet birikir; koşu panelinde takıldığı adım ve hata metni, düğümün üstünde kırmızı not görünür. E-posta/SMS aksiyonu kullanan bir akışı kaydederken ilgili servis kapalıysa uyarı alırsınız. Editördeki Etkin anahtarı kaydetmeden anında uygulanır.

Uygulama-içi yapay zekâ asistanı

Sağlayıcı seçimi (bulut ve yerel LLM), yetkiler ve izin akışı, ağ teşhisi, SCADA ekran tasarımı, ekler, beceriler, bağlam yönetimi.

MergenHub'ın içinde, **oturumdaki kullanıcının yetkileriyle** çalışan bir yapay zekâ asistanı vardır. Üst çubuktaki ✨ düğmesi ya da Eklentiler > **Yapay Zekâ Asistanı** ile sağ tarafta bir panel (dock) olarak açılır; istenirse ayrı pencereye çıkarılır (masaüstü uygulamasında yerel ikinci pencere). Dışarıdan bağlanan asistanlar için ayrıca [MCP sunucusu](#) vardır — ikisi aynı araç kataloğunu ve aynı yetki denetimini kullanır.

Sağlayıcı seçimi

Yönetici (Ayarlar yetkisi) asistan panelindeki ⚙ > **Sağlayıcılar** sekmesinden bir ya da birden çok model sağlayıcısı tanımlar:

- **Bulut:** OpenAI (ChatGPT), Anthropic (Claude), OpenRouter, Google Gemini, Mistral, DeepSeek, Groq, xAI, Together, Fireworks, Perplexity, Cerebras, Cohere, Azure OpenAI.
- **Yerel (anahtarsız):** Ollama, LM Studio, vLLM, llama.cpp server (Unsloth GGUF), LocalAI, Jan, text-generation-webui, KoboldCpp, LiteLLM/OpenWebUI vekili. "**Yerel sunucuları tara**" ayakta olanları bulur, tek tıkla profil ekler.
- **Özel:** herhangi bir OpenAI-uyumlu ya da Anthropic-uyumlu uç.

Her profilde Base URL, API anahtarı (yalnız sunucuda saklanır, maskeli döner), varsayılan model ("Modelleri getir" ile listeden), bağlam penceresi (bilinmiyorsa elle) ve etkin/varsayılan işareti vardır. Sohbet başlığındaki model çipinden profil ve model **aramalı listeden** seçilir; seçim konuşmaya kaydedilir.

İlk kurulumda sağlayıcı yoksa asistan paneli yöneticiye **3 adımlı sihirbaz** gösterir (sağlayıcı → anahtar → model, test). Asistan kapalıysa menü girişi soluk görünür; tıklayınca nereden açılacağı söylenir.

Davranış ayarları

- **Asistan açık** — kapalıyken kimse sohbet edemez.
- **Asistan değişiklik yapabilir** — kapalıyken yalnız okuma araçları; açıkken yapılandırma/alarm/log/SCADA yazma araçları **kullanıcının kendi yetkisi ölçüsünde** kullanılır ve her yazma denetim günlüğüne `kullanıcı (AI asistan)` olarak düşer. Cihaza **canlı değer yazma aracı hiçbir durumda yoktur**.
- **Düşünme / muhakeme** (kapalı/düşük/orta/yüksek) — OpenAI o-serisi ve GPT-5, Claude (extended thinking), Gemini 2.5, OpenRouter, Groq, DeepSeek-R1 destekler; düşünce akışı katlanır blok olarak görünür.
- Sıcaklık, yanıt token üst sınırı, tur başına araç adımı, **yönetici talimatları** (tesise özel kurallar).

Asistan neler yapar

Bir SCADA/otomasyon uzmanı gibi çalışır: önce okur, sonra hedefli yazar, sonra doğrular.

- **Envanter ve teşhis:** kanal/cihaz/tag, canlı değer ve kalite, cihaz durumu, alarmlar, historian, olaylar, denetim günlüğü, kullanıcılar ve yetkiler, ayarlar — yetkiniz dahilinde hepsi.

- **Ağ testleri:** `ping` , TCP port testi (Modbus 502, S7 102, OPC UA 4840...), DNS çözümleme, yerel dinlenen portlar. "Cihaz haberleşmiyor" sorusuna tahminle değil ölçümle yanıt verir.
- **Yapılandırma:** kanal/cihaz/tag, virtual tag (ifade doğrulamalı), alias, alarm (ISA-18.2), log grubu, Data Bridge ajanı oluştur/düzenle/sil.
- **SCADA ekranı:** gerçek sembol kütüphanesi, boru hatları ve akış animasyonu, tank/pompa sahneleri, bağlamalar, önizleme. Yerleşik **"SCADA ekran tasarımı"** becerisi yerleşim ızgarası, ISA-101 renk dili, gezinme ve doğrulama kurallarını verir.
- **Sistem:** ayarları kısmi güncelleme (OPC UA portu, HTTPS...), HTTPS sertifikası (self-signed ya da sohbete eklenen PFX), Windows güvenlik duvarı portları, servis yeniden başlatma. Örnek: "alan adım X, 443'te HTTPS yayınla" → DNS → port boş mu → sertifika → ayar → güvenlik duvarı → restart → doğrulama.
- **Ekler:** görsel (PNG/JPG...), PDF, CSV/JSON/TXT, SVG ve PFX sohbete eklenir; görseller vision-destekli modele gider, `add_widget type=image` ile ekrana konabilir, CSV/JSON içeriği bağlam olur.

Yanıtlardaki `Kanal.Cihaz.Tag` referansları tıklanabilir — ana uygulamada o yere gider. Araç çağrıları argüman/sonuç/önizlemesiyle kart olarak görünür; "Düşünüyor / Ekran oluşturuyor / Portları test ediyor" durum satırı işin nerede olduğunu gösterir.

Yetki ve izin akışı

Güvenlik katmanları: (1) **"Yapay zekâ asistanını kullanma" yetkisi** — kullanıcı/grup bazında verilir, yoksa asistan kapalıdır; (2) her araç çağrısı kullanıcının **kendi yetkisiyle** sınırlıdır; (3) yönetici **yazma kapısı**; (4) **onay** — varsayılan olarak her değişiklik sohbette Onayla/Reddet ile bekler, asistan sormadan yapmaz; (5) **denetim günlüğü** — asistanın her araç çağrısı (okumalar dahil) ve sohbet mesajları loglanır; (6) canlı değer yazma aracı yoktur.

Token ekonomisi: modele her turda tüm araçlar değil, çekirdek gruplar + kompakt bir katalog gider; gereken gruplar konuşma sırasında yüklenir. Şablonlar: "bu cihaz tipini şablon yap" denince asistan beceri olarak kaydeder, sonra tek cümleyle o tipte kanal/cihaz/tag/alarm/ekran üretir.

Asistan token kullanmaz: **oturumdaki kullanıcı** kimdir, o yetkilerle çalışır. Bir araç yetki (ör. ConfigWrite) ya da yazma kapısı yüzünden reddedilirse asistan nedenini açıkça söyler ve sorar; yönetici kartta **"İzni aç ve yeniden dene"** ile kapıyı açabilir. Dolanma ya da uydurma yoktur.

Konuşmalar, bağlam ve beceriler

- Konuşmalar kullanıcıya özeldir; yeniden adlandırma, klasör, sabitleme, arşiv ve arama vardır.
- **Bağlam göstergesi** son turun token sayısını modelin penceresine oranlar; %75 dolunca eski turlar otomatik özetlenir (ekranda tam geçmiş kalır). Elle sıkıştırma göstergeye tıklayarak yapılır.
- **Beceriler** (`DataRoot/ai/skills/*.md`): tesise özel talimat paketleri. Yerleşik iki uzman beceri açık gelir (SCADA ekran tasarımı; alarm/veri kaydı/köprü/virtual tag mühendisliği); yönetici düzenler, kapatır, yenilerini ekler. Sistem istemine yalnız ad ve açıklama girer, içerik gerekince yüklenir.
- **Dış MCP sunucuları:** ERP, doküman deposu ya da başka bir MergenHub gibi sistemlerin araçları da asistana eklenebilir (Streamable HTTP + isteğe bağlı bearer token).

Veri ve gizlilik

Sohbetler, ekler ve beceriler MergenHub veri dizininde (`ai/`) tutulur; API anahtarları yalnız `settings.json` içinde saklanır ve arayüze maskeli döner. Sağlayıcıya yalnız o turda gereken metin/görsel gider; yerel sağlayıcı (Ollama, LM Studio...) seçilirse veri makineden çıkmaz.

Veri işleme onayı (KVKK / GDPR)

Bulut sağlayıcısı (OpenAI, Anthropic, OpenRouter, Google...) kullanılmadan önce yöneticinin **veri işleme onayı** vermesi gerekir; onay verilmeden bulut profiliyle sohbet açılmaz. Metin şunları kapsar: veri sorumlusunun siz olduğunuz ve sağlayıcının üçüncü taraf olduğu; aktarılan veriler (mesajlar, ekler, araç çıktıları: proje yapılandırması, tag değerleri, alarm/olay kayıtları, kullanıcı adları — anahtar ve parolalar maskelenir); yurt dışına aktarım (KVKK m.9) için hukuki dayanağın veri sorumlusunda olduğu; saklama; kullanıcıların aydınlatılması (KVKK m.10); güvenlik; onayın kim tarafından ve ne zaman verildiğinin kaydı ve geri alınabilirliği. Yerel sağlayıcılarda (Ollama, LM Studio, vLLM...) veri makineden çıkmaz; onay gerekmez.

Fiyat, maliyet ve model bilgisi

API anahtarı girildiği anda model listesi otomatik çekilir; her model için bağlam penceresi ve fiyat (1M token girdi/çıktı) gösterilir. Fiyatlar herkese açık OpenRouter kataloğundan alınır (24 saat önbellet), yerel modeller ücretsiz sayılır; bilinmeyen modelde dolar yerine yalnız token sayısı görünür. Her turun ve konuşmanın tahmini maliyeti sohbet altında ve konuşma listesinde görünür. Model, konuşma sırasında da değiştirilebilir; "Düşünme" çipiyle konuşma başına muhakeme düzeyi seçilir.

Onay kipleri

Değiştiren araçlar için üç kip: **Her değişiklikte sor** (varsayılan), **Konuşmada bir kez onaylanana hatırla**, **Sormadan uygula** (yalnız yönetici; kapalı ortamlar için). Onay kartında "bu konuşmada bir daha sorma" ve "her zaman izin ver" seçenekleri vardır; kalıcı izinler Ayarlar > Davranış'ta listelenir ve kaldırılabilir. Her kipte tüm araç çağrıları denetim günlüğüne yazılır.

Betikler, ekran görme ve yetkiler

Asistan, hesap/rapor/dönüştürme gibi işler için **kum havuzunda** Python veya PowerShell betiği çalıştırabilir. Bu özellik **kapalı gelir** ve yalnız **sistem yöneticisi (Root)** açar: Ayarlar > Yapay Zekâ > Davranış > *Betik çalıştırma*. Açmadan önce riskleri anlatan metin okunup onaylanır; onay, kim ve ne zaman verdiğiyle birlikte silinemez uyum defterine yazılır.

Betikler nasıl sınırlanır:

- Ayrı ve **düşük yetkili bir Windows hesabıyla** çalışır (MergenHubSandbox), MergenHub veri klasörünü okuyamaz.
- Ağ varsayılan olarak kapalıdır**. Açarsanız izin listesi verirsiniz (IP, IP bloğu ya da alan adı); listede olmayan hiçbir adrese paket çıkamaz. Bu kısıt işletim sistemi güvenlik duvarında uygulanır.
- Betik **canlı tag yazamaz**. Sahaya yazma yalnız ayrı onay kartlarıyla yapılır.
- Zaman aşımı ve çıktı sınırı vardır; çalışma klasörü koşu sonunda silinir.
- Her çalıştırmada kod size gösterilir ve onay istenir**. "Sormadan uygula" kipi ve "her zaman izin ver" listesi bu araca işlemez. Her koşu (kodun özeti, çıkış kodu, süre) denetim günlüğüne ve uyum defterine yazılır.
- Acil durdurma** düğmesi çalışan tüm betikleri anında durdurur ve yeni betiği kilitler; kilidi yalnız sistem yöneticisi açar.

Asistan gerektiğinde **ekranınızı da görebilir**: "burası bozuk", "bu eğriyi yorumla" gibi durumlarda izin ister; onaylarsanız o anki ekranın görüntüsü yalnız o konuşmaya eklenir.

Yeni yetkiler (Ayarlar > Kullanıcılar > Yetkiler):

YETKİ	ANLAMI
Yapay zekâ asistanını kullanma	Asistanla sohbet
MCP köprüsü	Dış istemciyle (Claude Desktop, IDE) bağlanma — asistandan ayrıdır
Asistan betik çalıştırabilir	Tehlikeli; rol varsayılanyla gelmez, yalnız Root tek tek verir
Asistan/MCP güvenlik yönetimi	Betik özelliği, onay kipi, kalıcı izinler, dış MCP sunucusu — yalnız Root

Asistan ayrıca **lisans ve paket durumunuzu** okuyabilir: kaç tag ekleyebilirsiniz, lisans süresiz mi abonelik mi, bakım (yeni sürüm hakkı) ne zaman bitiyor, daha yeni sürüm var mı ve yenileme nereden yapılır. "Kaç tag hakkım kaldı?" diye sorabilirsiniz.

Sorun giderme

Cihaz teşhisi ve paket izleri, sık hatalar ve çözümleri, önce nereye bakılır.

Soruların çoğunu üç yer yanıtlar: **Olay günlüğü** (... > Olay Günlüğü), cihazın **Teşhis** paneli ve

C:\ProgramData\MergenHub\logs altındaki günün **dosya logu**. Bkz. [Loglar ve teşhis verileri](#).

Hata filtreli olay günlüğü – neyin ters gittiğini görmenin en hızlı yolu

Cihaz Bad kalite gösteriyor

KONTROL	NEREDE
Cihaz erişilebilir mi?	Sunucudan <code>ping</code> , <code>Test-NetConnection <ip> -Port 502</code>
Unit ID / rack-slot / adresler doğru mu?	Cihaz ayarları; ?adres biçimleri
Cihaz ne yanıtlıyor?	Cihaz > Teşhis : istek/yanıt hex, istisna kodları (Modbus 02 = geçersiz adres, 03 = geçersiz değer)
Blok çok mu büyük?	Modbus cihazında *maksimum blok*u düşürün; bazı RTU'lar 64 register üstünde resetlenir
GSM/telsizde zaman aşımı?	İstek zaman aşımını artırın, yeniden denemeyi azaltın, talep güdümlü taramayı açın
Deneme penceresi mi doldu?	Üst çubukta <i>DENEME · süre doldu</i> — yeni pencere başlatın ya da lisans girin (Lisanslama)

OPC UA istemcisi bağlanamıyor

- Endpoint URL'si *Ayarlar* > *OPC UA Sunucu*'daki bir endpoint'le eşleşmeli (host adı ve port).
- Sertifika hataları: istemciyi *Ayarlar* > *Güvenilen istemciler*'de güvenilir yapın (ya da geçici otomatik kabul açın); istemcide sunucu sertifikasına güvenin.
- Anonim kapalıysa OPC UA kullanıcısı verin.
- Port kullanımdaysa olay günlüğü *OPC UA server failed to start* gösterir; portu değiştirin.

Web arayüzüne başka bilgisayarlardan ulaşılamıyor

Host `localhost` (`0.0.0.0` yapın), güvenlik duvarı kuralı eksik ([Güvenlik duvarı](#)) ya da reverse proxy yanlış ([Yayınlama](#)).

Servis başlamıyor

`sc query MergenHub` → *STOPPED*. `logs\mergenhub-YYYYAAGG.log` dosyasında açılıştaki istisnayı bulun (genelde port çakışması ya da bozuk `settings.json`); Windows Olay Görüntüleyicisi > Application > *MergenHub*. Program dosyalarını onarmak için kurucuyu mevcut kurulumun üzerine çalıştırın; veri korunur.

Yükseltmeden sonra OPC UA istemcileri sertifika geçersiz diyor

Sunucu sertifikası yeniden üretilmiş. İstemcide yeniden güvenin ya da yedekten `pki\own` klasörünü geri yükleyip yeniden başlatın.

Alarmlar e-posta / SMS göndermiyor

Ayarlar > *E-posta* > *Test gönder*; SMTP port/TLS ve kullanıcının bildirim tercihlerini kontrol edin; SMS sağlayıcı kimliği *Ayarlar* > *SMS*'te. Hatalar sağlayıcı yanıtıyla olay günlüğüne düşer.

Historian kaydetmiyor

Log grubu açık ve historian servisi çalışıyor olmalı (*Ayarlar* > *Servisler*); hedef veritabanı bağlantısını kontrol edin; ODBC'de sürücünün 64-bit olduğundan emin olun. *Bad* kaliteli değerler yalnız grup izin veriyorsa yazılır.

Başarım

- Yüksek CPU: seri hatlarda çok hızlı tarama ya da pahalı virtual tag'ler; değişimde hesaplamayı kullanın.
- Yüksek RAM: yüz binlerce tag ve çok OPC UA abonesiyle beklenir; 100K tag için 8 GB+ ayırın.
- Panonun kapasite bandı ve *Ayarlar* > *Servisler* CPU/RAM/disk gösterir.

Destek için bilgi toplama

1. ... > Olay Günlüğü > **Uygulama Raporu** (sürüm, ayar özeti, sağlık ve son olaylar tek dosyada)
2. `logs\mergenhub-<bugün>.log`
3. Sorunlu cihazın Cihaz > Teşhis izi
4. Müşteri portalından (*Destek*) gönderin — bakım anlaşmalı lisanslı müşteriler öncelik alır.

MergenHub Mobil (iOS ve Android)

Sunucu modu ve telefon modu, sunucuyu hazırlama, canlı değer, yazma, alarm onayı, SCADA ekranları, anlık bildirim, ekran editörü, Cihaz ekle sihirbazı, güvenlik, sorun giderme.

MergenHub Mobil, MergenHub'in iOS ve Android uygulamasıdır. İki modda çalışır: **sunucu modunda** testteki MergenHub sunucusuna bağlanıp canlı değerleri izler, değer yazar, alarmları onaylar ve SCADA ekranlarını açar; **telefon modunda** sunucu olmadan, sürücülerini telefonun kendisinde çalıştırarak PLC ve RTU'larla doğrudan konuşur. Uygulama ek lisans gerektirmez: sunucu modunda sunucunun lisansı geçerlidir, telefon modu ücretsizdir.



Ana sayfa – sunucu modu
ve telefon modu seçimi

Kurulum

Uygulama App Store'da ve Google Play'de **MergenHub** adıyla yayımlanır; mağaza bağlantıları [MergenHub Mobil](#) tanıtım sayfasındadır. Uygulama arayüzü Türkçedir.

Uygulamanın isteyebileceği izinler:

- **Yerel ağ** (iOS) — telefon modunda PLC'lere aynı ağ üzerinden bağlanmak için.
- **Bildirimler** — sunucu modunda alarm bildirimi almak için.
- **Kamera** — yalnız QR kodu okuturken.

- **Face ID / parmak izi** — yalnız uygulama kilidini ya da yazmada doğrulamayı açarsanız.

Uygulama Envest'e kişisel veri göndermez; reklam ve izleme içermez.

İki mod

	SUNUCU MODU	TELEFON MODU
Veri nereden gelir	MergenHub sunucusu (REST API)	Telefondaki proje ve telefonda çalışan sürücüler
Ne zaman çalışır	Sunucuda 7/24; telefon yalnız penceredir	Yalnız uygulama açıkken
Kayıt, alarm, bildirim	Sunucuda sürekli; telefona anlık bildirim	Uygulama açıkken; bildirim yok
Ekranlar	Web arayüzünde tasarlanan SCADA ekranları	Telefonda tasarlanan ekranlar
Yetkiler	Sunucudaki kullanıcıyla aynı	Telefonun sahibi her şeyi yapar
Lisans	Sunucunun lisansı	Gerekmez

Mod, ana sayfadaki iki büyük düğmeyle seçilir; seçim hatırlanır. Sunucu modu mavi, telefon modu pirinç rengiyle ayırır. Bir çalışma alanının içindeyken üst başlıkta modun adı (**SUNUCU MODU** ya da **TELEFON MODU**), kullanıcı ve sekme görünür; başlığa dokunmak ana sayfaya döndürür.

Sunucu modu

SUNUCUYU HAZIRLAMA

Telefonun bağlanabilmesi için sunucu tarafında şunlar gerekir (web arayüzünde, yönetici olarak):

1. **Ağdan erişim** — *Ayarlar > Web, REST API & Mobil* altında Host `localhost` olmamalıdır. Yerel ağ için `0.0.0.0` yazın ve MergenHub servisini yeniden başlatın. Kurucunun eklediği güvenlik duvarı kuralı 8090 portunu açar. Ayrıntı: [Domainde yayınlama](#), [Windows Güvenlik Duvarı ve portlar](#).
2. **Mobil erişim** — aynı sekmedeki *Mobil uygulama (iOS / Android)* kartında **Mobil uygulamayla girişe izin ver** açık olmalıdır. Kapatılınca yeni girişler ve açık mobil oturumlar reddedilir. **Oturum süresi (gün)** bir telefonun yeniden parola sormadan ne kadar bağlı kalacağını belirler (varsayılan 90).
3. **Kullanıcı bazında erişim** — *Kullanıcı bazında mobil erişim* tablosunda kullanıcının **Mobil giriş** kutusu işaretli olmalıdır. Mobil girişi kapatılan kullanıcı uygulamadan bağlanamaz; web erişimi etkilenmez.
4. **İnternet üzerinden erişim** — sunucuyu bir alan adıyla ve **HTTPS** ile yayınlayın. HTTP yalnız güvenilir yerel ağda kullanılmalıdır; parola ve veriler HTTP'de şifresiz gider.

Not Cihaz oturumu, anlık bildirim ve metin (String) tag yazma MergenHub 3.5 ve üzerini gerektirir. Daha eski sunuculara uygulama HTTP Basic ile bağlanır; bu durumda parola telefonun güvenli deposunda tutulur, oturum listesi ve bildirim yoktur. Sunucu ekranlarının web arayüzündekiyle birebir görünmesi için 3.5.1 ve üzeri gerekir.

SUNUCU EKLEME VE GİRİŞ

1. Ana sayfada **Sunucu modu**'nu seçip **Sunucu ekle**'ye dokununuz.
2. **Adres** alanına IP ya da alan adını yazın. IP yazarsanız sunucunun 8090 portu denenir; alan adı yazarsanız önce HTTPS denenir. Şemayı (`http://` ya da `https://`) ve portu kendiniz de yazabilirsiniz.

- İsterseniz **Görünen ad** verin (ör. "Kayseri Terfi"); boş bırakılırsa sunucunun firma adı ya da adresi gösterilir.
- Kullanıcı adı** ve **Parola**'yı girin. Kullanıcı, web arayüzündeki hesabınızdır.
- Bağlantıyı dene** sunucunun adını, tag sayısını, durumunu ve bağlantının HTTPS mi HTTP mi olduğunu gösterir. **Kaydet** girişi yapar.

Sunucuya yalnız HTTP ile ulaşabiliyorsa uygulama **Şifresiz bağlantı (HTTP)** uyarısı gösterir ve devam etmek için onay ister. Hesap ilk girişte parola değişikliği istiyorsa sunucuyu açtığınızda yeni parola ekranı gelir.

Giriş bir kez yapılır: sunucu bu telefona özel bir **oturum belirteci** verir, parola telefonda saklanmaz. Belirteç telefonun şifreli deposunda (Keychain / Keystore) tutulur.

Ana sayfadaki sunucu listesinde her sunucunun durumu (**Çevrimiçi / Erişilemiyor**) on saniyede bir denetlenir. Adresi, görünen adı ya da kullanıcıyı değiştirmek için sunucuya basılı tutun; **Sunucuyu kaldır** sunucudaki cihaz oturumunu da kapatır. Birden çok sunucu ekleyebilir, aralarında ana sayfadan geçebilirsiniz.

YETKİLER

Uygulamada yapabilecekleriniz sunucudaki yetkilerinize ve kaynak erişim listelerine (ACL) bağlıdır; erişiminiz olmayan kanal ve cihazlar görünmez.

İŞLEM	GEREKEN YETKİ
Ağaç, canlı değerler, cihaz durumu	herhangi bir rol (Viewer)
Olay günlüğü	ConfigRead
Kanal, cihaz, grup, tag ekleme / düzenleme / silme; Cihaz ekle sihirbazı	ConfigWrite
Değer yazma, buton ve anahtarlar	RuntimeControl
Alarmları görme ve onaylama, alarm bildirimi	Alarms
Trend geçmişi (Data Logger)	DataLogging
SCADA ekranları	Scada
Lisans anahtarı girme	Settings
Yapay zekâ asistanı	Ai

GENEL

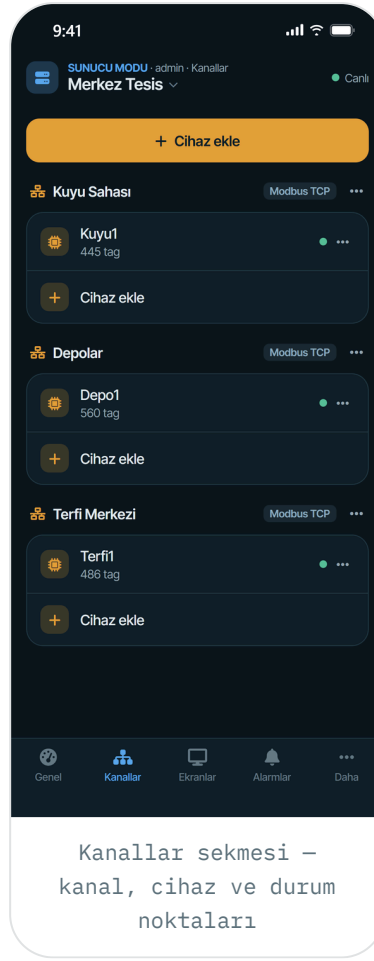


Sunucuya bağlanınca açılan özet sayfasıdır:

- **Kutular** — tag sayısı, bağlı / toplam cihaz, hatalı cihaz sayısı, aktif ve onaysız alarm sayısı.
- **Aktif alarmlar** — en yeni üç alarm; **Tümü** Alarmlar sekmesine geçer.
- **Cihazlar** — her cihazın durumu: **Bağlı**, **Hatalı**, **Bağlı değil** ya da **Kapalı**; son hata satırın altında yazar. **Ayrıntı** okuma/yazma sayaçlarını gösteren *Cihaz durumu* sayfasını açar.
- **İzleme listesi** — sık baktığınız tag'ler. Bir tag'in ayrıntısında yıldız basarak eklenir; liste sunucu başına ayrı tutulur.

Başlığın sağında **Canlı** yeşil noktası değerlerin aktığını gösterir; bağlantı koparsa **Bağlantı yok** yazar. Değerler varsayılan olarak saniyede bir yenilenir (*Ayarlar > Sunucu modu*).

KANALLAR, CİHAZLAR VE TAG'LER



Kanallar sekmesi web arayüzündeki proje ağacıdır: kanallar, altında cihazlar, her cihazda tag sayısı ve durum noktası. Cihaza dokunmak cihaz sayfasını açar.

ConfigWrite yetkiniz varsa:

- Kanal adına dokununuz: **Cihaz ekle**, **Kanal ayarları**, **Kanalı kopyala**, **Kanalı sil**.
- Cihazın ... düğmesine dokununuz ya da cihaza basılı tutun: **Tag'leri aç**, **Tag ekle**, **Cihaz ayarları**, **Cihazı kopyala**, **Cihazı sil**.
- Sayfanın üstündeki **Cihaz ekle** düğmesi [Cihaz ekle sihirbazını](#) açar.

Kanal ya da cihaz ayarlarını kaydetmek o kanalı yeniden başlatır; iletişim birkaç saniye kesilebilir. Ayar alanları ve anahtarları web arayüzündekiyle aynıdır; bkz. [Kanallar, cihazlar ve tag'ler](#).

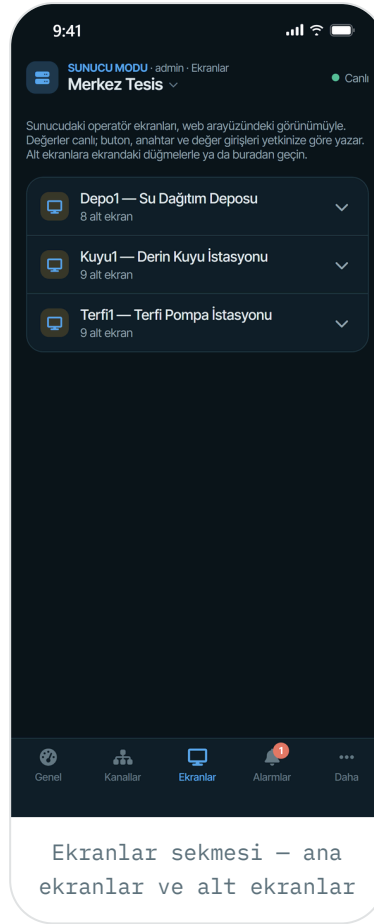


Cihaz sayfası cihazın tag'lerini ve gruplarını canlı değerleriyle listeler; üstte okuma, hata ve yazma sayaçları ile cihaz içi arama kutusu vardır. Yetkiniz varsa buradan tag ve grup eklenir, cihaz ayarları açılır.

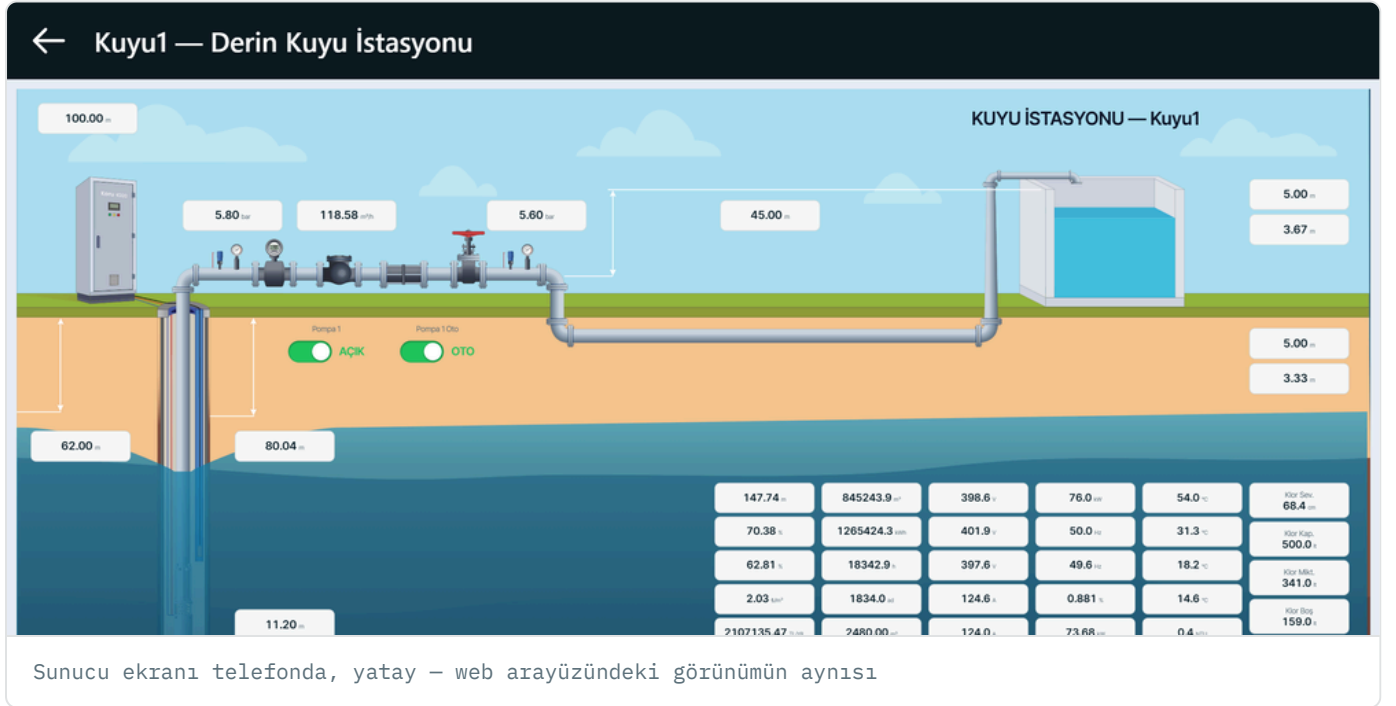
Tag ayrıntısı bir tag'e dokununca açılır:

- Değer, kalite (**Good**, **Uncertain**, **Bad**) ve zaman damgası.
- **Değer yaz** — Boolean tag'lerde **AÇIK** / **KAPALI** düğmeleri, diğerlerinde değer kutusu ve **Yaz**. Yalnız yazılabilir tag'lerde ve RuntimeControl yetkisiyle görünür.
- **Trend** — 5 dk, 15 dk, 1 sa, 6 sa ya da 24 sa. Geçmiş Data Logger'dan gelir (DataLogging yetkisi); sayfa açıkken toplanan örnekler de eklenir.
- Adres, veri tipi, erişim, tarama hızı, ölçekleme ve açıklama.
- Sağ üstteki yıldız tag'i izleme listesine ekler; menüden **Düzenle**, **Kopyala**, **Sil**.

EKRANLAR



Ekranlar sekmesi sunucudaki SCADA ekranlarını web arayüzündeki düzenle listeler: ana ekranlar üstte, alt ekranlar (ayar ekranları vb.) okla açılan dalda. Sekme Scada yetkisi ister.



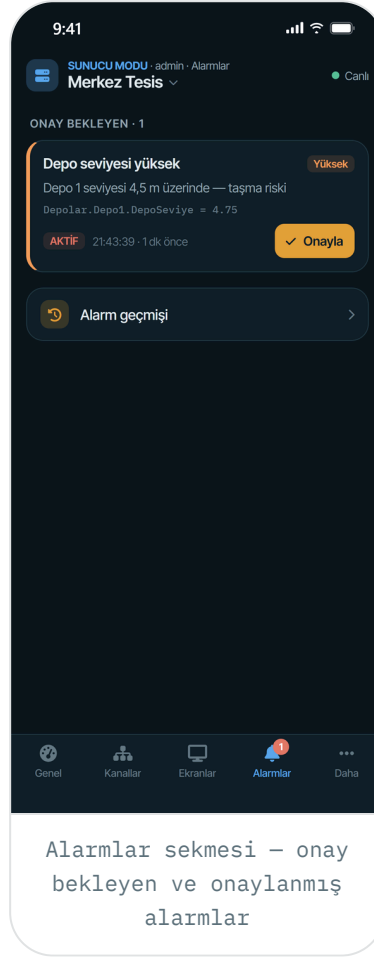
Bir ekrana dokununca ekran, sunucunun kendi görüntüleyicisiyle operatör (çalıştırma) görünümünde açılır: semboller, akış animasyonları, 3B altlıklar ve alt ekran düğmeleri web arayüzündekiyle aynıdır. Değerler canlıdır;

buton, anahtar ve değer girişleri yetkinize göre yazar.

Geniş istasyon ekranları dik tutulan telefonda ekran genişliğine sığdırılır. Sağ üstteki telefon simgesi ekranı **yatay** çevirir; iki parmakla yakınlaştırıp tek parmakla kaydırabilirsiniz. Tablette ve yatayda da görünüm operatör görünümü kalır; ekran tasarımı yalnız web arayüzündedir.

Ekranlar web arayüzünde tasarlanır ([SCADA / HMI](#)); sunucu modunda telefonda ekran düzenlenmez. MergenHub 3.5.0 ve eski sunucularda ekranlar sadeleştirilmiş çizilir: makro, SVG sembol kütüphanesi ve harita nesneleri soluk kutu olarak görünür.

ALARMLAR



Sekme simgesindeki kırmızı sayı onay bekleyen alarm sayısıdır. Alarmlar iki grupta listelenir: **Onay bekleyen** ve **Onaylanmış, aktif**. Her kartta:

- Alarm adı ve önem: **Kritik** (800 ve üstü), **Yüksek** (600–799), **Orta** (400–599), **Düşük**.
- İlet, tag yolu ve anlık değer — tag satırına dokunmak tag ayrıntısını açar.
- AKTİF** ya da **NORMALE DÖNDÜ**, oluşma zamanı ve onaylayan kullanıcı.
- Onayla** düğmesi. Onay, sunucunun denetim kaydına kullanıcınız ve **Mobil** kaynağıyla yazılır.

Alarm geçmişi alarm olaylarını (oluştur, normale döndü, onaylandı) zaman, tag değeri ve kullanıcıyla listeler. Alarm tanımları sunucuda web arayüzünden yönetilir; bkz. [Alarmlar ve bildirimler](#).

DAHA

Daha sekmesi sunucu modunda şunları içerir:

- **Tag ara** — ad, yol ya da adresle tüm sunucuda arama.
- **Cihaz durumu** — her cihazın bağlantısı, okuma/yazma ve hata sayaçları.
- **Olay günlüğü** — sunucunun olay günlüğü (ConfigRead yetkisi).
- **AI asistanı** — yalnız AI yetkiniz varsa görünür; aşağıya bakın.
- **Cihaz ekle / kütüphane** — ConfigWrite yetkiniz varsa; [Cihaz ekle sihirbazı](#).
- **Sunucu kaynakları** — işlemci, bellek, disk, MergenHub'ın belleği, tag ve taranan tag sayısı, OPC istemcileri, izlenen öğeler, iş parçacığı ve tanıtıcı sayısı ile servislerin durumu; beş saniyede bir yenilenir.
- **Cihazlarım ve bildirimler** — açık mobil oturumlarınız ve bu telefonun bildirim kaydı.
- **Sunucu lisansı** — müşteri, anahtar, tag sınırı, geçerlilik, bakım tarihi ve makine parmak izi. Settings yetkisi olan kullanıcı buradan lisans anahtarı girip etkinleştirebilir.
- **Sunucu bağlantısını düzenle** ve **Bu sunucunun projesini telefona al**.
- **Ayarlar** ve **Çalışma alanını değiştir** (ana sayfaya döner).

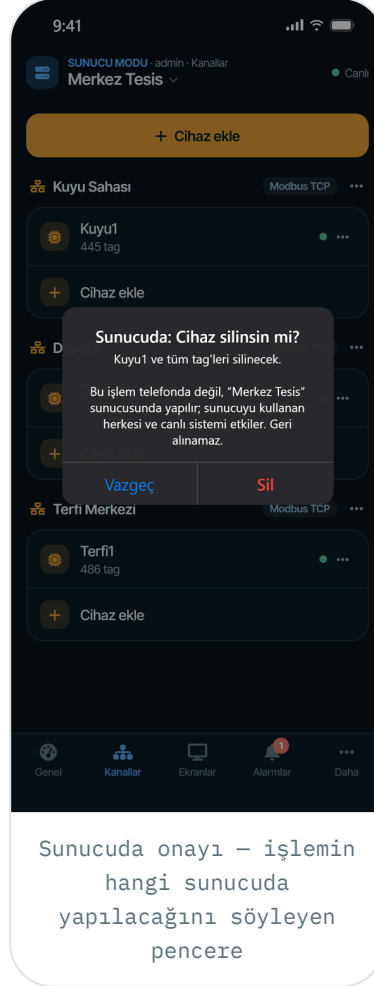


Oturumlar. *Cihazlarım ve bildirimler* sayfası kullanıcınızın açık oturumlarını (cihaz adı, platform, uygulama sürümü, son görülme, IP) listeler; kullandığınız telefon **BU CİHAZ** olarak işaretlidir. Kaybettiğiniz bir telefonun oturumunu buradan **Kapat** ile sonlandırabilirsiniz. **Bu cihazda oturumu kapat** sunucuyu listeden kaldırır; yeniden bağlanmak için parola gerekir.

AI asistanı. Sunucuda uygulama-içi yapay zekâ asistanı açık ve etkin bir sağlayıcı varsa telefonda sohbet edebilirsiniz; etkin sağlayıcı ve modeller arasından seçim yapılır. Asistan bir işlem için onay ya da izin isterse mesajın altında onay ve **Reddet** düğmeleri çıkar. Asistan ayarları web arayüzündedir; bkz. [Uygulama-içi yapay zekâ asistanı](#).

“SUNUCUDA” ONAYLARI

Sunucu modunda yaptığınız değişiklik telefonda değil, sunucuda gerçekleşir ve sunucuyu kullanan herkesi etkiler. Uygulama bunu onay penceresinde açıkça söyler: pencere başlığı **Sunucuda:** ile başlar ve metnin sonunda işlemin hangi sunucuda yapılacağı yazar.



Bu pencere şu işlemlerde çıkar:

- Kanal, cihaz, grup ya da tag **silme** — metne "Geri alınamaz." eklenir.
- Kanal, cihaz ya da tag **oluşturma** ve **değişiklikleri kaydetme**; grup oluşturma ve yeniden adlandırma. Kanal ve cihaz kaydında kanalın yeniden başlayacağı da yazar.
- Kanal, cihaz ya da tag **kopyalama** — kopya sunucuda hemen taranmaya başlar.
- Telefon modunda *Proje > Sunucuya gönder* — hangi sunucuda kaç cihaz ve tag oluşacağı yazar.
- Cihaz ekle sihirbazında **kurulum** — kurulacak cihazlar ve sunucuda oluşacaklar (kanal, cihaz, tag'ler, hesaplanan tag'ler, ekranlar) listelenir.
- **Değer yazma** — *Yazmadan önce onay iste* ayarı açıksa, tag ayrıntısından yazarken.

Telefon modunda aynı işlemler başlıkta "Sunucuda" olmadan, düz onayla sorulur.

ANLIK BİLDİRİMLER

Alarm oluşunca sunucu, Expo Push hizmeti üzerinden telefona bildirim gönderir. Telefon kilitliyken de gelir. Bildirimin gelmesi için şunların hepsi gerekir:

1. Sunucu **MergenHub 3.5** ve üzeri olmalı, telefon sunucuya cihaz oturumuyla bağlanmış olmalı.
2. *Ayarlar* > *Web, REST API & Mobil*'de **Mobil uygulamayla girişe izin ver** açık olmalı.
3. *Ayarlar* > *Mobil bildirim*'de **Mobil anlık bildirim gönder** açık olmalı.
4. Kullanıcı için **Bildirim alsın** işaretli olmalı (*Ayarlar* > *Mobil bildirim* > *Kullanıcı bazında bildirim*).
5. Kullanıcının **Alarms** yetkisi ve alarmin bağlı olduğu tag'e erişimi olmalı.
6. Alarm tanımında bildirim kanallarından **Mobil bildirim** seçili olmalı (varsayılan açıktır).
7. Sunucunun internete **HTTPS çıkışı** olmalı (`exp.host` , port 443). Telefonun sunucuya erişmesi gerekmez; bildirim Expo üzerinden gelir.
8. Telefonda MergenHub'a bildirim izni verilmiş olmalı.

Telefonun kayıt durumunu uygulamada *Daha* > *Cihazlarım ve bildirimler* gösterir ("Bu cihaz alarm bildirimlerine kayıtlı" ya da kaydın neden yapılamadığı); **Yeniden kaydet** kaydı tekrarlar. Yönetici *Ayarlar* > *Mobil bildirim* sayfasında bildirim alan cihazları görür, test bildirimi ve duyuru gönderebilir.

Telefon modunda bildirim yoktur.

Telefon modu

Telefon modunda proje telefonda tutulur ve sürücüler telefonda çalışır; sunucu ve lisans gerekmez. Sahada devreye alma, adres doğrulama ve test için tasarlanmıştır.



1. Ana sayfada **Telefon modu**'nu seçin ve **Projeye başla**'ya (proje varsa **Projeyi aç**'a) dokununuz.
2. **Cihaz ekle** ile kütüphaneden hazır bir cihaz seçin ya da kanalı elle oluşturun.
3. Başlığın sağındaki **Durdu** / **Çalışıyor** düğmesi taramayı başlatır ve durdurur. Tarama başlayınca sürücüler cihazlara bağlanır ve değerler canlı akar.

Uyarı Telefon modu yalnız uygulama açık ve ön plandayken çalışır. Telefon uykuya geçince ya da uygulama arka plana alınınca tarama durur, alarmlar izlenmez ve alarm geçmiş i uygulama kapanınca silinir. Kesintisiz izleme, kayıt ve bildirim için projeyi MergenHub sunucusunda çalıştırınız. *Ayarlar > Tarama sürerken ekranı açık tut* (varsayılan açık) ekranın kendiliğinden kapanmasını önler.

SÜRÜCÜLER

Telefonda MergenHub'ın saha sürücülerinin tamamı ve simülatör vardır. Sürücü adları, cihaz kimliği biçimi, adres sözdizimi ve ayar anahtarları sunucuyla aynıdır; proje sunucu ile telefon arasında olduğu gibi taşınır. *Daha > Desteklenen protokoller* her sürücünün adres biçimlerini örnekleriyle gösterir.

SÜRÜCÜ	NOT
Simülatör	Sinüs, rampa, kare dalga, rastgele değer, sayaç, yazılabilir register
Modbus TCP/IP Ethernet	Holding/giriş register, bobin, ayrık giriş, register biti, dize
Modbus RTU Serial / Modbus ASCII Serial	Seri sunucu ya da ağ geçidi üzerinden (Ethernet kapsülleme)
Siemens S7 TCP/IP	S7-200/300/400/1200/1500, DB, M, I, Q alanları
OMRON FINS Ethernet	Telefonda FINS/TCP; UDP ayarlı cihaz TCP ile bağlanır
OMRON NJ Ethernet	NJ/NX, EtherNet/IP CIP sembolik değişkenler
MQTT Client	MQTT 3.1.1, TCP ya da WebSocket; JSON, CSV, XML, ikili, regex çıkartımı
DNP Client Ethernet	Integrity ve Class 1/2/3 sorguları, CROB ve analog çıkış
IEC 60870-5-104 Client	Genel sorgulama, spontane olaylar, komutlar
IEC 61850 MMS Client	Okuma, yazma ve kontrol (Oper)
OPC UA Client	Yalnız güvenliksiz (None) uç nokta; anonim ya da kullanıcı adı

Telefonun cihazlarla aynı ağda (Wi-Fi) olması gerekir. Mağaza sürümünde ham TCP desteği uygulamanın içindedir: *Ayarlar > İstemci modu* altında **Ham TCP** rozeti **NATIVE** görünür ve ek bir şey gerekmez. Aynı bölümdeki **TCP köprüsü** alanı yalnız geliştirme derlemeleri içindir; mağaza uygulamasında boş bırakın.

KANAL, CİHAZ VE TAG DÜZENLEME

Kanallar sekmesi ve menüleri sunucu modundakiyle aynıdır. Telefon moduna özgü noktalar:

- Yeni kanalda sürücü seçilir; kanal oluşunca doğrudan ilk cihaz eklenir.
- Yeni cihazda **Örnek tag'ler ekle** açıksa sürücünün örnek tag'leri gelir.
- Tag formu: ad, adres, veri tipi, **Yazılabilir**, tarama hızı, **Açıklama / birim** (6 karakter ve altıysa birim olarak gösterilir, ör. `bar`, `m3/h`) ve **Ölçekleme** (ham ve ölçekli alt/üst, sınırdaki kırpm, işaret çevirme). **Kaydedince yenisini ekle** art arda tag girmeyi hızlandırır.

- Cihaz menüsünde **Şablon olarak kaydet / paylaş** vardır; bkz. [Şablonlar ve paylaşma](#).

Ölçeklenmiş tag'e yazarken ölçekleme ters uygulanır.

ALARMLAR VE HESAPLANAN TAG'LER

Alarm tanımları (*Daha › Alarm tanımları* ya da Alarmlar sekmesinin altı) telefondaki alarmları tanımlar:

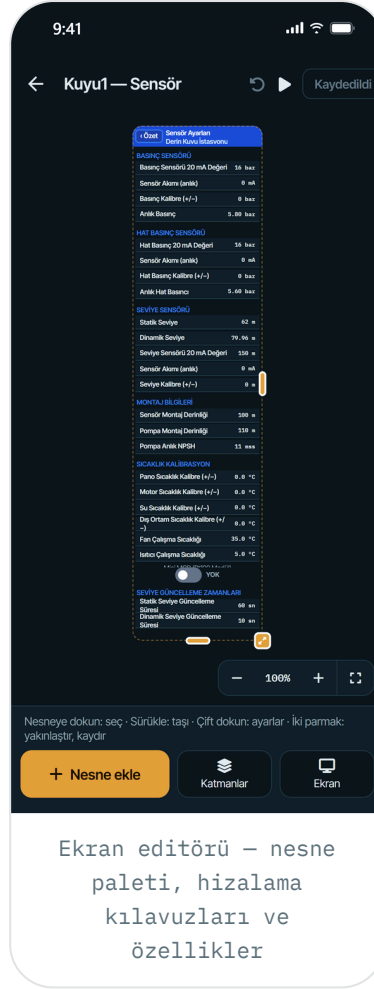
- **Koşul** — Çok yüksek, Yüksek, Düşük, Çok düşük, Sapma +, Sapma –, Artış hızı, Düşüş hızı.
- **Sınır** — sapmada ayrıca **Ayar noktası**; hız alarmlarında sınır birim/saniyedir.
- **Ölü bant** — Yok, Mutlak ya da Yüzde (yüzdede aralık alt/üst), **Açılma** ve **Kapanma gecikmesi** (sn).
- **Önem** (Düşük, Orta, Yüksek, Kritik), **İleti**, **Alan**, **Etkin**.

Gelişmiş tag'ler (*Daha › Gelişmiş tag'ler*) başka tag'lerden hesaplanan tag'lerdir; yolları `_AdvancedTags.<ad>` biçimindedir:

- **İfade** — `{Kanal.Cihaz.Tag}` referansları, aritmetik, karşılaştırma ve `ROUND`, `MIN`, `MAX`, `IF`, `POLY` gibi fonksiyonlar. Fonksiyon listesi sunucuyla aynıdır; bkz. [Advanced \(virtual\) tag'ler](#) ve [Cihaz şablonu oluşturma](#).
- **Totalizer** — değer × saniye toplamı (m³/h girişte sonucu 3600'e bölün).
- **Ortalama, Maks, Min** — kaynak tag'in son **Örnek sayısı** kadar değeri üzerinden.

EKRANLAR VE EKRAN EDITÖRÜ

Telefon modunda **Ekranlar** sekmesi bu projenin operatör ekranlarını listeler. Ekran dokunmak çalıştırır; **Düzenle** editörü açar; menüde **Çalıştır**, **Düzenle**, **Kopyala** ve **Sil** vardır. **Ekran ekle** yeni ekranı taslak olarak açar; ekran ancak **Kaydet** ile projeye eklenir.



Ekran editörü – nesne paleti, hizalama kılavuzları ve özellikler

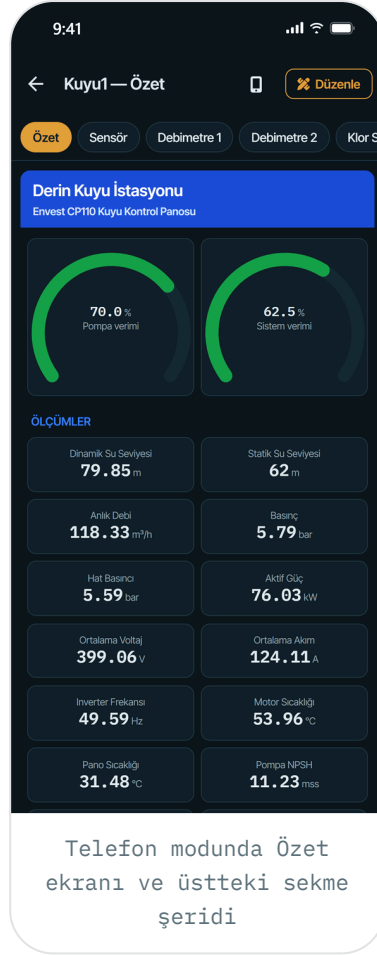
Editörde:

- **Nesne ekle** paleti: değer göstergesi, lamba, buton, anahtar, değer girişi, ayar satırı, kadran, depo ve seviye çubuğu, pompa, vana, motor, fan, boru, yazı, panel, resim ve 260'tan fazla endüstriyel sembol.
- Nesneye dokunun ve sürükleyin; hizalama kılavuzları çıkar. Köşe ve kenar tutamakları enine boyuna boyutlandırır, üstteki yuvarlak tutamak döndürür. Boş alanda tek parmak kaydırır, iki parmak yakınlaştıır.
- Seçili nesnede: tag bağlama, etiket, birim, ondalık, renkler; **Durum** (nesnenin ne zaman "açık" sayılacağı), **Kumanda** (açık/kapalı değerleri), **Seviye (dolum)** için en az ve en çok; **Koşullu kurallar** (Renk değiştir, Yanıp sön, Gizle; koşul $>$, $\geq$, $=$, $\neq$, $<$, $\leq$ ya da arasında); **Dokununca** başka ekrana geçme.
- **Katmanlar**, **Öne / Arkaya**, **Çoğalt**, **Sil** ve geri al.
- **Ekran ayarları**: ad, boyut (Telefon dikey 400×700, Telefon uzun, Telefon yatay, Tablet, Geniş ekran) ve zemin rengi.

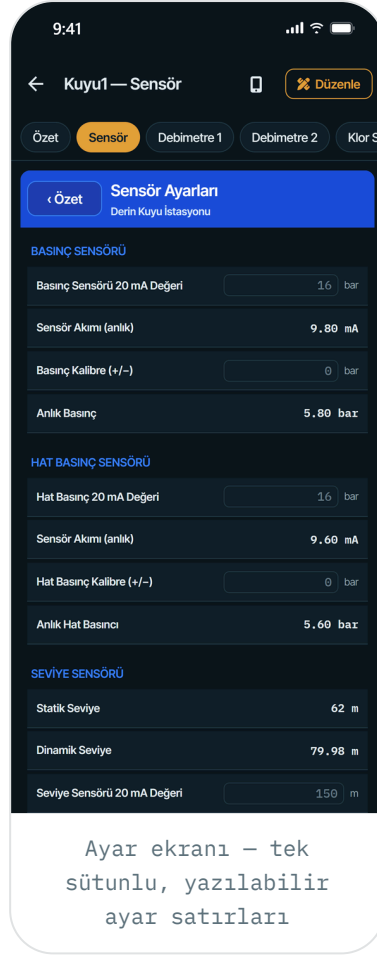
Ekranlar sunucudaki SCADA ekranlarıyla aynı yapıda saklanır; proje paket olarak sunucuya taşındığında ekranlar da gider.

KÜTÜPHANE CİHAZLARINDA ÖZET VE SEKMELER

Cihaz kütüphanesinden telefon moduna kurulan bir cihaz, telefon için hazırlanmış ekran takımıyla gelir: dikey bir **Özet** ekranı (ölçümler, kumanda ve ayar düğmeleri) ve tek sütunlu ayar ekranları.



Özet açıldığında üstte bir sekme şeridi görünür: **Özet** ve cihazın ayar ekranları (ör. Sensör, PID, Haberleşme). Sekmeye ya da Özet'teki ayar düğmesine dokunmak o ekrana geçer.



Ayar satırlarında değere dokunup yeni değer yazılır; yazma, ayarlarınıza göre onay ve kimlik doğrulaması ister.

PROJEYİ TELEFON İLE SUNUCU ARASINDA TAŞIMA

Telefondan sunucuya:

- *Daha > Proje > Sunucuya gönder* — kayıtlı bir sunucu ve kanal seçin; kanal, cihazları ve tag'leriyle sunucuda yeni kanal olarak oluşturulur. Ekranlar, alarmlar ve gelişmiş tag'ler bu yolla gitmez; sunucuda aynı adlı kanal olmamalıdır.
- *Proje > Projeyi dışa aktar* ile MergenHub paketi (`.mhpkg.json`) oluşturup web arayüzünde *Cihaz Kütüphanesi > Dosyadan içe aktar* ile kurun. Bu yol ekranları, alarmları ve hesaplanan tag'leri de taşır.
- *Proje > Kepware/MergenHub JSON olarak paylaş* — web arayüzünde *Proje > İçe aktar* ile yüklenen klasik proje JSON'u (yalnız kanal, cihaz, tag).

Sunucudan telefona: telefon modunda *Daha > Sunucudan proje al* ya da sunucu modunda *Daha > Bu sunucunun projesini telefona al*. Alınacak kanalları seçin; **Aynı adlı kanalın yerine geç** kapalıysa kanal yeni adla eklenir. Kanal, cihaz ve tag'ler gelir.

Cihaz ekle sihirbazı

Sihirbaz iki modda da aynı akışla çalışır. Sunucu modunda cihaz **sunucuya** kurulur (ConfigWrite yetkisi gerekir); telefon modunda **telefondaki projeye**. Açmak için: Kanallar sekmesindeki **Cihaz ekle** düğmesi, boş projede **Cihaz ekle** kartı ya da *Daha > Cihaz ekle / kütüphane*.

KAYNAK SEÇİMİ

- **Cihaz kütüphanesi** — Envest'in hazırladığı, test edilmiş cihaz paketleri: marka, model ve yazılım sürümüyle; tag listesi, hesaplanan tag'ler ve ekranlarıyla. Birden çok cihaz seçilebilir.
- **Telefondaki şablonlarım** — kaydettiğiniz ya da size gönderilen şablonlar.
- **Dosya, pano, QR ya da adres** — MergenHub paketi: dosyadan, panodan yapıştırarak (ör. bir yapay zekânın verdiği metin), QR kodla ya da `https://.../cihaz.mhpkg.json` adresinden.
- **Elle oluştur** — sürücüyü seçip kanal ve cihazı kendiniz tanımlayın; tag'leri sonra ekleyin.

KÜTÜPHANEDEN SEÇME



Kütüphane listesi siteden indirilir. Arama kutusu marka, model, yazılım sürümü ve protokolde arar; kategori düğmeleri listeyi daraltır. Her kartta protokol, tag sayısı ve kurulacak ekran sayısı yazar. Kartın sağındaki kutu cihazı seçime ekler; **Seçilenleri kur** seçilenlerin hepsini birlikte kurar.

İnternet yoksa son indirilen liste gösterilir ("Çevrimdışı"); kurmak için paket indirilmesi gerektiğinden internet gerekir. Sunucu modunda paketi telefon indirir ve sunucuya kurar; sunucunun internete çıkması gerekmez.



Karta dokunmak ayrıntıyı açar: fotoğraf, kategori, **Ne yapar?**, **Nasıl çalışır?**, **Paketin içeriği**, **Kurulumda sorulanlar** ve ürün sayfası. **Bu cihazı kur** yalnız bu cihazı, **Seçime ekle** birden çok kurulum için seçer.

Aradığınız cihaz yoksa listenin altındaki **Ekleme isteği gönderin** bağlantısı sitedeki müşteri panelini açar; bkz. [Cihaz şablonu oluşturma](#).

KURULUM BİLGİLERİ

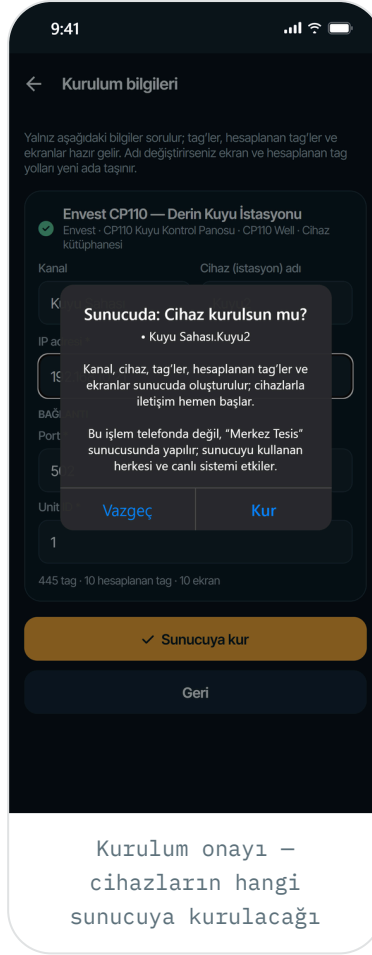
Her cihaz için yalnız paketin sorduğu bilgiler istenir; tag'ler, hesaplanan tag'ler ve ekranlar hazır gelir:

- **Kanal ve Cihaz (istasyon) adı** — paketteki adlar önerilir; ad doluysa boştaki ilk numara verilir (Kuyu1 doluysa Kuyu2). Adı değiştirdiğinizde ekranlardaki tag yolları, alarmlar, hesaplanan tag'lerin adları ve ifadeleri ile ekran adlarındaki istasyon adı yeni ada taşınır.
- **Paket parametreleri** — IP adresi, port, unit ID, FINS düğümleri ya da istasyon parametreleri gibi; ★ işaretli zorunludur.

Kartın altında kurulacakların özeti (tag, hesaplanan tag, ekran ve alarm sayısı) ya da düzeltilmesi gereken sorunlar yazar. Kurulum ancak hiç sorun kalmayınca açılır. Denetlenenler:

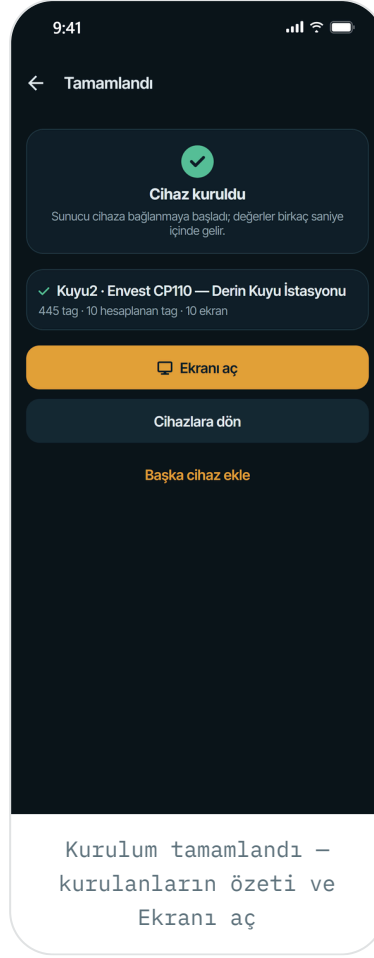
- Zorunlu parametre boş ya da yer tutucusu doldurulmamış ({{ip}} gibi).
- Kanal adı sunucuda (ya da projede) başka bir sürücüyle kullanılıyor — farklı kanal adı verin. Aynı sürücüyle kullanılıyorsa cihaz o kanala eklenir.
- Kanalda aynı adla cihaz var ya da listede aynı adla iki cihaz var.
- Hesaplanan tag adı zaten var. Hesaplanan tag adları sunucu (ya da proje) genelinde tekildir; kütüphane paketlerinde bu adlar cihaz adıyla başladığı için farklı bir cihaz adı vermek çakışmayı çözer.
- Paketin adresleri ve sürücü ayarları sürücüyle doğrulanır.

Sunucuya kur (telefon modunda **Projeye kur**) kurulumu başlatır; sunucu modunda önce **Sunucuda:** onayı gelir ve kurulacak cihazları listeler.



Kurulum onayı –
cihazların hangi
sunucuya kurulacağı

KURULUM VE SONUÇ

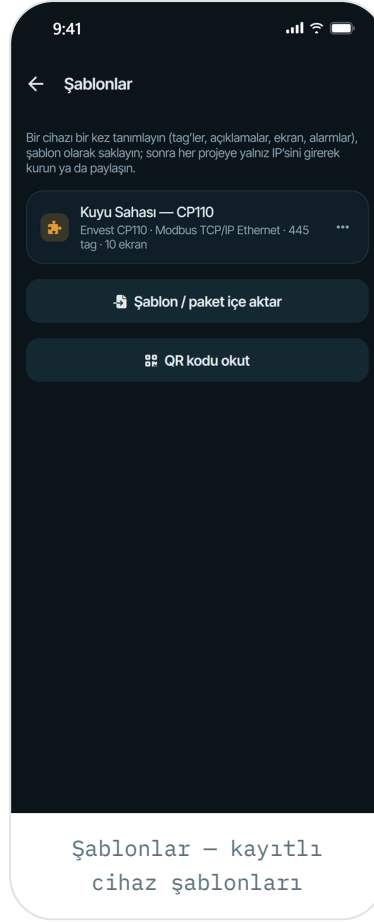


Sunucu modunda kurulum sırası: kanal (yoksa oluşturulur), cihaz ve tüm tag'leri, hesaplanan tag'ler, çalışma zamanı yenileme, alarmlar, ekranlar. Yüzlerce tag'li cihazlarda birkaç saniye sürer; ekrandan ayrılmayın. Sonuç kartı her cihaz için kurulan tag, hesaplanan tag, ekran ve alarm sayısını ve varsa kurulamayan parçaları gösterir. **Ekranı aç** kurulan ana ekranı açar.

Paketlerde iki ekran takımı bulunabilir. **Sunucuya** sunucu takımı kurulur: tek bir ana ekran ve onun altında ayar ekranları. **Telefona** telefon takımı kurulur: Özet ve sekmelerdeki ayar ekranları. Ekran kimlikleri kurulumda yenilenir; aynı paketi ikinci kez kurmak mevcut ekranların üzerine yazmaz.

Şablonlar ve paylaşma

Bir cihazı bir kez tanımlayıp (tag'ler, açıklamalar, ekranlar, alarmlar) şablon olarak saklayabilir, sonra başka projelere yalnız IP adresini girerek kurabilir ya da başkalarıyla paylaşabilirsiniz. Şablonlar ve paketler aynı biçimdedir (`.mhpkg.json`); ayrıntılar: [Cihaz şablonu oluşturma](#).



Şablon kaydetme (telefon modu): Kanallar sekmesinde cihazın ... menüsünden **Şablon olarak kaydet / paylaş**. **Ad**, **Açıklama**, **Üretici**, **Model** ve **Hazırlayan** girin; **Ekranları da al** açıksa bu cihaza bağlı ekranlar da pakete girer. Cihaz kimliğindeki ve ayarlardaki IP adresleri kendiliğinden `{{ip}}` parametresine çevrilir. Ardından:

- **Dosya olarak paylaş** — telefonun paylaşım menüsüyle (e-posta, mesajlaşma, dosyalar).
- **Panoya kopyala** — metin olarak; bir yapay zekâya vermek için uygundur.
- **Şablonlarıma kaydet** — telefondaki şablon listesine.

Tüm projeyi paket yapmak için *Daha > Proje > Projeyi dışa aktar*.

Şablonlar sayfası (telefon modunda *Daha > Şablonlar ve paketler*): şablona dokunmak kurar; menüde **Projeye kur**, **Dosya olarak paylaş**, **Panoya kopyala**, **Sil** vardır. **Şablon / paket içe aktar** dosya, pano, QR ya da adresten paket alır; proje paketlerinde **Projeye ekle** (aynı adlı cihazlar güncellenir) ya da **Projeyi bununla değiştir** seçilir.

QR kodla paylaşma. Uygulama QR kodu okur ama üretmez. Paketi erişilebilir bir adrese koyun ve adres için herhangi bir QR üreticisiyle kod oluşturun. Uygulamanın tanıdığı QR içerikleri: paketin `https://` adresi, `mergenhub://paket?url=<adres>` ya da küçük paketlerde JSON metninin kendisi.



Ayarlar (ana sayfadaki dişli ya da *Daha > Ayarlar*):

- **Görünüm** — Mergen Gece, Mergen Gündüz ya da Otomatik (telefonun temasına uyar).
- **Sunucu modu > Canlı değer yenileme aralığı** — 0,5 / 1 / 2 / 5 sn. Sunucu, izlenen cihazı son istekten sonra 8 sn daha tarar; aralık bunun altında tutulur.
- **İstemci modu** — Ham TCP durumu ve geliştirme için TCP köprüsü.
- **Güvenlik > Uygulama kilidi** — açılışta ve arka planda belirli bir süre kaldıktan sonra Face ID, Touch ID, parmak izi ya da telefon şifresi ister. Yeniden kilitleme süresi: Hemen, 1, 5 ya da 15 dakika.
- **Güvenlik > Değer yazarken doğrula** — cihaza değer yazmadan ya da komut göndermeden önce kimlik doğrulaması ister.
- **Tarama sürerken ekranı açık tut** — telefon modunda taramanın durmaması için.
- **Yazmadan önce onay iste** — varsayılan açık; her yazmada değeri gösteren onay penceresi.

Güvenlik ayarlarından birini açarken uygulama bir kez kimlik doğrulaması ister; telefonda ekran kilidi tanımlı değilse ayar açılmaz.

Güvenlik ilkeleri:

- **Parola telefonda saklanmaz.** Telefon yalnız sunucunun o cihaza verdiği oturum belirtecini şifreli depoda tutar.
- **Yönetici her oturumu görür.** *Ayarlar > Web, REST API & Mobil* sayfasında aktif kullanıcılar (web, mobil, API), mobil cihaz oturumları ve kullanıcı bazında mobil erişim yönetilir; kaybolan telefonun oturumu tek tıkla kapatılır ya da kullanıcının mobil erişimi kapatılır. Web erişimi etkilenmez.
- **Denetim kaydı** her işlemin kaynağını gösterir; mobilden yapılanlar **Mobil** olarak görünür.
- **Uygulama sunucunun mobil ayarlarını değiştiremez;** bunlar yalnız web arayüzünden yönetilir.
- **Yetkiler sunucudakiyle aynıdır;** uygulama hiçbir yetkiyi genişletmez.

Sınırlar

- Telefon modu yalnız uygulama ön plandayken çalışır; bildirim göndermez, alarm geçmişini kalıcı tutmaz. Trend son birkaç yüz örnekle (en çok 1 saat) sınırlıdır.
- Telefon modunda OPC UA yalnız güvenli (None) uç noktaya bağlanır; MQTT'de TLS yalnız WebSocket (`wss://`) üzerindendir; diğer TCP sürücülerinde TLS yoktur; OMRON FINS yalnız TCP ile bağlanır.
- Sunucu modunda ekranlar telefonda düzenlenmez; tasarım web arayüzündedir.
- MergenHub 3.5 öncesi sunucularda cihaz oturumu, oturum listesi, anlık bildirim ve metin tag yazma yoktur; 3.5.0'da sunucu ekranları sadeleştirilmiş çizilir.
- Uygulama arayüzü Türkçedir.

Sorun giderme

"Sunucuya ulaşılamadı" ya da listede "Erişilemiyor". Telefon sunucuyla aynı ağda mı? Sunucuda Host `localhost` ise yalnız o bilgisayardan erişilir; `0.0.0.0` yapıp servisi yeniden başlatın. Güvenlik duvarında 8090 portunun açık olduğunu denetleyin. Telefonun tarayıcısında `http://<sunucu-ip>:8090` açılıyorsa uygulama da bağlanır. **Bağlantıyı dene** denediği adresleri ve her birinin hatasını listeler. İnternette bağlanıyorsanız alan adı ve HTTPS kullanın.

"Bu sunucuda mobil erişim kapalı." Yönetici *Ayarlar > Web, REST API & Mobil* de **Mobil uygulamayla girişe izin ver**'i açmalıdır. Hata kullanıcıya özgüyse kullanıcının **Mobil giriş** kutusu kapalıdır.

Bağlıyken birden oturum düştü, parola istiyor. Oturum yönetici tarafından kapatılmış, kullanıcının mobil erişimi kapatılmış ya da oturum süresi dolmuş olabilir. Sunucuya basılı tutup **Parola**'yı yeniden girin.

Bildirim gelmiyor. [Anlık bildirimler](#) bölümündeki listeyi sırayla denetleyin. En sık nedenler: telefonda bildirim izni verilmemiş, sunucunun `exp.host` 'a internet çıkışı yok, alarm tanımında Mobil bildirim kapalı, kullanıcının o tag'e erişimi yok. Yöneticinin *Ayarlar > Mobil bildirim* sayfasındaki test düğmesi zinciri uçtan uca dener.

Değerler "Bad" ya da "—" görünüyor. Sunucu modunda sorun cihaz ile sunucu arasındadır: *Daha > Cihaz durumu*'nda son hataya bakın, web arayüzünde cihazın *Teşhis* ekranını açın ([Sorun giderme](#)). Telefon modunda: tarama çalışıyor mu (başlıkta **Çalışıyor**)? Telefon cihazla aynı ağda mı? iOS'ta MergenHub'a **Yerel ağ** izni verilmiş mi (telefon *Ayarlar > Gizlilik ve Güvenlik > Yerel Ağ*)? Cihaz kimliği (IP, unit ID, rack/slot) ve port doğru mu?

Değer yazılamıyor. RuntimeControl yetkiniz, tag'in yazılabilir olması ve tag'e yazma erişiminiz (ACL) gerekir. Yazma düğmesi görünmüyorsa bunlardan biri eksiktir.

Ekranlar sekmesinde "SCADA yetkisi yok". Kullanıcınıza sunucuda Scada yetkisi verilmelidir.

Sunucu ekranı sadeleştirilmiş görünüyor. Sunucu MergenHub 3.5.0 ya da eskidir; 3.5.1 ve üzerine güncelleyin.

Kütüphane açılmıyor. Kütüphane internetten indirilir; çevrimdışıyken son liste gösterilir ama kurulum yapılamaz. Paketi siteden indirip **Dosya, pano, QR ya da adres** ile kurabilirsiniz.

"Bu adla hesaplanan tag zaten var." Aynı cihaz paketi aynı cihaz adıyla daha önce kurulmuştur. Farklı bir **Cihaz (istasyon)** adı verin.

Telefon modunda tarama kendiliğinden duruyor. Uygulama arka plana alındığında ya da telefon uykuya geçtiğinde tarama durur. *Ayarlar › Tarama sürerken ekranı açık tut*'u açık tutun ve uygulamayı ön planda bırakın.

Cihaz şablonu (paket) oluşturma

.mhpkg.json biçimi: parametreler, kanal-cihaz-tag, hesaplanan tag ve ifade fonksiyonları, alarmlar, sunucu ve telefon ekran takımları, doğrulama, tam örnek, yapay zekâ becerisi, kütüphaneye cihaz ekletme.

Bir **cihaz şablonu** (MergenHub paketi), bir cihazı tüm ayrıntısıyla taşıyan tek bir JSON dosyasıdır: kanal ve sürücü ayarları, cihaz, register haritası (tag'ler), hesaplanan tag'ler, alarmlar ve operatör ekranları. Kurulumda yalnız sahaya özgü bilgiler (IP adresi, port, unit ID...) sorulur; geri kalan her şey hazır gelir. Dosya uzantısı `.mhpkg.json` 'dur.

Bu sayfa bir paketin nasıl hazırlandığını, alanlarını ve kurulumda neler olduğunu anlatır.

Paketler nerede kullanılır

- **MergenHub web arayüzü** — *Cihaz Kütüphanesi* penceresi (proje ağacının başlığındaki kitap simgesi, *Bağlantılar* görünümündeki **Cihaz Kütüphanesi** düğmesi ya da SCADA'da yeni ekran seçicisindeki kart). Kütüphaneden kurar; **Dosyadan içe aktar** ile elinizdeki `.mhpkg.json` dosyasını kurar.
- **MergenHub Mobil** — *Cihaz ekle* sihirbazı (kütüphane, şablonlarım, dosya, pano, QR, adres), *Şablonlar* sayfası ve *Paket / şablon içe aktar*. Bkz. [MergenHub Mobil](#).
- **Cihaz Kütüphanesi** — Envest'in hazırlayıp test ettiği paketler [sitede](#) yayımlanır ve uygulamalar bunları herkese açık bir API'den okur: `GET /api/v1/library` (katalog), `GET /api/v1/library/{slug}` (tek cihaz), `GET /api/v1/library/{slug}/package` (paket dosyası).

Cihaz Kütüphanesi
×

Marka, model, yazılım sürümü, protokol...

Dosyadan içe aktar

Tümü · 11

Su Yönetimi · 11

SONDAJ KUYUSU
Envest CP110 — Derin Kuyu İstasyonu
 Envest · CP110 Kuyu Kontrol Panosu · CP110 Well
 Modbus TCP/IP Ethernet · 445 tag · 10 hesaplanan · 10 ekran

☐ Seç

SU DEPOSU
Envest CP110 — Su Dağıtım Deposu
 Envest · CP110 Depo Kontrol Panosu · CP110 Store
 Modbus TCP/IP Ethernet · 560 tag · 4 hesaplanan · 9 ekran

☐ Seç

TERFİ İSTASYONU
Envest CP110 — Terfi Pompa İstasyonu
 Envest · CP110 Terfi Kontrol Panosu · CP110 Riser
 Modbus TCP/IP Ethernet · 486 tag · 7 hesaplanan · 10 ekran

☐ Seç

SONDAJ KUYUSU
AQUA CNT 100+ — Kuyu Pompa Kontrolü
 Envest · AQUA CNT 100+ · AquaCnt V2 — Kuyu
 Modbus TCP/IP Ethernet · 159 tag · 11 hesaplanan · 10 ekran

☐ Seç

SU DEPOSU
AQUA CNT 100+ — Depo İzleme
 Envest · AQUA CNT 100+ · AquaCnt V2 — Depo
 Modbus TCP/IP Ethernet · 159 tag · 5 hesaplanan · 9 ekran

☐ Seç

TERFİ İSTASYONU
AQUA CNT 100+ — Terfi Pompa Kontrolü

☐ Seç

Kaynak: mergenhub.envest.com.tr/devices · Cihazınız yok mu? Ekleme isteği gönderin

İptal

Seçilenleri kur

MergenHub web arayüzünde Cihaz Kütüphanesi penceresi

İnternete çıkamayan bir sunucuda paketi siteden indirip **Dosyadan içe aktar** ile kurun.



Kurulum bilgileri

Yalnız aşağıdakiler sorulur; tag'ler, hesaplanan tag'ler ve ekranlar hazır gelir. Adı değiştirirseniz ekran ve hesaplanan tag yolları yeni ada taşınır.

Envest CP110 — Su Dağıtım Deposu
Envest · CP110 Depo Kontrol Panosu · CP110 Store · envest-cp110-depo.mhpkq.json

Kanal

Cihaz (istasyon) adı

Korubin-Depo

Depo2

IP adresi *

192.168.1.50

BAĞLANTI

Port *

Unit ID *

502

1

• IP adresi: gerekli

İptal
Kur

Dosyadan içe aktar — kurulum bilgileri ve denetim sonucu

Hızlı başlangıç

En kolay yol, çalışan bir cihazdan başlamaktır:

1. MergenHub Mobil'i **telefon modunda** açın, cihazı kanal, cihaz ve birkaç tag'ile tanımlayın, değerlerin geldiğini görün.
2. Kanallar sekmesinde cihazın ... menüsünden **Şablon olarak kaydet / paylaş**'ı seçin. Ad, açıklama, üretici ve model girin; ekranlar da gelsin istiyorsanız **Ekranları da al**'ı açık bırakın.
3. **Dosya olarak paylaş** ya da **Panoya kopyala** ile paketi alın. Cihaz kimliğindeki ve ayarlardaki IP adresleri kendiliğinden `{{ip}}` parametresine çevrilmiştir.

4. Dosyayı bir metin düzenleyicide ya da bir yapay zekâyla düzenleyin (tag ekleyin, açıklama yazın, alarm ve ekran ekleyin). Dışa aktarılan dosyanın `guide` bölümünde biçim kuralları, kullanılan sürücünün ayar anahtarları, adres örnekleri ve sembol listesi yazılıdır.
5. Paketi geri alın: telefonda *Cihaz ekle > Dosya, pano, QR ya da adres*, sunucuda *Cihaz Kütüphanesi > Dosyadan içe aktar*.

Tüm projeyi paket yapmak için telefonda *Daha > Proje > Projeyi dışa aktar* kullanılır (`kind: "project"`).

Dosya yapısı

Paket tek bir JSON nesnesidir:

```
{
  "format": "mergenhub-package",
  "formatVersion": 1,
  "kind": "template",
  "meta": { "name": "..." },
  "parameters": [],
  "channels": [],
  "derived": [],
  "alarms": [],
  "screens": [],
  "notes": "..."
}
```

ALAN	ZORUNLU	AÇIKLAMA
format	evet	Her zaman "mergenhub-package"
formatVersion	önerilir	Şu an 1 ; uygulamanın bildiğinden büyükse paket reddedilir
kind	evet	"template" (cihaz şablonu, kurulumda parametre sorulur) ya da "project" (tüm proje)
meta	evet	Paketin adı ve tanıtım bilgileri
parameters	hayır	Kurulumda sorulacak değerler
channels	evet	Kanal → cihaz → tag ağacı (en az bir kanal)
derived	hayır	Hesaplanan (gelişmiş) tag'ler
alarms	hayır	Alarm tanımları
screens	hayır	Operatör ekranları
notes	hayır	Cihazın ne işe yaradığını anlatan serbest metin (Markdown)
guide	hayır	Dışa aktarırken yazılan açıklama bölümü; içe alırken yok sayılır

Bir cihaz şablonunda tek kanal ve tek cihaz bulunur. Yalnız bu durumda kurulumda kanal ve cihaz adı sorulur ve ad değişince yollar taşınır.

META

ALAN	AÇIKLAMA
name	Paketin görünen adı (zorunlu), ör. "Örnek EA-100 Enerji Analizörü"
description	Bir iki cümlelik özet
vendor , model , version	Üretici, model ve cihaz yazılım sürümü; listelerde ve aramada görünür
author	Hazırlayan
keywords	Arama için anahtar kelimeler (dizi)
homepage	Ürün sayfası adresi
createdAt , createdWith	Dışa aktarırken kendiliğinden yazılır

Parametreler ve yer tutucular

parameters kurulumda kullanıcıya sorulacak alanlardır. Paketin herhangi bir yerindeki **dizge** değerde {{anahtar}} yazılan yere kullanıcının girdiği değer konur.

```
"parameters": [
  { "key": "ip", "label": "IP adresi", "type": "text", "required": true, "placeholder": "192.168.1.50" },
  { "key": "port", "label": "Port", "type": "number", "default": 502, "group": "Bağlantı" },
  { "key": "unit", "label": "Unit ID", "type": "number", "default": 1, "group": "Bağlantı" }
]
```

ALAN	AÇIKLAMA
key	Harf ya da _ ile başlar; harf, rakam ve _ içerir (ör. ip , port , unit , fins_node)

ALAN	AÇIKLAMA
label	Formda görünen ad
type	text (varsayılan), number ya da select
default	Varsayılan değer
required	true ise boş bırakılamaz
placeholder , hint	Alan içi örnek ve alt açıklama
group	Formda başlık; aynı gruptakiler birlikte gösterilir
options	select için [{ "value": 1, "label": "TCP" }, ...]

Kurallar:

- Yer tutucu yalnız dizge değerlerin içinde çalışır. Sayısal bir ayarı parametreye bağlamak için dizge yazın: "modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER": "{{port}}". Kurulumda sürücünün sayısal ayarları sayıya çevrilir.
- Cihaz kimliği tipik olarak parametrelerden kurulur: "id": "<{{ip}}>{{unit}}".
- Kurulumdan sonra doldurulmamış bir {{...}} kalırsa kurulum açılmaz.
- IP adresinin varsayılanını boş bırakıp required: true yapın; örnek adresi placeholder 'a yazın. Böylece kullanıcı adresi girmeden kuramaz.

Kanallar, cihazlar ve tag'ler

```
"channels": [
  {
    "name": "Enerji",
    "driver": "Modbus TCP/IP Ethernet",
    "settings": {},
    "devices": [
      {
        "name": "Analizor1",
        "id": "<{{ip}}>{{unit}}",
        "scanMs": 1000,
        "timeouts": { "connectS": 3, "requestMs": 1000, "retries": 3 },
        "settings": { "modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER": "{{port}}" },
        "tags": [
          { "name": "GerilimL1", "address": "30001", "type": "Float", "access": "read", "scanMs": 1000, "units": "V" }
        ],
        "groups": []
      }
    ]
  }
]
```

Kanal

ALAN	AÇIKLAMA
name	Kanal adı (zorunlu, nokta içeremez)
driver	Sürücünün tam adı (aşağıdaki tablo)

ALAN	AÇIKLAMA
settings	Kanal düzeyindeki sürücü ayarları (Kepware uyumlu anahtarlar)
description	Açıklama
devices	Cihaz dizisi (zorunlu)

Cihaz

ALAN	VARSAYILAN	AÇIKLAMA
name	—	Cihaz adı (zorunlu, nokta içeremez)
id	sürücüye göre	Cihaz kimliği; biçimi sürücüye bağlıdır
description	—	Açıklama
enabled	true	false ise cihaz taranmaz
scanMs	1000	Cihaz tarama hızı (ms)
scanMode	4	0 istemciye uy, 1 en hızlı, 2 sabit hız, 3 talep, 4 tag hızına uy
timeouts	3 / 1000 / 3	connectS bağlantı zaman aşımı (sn), requestMs istek zaman aşımı (ms), retries deneme sayısı
settings	sürücü varsayılanları	Cihaz düzeyindeki sürücü ayarları
tags , groups	—	Tag'ler ve tag grupları

Tag

ALAN	VARSAYILAN	AÇIKLAMA
name	—	Tag adı (zorunlu, nokta içeremez, cihaz ya da grup içinde tekil)
address	—	Sürücünün adres sözdizimiyle adres (zorunlu)
type	Default	Veri tipi (aşağıda)
access	read	read ya da readwrite
scanMs	1000	Tag tarama hızı (ms)
description	—	Açıklama; 6 karakter ve altıysa uygulamada birim olarak gösterilir
units	—	Birim; description boşsa açıklama olarak kullanılır
scaling	—	Ölçekleme (aşağıda)

Veri tipleri: Default , Boolean , Char , Byte , Short , Word , Long , DWord , Float , Double , String , BCD , LBCD , Date , LLong , QWord . Default sürücünün adrese göre seçtiği tiptir.

Ölçekleme ham değeri mühendislik birimine çevirir; yazarken ters uygulanır:

```
"scaling": { "type": "linear", "rawLow": 0, "rawHigh": 10, "low": 0, "high": 1, "resultType": "Double", "units": "kWh" }
```

`type` `linear` ya da `sqrt` (karekök); `rawLow` / `rawHigh` ham aralık, `low` / `high` ölçekli aralık; `resultType` ölçekli veri tipi (varsayılan `Double`); `clampLow` , `clampHigh` sınırdan kırpar, `negate` işaret çevirir.

Gruplar tag'leri cihaz içinde klasörler; iç içe olabilir. Gruptaki tag'in yolu `Kanal.Cihaz.Grup.Tag` olur.

```
"groups": [ { "name": "Enerji", "description": "Sayaçlar", "tags": [ ... ], "groups": [ ... ] } ]
```

SÜRÜCÜLER VE ADRES SÖZDİZİMİ

`driver` alanına sürücünün tam adı yazılır. Adres sözdizimi ve ayar anahtarları MergenHub sunucusu ile MergenHub Mobil'de aynıdır; ayrıntılar için [Protokol sürücüler](#).

SÜRÜCÜ (DRIVER)	CİHAZ KİMLİĞİ (ID)	ADRES ÖRNEKLERİ
Simulator	cihaz numarası, ör. 1	SINE:10:0:100 , RAMP:30:0:500 , RANDOM:20:80 , REG:1
Modbus TCP/IP Ethernet	<IP>.Unit , ör. <192.168.1.10>.1	40001 holding, 30001 giriş register, 00001 bobin, 10001 ayrık giriş, 40001.3 bit, 40010.16H dize
Modbus RTU Serial , Modbus ASCII Serial	köle adresi 1 – 247	Modbus TCP ile aynı
Siemens S7 TCP/IP	<IP>.Rack.Slot , ör. <192.168.0.1>.0.1	DB1.DBD4 , DB1.DBW2 , DB1.DBB0 , DB1.DBX0.3 , MW10 , IO.0 , Q0.0
OMRON FINS Ethernet	<IP>.Düğüm , ör. <192.168.1.10>.1	D100 , D0.05 , CI010 , W5.00 , H2 , A3 , E100
OMRON NJ Ethernet	<IP>	PompaDurum , Hat1.Vana[3] , Matris[1,2] , Sayac.5
MQTT Client	broker host:port (kanalda broker verilmişse yok sayılır)	sensors/temp , sensors/pump1:/flow , plant/csv;idx=2;sep=;
DNP Client Ethernet	outstation IP	AI0 , BI5 , CTR2 , AO1 , BO3 , 30.1.51.Value
IEC 60870-5-104 Client	RTU IP	100 , IOA100 , M_ME_NC.100.VALUE , C_SC_NA.3000.DIRECTVALUE
IEC 61850 MMS Client	IED IP	LD0/MMXU1.MX.TotW.mag.f , LD0/XCBR1.ST.Pos.stVal
OPC UA Client	kullanılmaz (endpoint kanal ayarında)	ns=2;s=Kanal.Cihaz.Tag , ns=2;i=1001

Sık kullanılan ayar anahtarları:

- Modbus TCP** (cihaz): `modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER` (502), `modbus_ethernet.DEVICE_ZERO_BASED_ADDRESSING` , `modbus_ethernet.DEVICE_FIRST_WORD_LOW` (32 bitte word sırası), `modbus_ethernet.DEVICE_HOLDING_REGISTERS` (blok boyutu, 1–125).
- Modbus RTU/ASCII** (cihaz): `servermain.DEVICE_ETHERNET_COMMUNICATIONS_IP` ve `servermain.DEVICE_ETHERNET_COMMUNICATIONS_PORT` (seri sunucu), `modbus.DEVICE_...` anahtarları.
- Siemens S7** (cihaz): `siemens_tcpip_ethernet.DEVICE_COMMUNICATIONS_PORT_NUMBER` (102), `siemens_tcpip_ethernet.DEVICE_S7_COMMUNICATIONS_CPU_RACK` , `..._CPU_SLOT` , model için `servermain.DEVICE_MODEL` (0 S7-200, 1 S7-300, 2 S7-400, 3 S7-1200, 4 S7-1500).

- **OMRON FINS:** kanalda `omron_fins_ethernet.CHANNEL_PORT_NUMBER` (9600); cihazda `omron_fins_ethernet.DEVICE_PROTOCOL` (0 UDP, 1 TCP) ve düğüm numaraları.
- **MQTT** (kanal): `mqtt.BROKER_HOST`, `mqtt.PORT`, `mqtt.USERNAME`, `mqtt.PASSWORD`, `mqtt.QOS`, `mqtt.TRANSPORT`.
- **OPC UA istemci** (kanal): `opcuaclient.CHANNEL_UA_SERVER_ENDPOINT_URL`.

Yazılmayan ayar sürücünün varsayılanını alır. Bir sürücünün tüm ayar anahtarları, o sürücüyle dışa aktarılmış bir paketin `guide.drivers` bölümünde açıklamalarıyla yazılıdır.

Hesaplanan tag'ler

`derived` dizisi başka tag'lerden hesaplanan tag'leri tanımlar. Kurulunca yolları `_AdvancedTags.<ad>` olur; ekranlar ve alarmlar bu yolu kullanır.

```
"derived": [
  {
    "name": "Analizor1_GorunurGuc",
    "type": "Derived",
    "expression": "ROUND({Enerji.Analizor1.GerilimL1} * {Enerji.Analizor1.AkimL1} / 1000, 2)",
    "scanRateMs": 1000,
    "description": "Görünür güç (kVA)"
  }
]
```

ALAN	AÇIKLAMA
<code>name</code>	Ad; sunucu genelinde tekildir, cihaz adıyla başlamalıdır (<code><Cihaz>_...</code>)
<code>type</code>	<code>Derived</code> (ifade), <code>Totalizer</code> , <code>Average</code> , <code>Maximum</code> , <code>Minimum</code>
<code>expression</code>	<code>Derived</code> 'da ifade; diğerlerinde kaynak tag'in yolu (süslü parantezsiz)
<code>sampleSize</code>	Ortalama / maks / min için örnek sayısı (varsayılan 10)
<code>scanRateMs</code>	Güncelleme hızı (ms, varsayılan 1000)
<code>enabled</code>	Varsayılan <code>true</code>
<code>description</code>	Açıklama
<code>dataType</code>	Sunucuda sonuç veri tipi kodu (varsayılan 9 = Double); telefon yok sayar

Totalizer, değer × saniye toplar; m³/h girişte sonucu 3600'e bölün.

FONKSİYON BAŞVURUSU

İfadede tag referansı `{Kanal.Cihaz.Tag}` biçimindedir (grup varsa `{Kanal.Cihaz.Grup.Tag}`). İşleçler: `+`, `-`, `*`, `/`, `%`, karşılaştırma `<`, `<=`, `>`, `>=`, `==`, `!=` (doğruysa 1, yanlışsa 0), mantık `&&`, `||`, `!` ve parantez. Ondalık ayırıcı her zaman noktadır (`1.5`). Fonksiyon adları büyük/küçük harf duyarlıdır; argümentsiz fonksiyonlar da parantez ister (`PI()`). İfade en çok 8000 karakter olabilir. Fonksiyon listesi sunucu ve telefonda aynıdır.

FONKSİYON	AÇIKLAMA
<code>ABS(x)</code>	Mutlak değer
<code>SQRT(x)</code>	Karekök (negatif girişte 0)

FONKSİYON	AÇIKLAMA
POW(x, y)	x üssü y
EXP(x) , LN(x)	e üssü x, doğal logaritma
LOG(x) , LOG(x, taban)	10 tabanlı ya da verilen tabanda logaritma
ROUND(x) , ROUND(x, basamak)	Yuvarlama (yarı yukarı; basamak 0-15)
FLOOR(x) , CEIL(x)	Aşağı / yukarı yuvarlama
MOD(a, b)	Bölümden kalan (b = 0 ise 0)
SIGN(x)	İşaret: -1, 0, 1
MIN(a, b, ...) , MAX(a, b, ...)	En küçük / en büyük
AVG(a, b, ...) , SUM(a, b, ...)	Ortalama / toplam
IF(koşul, a, b)	Koşul 0 değilse a, değilse b
CLAMP(x, alt, üst)	Aralığa sıkıştırma
MAP(x, girişAlt, girişÜst, çıkışAlt, çıkışÜst)	Doğrusal ölçekleme
LERP(a, b, t)	a ile b arasında doğrusal ara değer
POLY(x, c0, c1, c2, ...)	Polinom: c0 + c1·x + c2·x ² + ...
HYPOT(a, b)	$\sqrt{a^2 + b^2}$
SIN , COS , TAN , ASIN , ACOS , ATAN	Trigonometri (radyan)
ATAN2(y, x)	İki argümanlı arktanjan
SINH , COSH , TANH	Hiperbolik fonksiyonlar
DEG(x) , RAD(x)	Radyandan dereceye, dereceden radyana
PI() , E()	Sabitler
RANDOM() , RANDOM(üst) , RANDOM(alt, üst) , RND(...)	Rastgele sayı; yalnız test için

Örnekler: ($\{K.C.A\} + \{K.C.B\}$) / 2 , IF($\{K.C.Basinc\} > 6$, 1, 0) , MAX($\{K.C.Seviye\} - \{K.C.Statik\}$, 0) , MAP($\{K.C.Ham\}$, 4, 20, 0, 10) .

Alarmlar

```
"alarms": [
  { "name": "Yüksek gerilim L1", "tag": "Enerji.Analizor1.GerilimL1", "conditionType": "Hi",
    "limit": 250,
    "deadband": 2, "severity": 700, "message": "Faz 1 gerilimi 250 V üstünde", "enabled": true }
]
```

ALAN	VARSAYILAN	AÇIKLAMA
name	—	Alarm adı
tag	—	Paketteki bir tag'in tam yolu ya da <code>_AdvancedTags.<ad></code> (zorunlu, pakette bulunmalı)

ALAN	VARSAYILAN	AÇIKLAMA
conditionType	—	HiHi , Hi , Lo , LoLo , DevHi , DevLo (sapma), RocHi , RocLo (değişim hızı, birim/sn)
limit	—	Sınır
setpoint	0	Sapma alarmlarında ayar noktası
deadbandType , deadband	Absolute , 0	Ölü bant: None , Absolute , Percent
rangeLow , rangeHigh	0, 100	Yüzde ölü bant için aralık
onDelaySec , offDelaySec	0	Açılma ve kapanma gecikmesi (sn)
severity	500	1-1000; 800 ve üstü kritik
message , area	—	İleti ve alan
enabled	true	Etkin

Ekranlar

Ekranlar sunucudaki SCADA ekranlarıyla aynı yapıdadır. Bir ekran bir koordinat alanında (width × height) konumlanmış nesnelerden (widgets) oluşur.

ALAN	AÇIKLAMA
id	Paket içinde tekil kimlik (zorunlu); kurulumda yenilenir, bağlar taşınır
name	Ekran adı; cihaz adını içerirse kurulumda yeni ada çevrilir
target	"server" (yalnız sunucuya kurulur), "phone" (yalnız telefon moduna kurulur) ya da yok (ikisine de)
parentId	Üst ekranın id 'si; boşsa ana ekrandır
order	Alt ekranlar arasındaki sıra
width , height	Koordinat alanı; telefon dikey 400 × 700 (uzun ekranda yükseklik artar), sunucu ör. 1200 × 700
background	Zemin rengi (#eef2f7); telefon ekranında boş bırakılırsa temaya uyur
group , zoneId	Sunucuda ekran grubu ve bölge; şablonlarda boş bırakın
widgets	Nesneler

İKİ EKRAN TAKIMI

Bir cihaz için iyi bir paket iki takım taşır:

- **Sunucu takımı** ("target": "server") — **tek bir ana ekran** (parentId boş) ve onun altında ayar ekranları (parentId = ana ekranın id 'si). Web arayüzünde ana ekran listede görünür, ayar ekranları onun alt dalındadır. Ana ekrandan ayar ekranlarına ve geri düğmeler (nav) koyun.
- **Telefon takımı** ("target": "phone") — dikey bir **Özet** ekranı (ölçümler, kumanda ve ayar ekranlarına düğmeler) ve onun alt ekranları olarak tek sütunlu ayar ekranları. Telefon modunda Özet açılınca üstte **Özet** ve alt ekranlardan oluşan bir sekme şeridi çıkar. Ekran adlarını <Cihaz> – Özet , <Cihaz> – Ayarlar biçiminde verin; sekmelerde "—" işaretinden sonraki kısım görünür.

Kurulumda yalnız hedefe uyan takım kurulur. Diğer takımdaki bir üst ekrana bağlı ekran ana ekran olur; diğer takıma giden `nav` bağları kaldırılır. `target` vermezseniz ekran iki yere de kurulur.

NESNELER

```
{ "id": "w2", "type": "display", "x": 24, "y": 80, "w": 220, "h": 80,
  "tag": "Enerji.Analizor1.GerilimL1", "props": { "label": "Gerilim L1", "units": "V", "decimals":
1 } }
```

ALAN	AÇIKLAMA
<code>id</code>	Paket içinde tekil (ör. <code>w1</code> , <code>w2</code> ...)
<code>type</code>	Nesne tipi (aşağıda)
<code>x</code> , <code>y</code> , <code>w</code> , <code>h</code>	Konum ve boyut (ekran koordinatlarında)
<code>rot</code>	Döndürme açısı (derece)
<code>tag</code>	Bağlı tag yolu: <code>Kanal.Cihaz.Tag</code> ya da <code>_AdvancedTags.<ad></code>
<code>props</code>	Tipe göre özellikler
<code>anim</code>	Koşullu kurallar
<code>onClick</code> , <code>nav</code>	Dokununca başka ekrana geçmek için <code>"onClick": "nav"</code> ve <code>"nav": "<ekran id>"</code>
<code>popupScreen</code>	Açılır pencere olarak gösterilecek ekran (sunucu)

TİP	NE YAPAR	ÖNEMLİ PROPS
<code>display</code>	Değer göstergesi	<code>label</code> , <code>units</code> , <code>decimals</code> , <code>color</code> , <code>fontSize</code>
<code>label</code>	Sabit yazı (tag yok)	<code>text</code> , <code>fontSize</code> , <code>color</code> , <code>align</code>
<code>lamp</code>	Lamba	<code>label</code> , <code>onColor</code> , <code>offColor</code> , <code>onText</code> , <code>offText</code> , <code>blinkOn</code> , <code>onOp</code> / <code>onA</code> / <code>onB</code>
<code>button</code>	Basınca <code>onValue</code> / <code>offValue</code> yazar	<code>label</code> , <code>onValue</code> , <code>offValue</code> , <code>onText</code> , <code>offText</code> , <code>onColor</code> , <code>offColor</code>
<code>switch</code>	Anahtar (buton gibi yazar)	<code>label</code> , <code>onValue</code> , <code>offValue</code> , <code>onText</code> , <code>offText</code> , <code>onColor</code>
<code>radio</code>	Seçim düğmesi	<code>label</code> , <code>onValue</code> , <code>offValue</code>
<code>input</code>	Değer girişi (yazılabilir tag)	<code>label</code> , <code>decimals</code>
<code>sfield</code>	Ayar satırı: etiket + değer (+ birim)	<code>label</code> , <code>units</code> , <code>decimals</code> , <code>editable</code> , <code>step</code>
<code>gauge</code>	Kadran	<code>label</code> , <code>min</code> , <code>max</code> , <code>units</code> , <code>decimals</code> , <code>fillColor</code>
<code>tank</code> , <code>levelbar</code>	Depo seviyesi, seviye çubuğu	<code>label</code> , <code>min</code> , <code>max</code> , <code>units</code> , <code>decimals</code> , <code>fillColor</code>
<code>pump</code> , <code>motor</code> , <code>fan</code> , <code>valve</code>	Makine simgeleri (çalışırken döner / renk değiştirir)	<code>label</code> , <code>onColor</code> , <code>offColor</code> , <code>onText</code> , <code>offText</code> , <code>spin</code> , <code>blinkOn</code> , <code>onOp</code> / <code>onA</code> / <code>onB</code>
<code>pipe</code>	Boru; akış varken renklenir	<code>thickness</code> , <code>flowColor</code> , <code>onOp</code> / <code>onA</code>
<code>rect</code>	Panel / dikdörtgen (tag yok)	<code>fill</code> , <code>stroke</code> , <code>radius</code>

TİP	NE YAPAR	ÖNEMLİ PROPS
image	Resim (tag yok)	src (data URI ya da adres)
svg	Sembol kütüphanesinden nesne	lib2 (sembol kimliği), label , onOp / onA / onB , onColor , offColor

Açık durumu. Lamba, makine ve sembollerde nesnenin ne zaman "açık" sayılacağı onOp (gt , ge , eq , ne , lt , le , between), onA ve onB ile verilir; verilmezse değer 0,5'ten büyükse açıktır.

Koşullu kurallar (anims): { "kind": "color", "tag": "", "op": "gt", "a": 80, "b": null, "p2": "#ef4444" } . kind : color (rengi p2 yapar), blink (yanıp söner), visible (koşul doğruyken nesneyi gizler), level (dolum: a en az, b en çok, p2 renk, op always). tag boşsa nesnenin kendi tag'i kullanılır.

Semboller. svg nesnesinin lib2 alanı yerleşik sembol kütüphanesindeki kimliktir (ör. en-pumps/en_pump_submersible_1 , en-meters/en_flowmeter_1 , pumps_centrifugal_pump_4). Kimliklerin tam listesi dışa aktarılan paketlerin guide.symbols bölümündedir.

Sunucu ekranlarında web SCADA editörünün diğer nesne tipleri de kullanılabilir; telefon bunları soluk kutu olarak gösterir ve denetim uyarı verir. İki yerde çalışması gereken ekranlarda tablodaki tipleri kullanın.

Adlandırma ve kurulumda yeniden adlandırma

- Kanal, cihaz, grup ve tag adlarında **nokta (.) kullanılmaz**; nokta yol ayırıcıdır. Adlar bulunduğu düzeyde tekil olmalıdır (büyük/küçük harf duysanız).
- Tag yolu** Kanal.Cihaz.Tag (grupta Kanal.Cihaz.Grup.Tag) biçimindedir. Ekranlarda, alarmlarda ve ifadelerde her zaman tam yol yazılır.
- Cihaz adı** yalnız kanal içinde tekildir; aynı paket aynı kanala farklı cihaz adlarıyla birçok kez kurulabilir.
- Hesaplanan tag adları sunucu genelinde tekildir.** Bu yüzden her hesaplanan tag'in adını cihaz adıyla başlatın: Analizor1_GorunurGuc . Kurulumda cihaz adı değişince bu ön ek de değişir (Analizor2_GorunurGuc) ve aynı paket ikinci kez çakışmadan kurulur.

Kurulumda kullanıcı kanal ve cihaz adını değiştirirse paket şöyle güncellenir:

- Tüm Kanal.Cihaz ile başlayan yollar (ekran nesnelerinin tag 'i, kurallar, alarm tag 'leri, ifadelerdeki {...} referansları) yeni ada taşınır.
- Adı <EskiCihaz>_ ile başlayan hesaplanan tag'ler <YeniCihaz>_ ile yeniden adlandırılır; _AdvancedTags.... referansları da taşınır.
- Ekran adlarında ve label nesnelerinin text 'inde geçen eski cihaz adı (tam kelime olarak, en az 3 karakterse) yeni adla değiştirilir.

Kurulumda denetlenen çakışmalar:

- Kanal adı hedefte başka bir sürücüyle kullanılıyorsa kurulum açılmaz; aynı sürücüyle kullanılıyorsa cihaz mevcut kanala eklenir.
- Kanalda aynı adla cihaz varsa kurulum açılmaz; uygulama boştaki ilk numaralı adı önerir (Analizor1 → Analizor2).
- Hesaplanan tag adı hedefte varsa kurulum açılmaz; farklı bir cihaz adı çözer.

Doğrulama ve sık hatalar

MergenHub Mobil paketi kurmadan önce denetler; sorunlar kurulum kartında listelenir. Web arayüzünün **Dosyadan içe aktar**'ı parametre, ad ve çalışma denetimlerini yapar ama adresleri sürücüyü doğrulamaz; yeni bir paketi önce telefonda denemek iyi bir alışkanlıktır.

Hatalar (kurulum açılmaz):

- Bu bir MergenHub paketi değil — format alanı eksik ya da yanlış.
- JSON hatası (satır ..., sütun ...) — sözdizimi hatası; konum ve çevresindeki metin gösterilir.
- Bilinmeyen sürücü "..." — driver alanında sürücünün tam adını yazın (ör. Modbus TCP/IP Ethernet).
- Adres geçersiz (...) — adres sürücünün sözdizimine uymuyor; ileti nedenini söyler (ör. Modbus'ta alan rakamı 0, 1, 3 ya da 4 olmalı).
- Bilinmeyen veri tipi "..." — tip adları yukarıdaki listedeki gibi yazılır (Float , float32 değil).
- access "read" ya da "readwrite" olmalı .
- ... adı tekrar ediyor , ... adında nokta olamaz , ... adı boş .
- Alarm ...: Tag bulunamadı — alarmın tag'i pakette yok.
- Geçersiz parametre anahtarı — anahtar harf, rakam ve _ dışında karakter içeriyor.
- {{...}} için değer girilmedi — bir yer tutucunun parametresi yok ya da boş.

Uyarılar (kurulum yapılır, ama bakın):

- Paket içinde olmayan tag'e bağlı — ekran nesnesi, kural ya da ifade pakette olmayan bir yolu kullanıyor; çoğunlukla yazım hatasıdır.
- Gidilecek ekran pakette yok — nav bir ekran id 'sine karşılık gelmiyor.
- Bu uygulamanın tanımadığı nesne tipi — telefon bu nesneyi çizemez; sunucuda çalışabilir.
- Sembol kütüphanede yok — lib2 kimliği yanlış.

İfadeler kurulumda sunucu tarafından derlenir; hatalı bir ifade (bilinmeyen fonksiyon, yanlış argüman sayısı, 1, 5 gibi virgüllü sayı) kurulum sonucunda "kurulamayan" olarak listelenir, diğer parçalar kurulur.

Uygulama // ve /* */ yorumlarını ve sondaki fazla virgülleri tolere eder, ama web arayüzü yorumları kabul etmez. Paketleri yorumsuz, geçerli JSON olarak yazın.

Tam örnek

Tek bir Modbus TCP enerji analizörü: altı tag (biri ölçeklenmiş, ikisi yazılabilir), bir hesaplanan tag, bir alarm, sunucu takımında ana ekran ve ayar ekranı, telefon takımında Özet ve ayar ekranı. Bu paket MergenHub Mobil'in denetiminden hatasız ve uyarısız geçer.

```

{
  "format": "mergenhub-package",
  "formatVersion": 1,
  "kind": "template",
  "meta": {
    "name": "Örnek EA-100 Enerji Analizörü",
    "description": "Tek fazlı enerji ölçümü, röle çıkışı ve akım trafosu ayarı.",
    "vendor": "Örnek",
    "model": "EA-100",
    "version": "1.0",
    "author": "Envest",
    "keywords": ["enerji", "analizör", "Modbus TCP"]
  },
  "parameters": [
    { "key": "ip", "label": "IP adresi", "type": "text", "required": true, "placeholder": "192.168.1.50", "hint": "Analizörün ağdaki adresi" },
    { "key": "port", "label": "Port", "type": "number", "default": 502, "group": "Bağlantı" },
    { "key": "unit", "label": "Unit ID", "type": "number", "default": 1, "group": "Bağlantı" }
  ],
  "channels": [
    {
      "name": "Enerji",
      "driver": "Modbus TCP/IP Ethernet",
      "devices": [
        {
          "name": "Analizor1",
          "id": "<{{ip}}>{{unit}}",
          "scanMs": 1000,
          "timeouts": { "connectS": 3, "requestMs": 1000, "retries": 3 },
          "settings": { "modbus_ethernet.DEVICE_ETHERNET_PORT_NUMBER": "{{port}}" },
          "tags": [
            { "name": "GerilimL1", "address": "30001", "type": "Float", "access": "read", "scanMs": 1000, "units": "V" },
            { "name": "AkimL1", "address": "30003", "type": "Float", "access": "read", "scanMs": 1000, "units": "A" },
            { "name": "AktifGuc", "address": "30005", "type": "Float", "access": "read", "scanMs": 1000, "units": "kW" },
            { "name": "ToplamEnerji", "address": "30007", "type": "DWord", "access": "read", "scanMs": 5000, "scaling": { "type": "linear", "rawLow": 0, "rawHigh": 10, "low": 0, "high": 1, "resultType": "Double", "units": "kWh" } },
            { "name": "RoleCikis", "address": "00001", "type": "Boolean", "access": "readwrite", "scanMs": 1000, "description": "Röle çıkışı (1 = çekili)" },
            { "name": "AkimTrafoOrani", "address": "40001", "type": "Word", "access": "readwrite", "scanMs": 5000, "description": "Akım trafosu oranı (ör. 100 = 100/5 A)" }
          ]
        }
      ]
    }
  ],
  "derived": [
    {
      "name": "Analizor1_GorunurGuc",
      "type": "Derived",
      "expression": "ROUND({Enerji.Analizor1.GerilimL1} * {Enerji.Analizor1.AkimL1} / 1000, 2)",
      "scanRateMs": 1000,
      "description": "Görünür güç (kVA) = gerilim × akım / 1000"
    }
  ],
  "alarms": [
  ]
}

```

```

        "name": "Yüksek gerilim L1",
        "tag": "Enerji.Analizor1.GerilimL1",
        "conditionType": "Hi",
        "limit": 250,
        "deadband": 2,
        "severity": 700,
        "message": "Faz 1 gerilimi 250 V üstünde",
        "enabled": true
    }
},
"screens": [
    {
        "id": "ea100-ana", "name": "Analizor1 – Enerji Analizörü", "target": "server", "parentId":
"", "order": 0,
        "width": 1200, "height": 700, "background": "#eef2f7",
        "widgets": [
            { "id": "w1", "type": "label", "x": 24, "y": 16, "w": 700, "h": 40, "props": { "text":
"ENERJİ ANALİZÖRÜ – Analizor1", "fontSize": 24, "color": "#0f172a" } },
            { "id": "w2", "type": "display", "x": 24, "y": 80, "w": 220, "h": 80, "tag":
"Enerji.Analizor1.GerilimL1", "props": { "label": "Gerilim L1", "units": "V", "decimals": 1 } },
            { "id": "w3", "type": "display", "x": 264, "y": 80, "w": 220, "h": 80, "tag":
"Enerji.Analizor1.AkimL1", "props": { "label": "Akım L1", "units": "A", "decimals": 2 } },
            { "id": "w4", "type": "display", "x": 504, "y": 80, "w": 220, "h": 80, "tag":
"Enerji.Analizor1.AktifGuc", "props": { "label": "Aktif güç", "units": "kW", "decimals": 2 } },
            { "id": "w5", "type": "display", "x": 744, "y": 80, "w": 220, "h": 80, "tag":
"Enerji.Analizor1.ToplamEnerji", "props": { "label": "Toplam enerji", "units": "kWh", "decimals":
1 } },
            { "id": "w6", "type": "gauge", "x": 24, "y": 180, "w": 240, "h": 220, "tag":
"_AdvancedTags.Analizor1_GorunurGuc", "props": { "label": "Görünür güç", "min": 0, "max": 50,
"units": "kVA", "decimals": 2 } },
            { "id": "w7", "type": "switch", "x": 284, "y": 180, "w": 200, "h": 70, "tag":
"Enerji.Analizor1.RoleCikis", "props": { "label": "Röle", "onValue": 1, "offValue": 0, "onText":
"AÇIK", "offText": "KAPALI", "onColor": "#22c55e" } },
            { "id": "w8", "type": "rect", "x": 24, "y": 620, "w": 200, "h": 56, "props": { "fill":
"#1d4ed8", "stroke": "#1d4ed8", "radius": 10 }, "onClick": "nav", "nav": "ea100-ayar" },
            { "id": "w9", "type": "label", "x": 24, "y": 620, "w": 200, "h": 56, "props": { "text":
"Ayarlar", "fontSize": 16, "color": "#ffffff" }, "onClick": "nav", "nav": "ea100-ayar" }
        ]
    },
    {
        "id": "ea100-ayar", "name": "Analizor1 – Ayarlar", "target": "server", "parentId": "ea100-
ana", "order": 1,
        "width": 1200, "height": 700, "background": "#eef2f7",
        "widgets": [
            { "id": "w10", "type": "label", "x": 24, "y": 16, "w": 700, "h": 40, "props": { "text":
"AYARLAR – Analizor1", "fontSize": 24, "color": "#0f172a" } },
            { "id": "w11", "type": "sfield", "x": 24, "y": 80, "w": 520, "h": 44, "tag":
"Enerji.Analizor1.AkimTrafoOrani", "props": { "label": "Akım trafosu oranı", "decimals": 0,
"editable": true, "step": 5 } },
            { "id": "w12", "type": "rect", "x": 24, "y": 620, "w": 200, "h": 56, "props": { "fill":
"#1d4ed8", "stroke": "#1d4ed8", "radius": 10 }, "onClick": "nav", "nav": "ea100-ana" },
            { "id": "w13", "type": "label", "x": 24, "y": 620, "w": 200, "h": 56, "props": { "text":
"< Ana ekran", "fontSize": 16, "color": "#ffffff" }, "onClick": "nav", "nav": "ea100-ana" }
        ]
    },
    {
        "id": "ea100-tel", "name": "Analizor1 – Özet", "target": "phone", "parentId": "", "order":
0,
        "width": 400, "height": 700, "background": "",
        "widgets": [
            { "id": "w14", "type": "rect", "x": 0, "y": 0, "w": 400, "h": 64, "props": { "fill":
"#1d4ed8", "stroke": "#1d4ed8", "radius": 0 } },

```

```

    { "id": "w15", "type": "label", "x": 14, "y": 8, "w": 372, "h": 48, "props": { "text":
"Enerji Analizörü", "fontSize": 18, "color": "#ffffff", "align": "left" } },
    { "id": "w16", "type": "display", "x": 8, "y": 76, "w": 188, "h": 62, "tag":
"Enerji.Analizor1.GerilimL1", "props": { "label": "Gerilim L1", "units": "V", "decimals": 1 } },
    { "id": "w17", "type": "display", "x": 204, "y": 76, "w": 188, "h": 62, "tag":
"Enerji.Analizor1.AkimL1", "props": { "label": "Akım L1", "units": "A", "decimals": 2 } },
    { "id": "w18", "type": "display", "x": 8, "y": 146, "w": 188, "h": 62, "tag":
"Enerji.Analizor1.AktifGuc", "props": { "label": "Aktif güç", "units": "kW", "decimals": 2 } },
    { "id": "w19", "type": "display", "x": 204, "y": 146, "w": 188, "h": 62, "tag":
"Enerji.Analizor1.ToplamEnerji", "props": { "label": "Toplam enerji", "units": "kWh", "decimals":
1 } },
    { "id": "w20", "type": "gauge", "x": 106, "y": 220, "w": 188, "h": 178, "tag":
"_AdvancedTags.Analizor1_GorunurGuc", "props": { "label": "Görünür güç", "min": 0, "max": 50,
"units": "kVA", "decimals": 2 } },
    { "id": "w21", "type": "switch", "x": 8, "y": 410, "w": 384, "h": 58, "tag":
"Enerji.Analizor1.RoleCikis", "props": { "label": "Röle", "onValue": 1, "offValue": 0, "onText":
"AÇIK", "offText": "KAPALI", "onColor": "#22c55e" } },
    { "id": "w22", "type": "rect", "x": 8, "y": 480, "w": 188, "h": 48, "props": { "fill":
"#1d4ed8", "stroke": "#1d4ed8", "radius": 10 } },
    { "id": "w23", "type": "label", "x": 8, "y": 480, "w": 188, "h": 48, "props": { "text":
"Ayarlar", "fontSize": 14, "color": "#ffffff" } },
    { "onClick": "nav", "nav": "ea100-tel-ayar" }
  ],
  {
    "id": "ea100-tel-ayar", "name": "Analizor1 – Ayarlar", "target": "phone", "parentId":
"ea100-tel", "order": 1,
    "width": 400, "height": 700, "background": "",
    "widgets": [
      { "id": "w24", "type": "label", "x": 12, "y": 12, "w": 376, "h": 30, "props": { "text":
"AKIM TRAFOSU", "fontSize": 13, "color": "#3b82f6", "align": "left" } },
      { "id": "w25", "type": "sfield", "x": 8, "y": 48, "w": 384, "h": 44, "tag":
"Enerji.Analizor1.AkimTraf0Orani", "props": { "label": "Akım trafosu oranı", "decimals": 0,
"editable": true, "step": 5 } }
    ]
  }
],
"notes": "# Örnek EA-100\\n\\nGerilim, akım, aktif güç ve toplam enerji okunur; röle çıkışı ve
akım trafosu oranı yazılabilir. Toplam enerji sayacı 0,1 kWh çözünürlüklüdür (ham değer / 10)."
}

```

Bu paketi kurarken cihaz adını `Analizor2` yaparsanız tag yolları `Enerji.Analizor2...`, hesaplanan tag `Analizor2_GorunurGuc`, ekran adları `Analizor2 – ...` olur.

Yapay zekâ ile çalışma

Paket biçimi yapay zekâ asistanlarının okuyup yazabileceği şekilde tasarlanmıştır. Bir cihazın register haritasını, kılavuzunu ya da veri sayfasını bir asistana verip paketi hazırlatabilirsiniz.

[MergenHub cihaz şablonu becerisini indirin](#) (zip). İçinde:

- `SKILL.md` — asistanın izleyeceği iş akışı ve kesin kurallar.
- `reference/format.md` — tüm alanların kısa başvurusu.
- `reference/drivers.md` — sürücü adları, cihaz kimliği biçimleri ve adres sözdizimi.
- `reference/expressions.md` — ifade fonksiyonları.
- `examples/minimal-modbus.mhpkg.json` — denetimden geçen tam örnek.

Claude uygulamasında zip dosyasını beceri (skill) olarak yükleyin ya da Claude Code'da `~/ .claude/skills/` klasörüne açın; beceri desteklemeyen bir asistanda dosyaları sohbete ekleyin. Sonra örneğin şunu isteyin: "Ekteki register tablosundan bu enerji analizörü için MergenHub şablonu hazırla; IP, port ve unit ID kurulumda sorulsun; gerilim için 250 V'ta alarm ekle." Asistanla Türkçe konuşabilirsiniz.

Asistanın verdiği dosyayı önce MergenHub Mobil'de *Cihaz ekle › Dosya, pano, QR ya da adres › Panodan yapıştır* ile açıp denetim sonucuna bakın; sonra sunucuya kurun.

Resmi kütüphaneye cihaz ekletme

Bir cihazın Cihaz Kütüphanesi'nde yer almasını istiyorsanız sitedeki müşteri panelinden istek gönderin: *Panel › Cihaz ekleme istekleri › Yeni istek*. MergenHub Mobil'deki kütüphane listesinin altındaki **Ekleme isteği gönderin** bağlantısı ve web arayüzündeki kütüphane penceresi de aynı sayfayı açar.

Formda marka, model, yazılım sürümü, kategori, haberleşme protokolü, ürün sayfası, cihazın ne yaptığı ve hangi verilerin izleneceği, bağlantı bilgisi istenir. En çok beş dosya eklenebilir (her biri 10 MB): fotoğraf, register haritası, kılavuz; hazırladığınız bir `.mhpkg.json` paketi de eklenebilir.

İstekler **otomatik yayımlanmaz**. Envest ekibi cihazı ve belgeleri doğrular, gerekirse sizden bilgi ister, paketi (tag'ler, ekranlar) hazırlayıp test eder ve kütüphaneye kendisi ekler. İsteğin durumu (alındı, inceleniyor, onaylandı — hazırlanıyor, yayında ya da uygun bulunmadı) ve ekibin notu panelde görünür; sonuç e-postayla da bildirilir. Bir firmanın aynı anda en çok on bekleyen isteği olabilir.

Genel Bakış

MergenHub Nedir

SÜRÜM 3.0

MergenHub, yazılım tabanlı bir **endüstriyel OPC UA aggregation (toplama) sunucusudur**. Çok sayıda saha protokolünden (Modbus, Siemens S7, OMRON, DNP3, IEC 60870-5-104, IEC 61850, MQTT...) veri okur; bunları tek bir **OPC UA** adres uzayında ve **REST Configuration API** üzerinden istemci uygulamalara sunar. Tek arayüz + eklenti (plug-in) mimarisi ile standart ve standart-dışı istemcilere tutarlı erişim sağlar.

Aynı zamanda gömülü modüller içerir: **SCADA/HMI ekran editörü**, **Advanced Tags** (türetilmiş tag / ifade motoru), **Veri Köprüsü (Data Bridge)**, **Historian**, **Alarms & Events** ve **Aliases**.

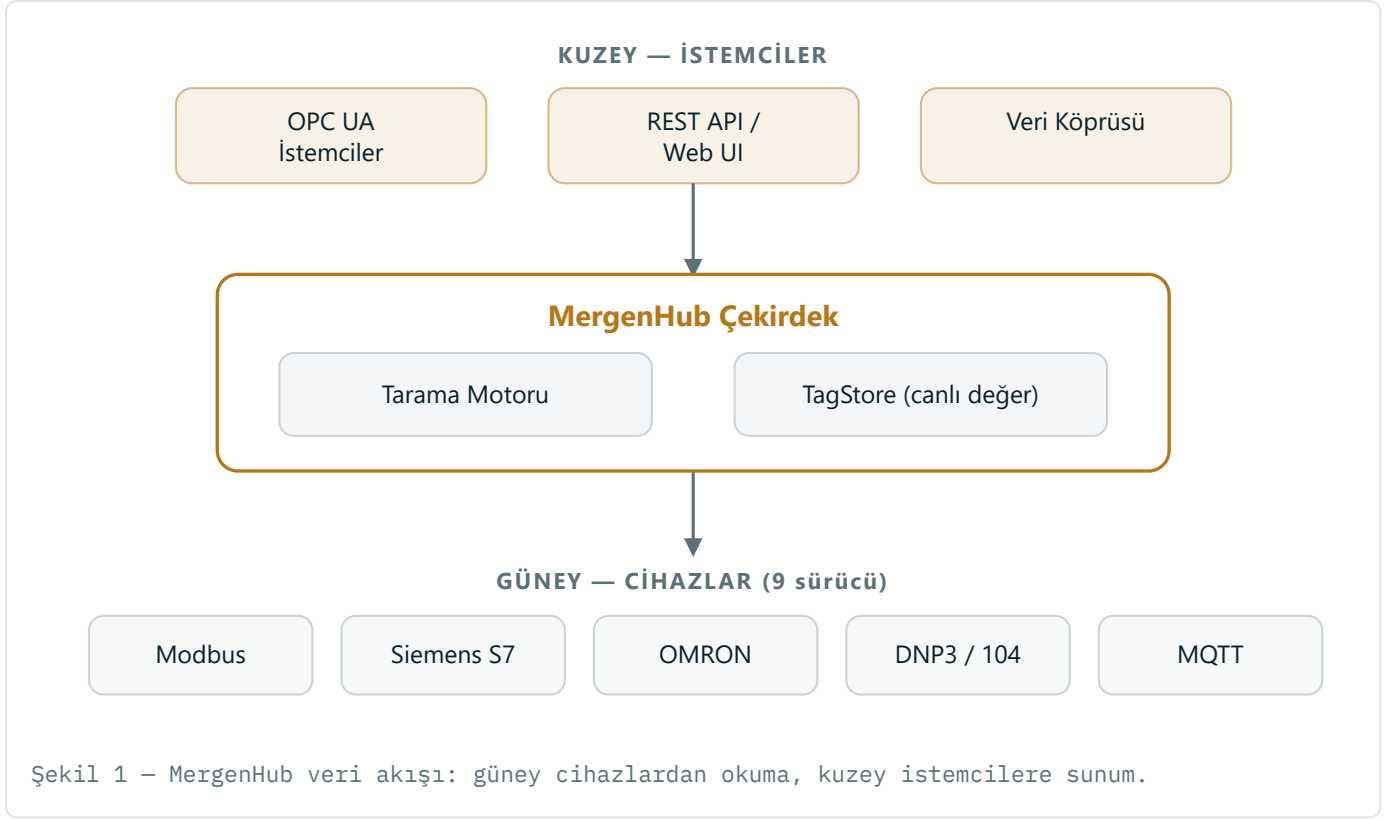
Hiyerarşi	Kanal → Cihaz → Tag
Kuzey yönü (northbound)	OPC UA Server + REST API + Veri Köprüsü (Data Bridge)
Güney yönü (southbound)	9 saha sürücüsü (TCP/seri)
Ölçek	450.000+ tag doğrulandı

Not Uygulama verileri (proje, ayarlar, loglar) varsayılan olarak `C:\ProgramData\MergenHub` altında saklanır. Windows standart kullanıcıların bu klasöre okuma/yazma yetkisi olmalıdır.

Ayrıca bkz [Mimari](#) [Bağlanabilirlik](#) [Kurulum](#)

Mimari

MergenHub, tek bir çekirdek etrafında **kuzey (northbound)** istemciler ile **güney (southbound)** saha cihazlarını köprüler:



Çekirdek ve modüller ayrı Windows servisleri olarak çalışır:

MergenHub	Çekirdek: tarama motoru + OPC UA Server + REST Configuration API + Web UI.
MergenHubDataBridge	Veri Köprüsü / Data Bridge (ayrı süreç, çekirdekten beslenir).
MergenHubHistorian	Historian — tag geçmişi kaydı (ayrı süreç).

Veri akışı: Sürücü cihazdan okur → değer **TagStore**'a yazılır → OPC UA aboneleri, REST `/runtime/v1/values`, IoT/Logger/Alarm ve SCADA aynı canlı değeri paylaşır.

İpucu Yazma ters yönde çalışır: OPC/REST/SCADA → yazma kuyruğu → sürücü cihaza yazar → değer **optimistic** olarak cache'e anında yansır, sonraki taramada cihazdan doğrulanır.

Ayrıca bkz [Tarama Modu](#) [OPC UA Server](#)

Arayüz Turu

Sol menüden ana bölümlere ulaşılır:

Bağlanabilirlik	Kanal/cihaz/tag ağacı; ekle-düzenle-sil, canlı değer/kalite.
OPC UA İstemci	Gömülü QuickClient: keşfet, browse, oku/yaz, abone ol.
Advanced Tags / Veri Köprüsü / Logger / Alarmlar / Aliases	Modüller.
SCADA	Ekran editörü + çalıştır.
Ayarlar	OPC UA server, güvenlik, sertifika, REST API, e-posta, firewall.

Her modalde **Yardıml** düğmesi o ekranın hızlı yardımını açar; bu **Kılavuz** ise tüm sistemi kapsar.

Bağlanabilirlik

Kanallar (Channel)

Proje üç seviyeli bir hiyerarşidir. Bir **kanal**, tek bir iletişim yolunu ve sürücüyü temsil eder; ayarları o kanaldaki tüm cihazlar için ortak ağ/seri parametrelerini taşır.



Şekil 2 – Kanal → Cihaz → Tag hiyerarşisi.

Sürücü	Kanalın protokolü (bir kez seçilir).
Yazma optimizasyonu	Yöntem + duty cycle (tur başına en fazla N yazma).
Soket paylaşımı	Tek-soket cihazlarda kanal başına tek bağlantı (One-Socket-per-Channel).

Uyarı Aynı sahaya (ör. tek-soket hücresel gateway) hem MergenHub hem başka bir sunucu aynı anda bağlanırsa biri reddedilebilir (ECONNREFUSED). Tek-soket cihazlarda tek istemci kullanın.

Sınır Her sürücüyle açılacak kanal sayısı sınırlıdır (ör. Modbus TCP 1024, MQTT 60). Ayrıntı: [Kanal ve Cihaz Sınırları](#).

Ayrıca bkz [Sınırlar](#) [Cihazlar](#) [Tag'ler](#) [Protokoller](#)

Cihazlar (Device)

Bir **cihaz**, kanaldaki fiziksel bir PLC/RTU/gateway'dir. Adres/kimlik (IP, unit id, rack/slot...) ve zamanlama cihaz seviyesindedir.

Data Collection	Cihaz taraması aç/kapa (<code>servermain.DEVICE_DATA_COLLECTION</code>).
Simulated	Donanımsız sentetik veri (test).
Tag Generation	Sürücü destekliyorsa otomatik tag üretimi (adres doğrulamalı).
Kopyala	Cihazı tüm tag/gruplarıyla çoğaltma.
Sınır	Bir kanala eklenebilecek cihaz sayısı sürücüye göre sınırlıdır (ör. FINS 32, S7 1024, Modbus TCP 8192) — bkz. Kanal ve Cihaz Sınırları .

Kanal ve Cihaz Sınırları

Her sürücü için **açılabilir en fazla kanal** ve **bir kanaldaki en fazla cihaz** sayısı sınırlıdır. Değerler endüstri standardı OPC sunucusunun sürücü kılavuzlarındaki sınırlarla aynıdır; kanal tavanı ayrıca 1024 ile kapatılmıştır. Sınır **sürücü başıdır**, proje toplamı değildir: farklı sürücülerle toplamda daha fazla kanal açılabilir.

Sürücü	En fazla kanal · kanal başına en fazla cihaz
Modbus TCP/IP Ethernet	1024 kanal · 8192 cihaz
Modbus RTU Serial	1024 kanal · 255 cihaz
Modbus ASCII Serial	100 kanal · 247 cihaz
DNP Client Ethernet	1024 kanal · 1024 cihaz
IEC 60870-5-104 Client	1024 kanal · 1024 cihaz (istasyon)
IEC 61850 MMS Client	256 kanal · 256 cihaz
MQTT Client	60 kanal (broker bağlantısı) · 1024 cihaz
OMRON FINS Ethernet	512 kanal · 32 cihaz
OMRON NJ Ethernet	256 kanal · 1024 cihaz
Siemens S7 TCP/IP	1024 kanal · 1024 cihaz
OPC UA Client	256 kanal (oturum) · 256 cihaz
Simulator	1024 kanal · 999 cihaz

Ne zaman denetlenir: yalnız **ekleme** sırasında — kanal/cihaz oluşturma (arayüz, REST API, AI asistanı/MCP), kopyalama ve içe aktarma (Kepware JSON, proje paketi, şablon). Sınır doluyorsa işlem reddedilir, nedeni ekranda yazar ve proje **hiç değişmez**. Kayıtlı mevcut proje yüklenirken hiçbir şey reddedilmez ya da silinmez; sınırı aşan eski bir proje çalışmaya, düzenlenmeye ve silinmeye devam eder, yalnız o sürücüye yeni ekleme yapılamaz.

Doğrulama: her değer, çalışan bir MergenHub'a karşı protokol simülatörleriyle sınırın kendisinde denendi: tüm tag'ler Good oldu, değerler kesintisiz güncellendi, sınırın bir fazlası reddedildi, silince tüm bağlantılar bırakıldı. Sınırlar sürücü başına olduğundan, tüm sürücüler **aynı anda** sınırındayken de denendi.

Tarama süresi Bir kanal cihazlarını **sırayla** tarar. Sınıra yakın kalabalık bir kanalda tarama turu uzar: testte tek kanalda 8192 Modbus TCP cihazı ~40 sn'de bir tazelandı (tag'in tarama hızı 1 sn olsa bile). Hızlı güncelleme gereken cihazları **birden çok kanala** bölün — kanallar birbirinden bağımsız ve paralel çalışır (1024 kanallı testlerde her değer ≤ 1 sn'de tazelandı).

İpucu Bazı sürücüler aynı uca giden bağlantıları birleştirir: IEC 104'te aynı IP:port'taki istasyonlar tek soketi paylaşır (ortak adres her istasyonda farklı olmalı), MQTT'de aynı broker'a giden kanallar tek bağlantı kullanır.

Ayrıca bkz [Kanallar](#) [Cihazlar](#) [Proje Taşıma](#)

Tag'ler

Bir **tag**, cihazdaki tek bir adrese bağlı değişkendir. Adres biçimi sürücüye özeldir (bkz. Protokoller).

Veri tipi	Boolean, Word, Short, DWord, Long, Float, Double, String...
Erişim	Read Only / Read-Write (endüstriyel standart: 0=RO, 1=RW).
Tarama hızı	Tag/grup bazlı; client-driven da olabilir.
Ölçekleme	Ham cihaz değeri → mühendislik değeri (bkz. Ölçekleme).

Toplu ekleme: tag'ler dizi olarak POST edilebilir (batch). Geçersiz tag atlanır, atlanma raporu tutulur (lenient import).

Tag Grupları

Tag'ler cihaz altında **gruplara** ("Tag Group") ayrılabilir; gruplar iç içe (nested) olabilir. Her grup OPC UA adres uzayında bir **klasör** olarak görünür: istemci `Kanal.Cihaz.Grup.Tag` yolundan erişir.

Amaç	Büyük cihazları mantıksal alt-kümelere böler (ör. "Motorlar", "Sıcaklıklar") — 1000'lerce tag'te gezinmeyi kolaylaştırır.
Parent Group	Otomatik tag üretiminde üretilen tag'lerin altına konacağı hedef grup.
Kopyala / taşı	Grup, içindeki tüm tag'lerle birlikte çoğaltılabilir.

İpucu Grup adları da OPC yoluna girdiğinden, istemci tarafında adres kararlılığı için grup adlarını erkenden belirleyin — sonradan yeniden adlandırma istemci yollarını kırar (bunun için **Aliases** kullanın).

Ölçekleme (Scaling)

Ölçekleme, ham cihaz değerini mühendislik birimine çevirir ("Scaling") ve bu tag'i özellik editöründeki **Scaling** grubunda ayarlanır.

Tip	Yok (ham), Lineer , Karekök (Square-Root — akış/basınç sensörleri).
Raw Lo/Hi	Cihazdan gelen ham aralık (ör. 0–65535 ya da 4–20 mA'nın ham karşılığı).

Scaled Lo/Hi	Mühendislik aralığı (ör. 0–100 m³/h).
Scaled Data Type / Clamp / Negate	Ölçekli tip, alt/üst sınırlama (clamp), işaret çevirme.

Lineer: $eng = (raw - rawLo) / (rawHi - rawLo) \times (engHi - engLo) + engLo$. Kapalıysa ham değer kullanılır. Ölçekleme yalnız **açık (abone/taranan)** tag'lerde uygulanır (performans için slot-bazlı).

İpucu 4–20 mA döngüsünde ham 0–65535 için: Raw Lo/Hi = sensörün 4 mA / 20 mA'daki ham değeri, Scaled Lo/Hi = mühendislik min/maks. Clamp ile aralık dışı gürültü kırılır.

Zamanlama (Timing)

Cihaz zamanlaması referans sunucuyla birebir:

Request Timeout	Tek istek için bekleme (ms). Aşılsa yeniden dener.
Attempts Before Timeout	Bağlantı ve istek denemelerine uygulanır.
Inter-Request Delay	İstekler arası nefes; yavaş cihazlar için.
Auto-Demotion / Failure Backoff	Ard arda hata veren cihazı geçici deviren geri-çekilme.

Tarama Modu: Eager vs Talep-Güdümlü

İki tarama davranışı vardır (Ayarlar → Tarama):

Eager (endüstriyel parite)	Tüm etkin kanallar açılışa bağlanır ve sürekli taranır — tüketici olmasa da. Kanallar "hep okur".
Talep-güdümlü	Cihaz yalnız bir tüketicisi (OPC abonesi / UI / Logger) varken taranır; boşta cihaz soket/CPU/RAM harcamaz.

Tüm tag'e abone olan bir OPC istemcisinde ikisi de kararlı durumda ~aynı sonucu verir; fark açılışa ve tüketicisiz durumdadır. `_System._DemandPoll` ve bekleyen yazma her iki modda cihazı uyandırır.

Protokoller (Sürücüler)

Adresleme — Genel

Her sürücünün kendi adres biçimi vardır. Geçersiz adres yazan tag reddedilir (referans sunucuyla aynı kurallar). Gerçek OPC sunucu JSON export'ları import edilebilir; sürücü adları ve adres biçimleri birebir korunur.

Modbus (TCP / RTU / ASCII)

Sürücüler: **Modbus TCP/IP Ethernet, Modbus RTU Serial, Modbus ASCII Serial.**

0xxxx	Coil (okuma/yazma bit) — F01/F05/F15
1xxxx	Discrete Input (salt-okunur bit) — F02
3xxxx	Input Register (salt-okunur word) — F04
4xxxx	Holding Register (okuma/yazma word) — F03/F06/F16

Adres = alan öneki + ofset (5–7 hane; 6-digít adreslemede 4000011 gibi 7 hane kabul edilir). Word içi bit (40001.3) ve string (40001.10H = 10 karakter, H=HiLo/L=LoHi) desteklenir. Register biti yazımı **FC22 Mask-Write** (atomik) kullanır. Byte order (First Word/DWord Low, Modicon bit sırası, Treat Longs as Decimals), blok boyutu (64/125 register), IP Protocol (TCP/UDP), Timeout'ta Soketi Kapat ve Illegal Address'te tag pasifleştirme ayarlanabilir. Çıplak IP cihaz kimliği ve unit'siz <IP> desteklenir.

İpucu Blok boyutu bazı cihazları resetler: 125-register blok yerine **64** kullanmak (yaygın endüstriyel varsayılan) uyumu artırır. Cihaz "Bad" veriyorsa blok boyutunu 64'e düşürün.

Sınır Modbus TCP 1024 kanal · kanal başına 8192 cihaz; RTU 1024 · 255; ASCII 100 · 247. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

Ayrıca bkz [Zamanlama](#) [Ölçekleme](#) [Sık Sorunlar](#)

Siemens S7

Sürücü: **Siemens S7 TCP/IP** (S7-300/400/1200/1500, ISO-on-TCP / RFC1006). Cihaz kimliğinde IP + **Rack/Slot** (1200/1500 için tipik Rack 0, Slot 1).

DBn, type offset	Data block. Ör. DB1, DBW20 (word), DB1, DBD8 (dword/real), DB1, DBX0.3 (bit)
M / MB / MW / MD	Merker (bit/byte/word/dword). Ör. M10.3 , MW14
I / E , Q / A	Giriş / çıkış. Ör. I0.0 , Q2.1
T / C	Timer / Counter

```
{ "common.ALLTYPES_NAME": "Sicaklik", "servermain.TAG_ADDRESS": "DB1,DBD8",  
  "servermain.TAG_DATA_TYPE": "Float" }
```

Sınır 1024 kanal · kanal başına 1024 cihaz. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

OMRON (FINS / NJ CIP)

İki sürücü:

OMRON FINS Ethernet	CS/CJ/CP serisi. Alanlar: CIO, W, H, D, A . Ör. D100, CI00.05 (bit), W20 . FINS SA1/DA1 ve FirstWordLow ayarları kanal/cihazda.
OMRON NJ Ethernet	NJ/NX serisi. EtherNet/IP CIP sembolik — PLC'deki sembol adıyla erişim (UTF-8, Türkçe ad destekli). Ör. adres = Pompa1.Debi .

Not FINS'te word/bit sırası cihaza göre değişebilir; değer ters/kaymış geliyorsa **FirstWordLow** ayarını değiştirin. Gerçek NJ/CJ PLC'lerle canlı doğrulandı.

Sınır FINS Ethernet 512 kanal · kanal başına 32 cihaz; NJ Ethernet 256 · 1024. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

DNP3

Sürücü: **DNP Client Ethernet**. Adres biçimi `grup.varyasyon.index ( g.v.i )`.

<code>1.x.i</code> · B <i>I</i> i	Binary Input (i = index)
<code>10.x.i</code> · B0 <i>i</i>	Binary Output (yazılabilir, kontrol)
<code>30.x.i</code> · A <i>I</i> i	Analog Input
<code>40.x.i</code> · A0 <i>i</i>	Analog Output (yazılabilir)
<code>DS0.1</code> · DA250	Data Set üyesi · Device Attribute (g0)

Kısa mnemonik (AI0) ve DNP3 `g.v.i` biçimleri birlikte kabul edilir. Cihaz ID yalnız **IP**'dir; master/outstation adresi + port cihaz/kanal özellikleridir.

Wire edilen endüstri-parite özellikleri: Operate Mode (Direct / SBO / No-Ack, operate STATUS doğrulamalı — sahte-Good yok), sınıf tarama (Class 0/1/2/3) + integrity, unsolicited (mode 1/2/3), zaman senkronizasyonu (UTC/DST/Honor/LAN·Serial), Event Playback (olay tamponu + oynatma), Data Sets (g85–88), Device Attributes (g0), File Control (g70), Secure Authentication (IEC 62351-5 / SAV5) ve yol yedekliliği (Redundancy). CRC/çerçeve doğrulaması dahili.

Sınır 1024 kanal · kanal başına 1024 cihaz. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

IEC 60870-5-104

BETA Bu sürücü henüz tam saha testinden geçmedi; üretimde dikkatli kullanın.

Sürücü: **IEC 60870-5-104 Client**. Adres biçimi `TYPE.IOA.attr` — TypeID + Information Object Address.

<code>M_SP.IOA</code>	Tek nokta bilgi (telesignal)
<code>M_ME.IOA</code>	Ölçülen değer (telemetry)
<code>C_SC.IOA</code>	Tek komut (telekumanda, yazma)

Cihaz ID: `IP:2404,commonAddress`. ASDU/APCI katmanı, k/w pencere, t1-t3 zamanlayıcıları dahili. Başlangıçta ve periyodik **Genel Sorgu (GI)** ile tüm noktalar okunur. **Komut/set-point yazımı** (`C_SC/C_DC/C_SE`) **ACT** → **ACT-CON** onay el sıkışmasını yürütür: cihaz olumsuz onay (COT P/N) dönerse tag Bad olur, sahte-Good yazılmaz.

Sınır 1024 kanal · kanal başına 1024 istasyon (aynı IP:port'taki istasyonlar tek soketi paylaşır; ortak adresler farklı olmalı). Ayrıntı: [Kanal ve Cihaz Sınırları](#).

IEC 61850 (MMS)

Sürücü: **IEC 61850 MMS Client**. Tam ISO yığını (MMS/ACSE/Presentation/Session/COTP/TPKT). Adres = nesne referans yolu:

`LD/LN.DO.DA` → örneğin `RELAY/MMXU1.TotW.mag.f` (aktif güç)
`RELAY/XCBR1.Pos.stVal` (kesici durumu)

Mantıksal cihaz (LD) → mantıksal düğüm (LN) → veri nesnesi (DO) → veri özneliği (DA). Cihaz ID: `IP:102`. Okuma LN/DO/DA seviyelerinde desteklenir. **Kontrol yazımı** tam kontrol modeliyle (orCat / orIdent / ctrlNum / Test / Check) **select**→**operate** el sıkışmasını yürütür ve IED'nin **AddCause** yanıtını doğrular — reddedilirse tag Bad döner (sahte-Good yazılmaz). SCL/ICD içe aktarma + AP-Title/TSEL seçici kimlik doğrulaması desteklenir.

Sınır 256 kanal · kanal başına 256 cihaz (IED). Ayrıntı: [Kanal ve Cihaz Sınırları](#).

MQTT Client

Sürücü: **MQTT Client** (3.1.1 / 5.0). Bir MQTT broker'a abone olup gelen topic/payload'ı tag olarak okur ("MQTT'yi Modbus gibi kullan"). **Yeni model (104 paritesi)**: broker bağlantısı (host/port, TLS 1883/8883, CA + istemci sertifikası, QoS, kullanıcı/parola, Client ID) **KANAL** seviyesindedir; aynı kanaldaki cihazlar tek paylaşılan broker soketini kullanır. **Cihaz** yalnız topic kümesidir — Cihaz ID opsiyoneldir (boş ya da etiket). Cihaz ayrıca canlılık (**Arrival Watchdog**: beklenen periyot × 1.5 → Uncertain, × 3 → Bad; reconnect grace) ve tag keşif ayarlarını taşır.

Adres = **topic + payload çözümü**. Tag özellik editöründe ham adres yerine **sihirbaz** çıkar (Okuma + ayrı Yazma alanları); adres dizesini arka planda o kurar:

Çıplak değer `values/debi` — payload doğrudan sayı/metin (45 / 45,7 / ON)

JSON alanı `telem:/flow/max` — / ile alt alanlara inilir; dizi: `/data/0/x`

Anahtar-değer	<code>raw;key=debimetre</code> — "debimetre: 45" metninden çeker
Ayraçlı (CSV)	<code>csv;idx=2;sep=;</code> — "10;20;30" → 30 · sep= TEK karakter: <code>:</code> <code>#</code> <code>tab</code> <code>space</code>
XML / Regex / İkili	<code>xml=data/flow</code> · <code>re=debi=([0-9.]+)</code> · <code>bin=f32le@0</code>
Joker	<code>sens/+ /temp</code> (tek seviye), <code>#</code> (çok seviye)
Yazma	<code>;w=cmd/sp</code> (farklı topic), <code>;wp=/set</code> (JSON'a göm), <code>;wr=1</code> (retain), <code>;wbool=1</code> (true/false)

Akıllı keşif: "Cihazdan Tag Üret" gelen payload'ı parçalar (JSON/XML/CSV/anahtar-değer) ve JSON-pointer'lı tag'ler önerir; derin/devasa payload'larda 500+ yaprak tag çıkarabilir.

Not MQTT ayrıca **kuzey yönünde** Veri Köprüsü'nde (Data Bridge) de agent olarak vardır — orada tag'leri broker'a *publish eder*; burada ise broker'dan *okur*.

Sınır 60 kanal (broker bağlantısı) · kanal başına 1024 cihaz. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

Simulator

Sürücü: **Simulator**. Donanımsız sentetik veri — test, demo ve SCADA/alarm doğrulaması için. Ağ/soket kullanmaz; cihaz ID'si IP değil düz numaradır (1–999).

<code>SINE:10:0:100</code>	Sinüs — periyot 10 sn, 0..100 arası
<code>RAMP:5:0:100</code>	Testere/rampa (0..100)
<code>SQUARE:2:0:1</code>	Kare dalga (Boolean)
<code>RANDOM:0:100</code>	Rastgele 0..100
<code>COUNTER</code>	Sürekli artan sayaç

Adres biçimi **FONKSİYON:periyot:min:max**. Simulator import'u ile uyumludur (R/K/B/S register'ları).

Sınır 1024 kanal · kanal başına 999 cihaz. Ayrıntı: [Kanal ve Cihaz Sınırları](#).

OPC UA Server

Endpoint & Bağlantı

OPC UA Server, tüm tag'leri `opc.tcp://host:48010/MergenHub` adresinde sunar (Ayarlar → OPC UA).

Host	<code>localhost</code> (yalnız yerel) veya <code>0.0.0.0</code> /makine adı (LAN).
Port / EndpointPath	Varsayılan 48010 / MergenHub.
Max. connections / session timeout	proje özellikleri paritesi (çok-oturulmuş istemciler için uzun timeout).

Güvenlik & Sertifikalar

Endpoint güvenlik policy'leri seçilebilir: **None**, **Basic256Sha256** (Sign / Sign&Encrypt) vb. Anonim veya kullanıcı adı/şifre token'ı. Güvenilmeyen istemci sertifikaları "rejected" listesine düşer; **Ayarlar** → **Certificates**'tan "trusted" yapılır. Sunucunun kendi instance sertifikası ilk açılışta üretilir.

_System Tag'leri

Cihaz ve sunucu istatistikleri `_System` / `_Statistics` tag'leri olarak OPC'de sunulur:

<code>_System._ActiveTagCount</code>	Aktif (abone/taranan) tag sayısı.
<code><cihaz>._System._DemandPoll</code>	Yazınca cihazı bir kez zorla taratır (write-triggered poll).
<code><cihaz>._System._Enabled</code>	Cihaz taraması aç/kapa (yaz).
<code>._Statistics._SuccessfulReads / _FailedReads / _SuccessfulWrites</code>	Sayaçlar.
<code>._Statistics._Error</code>	Cihaz hata durumu (0/1).
<code><cihaz>._System._Error / _NoError</code>	Cihazın haberleşme sağlığı . Yalnız "son işlem hatası" değil, gerçek veri akışı da hesaba katılır: bağlantı açık ama veri gelmiyorsa (zombi oturum) <code>_NoError=0</code> olur.

İpucu Bir istasyonun sağlığını izlemek için `_NoError` 'a abone olmak yeterlidir — abonelik cihaz için tarama talebi oluşturur, cihaz gerçekten yoklanır. Devre dışı ya da hiç tüketicisi olmayan (bilerek boşta) cihaz hata sayılmaz.

Not Bu sistem tag'leri de aboneliğe/monitored-item'a dahil olduğundan, OPC panelindeki abone/item sayısı ham cihaz tag sayısından birkaç fazla görünür — normaldir.

Desteklenen Servisler (profil)

MergenHub bir **Standard UA Server**'dir: **Data Access + Historical Access** (ham & aggregate) + **Method Call**. Aşağıdaki tablo neyin sunulduğunu ve neyin sunulmadığını gösterir.

Browse / BrowseNext / TranslateBrowsePath	Var — yetkiye göre süzülür.
Read / Write	Var — değer canlı TagStore'dan; yazma cihaz kuyruğuna gider.
Subscribe (MonitoredItem / Publish)	Var — veri değişimi push edilir.
HistoryRead	Var — Raw + Processed (Average/Minimum/Maximum/Count). Bkz. "Geçmiş Okuma".
Call (metot)	Var — <code>_Methods</code> klasörü. Bkz. "Sunucu Metotları".
HistoryUpdate (geçmiş yaz/sil)	Yok — geçmiş salt-okunurdur.
OPC UA Alarms & Conditions (event modeli)	Var — <code>AlarmConditionType</code> koşulları. Bkz. "Alarm & Events (A&E) Yayını".
OPC UA PubSub (UADP/JSON)	Yok — bulut/mesajlaşma için Veri Köprüsü (MQTT/REST).
AddNodes / DeleteNodes, Query	Yok — adres uzayı config'ten üretilir (REST Configuration API).
LDS'e kayıt, X509 kullanıcı token'i	Yok — doğrudan endpoint URL'i; UserName/Password + kanal sertifikası.

Ayrıntı Tam profil belgesi: `docs/opc-ua-server.md` .

Tag Bazlı Yetki (ACL)

Ayarlar → **OPC UA** → **Kullanıcılar** → **Yetki**'den kullanıcı (ve anonim oturum) başına kanal/cihaz/tag yolu üzerinde erişim **yok** · **okuma** · **okuma-yazma** verilir. Çözümleme **en uzun önek kazanır**.

Browse	Yetkisiz düğüm hiç listelenmez (erişilebilir tag'in ata klasörleri görünür kalır).
Read / Write / Abonelik	<code>BadUserAccessDenied</code> .
HistoryRead	<code>BadUserAccessDenied</code> — okuyamadığı tag'in geçmişini de okuyamaz .
Call (metot)	Hedef üzerinde okuma-yazma yoksa <code>BadUserAccessDenied</code> .

Hiçbir kullanıcıya yetki atanmamışsa kısıt yoktur (geriye uyum). Her red denetim günlüğüne düşer (spam koruması ile).

Geçmiş Okuma (Historical Access)

OPC UA istemcileri **geçmiş veriyi** doğrudan sunucudan okuyabilir. Kaynak **Data Logger**'dir — OPC için ayrı bir depolama kurulmaz.

Hangi tag?	Yalnız bir Data Logger grubuna eklenmiş tag'ler. Onların node'unda <code>Historizing = true</code> ve <code>HistoryRead</code> erişim biti bulunur; diğerleri <code>BadHistoryOperationUnsupported</code> döner.
Ham (raw)	<code>StartTime</code> / <code>EndTime</code> (bitiş hariç), <code>NumValuesPerNode</code> , <code>ReturnBounds</code> . <code>StartTime > EndTime</code> → yeniden eskiye.
Aggregate	Average · Minimum · Maximum · Count . Kova zamanı kova başlangıcıdır; boş kovada Count 0, diğerleri <code>BadNoData</code> .

Sayfalama	Çağrı başına değer sınırı Ayarlar → OPC UA 'dan (varsayılan 5000). Sınıra ulaşırsa devam noktası döner; istemci kalanı onunla alır.
Kalite / zaman	Kalite Good/Uncertain/Bad olarak; kaynak zaman damgası sürücünden gelen örnek zamanıdır.
Desteklenmeyen	ReadAtTime, ReadModified, event geçmiş, HistoryUpdate (geçmişe yazma/silme).

UaExpert ile: düğümü seçin (`Historizing` true olmalı) → **Document** → **Add** → **History Trend View** → aralığı seçin. Aggregate için görünümün **Configuration** bölümünden `Average` + `Processing Interval` girin.

Kapatma Ayarlar → **OPC UA** → **Geçmiş okuma** kapatılırsa bayraklar hiç konmaz ve HistoryRead desteklenmez (özellik öncesi davranış).

Ayrıca bkz [Data Logger](#)

Sunucu Metotları (Call)

Sunucu kökündeki `_Methods` klasöründe çağrılabilir metotlar bulunur. Metotlar cihaz/tag başına çoğaltılmaz — hedef, **argüman** olarak verilir (büyük projelerde adres uzayı maliyeti sabit kalır).

<code>ResetTotalizer(TagPath)</code>	Totalizer gelişmiş tag'inin birikimini sıfırlar (<code>_AdvancedTags.Ad</code> ya da yalnız <code>Ad</code>).
<code>AcknowledgeAlarm(AlarmId)</code>	Aktif alarmı onaylar — REST <code>alarms/active/{id}/ack</code> ile aynı işlem.
<code>ForceScan(DeviceKey)</code>	Cihazı zorla taratır — <code>._System._DemandPoll</code> yazmakla aynı (<code>Kanal.Cihaz</code>).
<code>ResetStatistics(DeviceKey)</code>	<code>_Statistics</code> sayaçlarını sıfırlar.

Hepsi (**Succeeded: Boolean, Message: String**) döner. İş sonucu başarısızlığı protokol hatası değildir: çağrı `Good` döner, neden `Message` 'da yazar.

Yetki + denetim Her çağrı, argümandan çözülen hedef üzerinde **okuma-yazma** yetkisi ister (yoksa `BadUserAccessDenied`) ve başarılı/başarısız fark etmeksizin **denetim günlüğüne** düşer. Hedef çözülemezse çağrı `BadInvalidArgument` alır — yetki atlanamaz.

Bağımlılığı olmayan metot (ör. alarm modülü yoksa `AcknowledgeAlarm`) **adres uzayına hiç konmaz** — `BadNotImplemented` dönen hayalet düğüm bulunmaz. **Ayarlar** → **OPC UA** → **Sunucu metotları** kapatılırsa `_Methods` klasörü de yaratılmaz.

Ayrıca bkz [Tag Bazlı Yetki Alarmlar](#)

Alarm & Events (A&E) Yayını

MergenHub'ın **ISA-18.2 alarm motoru** OPC UA'da `AlarmConditionType` koşulları olarak yayınlanır. **İkinci bir alarm sistemi yoktur**: tanım, durum ve onay tek kaynaktan (Alarmlar modülü) gelir — web arayüzünden yapılan onay UA abonelerine, UA'dan yapılan onay web arayüzüne yansır.

Ayar `OpcUa.PublishAlarms` (varsayılan **açık**). Kapatılırsa `_Alarms` düğümleri hiç kurulmaz, adres uzayı eski hâliyle kalır.

Adres uzayı	Objects → _Alarms → [Alan] → <alarm> . Namespace: http://envest/mergenhub/ua/alarms .
Abonelik	İstemci Server düğümüne olay aboneliği açar (_Alarms Server'a HasNotifier ile bağlıdır). Dilerse yalnız bir Alan klasörüne abone olur.
ConditionRefresh	Bağlanır bağlanmaz çağrılır → aktif/onay bekleyen alarmların anlık listesi gelir.
Onay	AcknowledgeableConditionType.Acknowledge → mevcut ISA-18.2 onayı; denetim günlüğüne düşer.

ISA-18.2 ↔ UA durum eşlemesi:

Durum	ActiveState	AckedState	Retain
Normal	false	true	false
Alarm (onaysız)	true	false	true
Alarm (onaylı)	true	true	true
Normale döndü, onaysız	false	false	true

Koşul tipi ConditionName alanında taşınır: HIHI / HI / LO / LOLO / ROC_HI / ROC_LO / DEV_HI / DEV_LO . Severity motorla **aynı 1–1000** ölçeğindedir. SourceName tag yolu, SourceNode ise gerçek tag düğümüdür. Veri kesilmişse (BAYAT alarm) Quality = UncertainLastUsableValue olur.

Yetkilendirme A&E de **tag yetkisine tabidir**: kaynak tag'i **okuyamayan** kullanıcıya o alarm hiç yayınlanmaz (abonelik ve ConditionRefresh süzülür, düğüm browse'da görünmez); **onay** için okuma-**yazma** gerekir, yoksa BadUserAccessDenied . Yetki değişikliği açık oturumlara **canlı** yansır.

Sınırlar Tek olay tipi (AlarmConditionType) sunulur; branch/Confirm/Shelving desteklenmez; olay geçmişi UA HistoryRead ile değil GET /alarms/v1/history ile okunur. Durum değişimleri 500 ms çözünürlükle örneklenir.

Ayrıca bkz [Alarmlar \(A&E\)](#) [Alarmların OPC UA'dan Okunması](#)

OPC UA İstemci

Gömülü QuickClient

Gömülü OPC UA istemcisi ile dış OPC UA sunucularına bağlanıp test/entegrasyon doğrulaması yapılır. Adımlar:

1. Keşfet	Sunucu URL'i gir (<code>opc.tcp://host:port/path</code>) → endpoint'leri listele.
2. Endpoint & güvenlik	Policy seç (None / Basic256Sha256), token (anonim / kullanıcı).
3. Bağlan & Browse	Adres uzayını gez, düğümleri incele.
4. Oku / Yaz / Abone	Değer oku, yaz, canlı abonelik kur.

İpucu Sign&Encrypt bağlantıda ilk denemede el sıkışma reddedilirse, karşı sunucunun sertifikasını **Ayarlar** → **Certificates**'tan "trusted" yapın (ve gerekiyorsa kendi istemci sertifikamızı karşı tarafta güvenli yapın).

Advanced Tags

Türetilmiş Tag & İfade Motoru

Advanced Tags, mevcut tag'lerden **hesaplanmış** yeni tag'ler üretir (Advanced Tags muadili) ve `_AdvancedTags` altında OPC'ye sunar.

Derived (ifade)	Matematik/mantık ifadesi: <code>{Tag1}*2 + {Tag2}</code> . Rate/trigger modları.
Average / Max / Min	Run-kapılı toplama.
Delta Sayaç (Oracle Cumulative)	Birikimli bir sayacın ardışık iki okuması arasındaki fark (biriken toplam değil). İlk okuma ham değerdir; sayaç başa dönünce <code>(Maksimum - önceki) + şimdiki</code> . Örn Maksimum=100 için okunan 5, 45, 95, 4 → 5, 40, 50, 9. Biriken toplam için Totalizer kullanın.
Link (write-back)	Bir tag'i başka cihaz tag'ine yazar (Input→Output, Dead Value, Trigger/Link Mode/Rate).

Link Modu: "onChange" (kaynak değişince yaz), "onTrigger" (tetik tag'i ile yaz), "rate" (periyodik yaz — Link Rate). İfade editörü tag renklendirme + çoklu tag seçimi sunar.

Ayrıca bkz [İfade Sözdizimi Tag'ler](#)

İfade Sözdizimi & Örnekler

Derived tag ifadelerinde tag'ler `{Kanal.Cihaz.Tag}` ile referanslanır. Desteklenen operatör/fonksiyonlar:

Aritmetik	<code>+</code> <code>-</code> <code>*</code> <code>/</code> <code>%</code> ve parantez
Karşılaştırma / mantık	<code>></code> <code><</code> <code>>=</code> <code><=</code> <code>==</code> <code>!=</code> , <code>&&</code> <code> </code> <code>!</code>
Fonksiyonlar	<code>abs</code> , <code>min</code> , <code>max</code> , <code>sqr</code> , <code>pow</code> , <code>round</code> , <code>floor</code> , <code>ceil</code> vb.
Koşul	Üçlü: <code>{a} > 10 ? 1 : 0</code>

ÖRNEK İFADELER

```
// İki debinin toplamı (m³/h)
{Modbus_1.1001.Debi_A} + {Modbus_1.1001.Debi_B}

// Yüzde doluluk
({Depo.1002.Seviye} / 5.0) * 100

// Yüksek-basınç alarm bayrağı (1/0)
{Kuyu.1003.Basinc} > 8 ? 1 : 0
```

İpucu Link (write-back) ile bir hesap sonucunu başka bir cihazın setpoint tag'ine yazabilirsiniz: Input = ifade/kaynak tag, Output = hedef cihaz tag'i, Link Mode = onChange, Dead Value ile gürültü eşiği.

Veri Köprüsü (Data Bridge)

Agent'lar

Veri Köprüsü (Data Bridge), tag değerlerini kuzey yönünde dış sistemlere köprüler. Ayrı Windows servisi (`MergenHubDataBridge`) olarak çalışır; çekirdek REST'ten beslenir (X-MergenHub-Token). Her şey bir **agent**'tır — varsayılan açık hiçbir şey yoktur, REST Server dahil.

REST Server	Gelen bağlantı: dış sistem <code>/iotgateway/browse read write</code> uçlarından tag okur/yazar (endüstriyel standart JSON <code>{id,s,r,v,q,t}</code>). Kapalıysa uç 404'tür.
MQTT Client	Giden: seçili tag'leri bir broker'a publish eder (<code>[{id,v,q,t}]</code>); write-topic ile çift yönlü. Saha portu açmadan bulut entegrasyonu.
REST Client	Giden: tag değerlerini bir REST uca gönderir (POST/PUT/PATCH; Basic / Bearer / API-Key kimlik doğrulama; periyot ya da değişince). Test isteği gönder ile kaydetmeden denenebilir.

Yayın tetiği: periyot (`publishRateMs`) veya yalnız değişince (`publishOnChangeOnly`). Bağlantı durumu (bağlı/yayınlanan/son hata) her agent'ın Durum sekmesinde canlı izlenir.

Not API route'ları (`/iotgateway`), iç exe adı ve ayar anahtarları teknik uyumluluk için **IoT Gateway** kimliğini korur; Windows servis adı `MergenHubDataBridge` , ürün adı ise **Veri Köprüsü / Data Bridge**'dir.

Ayrıca bkz [Agent Örnekleri](#)

Agent Örnekleri & Uçlar

REST SERVER — OKUMA/YAZMA

```
# Değer oku (id = tag yolu)
GET /iotgateway/read?ids=Modbus_1.1001.Debi
→ [{"id":"Modbus_1.1001.Debi","v":42.5,"q":true,"t":1712345678}]

# Değer yaz
POST /iotgateway/write
[{"id":"Modbus_1.1001.Setpoint","v":50}]
```

MQTT CLIENT AGENT (YAPILANDIRMA)

```
{
  "name": "Broker1", "type": "MqttClient", "enabled": true,
  "brokerHost": "10.0.0.20", "brokerPort": 1883,
  "topic": "saha/veri", "qos": 1,
  "publishRateMs": 1000, "publishOnChangeOnly": true,
  "tags": ["Modbus_1.1001.Debi", "Modbus_1.1001.Basinc"],
  "writeTopic": "saha/yaz"
}
```

REST CLIENT AGENT – GÖNDERİLEN ZARF & KİMLİK DOĞRULAMA

```
PATCH https://sunucu.example.com/api/veri
Content-Type: application/json
X-API-Key: ***** ← Auth: API Key (başlık adı seçilebilir)

{"timestamp": 1723094400000,
 "values": [
   {"id": "Kanal.Cihaz.Debi", "v": 42.5, "q": true, "t": 1723094399870}
 ]}
```

Kimlik doğrulama türleri: **Basic** (kullanıcı+parola), **Bearer** (token), **API Key** (özel başlık) — başlığı elle yazmak gerekmez; ek başlıklar ayrıca satır satır girilebilir. Formdaki **"Nasıl çalışır? + örnek"** bölümü, girdiğiniz değerlerle eşdeğer `curl` komutunu üretir.

İpucu Test isteği gönder, ayarları KAYDETMEYEN gerçek zarfla hedefe tek istek atar; HTTP durum kodu, süre ve yanıt gövdesi ekranda görünür — entegrasyonu dakikalar içinde doğrularsınız.

Not Yayın payload'ı `[{"id", "v", "q", "t"}]` biçimindedir. `writeTopic` boşsa yazma kapalıdır. TLS için 8883 portu + CA/istemci sertifikası.

Historian

Veri Kaydı

Seçili tag'leri bir veritabanına tarihsel olarak kaydeder (Historian). Aynı Windows servisi (`MergerHubHistorian`).

Motorlar	SQLite (yerleşik), MySQL, SQL Server (native) ve ODBC (DSN + tablo).
Store-and-forward	DB erişilemezse yerelde tampon, sonra iletir.
Downsample	10 yıl ölçeğinde seyreltme; trend/CSV dışı aktarım.

Ayrıca bkz [Grup & Sorgu](#)

Log Grubu, Sink & Sorgu

Bir **log grubu** = tag listesi + hedef (sink) + tablo. Tetik: periyodik veya değişince.

Sqlite	Yerleşik dosya DB (kurulum gerektirmez). Varsayılan.
MySQL / SqlServer	Native bağlantı (host, DB, kullanıcı/şifre, tablo).
Odbc	DSN + tablo (herhangi bir ODBC kaynağı).

ÖRNEK – LOG GRUBU

```
POST /config/v1/logger/groups
{
  "name": "Debi Kayıt", "sink": "Sqlite", "tableName": "debi_log",
  "tags": ["Modbus_1.1001.Debi"], "triggerMode": "OnChange"
}
```

GEÇMİŞ SORGU (TREND)

```
GET /config/v1/logger/query?tag=Modbus_1.1001.Debi&from=1712300000000&to=1712345678000
```

İpucu DB erişilemezse **store-and-forward** yerelde tamponlar; bağlantı gelince otomatik iletilir — veri kaybı olmaz.

Alarmlar (A&E)

Alarm & Event (ISA-18.2)

Alarms & Events modülü ISA-18.2 durum makinesini uygular: **raise / clear / acknowledge**, deadband, geçmiş (history).

Koşullar	HiHi / Hi / Lo / LoLo, RoC (değişim hızı), Deviation.
Alan → Kaynak → Koşul	A&E hiyerarşisi.
E-posta	SMTP ile alarma alıcı bildirimi (Ayarlar → E-posta).

Ayrıca bkz [Koşul Tipleri](#)

Koşul Tipleri & Deadband

Alan → Kaynak → Koşul hiyerarşisi. Bir kaynak (tag) üzerinde birden çok koşul tanımlanır:

HiHi / Hi	Değer üst limitleri aşınca (kritik / uyarı).
Lo / LoLo	Değer alt limitlerin altına inince.
RoC (Rate of Change)	Değişim hızı eşiği aşınca (ani sıçrama).
Deviation	Bir hedeften sapma bandı aşıncı.

Deadband: alarmin "titremesini" (chatter) önler — değer eşiğin deadband kadar altına dönmeden alarm temizlenmez. Durum makinesi: **Normal** → **Active(Unacked)** → **Active(Acked)** → **Normal**.

İpucu Gürültülü analog sinyallerde deadband'i ölçüm gürültüsünün biraz üstünde tutun; aksi halde alarm sürekli raise/clear yapıp geçmişini doldurur.

Alarmların OPC UA'dan Okunması

Alarmlar arayüz ve REST dışında **OPC UA** üzerinden de sunulur (SCADA/DCS istemcileri standart A&E arayüzüyle bağlanır). Aynı motor, ikinci bir alarm sistemi **yoktur**.

1. Bağlan	<code>opc.tcp://<host>:48010/MergenHub</code>
2. Abone ol	<code>Server</code> düğümüne olay aboneliği (EventNotifier). İstersen yalnız <code>_Alarms</code> → <code><Alan></code> klasörüne.
3. Anlık liste	<code>ConditionRefresh</code> → aktif/onay bekleyen tüm alarmlar.
4. Onayla	Koşul üzerinde <code>Acknowledge(EventId, Comment)</code> .

Alan (Area) klasör olarak, koşul tipi `ConditionName` (`HIHI/HI/LO/LOLO/ROC_HI/ROC_LO/DEV_HI/DEV_LO`) olarak, şiddet `Severity` (1-1000) olarak taşınır.

Yetki Kaynak tag'ini okuma yetkisi olmayan OPC kullanıcısına alarm **gönderilmez**; onay için okuma-yazma gerekir. Kapatmak için `OpcUa.PublishAlarms = false` .

Ayrıca bkz [OPC UA – A&E Yayını](#)

Aliases

Takma Adlar

Aliases, uzun/teknik tag yollarına kısa, anlamlı **takma adlar** verir ve `_Aliases` altında OPC'ye sunar. İstemciler cihaz yolunu bilmeden alias üzerinden erişir; saha yeniden adreslense bile istemci adresi değişmez (soyutlama katmanı).

Gerçek yol

Modbus_1.1001.XA_Deger12

Alias

Kuyu1_Debi → istemci `_Aliases.Kuyu1_Debi` okur

İpucu SCADA/rapor istemcilerini alias'lara bağlayın; cihaz/kanal adı değişse bile yalnız alias eşlemesini güncellemeniz yeter, tüm istemciler çalışmaya devam eder.

Proje Taşıma (Dışa/İçe Aktarma)

Ne Taşınır

Proje **tek bir dosyada**, MergenHub'ın kendi paket formatında dışa/İçe aktarılır. Amaç: yedekleme, başka makineye taşıma, saha şablonu paylaşımı ve WinCC/Kepware'den geçiş.

Bağlantılar → **Dışa Aktar** iki kapsam sunar:

Tam proje paketi	Kanal · cihaz · tag · tag grubu, Advanced Tags , Aliases , modüller (Historian grupları, alarm tanımları, Veri Köprüsü ajanları, SCADA ekran/bölge/sembolleri) ve taşınabilir sunucu ayarları. <i>Önerilen.</i>
Yalnız yapılandırma	Kanal/cihaz/tag + Advanced Tags + Aliases. Kepware JSON ile uyumlu.

Tek bir **kanal** ya da **cihaz** ağaçtan sağ tıkla ayrıca dışa aktarılabilir (*Dışa Aktar (JSON)*).

Sürücü sınırları içe aktarmada da geçerlidir: içe aktarma sonunda bir sürücünün kanal sayısı ya da bir kanalın cihaz sayısı **sınırı** aşacaksa içe aktarma bütünüyle reddedilir ve mevcut proje hiç değişmez (kısmi içe aktarma yapılmaz).

Format kimliği	"format": "mergenhub-project"
Format sürümü	"version": 2 — okunamayan sürümde net hata verilir
Uygulama sürümü	"app" , "exportedAt" , "exportedBy"
Kapsam	"scope" : project / channel / device

Hassas Veri Politikası

Kural Parolalar, API anahtarları, token'lar ve bağlantı dizelerindeki kimlik bilgileri **VARSAYILAN OLARAK DIŞA AKTARILMAZ**.

Kapsanan alanlar: sürücü parolaları (`mqtt.PASSWORD` , `opcuaclient..._PASSWORD` , `iec_61850_mms_client.DEVICE_AUTHENTICATION_PASSWORD` ...), Historian DB parolası ve ODBC dizesindeki `PWD=` , SMTP/SMS parolaları, Veri Köprüsü ajan parolaları ve OPC UA kullanıcı parolaları. Kural **anahtar adı** üzerinden çalışır (`PASSWORD` , `SECRET` , `APIKEY` , `TOKEN` , `PRIVATE_KEY` içeren her alan) — sonradan eklenen sürücü alanları da otomatik kapsanır.

Paket `"redactedFields"` listesiyle **hangi alanların çıkarıldığını** bildirir; hedef sistemde yalnız onları elle girmeniz yeterlidir. Sunucu adresi, kullanıcı adı, port gibi bağlantıyı kurmak için gereken bilgiler korunur.

Kullanıcı açıkça **"Kimlik bilgilerini de dahil et"** derse parolalar düz metin olarak yazılır; paket `"includesSecrets": true` damgası taşır ve işlem **denetim günlüğüne** ayrıca kaydedilir.

Hiçbir koşulda taşınmayanlar Sertifika ÖZEL ANAHTARLARI (OPC UA / web PKI) — makineye bağlıdır, hedefte yeniden üretilir. Kullanıcı hesapları ve parola hash'leri — kimlik yönetimi hedef sistemin sorumluluğudur. MCP/iç servis token'ları.

Hangi Ayarlar Taşınır

Taşınır OPC UA sunucusu (endpoint'ler, güvenlik politikaları, kullanıcı listesi — parolasız, oturum limitleri) · Veri Köprüsü ajanları · e-posta/SMS sunucu ayarları · mesaj şablonları · tarama/izleme/saklama parametreleri · markalama (logo, şirket adı) · SCADA harita döşeme URL'i · Windows servisi başlangıç tercihi

Taşınmaz `api` bölümü: web dinleme URL'i, HTTPS portu, sertifika bağlaması

Neden `api` taşınmaz Dinleme adresi makineye özgüdür. İçer aktarılırsa hedef sunucunun web dinleyicisi bozulur ve yönetici kendi arayüzünden **dışarı kilitlenir**. Migrasyonun en sık ve en tehlikeli hatası budur; bu yüzden bilerek kapsam dışıdır.

Ayarların içeri aktarılması **opsiyoneldir** — ön izleme ekranında ayrı bir kutuyla seçilir, varsayılan kapalıdır.

İçer Aktarma: Birleştir / Değiştir

İçer aktarma iki adımlıdır. Önce dosya **yüklenir ve taranır** (akış tabanlı — 450.000 tag'lik dosya belleğe alınmaz); bu adımda **hiçbir şey değişmez**. Ön izleme ekranı ne geleceğini (kaç kanal/cihaz/tag/gelişmiş tag/modül) ve **hangi adların çakıştığını** gösterir. Kip ve çakışma davranışı seçildikten sonra uygulanır.

Birleştir (varsayılan) Mevcut proje **korunur**; gelen yeni kanallar eklenir. Proje adı değişmez.

Değiştir Mevcut proje **tümüyle silinir**. Geri alınamaz; ayrı bir onay kutusu işaretlenmeden çalışmaz.

Ad çakışmasında üç davranış vardır — toplu seçilir, istenirse öge bazında değiştirilir:

Atla (varsayılan) Mevcut öge korunur, gelen atlanır.

Yeniden adlandır Gelen öge `Ad_1` olarak eklenir; ikisi de durur. Gelen Advanced Tag ifadeleri, alias'lar, Historian öğeleri ve alarm tag yolları yeni ada göre **otomatik düzeltilir**.

Üzerine yaz Mevcut öge silinir, gelen yazılır.

İşlem bitince **eklenen / üzerine yazılan / yeniden adlandırılan / atlanan** sayıları ve öge öge sonuç listesi gösterilir; aynı özet **denetim günlüğüne** yazılır.

Atomiktir Doğrulama hatasında hiçbir değişiklik uygulanmaz — yarım proje kalmaz. Bozuk adresli tag'ler (toplu içer aktarmada) atlanır ve raporlanır.

Tek cihaz aktarma Bir **cihaz** dosyası proje içe-aktarma ucuna verilirse net bir hatayla reddedilir (proje boşaltılmaz). Cihazlar hedef **kanalın** sağ-tık menüsündeki *İçe Aktar (JSON)* ile aktarılır; orada da atla/yeniden adlandır/üzerine yaz sorulur.

SCADA / HMI

Ekran Editörü

Sürükle-bırak görsel ekran editörü (WinCC benzeri endüstriyel HMI tarzı). **Tasarım** ve **Çalıştır** modları vardır.

Tuval	World-space SVG + ekran-uzayı overlay; zoom/pan, hizalama kılavuzları, eşit dağıt, Undo/Redo.
Snap / Grid	Izgara ve nesne yapışması; Ctrl basılıyken geçici serbest hareket.
Çoklu seçim	Taşı / resize / döndür (grup).
Ekranlar	Alt-ekranlar (üst tab şeridi), export/import.

FARE & KLAVYE

Ctrl + tekerlek	İmleç etrafında yakınlaş/uzaklaş
Boşluk / Alt + sürükle	Tuvalde gez (pan)
Ctrl (sürüklerken)	Geçici serbest hareket (snap kapalı)
Ctrl+Z / Ctrl+Y	Geri al / yinele
"Fit" / "100%"	Sayfayı görünür alana ortalama + sığdır

ÇİZİM (SVG) İÇİNDE SEÇİM – ILLUSTRATOR AKIŞI

Çift tık (çizime)	Çizimin içine girer; imlecin altındaki en üst grup seçilir.
Tek tık	O seviyedeki objeyi seçer (turuncu çerçeve + ad etiketi).
Çift tık (gruba)	Grubun bir kat içine girer; üstte yol çubuğu görünür (ör. <i>normal-hat > baglac</i>).
Dışarı tık	Tıklanan obje görünene kadar kat kat yukarı çıkar ve onu seçer.
Esc	Bir kat yukarı; en üstteyse seçimden çıkar.
Sağ tık (parçaya)	Parça menüsü: animasyon ekle, öne/arkaya al, renk, boyut/konum, üst gruba çık.

Not Fit tasarlanan ekran sayfasının tamamını viewport'a ortalayarak sığdırır — çalıştır (runtime) modunda operatör ekranı her çözünürlükte düzgün görünür.

Ayrıca bkz [Widget'lar Çalıştır Modu](#)

Widget'lar

Yerleşik widget'lar (varyantlı):

Temel	Etiket, dikdörtgen, gösterge, kadran, grafik, trend-log, lamba, buton, switch, giriş, ayar-alanı.
-------	---

Endüstriyel	Tank/Depo, 3B Depo, Pompa, Vana, Boru (düz / yuvarlak 3B / dirsek  ).
Harita	Harita/İstasyon (tile + istasyon pini, canlı renk/nav/popup).
Bağlantı	Pipe Link (iki nesneyi ortogonal, köşe-sabit boru ile bağlar).

Dirsek boru: köşe sabit kalır, kollar kutuyla uzar — bozulmadan yol çizme.

SVG Objeler (inline)

İçe aktarılan/kütüphane SVG'leri **inline vektör** olarak çizilir (raster değil): her zoom/döndürmede keskin kalır. Kusursuz davranış için SVG:

- `viewBox` içermeli; sabit `width/height` olmamalı.
- Animasyona bağlanacak parçalar `id` / `class` almalı (WinCC "SVG dynamization" — ör. dolum `id="fill"` , iğne `id="needle"`).
- İçinde gömülü raster (PNG) olmamalı; Illustrator/Inkscape "Optimized/Plain SVG".

Format standardı **SVG**'dir (WinCC Unified de native SVG kullanır).

Uyarı PNG/JPG (raster) semboller ölçeklenince bulanır — kaçının. Vektör olmayan görsel yalnız statik arka plan için uygundur.

Ayrıca bkz [Widget'lar](#) [Animasyon](#)

Animasyon & Tag Binding

Her widget bir **tag**'e bağlanır; canlı değere göre animasyon uygulanır. Bir widget'a birden çok animasyon eklenebilir (koşullu).

Renk	Değer eşiğine göre dolgu/çizgi rengi (ör. >8 kırmızı).
Görünürlük	Değere göre göster/gizle (ör. arıza lambası).
Seviye	Tank/depo dolum yüksekliği (min–max ölçekli).
Akış	Boru akış animasyonu — değer $\neq 0$ iken akar.
Döndürme	Pompa/motor rotoru döndürme (çalışırken).
Metin / değer	Sayısal değeri birim + ondalıkla gösterme.

ANİMASYON LİSTESİ (⚡)

Araç çubuğundaki ⚡ düğmesi ekrandaki **tüm** animasyonları tek panelde listeler: kullanıcı kuralları, basit widget animasyonları ve **şablon bağlamaları** ("şablon" rozeti — Koru1000 altlığının su seviyesi/akış/durum rengi bağları). Satıra tıklayınca obje tuvalde **ortalanır ve parlar**; üzerine gelince hedef parça vurgulanır; kullanıcı kuralları satırdan silinebilir.

Önizleme (▶): tasarımda tüm kuralları koşulsuz oynatır — Çalıştır'a geçmeden animasyonları görürsünüz (kaydedilen ekranı değiştirmez).

İpucu Tag alanlarında X seçimi tek tıkla kaldırır; uzun tag yolları kırpılsa da üzerine gelince tam yol görünür.

İpucu Setpoint yazımı "yaz-onayla" ile teyit edilir: girilen değer cihaza yazılır, geri okunur; okunan = yazılan ise ✓, değilse X gösterilir — operatör yazmanın gerçekten geçtiğini görür.

Ayrıca bkz [SVG Objeler](#) [İfade Sözdizimi](#)

Makro — kendi HTML objen + sistem SDK'sı

Makro, ekrana koyduğunuz kendi **HTML/CSS/JS** objenizdir: hazır widget'ların yetmediği yerde (tesise özgü tablo, özel gösterge, küçük hesap aracı, sistem panosu) kullanılır. Paletten **Makro** ile eklenir; sağdaki özellik panelinden düzenlenir.

Kum havuzunda (sandbox iframe) çalışır: uygulamanın sayfasına erişemez, **ağı kapalıdır** (CDN betiği, uzak yazı tipi/görsel yüklenmez — her şeyi gömülü yazın). Veriye iki kapıdan erişir: **data-mh-*** (tag değerleri) ve **MH.query** (sistem verisi).

1) DEĞER BAĞLAMA — JS YAZMADAN

Değeri gösterecek element bağıni kendisi taşır, motor doldurur:

```
<b data-mh-bind="Anlık güç" data-mh-tag="Kanal1.Pompa1.Guc" data-mh-unit="kW" data-mh-dec="1">--</b>
```

data-mh-kind	value (varsayılan) · state (on/off metni) · level (dolgu yüksekliği) · color (arka plan)
data-mh-unit / -dec	Birim ve ondalık basamak
data-mh-on / -off / -threshold	state ve color için metin/reng ve eşik
data-mh-min / -max	level için ölçek
data-mh-bind	Panelde görünecek ad — Bağlı elementler listesinde tag'i buradan değiştirirsiniz

Not Kalite gösterimi **otomatiktir**: kalitesi Good olmayan değer soluk + kesik çerçeve olur. Donmuş veriyi canlı sanmazsınız.

2) SİSTEM VERİSİ — MH.QUERY / MH.POLL

Alarm listesi, olay ve denetim günlüğü, sistem kaynakları, bağlı OPC UA istemcileri, cihaz durumu, bir cihazın okuma/yazma logu... hepsi makrodan okunabilir. Makro ağa çıkmaz: soruyu uygulama sorar, **sizin kendi oturumunuzla** — yani makro sizin göremediğinizi göremez. Bu makrolar **tag istemez**.

alarms · alarmDefs · alarmHistory	Aktif alarmlar, tanımlar, geçmiş
events · audit	Sistem olay günlüğü · kim ne yaptı (denetim)
metrics · status · subsystems · services	CPU/RAM/disk, tag ve OPC yükü, servis durumları

devices · diagnostics	Cihaz haberleşmesi · bir cihazın okuma/yazma trafiği
opcuaClients	Bağlı OPC UA istemcileri + abonelik/izlenen tag sayısı
tags	Tag arama: anlık değer + kalite
agents · loggerStatus · loggerGroups · advancedTags · license	Veri köprüsü ajanları, veri kaydı, sanal tag'ler, lisans

```
MH.poll('alarms', {}, 5000, function(list){
  govde.innerHTML = list.map(function(a){
    return '<tr><td>' + MH.esc(a.name) + '</td><td>' + MH.fmt(a.value,1) + '</td>'
      + '<td>' + (a.acked ? 'onaylı' : 'BEKLİYOR') + '</td></tr>';
  }).join('');
});
```

MH.poll sorguyu periyodik tekrarlar (ilk çağrı hemen; makro görünmezken sormaz). Yardımcılar: MH.fmt , MH.saate , MH.tarih ve MH.esc (sistem metnini HTML'e basmadan önce kaçırın). Yazma tek yoldan: MH.write(tag, değer) — uygulamanın yetki/onay ve denetim yolundan geçer.

3) ELEMENT MODU – İÇİNDEKİ PARÇALARI TEK TEK DÜZENLE

Makro dışarıdan tek bir kutudur; içindeki kartı/başlığı/değeri düzenlemek için **element moduna** girin: makroya **çift tıklayın** (sembollerdeki parça moduyla aynı el alışkanlığı) ya da panelde **Elementleri seç**. Modda üstte mavi bir şerit belirir.

Gezinme	Fareyi gezdirdiğiniz parça çerçevelenir ve adı görünür (bağlı olanlarda % işaretli).
Seçme	Tıklanan parça sağdaki panelde açılır. Yol çubuğu (kart › satır › b) tıklanabilir: kapsayıcıyı seçmek için üst adı tıklayın.
Veri	Seçili parçaya tag verin; gösterim (değer/durum/doluluk/reng), birim, ondalık buradan. JS gerekmez.
Görünüm	Yazı ve zemin rengi, punto, kalın, hizalama, iç boşluk, satırdaki genişlik payı.
Düzen	Yukarı/aşağı taşı, kopyala, sil; içine ya da altına hazır blok ekleyin.
Çıkış	Esc . Moddayken tıklamalar makronun içine gittiği için obje taşınmaz — çıkınca normale döner.

TUVAL ÜZERİNDE DÜZENLEME

Seçtiğiniz parçanın **üstünde küçük bir araç çubuğu** belirir — panele gitmeden orada düzenlersiniz:

A	Metni yerinde düzenle (parçaya çift tıklamak da aynı işi yapar; Enter kaydeder, Esc vazgeçer)
B · A- / A+	Kalın; yazıyı küçült / büyüt
^ · v · = · x	Yukarı/aşağı taşı, kopyala, sil
tag	Sağ paneldeki tag alanına atlar

Not Araç çubuğundaki işlemler de kaynağa aynı yoldan yazılır — panelden yapmışsınız gibi. **Ctrl+Z** hepsini geri alır.

YERLEŞİM – MAKRO KUTUSUNU TANIR

Makro kutusu sabit boyutludur; içerik uzarsa alttaki parçalar **görünmez** olurdu. Panelden **Yerleşim** kipini seçin:

Kutuya sığdır (yeni makrolarda varsayılan)	İçerik ölçülür ve kutuya tam oturacak şekilde ölçeklenir — hiç kırılmaz. Diğer SCADA objeleriyle aynı davranış; ekran Sığdır ile açıldığında da bozulmaz.
Akış	Klasik davranış: içerik akar, kutudan taşarsa kırılır (eski makrolar bu kipte kalır).
Dikey doldur	Kök parçalar kutunun yüksekliğini eşit paylaşır — kart/satır düzenleri için.
Kutu içeriği takip etsin	Kutu içeriğe göre büyür; gerekirse ekran sayfası da uzatılır (obje sayfanın altında kalıp kaybolmaz).

Ayrıca yeni eklenen blok **görünümüne kaydırılır ve kısa süre parlar**; "Akış" kipinde içerik taşarsa panelde **"İçerik kutudan taşıyor — kutuyu sığdır"** düğmesi belirir.

Not Ölçekleme kaynağınıza dokunmaz: HTML'iniz aynı kalır, motor yalnız görüntüyü kutuya oturtur.

Bloklar (Satır, Kolon, Kart, Değer, Durum rozeti, Başlık, Yazı, Tablo) makroyu parçalardan kurmanızı sağlar: seçili parçanın içine/altına ya da panelden en sona eklenir, stilleri bir kez otomatik gelir.

Not Değişiklikler doğrudan HTML metnine yazılır — kodunuz yeniden biçimlendirilmez, yalnız dokunduğunuz yer değişir. Yanlış bir şey olursa **Ctrl+Z** geri alır.

İpucu Kalıp düğmeleri artık varsayılan olarak **mevcut içeriğin altına ekler** (emeğiniz silinmez). "Kalıp uygulananca" seçeneğinden *Hep değiştir* yapabilirsiniz.

4) KOD DÜZENLEYİCİ

Renklendirme	HTML / CSS / JS ayrı ayrı, satır numaralı
Genişlet (↗)	Aynı düzenleyiciyi tam ekran açar
Otomatik tamamlama	<code>MH.query(' → veri listesi · sonucu değişkene alıp . → o tipin alanları · data-mh-tag=" → gerçek tag araması · data-mh- → öznitelikler. Ctrl+Boşluk ile de açılır.</code>
SDK ve örnekler	Her veri tipinin parametreleri, dönen alanları ve çalışır örneği; JS'e ekle ile doğrudan koda girer
Kalıplar	Değer: KPI kartı, değer listesi, durum rozeti, doluluk çubuğu — Sistem: alarm listesi, olay günlüğü, sistem kaynakları, OPC UA istemcileri, cihaz durumu, okuma/yazma logu

İpucu Renk sabitlemeyin: `var(--mh-primary)`, `--mh-card`, `--mh-border`, `--mh-success`, `--mh-warning`, `--mh-danger` temaya uyar; `MH.theme()` gece/gündüz döner.

Uyarı Kod hata verirse makronun altında **kırmızı kutuda** görünür (yakalanmamış Promise reddi dahil). Tag değerleri yalnız **Çalıştır** modunda akar; tasarımda örnek veri gösterilir ve köşede "ÖNİZLEME VERİSİ" etiketi durur.

Çalıştır Modu (Runtime)

Çalıştır modu temiz operatör görünümüdür: üst barlar gizlenir, tuval tam ekran, kontroller sağ üstte yüzen panelde.

Fit	Tasarlanan ekranın tamamını viewport'a sığdırır ve ortalı (oran korunur). Önerilen.
Stretch	Ekranı viewport'u tamamen dolduracak şekilde gerer (oran korunmaz).
1:1	Gerçek boyut; tekerlek/pan ile gezilir.

Widget tıklama aksiyonları: ekran değiştir, faceplate (yüzen pencerede tasarlanan ekranı canlı aç). Alt sekmeler: Alarmlar, Grafik (loglanan tag geçmişi).

İpucu Ekran mobilde/farklı çözünürlükte de düzgün görünsün diye **Fit** kullanın; tasarım sayfası boyutunu içeriğinize göre ayarlarsanız boşluk kalmaz.

Koru1000 Ekran Şablonları

Şablonla Dakikalar İçinde İstasyon

SCADA'da **Yeni Ekran** → **Koru1000 Ekran Şablonları**: hazır istasyon paketleri (Kuyu, Depo, Terfi, DMA, BKV — CP110 / AquaCnt-V2 / Omron varyantlarıyla). Şablonu seçip **istasyon adı + IP** girdiğinizde kurulum sizin yerinize her şeyi yapar:

Kanal + Cihaz	Doğru sürücü ve haberleşme ayarlarıyla oluşturulur.
Tag havuzu	Ürüne göre 113–560 hazır tag, tek istekte (adres/ölçek/tip hazır).
Hesaplanan tag'ler	Verim, maliyet, doluluk gibi türetilmiş değerler (Advanced Tags).
Çalışan ekran(lar)	Ana ekran + ayar alt ekranları; su seviyesi, akış, durum renkleri CANLI bağlı.

KURULUM KİPLERİ

Tam kurulum	Kanal + cihaz + tag + ekran (yeni istasyon için).
Mevcut cihaza ekle	Tag havuzunu VAR OLAN cihazınıza ekler (çakışan adlar atlanır), ekranı ona bağlar.
Yalnız ekran	Cihaz oluşturmaz; ekrandaki veri noktalarını kendi tag'lerinize sihirbazda eşlersiniz.

İpucu Şablonla gelen ekran normal, tamamen düzenlenebilir bir SCADA ekranıdır — obje ekleyin/silin, animasyon değiştirin; şablona bağımlılık kalmaz.

Altlık, Tema ve Su Ayarları

Şablon ekranının arka planı (altlık) profesyonel bir çizimdir ve **uygulama temasına uyar**: gündüz/gece/marine geçişinde altlık da tema karşılığıyla değişir; ekranın kayıtlı tanımı değişmez.

Su seviyesi bağları: tasarım modunda su grubuna tıklayınca özellik panelinde "Seviye tag'i / Yükseklik / Dolum payı" alanları çıkar. Depo yüksekliği PLC'de yoksa **elle sabit değer** girilebilir — animasyon doğru oranla çalışır.

Animasyon listesi (⚡) şablonun tüm canlandırmalarını "şablon" rozetiyle listeler: hangi tag hangi objeyi oynatıyor tek bakışta görünür, satıra tıklayınca obje tuvalde parlar.

Not Ekranı silerken "bu şablonla oluşan cihaz da silinsin mi?" seçeneği sunulur — yalnız o kurulumda oluşturulan cihaz için sorulur.

REST API

Genel & Kimlik Doğrulama

endüstriyel standart uyumlu Configuration API. Base URL: `http://<host>:8090/config/v1/project` .

Kimlik	HTTP Basic (kullanıcı/şifre) veya UI oturum cookie'si.
Biçim	JSON.
Yetki	Her uç bir capability ister (bkz. Güvenlik).
Durum kodları	201 (hepsi başarılı), 207 (kısmi), 400 (hepsi hatalı), 409 (çakışma).

```
curl -u admin:PAROLA http://localhost:8090/config/v1/project/channels
```

Başlıca Uçlar

GET	<code>/config/v1/project</code> — proje bilgisi
GET/POST	<code>/config/v1/project/channels</code> — kanallar
GET/POST	<code>.../channels/{ch}/devices</code> — cihazlar
PUT	<code>.../devices/{dev}</code> — cihaz güncelle
POST	<code>.../devices/{dev}/tags</code> — tag ekle (tekil/dizi=batch)
GET	<code>/runtime/v1/status</code> — sayaçlar
GET	<code>/runtime/v1/values?prefix=Kanal.Cihaz</code> — canlı değer
POST	<code>/runtime/v1/write</code> — <code>{"path":"K.C.Tag","value":1}</code>

Örnekler

```
# Kanaldaki cihazlar
curl -u admin:PAROLA http://localhost:8090/config/v1/project/channels/Modbus_1/devices

# Değer yaz
curl -u admin:PAROLA -X POST http://localhost:8090/runtime/v1/write \
  -H "Content-Type: application/json" -d '{"path":"Modbus_1.Dev1.Setpoint","value":42}'

# Canlı değer oku
curl -u admin:PAROLA "http://localhost:8090/runtime/v1/values?prefix=Modbus_1.Dev1"
```

JSON Property Referansı

Ortak Anahtarlar (common.*)

Her nesne (kanal/cihaz/tag) gövdesinde ortak alanlar. Property adları endüstriyel standartla birebirdir (interop).

<code>common.ALLTYPES_NAME</code>	Nesne adı (zorunlu). Kanal/cihaz/tag adı.
<code>common.ALLTYPES_DESCRIPTION</code>	Açıklama (opsiyonel).
<code>PROJECT_ID</code>	İyimser eşzamanlılık damgası. Her GET yanıtında gelir; PUT'ta geri gönderilir. Uyuşmazsa 409 .
<code>FORCE_UPDATE</code>	<code>true</code> → <code>PROJECT_ID</code> kontrolü atlanır (zorla yaz). DataExchanger tipi istemciler kullanır.

Not `servermain.CHANNEL_UNIQUE_ID` , `servermain.DEVICE_UNIQUE_ID` , `servermain.DEVICE_CHANNEL_ASSIGNMENT` , `PROJECT_ID` **server-yönetimli (read-only)**'dir — POST/PUT'ta göndersen de sonucu belirler.

Kanal Property'leri

<code>servermain.Multiple_Types_Device_Driver</code>	Sürücü adı (bkz. Sürücü Adları). Kanal başına bir kez.
<code>servermain.CHANNEL_UNIQUE_ID</code>	Otomatik benzersiz kimlik (read-only).
<code>servermain.CHANNEL_WRITE_OPTIMIZATIONS_METHOD</code>	Yazma optimizasyon yöntemi (yinelenenleri ele / en son yaz).
<code>servermain.CHANNEL_WRITE_OPTIMIZATIONS_DUTY_CYCLE</code>	Tur başına en fazla yazma sayısı.
<code>servermain.CHANNEL_NON_NORMALIZED_FLOATING_POINT_HANDLIN</code> G	NaN/Inf float işleme (ör. sıfırla).

ÖRNEK — KANAL OLUŞTUR

```
POST /config/v1/project/channels
{
  "common.ALLTYPES_NAME": "Modbus_1",
  "servermain.Multiple_Types_Device_Driver": "Modbus TCP/IP Ethernet"
}
```

Ayrıca bkz [Sürücü Adları](#) [Cihaz Property'leri](#)

Cihaz Property'leri

<code>servermain.DEVICE_ID_STRING</code>	Cihaz adresi/kimliği — sürücüye özel (bkz. Sürücü Adları).
<code>servermain.DEVICE_MODEL</code>	Model (sürücüye özel; ör. S7-1200).

<code>servermain.DEVICE_DATA_COLLECTION</code>	<code>true/false</code> — cihaz taraması aç/kapa.
<code>servermain.DEVICE_SIMULATED</code>	<code>true</code> → sentetik veri (donanımsız).
<code>servermain.DEVICE_SCAN_MODE</code> / <code>..._RATE_MS</code>	Tarama modu + hız (ms).
<code>servermain.DEVICE_REQUEST_TIMEOUT_MILLISECONDS</code>	İstek zaman aşımı (ms).
<code>servermain.DEVICE_RETRY_ATTEMPTS</code>	Zaman aşımı öncesi deneme (bağlantı+istek).
<code>servermain.DEVICE_INTER_REQUEST_DELAY_MILLISECONDS</code>	İstekler arası bekleme (ms).
<code>servermain.DEVICE_AUTO_DEMOTEION*</code>	Başarısızlıkta geri-çekilme: <code>ENABLE</code> / <code>DEMOTED_AFTER_SUCCESSIVE_TIMEOUTS</code> / <code>PERIOD_MS</code> / <code>DISCARD_WRITES</code> .
<code>servermain.DEVICE_TAG_GENERATION_*</code>	Otomatik tag üretimi: <code>ON_STARTUP</code> / <code>DUPLICATE_HANDLING</code> / <code>GROUP</code> / <code>ALLOW_SUB_GROUPS</code> .

ÖRNEK – CİHAZ OLUŞTUR

```
POST /config/v1/project/channels/Modbus_1/devices
{
  "common.ALLTYPES_NAME": "1001",
  "servermain.MULTIPLE_TYPES_DEVICE_DRIVER": "Modbus TCP/IP Ethernet",
  "servermain.DEVICE_ID_STRING": "192.168.1.50.1",
  "servermain.DEVICE_DATA_COLLECTION": true,
  "servermain.DEVICE_REQUEST_TIMEOUT_MILLISECONDS": 1000,
  "servermain.DEVICE_RETRY_ATTEMPTS": 3
}
```

Tag Property'leri & Ölçekleme

<code>servermain.TAG_ADDRESS</code>	Cihaz adresi (sürücüye özel; ör. Modbus <code>400001</code>).
<code>servermain.TAG_DATA_TYPE</code>	Veri tipi (Boolean/Word/Short/DWord/Long/Float/Double/String...).
<code>servermain.TAG_READ_WRITE_ACCESS</code>	0 = Read Only, 1 = Read/Write.
<code>servermain.TAG_SCAN_RATE_MILLISECONDS</code>	Tag tarama hızı (ms).
<code>servermain.TAG_SCALING_TYPE</code>	0=yok, 1=lineer, 2=karekök...
<code>..._RAW_LOW</code> / <code>_RAW_HIGH</code>	Ham cihaz aralığı.
<code>..._SCALED_LOW</code> / <code>_SCALED_HIGH</code>	Mühendislik aralığı.
<code>..._SCALED_DATA_TYPE</code> / <code>_CLAMP_LOW</code> / <code>_CLAMP_HIGH</code> / <code>_NEGATE_VALUE</code>	Ölçekli tip, sınırlama, negatifleme.

ÖRNEK – TAG'LER EKLE (BATCH, DİZİ)

```
POST /config/v1/project/channels/Modbus_1/devices/1001/tags
[
  { "common.ALLTYPES_NAME": "Basinc", "servermain.TAG_ADDRESS": "400001",
    "servermain.TAG_DATA_TYPE": "Float", "servermain.TAG_READ_WRITE_ACCESS": 0 },
  { "common.ALLTYPES_NAME": "Setpoint", "servermain.TAG_ADDRESS": "400010",
    "servermain.TAG_DATA_TYPE": "Float", "servermain.TAG_READ_WRITE_ACCESS": 1 }
]
```

Uyarı TAG_READ_WRITE_ACCESS tersine çevirmeyin: **0=RO, 1=RW**. Yanlış değer setpoint'leri salt-okunur yapıp yazmayı engeller.

Sürücü Adları & DEVICE_ID_STRING

Kanal `MULTIPLE_TYPES_DEVICE_DRIVER` ve cihaz `DEVICE_ID_STRING` değerleri:

Modbus TCP/IP Ethernet	<IP>.unit Ör. 192.168.1.50.1 ; unit'siz <192.168.1.50>
Modbus RTU Serial / Modbus ASCII Serial	COM portu + unit (kanal seri ayarları).
Siemens S7 TCP/IP	IP; Rack/Slot model/ID'de. Ör. 10.0.0.5
OMRON FINS Ethernet	IP (+FINS SA1/DA1 ayarları).
OMRON NJ Ethernet	IP (EtherNet/IP CIP sembolik).
DNP Client Ethernet	IP:port,masterAddr,slaveAddr
IEC 60870-5-104 Client	IP:2404,commonAddr
IEC 61850 MMS Client	IP:102
MQTT Client	broker host/port.
Simulator	ID gerekmez.

Ayrıca bkz [Protokoller \(adresleme\)](#).

Toplu İşlem & Durum Kodları

Tag'ler **dizi** olarak POST edilirse toplu (batch) işlenir. HTTP durum kodu sonuca göre:

201 Created	Hepsi başarılı.
207 Multi-Status	Kısmi — bazıları başarılı, bazıları hatalı (gövdede tek tek sonuç).
400 Bad Request	Hepsi hatalı. Kanal/cihaz oluşturmada sürücü sınırı doluyorsa da 400 döner (<code>property</code> + açıklama).
409 Conflict	PROJECT_ID uyumsuzluğu (FORCE_UPDATE ile atlanır).

İpucu Geçersiz tag'ler **atlanır** (lenient import); yanıt/olay günlüğünde atlananların nedeni raporlanır. Gerçek export'lar (binlerce tag) tek istekte içe aktarılabilir.

Ayrıca bkz [REST API](#) [Yetkiler](#)

Güvenlik & Yetkilendirme

Roller & Yetkiler (Capabilities)

Her API ucu bir **capability** ister; kullanıcının rolü (veya özel override) bunu sağlamalıdır.

ConfigRead / ConfigWrite	Proje okuma / yazma.
RuntimeControl	Değer yazma, reload, servis kontrol.
Settings / Users / Certificates / Firewall / Import	Özel alanlar.
Alarms / DataLogging / OpcClient	Modül yetkileri.

Uyarı Varsayılan `admin/admin` ile ilk girişte **parola değiştirme zorunludur** (sunucuda zorlanır — değiştirilmeden API tam yetkiyle kullanılamaz). Login rate-limit: 8 hatalı denemede kilit (HTTP 429). İnternete açık kurulumda güçlü parola şarttır.

Ayrıca bkz [HTTPS / Public Yayın](#) [REST API](#)

HTTPS / Public Yayın

Ayarlar → REST API → HTTPS: sertifika (self-signed üret veya PFX yükle), domain, HTTPS portu, HTTP→HTTPS yönlendirme. İnternete açarken: dinleme adresini `0.0.0.0` yapın, DNS'i sunucuya yönlendirin, firewall portlarını açın, AllowAnonymous'ı kapatın, güçlü parola zorunlu. Değişiklikler yeniden başlatmada uygulanır.

Kurulum & Dağıtım

Kurulum Sihirbazı

Windows installer (self-contained — hedefte .NET gerekmez). Kurulumda ayarlanabilir: OPC/REST/HTTPS portları, sertifika, admin şifresi, EULA. Üç servisi kurar: **MergenHub** (çekirdek) + **MergenHubDataBridge** + **MergenHubHistorian**. Domain DNS/dış-IP kontrol yardımcısı içerir.

Yönetici parolası sayfası: **Güçlü parola oluştur** 16 karakterlik güvenli bir parola üretir, iki alana yazar ve **panoya kopyalar**; **Kopyala** düğmesi yazdığınız parolayı istediğiniz an panoya alır; **Parolayı göster** kutusu maskeyi kaldırır (yazım hatasını görürsünüz). Kurulum localhost dışına açılıyorsa güçlü parola zorunludur.

Not Yükseltmede mevcut proje/kullanıcı verisi AYNEN korunur; "Temiz kurulum" kutusu işaretlenirse ProgramData\MergenHub sıfırlanır (geri alınamaz).

Web'de Yayınlama (domain, sertifika, proxy)

Bu bölüm tek soruyu cevaplar: **MergenHub arayüzüne nereden, hangi adresle ve hangi sertifikayla girilecek?** Önce aşağıdaki üç senaryodan birini seçin, sonra yalnız o senaryonun adımlarını uygulayın.

A · Yalnız yerel ağ	Saha/ofis ağı, internete kapalı. Adres: <code>https://192.168.1.50:8090</code> gibi. Sertifika: self-signed yeter (ya da hiç: düz HTTP). Domain gerekmez.
B · Doğrudan internete açma	MergenHub HTTPS'i kendisi sunar. Domain + gerçek sertifika (PFX) şarttır. Önünde başka web sunucusu yoktur.
C · Ters proxy arkasında	Sunucuda zaten IIS / nginx / Apache var. Sertifika ve 443 proxy tarafından ; MergenHub düz HTTP olarak yalnız <code>127.0.0.1</code> dinler.

DOMAIN GEREKLİ Mİ?

Yerel ağda **hayır** — IP ile girilir. İnternete açacaksanız **evet**: güvenilir sertifika IP'ye değil **ada** verilir. Domaini aldıktan sonra **A kaydını** sunucunun genel IP'sine yöneltin (IP değişkense DDNS kullanın).

SELF-SIGNED NE YAPAR, NE YAPMAZ

Ayarlar → Sunucu → SSL Sertifikası → Self-signed üret sertifikayı sizin adınıza yine siz imzalarsınız. Trafiği **şifreler**, kimliği **doğrulamaz**: tarayıcı her girişte "Güvenli değil" uyarısı verir. Yerel ağ ve test için uygundur; internete açık bir sistemde kullanmayın. Uyarıyı kaldırmak isterseniz üretilen sertifikayı her istemci bilgisayarda Windows **Güvenilen Kök Sertifika Yetkilileri** deposuna eklemeniz gerekir.

Not Self-signed üretirken sorulan **domain** alanına makineye hangi adla eriştiğinizi yazın (`mergenhub.firma.local` ya da IP). Adres ile sertifikadaki ad farklıysa tarayıcı ayrıca ad uyuşmazlığı hatası verir.

PFX NE ZAMAN GEREKİR, NASIL ELDE EDİLİR

PFX (.pfx / .p12) = **sertifika + özel anahtar** tek dosyada, parolayla korunur. MergenHub yalnız bu biçimi kabul eder (**PFX yükle** düğmesi). Gerçek sertifika kullandığınız her durumda (senaryo B) gerekir.

Ücretsiz (Let's Encrypt) `win-acme` (wacs.exe) çalıştırın, domaini doğrulayın; araç `.pfx` üretir. Geçerlilik **90 gün** — yenileme görevini kurmayı unutmayın.

Satın alınmış / kurumsal CA Size `.crt` + `.key` verildiyse PFX'e çevirin:
`openssl pkcs12 -export -out mergenhub.pfx -inkey ozel.key -in sertifika.crt -certfile ara-zincir.crt`

Windows'ta hazır sertifika `certlm.msc` → sertifika → Dışa Aktar → **özel anahtarı dahil et** → .pfx

Uyarı Ara sertifikayı (zinciri) mutlaka ekleyin. Zinciri eksik PFX tarayıcıda çalışır görünüp mobil cihazlarda ve API istemcilerinde "sertifika zinciri doğrulanamadı" hatası verir.

B • DOĞRUDAN YAYIN — ADIM ADIM

1. Domainin **A kaydını** sunucunun genel IP'sine yönlendirin.
2. Modem/güvenlik duvarında **443** → **sunucu:443** yönlendirmesi açın. (Let's Encrypt HTTP-01 doğrulaması için kurulum sırasında 80 de gerekir.)
3. Sertifikayı alın, PFX'e çevirin, **Ayarlar** → **Sunucu** → **SSL Sertifikası** → **PFX yükle** ile yükleyin.
4. Aynı sayfada **HTTPS'i aç** kutusunu işaretleyin, **HTTPS portu** 443 kalsın, **HTTP'yi HTTPS'e yönlendir** kutusunu işaretleyin.
5. **Dinleme adresi** alanını `http://0.0.0.0:8090` yapın — yalnız `localhost` yazılıysa dışarıdan erişilemez.
6. **Genel adres (public URL)** alanına `https://scada.firma.com` yazın; AI/MCP keşif adresleri ve bağlantılar bundan türetilir.
7. MergenHub servisini yeniden başlatın (ayarlar başlatmada uygulanır).
8. Dışarıdan doğrulayın: telefonda **mobil veriyle** `https://scada.firma.com` açın (aynı ağdan yapılan test yanlıtır).

İnternete açmadan önce Yönetici parolasını güçlendirin · kullanılmayan hesapları kapatın · **OPC UA portunu (48010) internete açmayın**, yalnız 443 açık olsun · denetim günlüğünü düzenli inceleyin · mümkünse VPN ya da güvenlik duvarında kaynak-IP kısıtı kullanın · sertifikanın bitiş tarihini takvime koyun.

C • TERS PROXY ARKASINDA YAYIN

MergenHub'da **HTTPS'i aç KAPALI** kalsın, dinleme adresi `http://127.0.0.1:8090` olsun (dışarıya kapalı). Sertifika ve 443 proxy tarafından. **Genel adres (public URL)** alanına dışarıdan görünen adresi yazın.

Proxy'de üç ayar kritiktir (1) **Tampon kapalı** olmalı: canlı değerler ve yapay zekâ yanıtları server-sent events ile akar, tamponlanırsa ekran donmuş görünür. (2) **Zaman aşımı en az 300 sn** olmalı ki uzun akışlar kesilmesin. (3) `Host` ve `X-Forwarded-Proto` başlıkları iletilmelidir.

```

nginx:
server {
    listen 443 ssl;
    server_name scada.firma.com;
    ssl_certificate /etc/ssl/certs/scada.crt;
    ssl_certificate_key /etc/ssl/private/scada.key;
    location / {
        proxy_pass http://127.0.0.1:8090;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_buffering off; # canli akis (SSE) icin SART
        proxy_read_timeout 300s;
    }
}

```

```

Apache (mod_proxy + mod_proxy_http + mod_headers etkin olmalı):
<VirtualHost *:443>
    ServerName scada.firma.com
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/scada.crt
    SSLCertificateKeyFile /etc/ssl/private/scada.key
    ProxyPreserveHost On
    RequestHeader set X-Forwarded-Proto "https"
    ProxyPass / http://127.0.0.1:8090/ flushpackets=on timeout=300
    ProxyPassReverse / http://127.0.0.1:8090/
</VirtualHost>

```

IIS: URL Rewrite + Application Request Routing (ARR) kurun, siteye ters proxy kuralı ekleyin (hedef `http://127.0.0.1:8090/`), ARR ayarlarında **Response buffer threshold = 0** yapın (canlı akış için), sertifikayı IIS site bağlamasına ekleyin.

SIK KARŞILAŞILANLAR

Dışarıdan açılmıyor	Dinleme adresi <code>localhost</code> kalmış · 443 yönlendirmesi yok · Windows güvenlik duvarı kuralı yok.
"Güvenli değil" uyarısı	Self-signed kullanıyorsunuz ya da adres sertifikadaki adla aynı değil.
Mobilde sertifika hatası	PFX içinde ara sertifika (zincir) yok.
Ekran donuk, değer gelmiyor (proxy)	Proxy tamponu açık: nginx <code>proxy_buffering off</code> , IIS ARR buffer 0.
Oturum sürekli düşüyor (proxy)	Zaman aşımı kısa; 300 sn yapın.
AI/MCP adresi yanlış görünüyor	Genel adres (public URL) boş kalmış; domaini oraya yazın.

Servis Yönetimi

Servisler `services.msc` 'den yönetilir. Çekirdek servis "MergenHub" adıyla kurulur (kurtarma senaryoları için önemli). Tray uygulaması durum gösterir. Ayarlar dosyası: `%ProgramData%\MergenHub\settings.json` ; proje: `mergenhub.db` .

Sorun Giderme

Tanılama (Diagnostics)

Cihaz bazlı tanılama: ham paket (Wireshark-tarzı hex TX/RX), okuma/yazma süreleri, hata sayaçları. Event Log (olay günlüğü) 7 gün kalıcıdır + dosya logu. UI durum çubuğunda cihaz noktası: yeşil=bağlı, sarı=demote, kırmızı=hata, gri=boşta.

Sık Karşılaşılanlar

Cihaz "Bad" / okumuyor	Bağlantı/timeout, blok boyutu (endüstriyel standart 64), tek-socket çekişmesi (referans sunucuyla aynı gateway'e aynı anda bağlanma).
Yazma gecikiyor	Talep-güdümlüde reconnect; Eager modda hep bağlı. Yazma optimistic olarak anında yansır.
Kanallar taranmıyor	Talep-güdümlü modda tüketici yoksa taranmaz; Eager modu açın.
Abone > toplam görünüyor	_System tag'leri + monitored-item ≥ tag (normal).