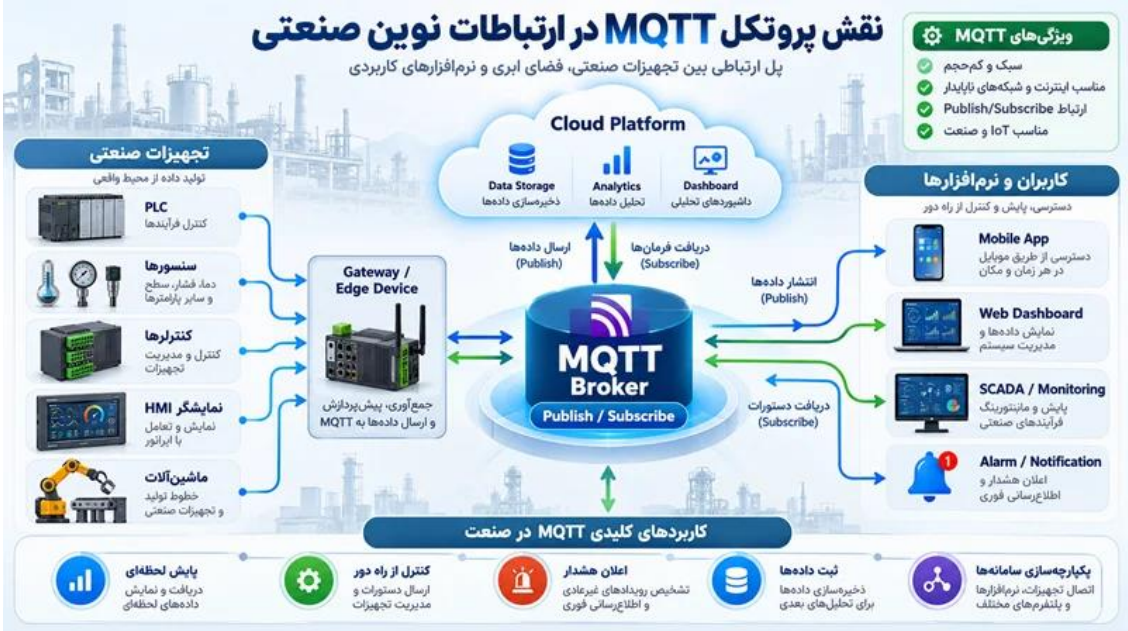
۱- پروتکل MQTT



اگر با تابلو برق، پی‌ال‌سی و اچ‌ام‌آی کار می‌کنید، احتمالاً انتقال یک مقدار از کنترلر به نمایشگر برای شما کار آشنایی است. مسئله زمانی گسترده‌تر می‌شود که همان مقدار را اتاق کنترل، واحد تعمیرات، سامانه ثبت اطلاعات و یک داشبورد مدیریتی هم بخواهند. پروتکل MQTT روشی برای رساندن پیام تجهیزات به مصرف‌کنندگان مختلف است. این فصل کمک می‌کند جایگاه آن را در یک پروژه صنعتی بشناسید و تفاوت انتقال اطلاعات با کنترل مستقیم ماشین را درک کنید.در مثال‌های این فصل، دمای تابلو، وضعیت موتور، شمارنده تولید و قرائت کنتور را دنبال می‌کنیم. هدف، شناخت مسیر داده و اجزای ارتباط است؛ نصب و مقایسه نرم‌افزارهای بروکر موضوع این فصل نیست.

نیاز کارخانه به تبادل اطلاعات میان تجهیزات و نرم‌افزارها

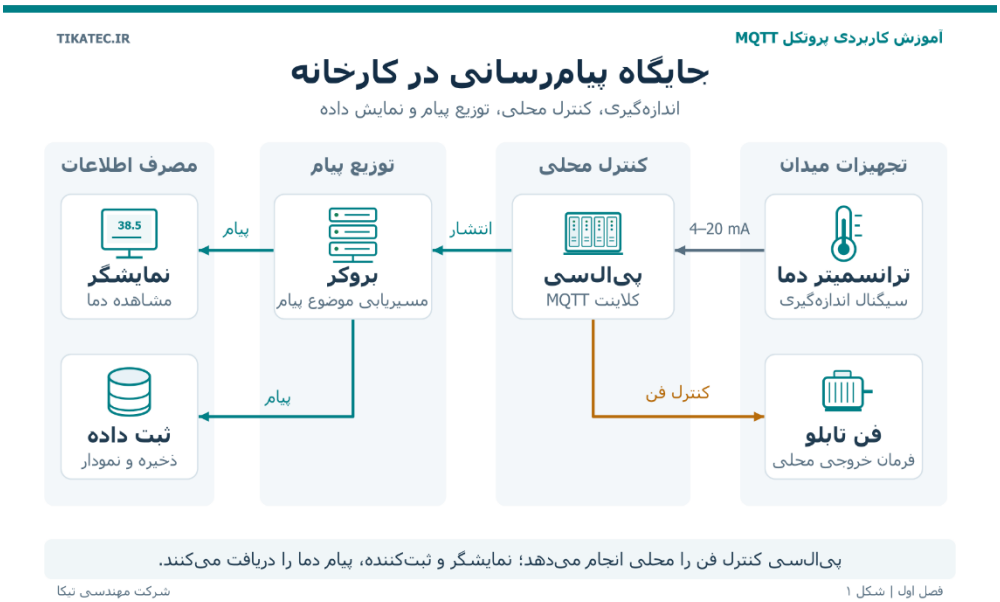
فرض کنید پی‌ال‌سی یک دستگاه بسته‌بندی، وضعیت موتور و تعداد بسته‌های تولیدشده را در حافظه دارد. اپراتور می‌خواهد روشن یا خاموش بودن موتور را روی اچ‌ام‌آی ببیند. مسئول تعمیرات به ساعت کارکرد و علت توقف نیاز دارد. سرپرست تولید نیز تعداد محصول در هر شیفت را می‌خواهد. در این حالت، یک تجهیز داده تولید می‌کند اما چند مصرف‌کننده با نیازهای متفاوت وجود دارند.راه ارتباط این مصرف‌کنندگان همیشه یکسان نیست. نمایشگر کنار دستگاه ممکن است از درایور اختصاصی سازنده استفاده کند؛ سامانه گزارش‌گیری هم ممکن است روی رایانه دیگری اجرا شود. اگر برای هر مصرف‌کننده، ارتباط و منطق جداگانه‌ای در کنترلر طراحی شود، گسترش پروژه و نگهداری آن دشوارتر خواهد شد.در معماری مبتنی برMQTT ، کنترلر یا واسط آن می‌تواند وضعیت موتور را با یک موضوع مشخص منتشر کند. نرم‌افزارهای مجاز که به این موضوع نیاز دارند، پیام را از بروکر دریافت می‌کنند. بنابراین هنگام طراحی، ابتدا می‌پرسیم «چه اطلاعاتی باید منتشر شود و چه کسی به آن نیاز دارد؟» و بعد سراغ انتخاب تجهیزات و تنظیمات می‌رویم.

آشنایی با پروتکل MQTT و هدف طراحی آن

پروتکل یعنی مجموعه‌ای از قواعد مشترک برای ارتباط. همان‌طور که در مدباس، طرفین باید معنی درخواست و پاسخ را بدانند، در MQTT هم قواعدی برای اتصال، انتشار پیام و درخواست دریافت پیام وجود دارد. این پروتکل نخستین‌بار در سال ۱۹۹۹ توسط اندی استنفورد کلارک و آرلن نیپر طراحی شد و برای ارتباط با سربار نسبتاً کم شناخته می‌شود.پیام می‌تواند یک مقدار ساده مانند دما، یک وضعیت مانند «موتور روشن است» یا مجموعه‌ای از اطلاعات باشد. سبک بودن پروتکل یعنی برای تبادل داده، سازوکار نسبتاً کم‌حجمی دارد؛ نه اینکه هر پیامی الزاماً دو بایت است یا با هر سرعت شبکه‌ای، بدون محدودیت کار می‌کند. نام موضوع، اندازه محتوا، تعداد تجهیزات، نرخ ارسال و تنظیمات تحویل، همگی بر مصرف شبکه و پردازنده اثر دارند.برای نمونه، اگر وضعیت یک فن فقط روشن یا خاموش است، معمولاً ارسال یک پیام کوتاه کافی است. اما اگر برنامه هر چند

میلی‌ثانیه همان وضعیت ثابت را تکرار کند، ترافیک بیهوده ایجاد می‌شود. انتخاب زمان ارسال و محتوای مناسب، وظیفه طراح برنامه است و صرف انتخاب پروتکل، این تصمیم‌ها را خودکار نمی‌کند.

جایگاه پروتکل MQTT در سیستم اتوماسیون صنعتی



در یک تابلو معمولی، سنسور مقدار فیزیکی را اندازه می‌گیرد، پی‌ال‌سی آن را پردازش می‌کند و براساس برنامه، خروجی مناسب را فعال می‌کند. اچ‌ام‌آی نیز اطلاعات را نمایش می‌دهد و درخواست اپراتور را به کنترلر می‌رساند. پیام‌رسانی می‌تواند مسیر دیگری برای انتقال داده از این مجموعه به نرم‌افزارهای ثبت و پایش فراهم کند.مثلاً ترانسمیتر، دمای داخل تابلو را اندازه می‌گیرد. پی‌ال‌سی دما را می‌خواند و روشن شدن فن را با منطق محلی کنترل می‌کند. هم‌زمان مقدار دما از طریق MQTT برای نمایشگر مرکزی و ثبت‌کننده ارسال می‌شود. در این طراحی، وظیفه اصلی کنترل فن وابسته به رسیدن پیام به سرور نیست.وجود این ارتباط، الزاماً به معنای تعویض همه ارتباط‌های فعلی نیست. اچ‌ام‌آی کنار دستگاه می‌تواند ارتباط قبلی خود را حفظ کند و فقط اطلاعات موردنیاز برای پایش کارخانه منتشر شوند. همین تفکیک در دستگاه بسته‌بندی هم کاربرد دارد: زمان‌بندی جک‌ها و دوخت بسته در کنترلر می‌ماند، اما شمارنده تولید و علت توقف به سامانه گزارش‌گیری می‌رسد.

بستر ارتباطی پروتکلMQTT

در کاربرد متداول، MQTTروی یک ارتباط TCP/IP اجرا می‌شود. اترنت و وای‌فای، راه اتصال تجهیزات به شبکه هستند؛ MQTT قواعد تبادل پیام روی این ارتباط را مشخص می‌کند. بنابراین داشتن سوکت شبکه به‌تنهایی نشانه پشتیبانی یک پی‌ال‌سی از این پروتکل نیست.

برای استفاده از آن، وجود اینترنت الزامی نیست. اگر پی‌ال‌سی، بروکر و نمایشگر در شبکه داخلی کارخانه به یکدیگر دسترسی داشته باشند، پیام‌ها می‌توانند همان‌جا مبادله شوند. اینترنت زمانی لازم می‌شود که مسیر ارتباط انتخاب‌شده از آن عبور کند؛ مثلاً سرور در مرکز داده بیرون از کارخانه قرار داشته باشد.در تنظیمات معمولاً نشانی بروکر، پورت و اطلاعات اتصال وارد می‌شود. پورت رایج برای ارتباط بدون TLS عدد ۱۸۸۳ و برای ارتباط دارای TLS عدد ۸۸۸۳ است، اما مدیر شبکه می‌تواند پورت دیگری تعیین کند. وارد کردن عدد ۸۸۸۳ به‌تنهایی رمزنگاری ایجاد نمی‌کند؛ هر دو طرف باید واقعاً برای TLS و بررسی گواهی مناسب تنظیم شده باشند.مثال ساده این است که یک رایانه صنعتی در اتاق کنترل میزبان بروکر باشد و کنترلر از طریق سوییچ کارخانه به آن وصل شود. در این حالت، قطع اینترنت لزوماً ارتباط داخلی را قطع نمی‌کند؛ اما قطع همان سوییچ یا مسیر شبکه داخلی می‌تواند انتقال پیام را متوقف کند.

اجزای اصلی یک شبکه پیام‌رسان صنعتی

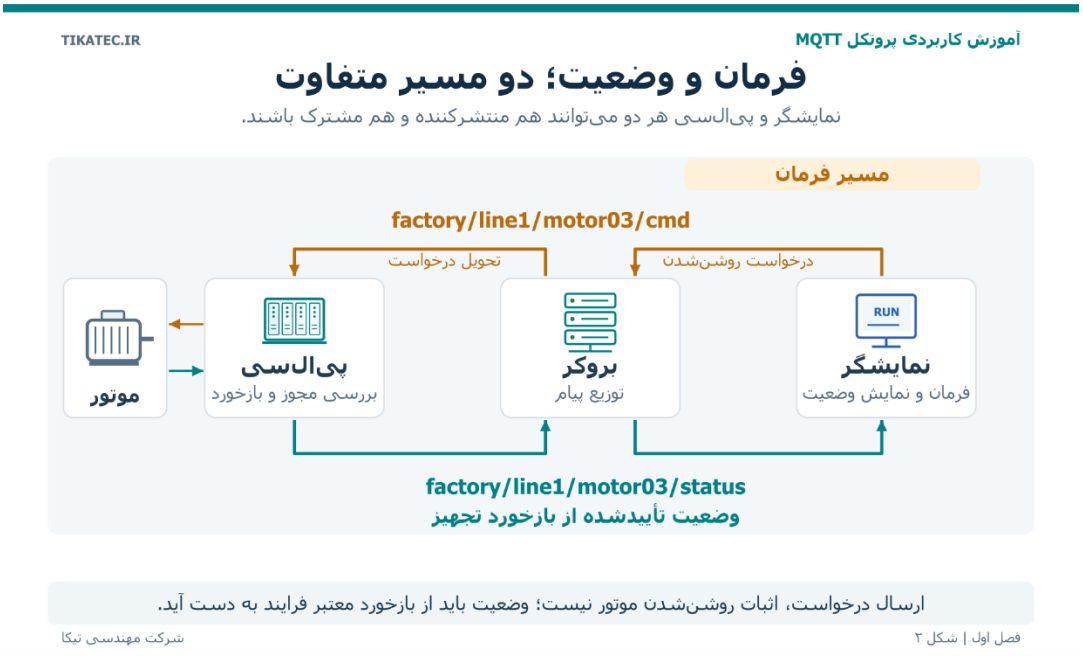
برای خواندن نقشه ارتباط، چهار اصطلاح زیر کافی است. این نام‌ها نقش نرم‌افزارها را توضیح می‌دهند و الزاماً چهار دستگاه فیزیکی جداگانه نیستند.

- کلاینت یا Client نرم‌افزاری است که به بروکر وصل می‌شود. این نرم‌افزار ممکن است داخل پی‌ال‌سی، اچ‌ام‌آی، دروازه ارتباطی یا رایانه اجرا شود.
- منتشرکننده یا Publisher کلاینتی است که پیام می‌فرستد؛ برای مثال، برنامه‌ای که دمای تابلو را منتشر می‌کند.
- مشترک یا Subscriber کلاینتی است که دریافت پیام‌های مربوط به یک موضوع را درخواست می‌کند. واژه اشتراک در اینجا به معنی اعلام علاقه به دریافت پیام است و ارتباطی با اشتراک پولی ندارد.
- بروکر یا Broker اتصال‌ها را مدیریت می‌کند و پیام‌ها را براساس موضوع، اشتراک‌ها و مجوزها به گیرندگان مناسب می‌رساند.

برای تصور اولیه، بروکر را مانند مرکز توزیع نامه در نظر بگیرید: فرستنده، موضوع پیام را مشخص می‌کند و مرکز توزیع آن را به گیرندگان مرتبط می‌رساند. البته بروکر معمولاً از محتوای پیام نتیجه نمی‌گیرد که مثلاً دما خطرناک شده است؛ تشخیص آلارم وظیفه برنامه‌ای است که برای این کار نوشته شده است.

همچنین بروکر را نباید به‌صورت خودکار معادل پایگاه داده در نظر گرفت. نگهداری بعضی پیام‌ها و وضعیت نشست، به قابلیت‌ها و تنظیمات مربوط است. ثبت تاریخچه تولید برای تهیه گزارش، معمولاً به یک مصرف‌کننده و سامانه ذخیره‌سازی مشخص نیاز دارد.

نقش هم‌زمان یک تجهیز به عنوان فرستنده و گیرنده



یک کلاینت می‌تواند هم پیام منتشر کند و هم مشترک پیام باشد. لازم نیست یک پی‌ال‌سی همیشه فقط فرستنده و یک اچ‌ام‌آی همیشه فقط گیرنده باشد. هر دو نقش می‌توانند در یک ارتباط با بروکر وجود داشته باشند.

مثلاً پی‌ال‌سی وضعیت موتور را منتشر می‌کند و اچ‌ام‌آی آن را دریافت می‌کند. در جهت مقابل، اچ‌ام‌آی درخواست روشن شدن موتور را منتشر می‌کند و پی‌ال‌سی مشترک موضوع فرمان است. موضوع «فرمان موتور» را از موضوع «وضعیت موتور» جدا می‌کنیم تا درخواست اپراتور با نتیجه واقعی اشتباه نشود.

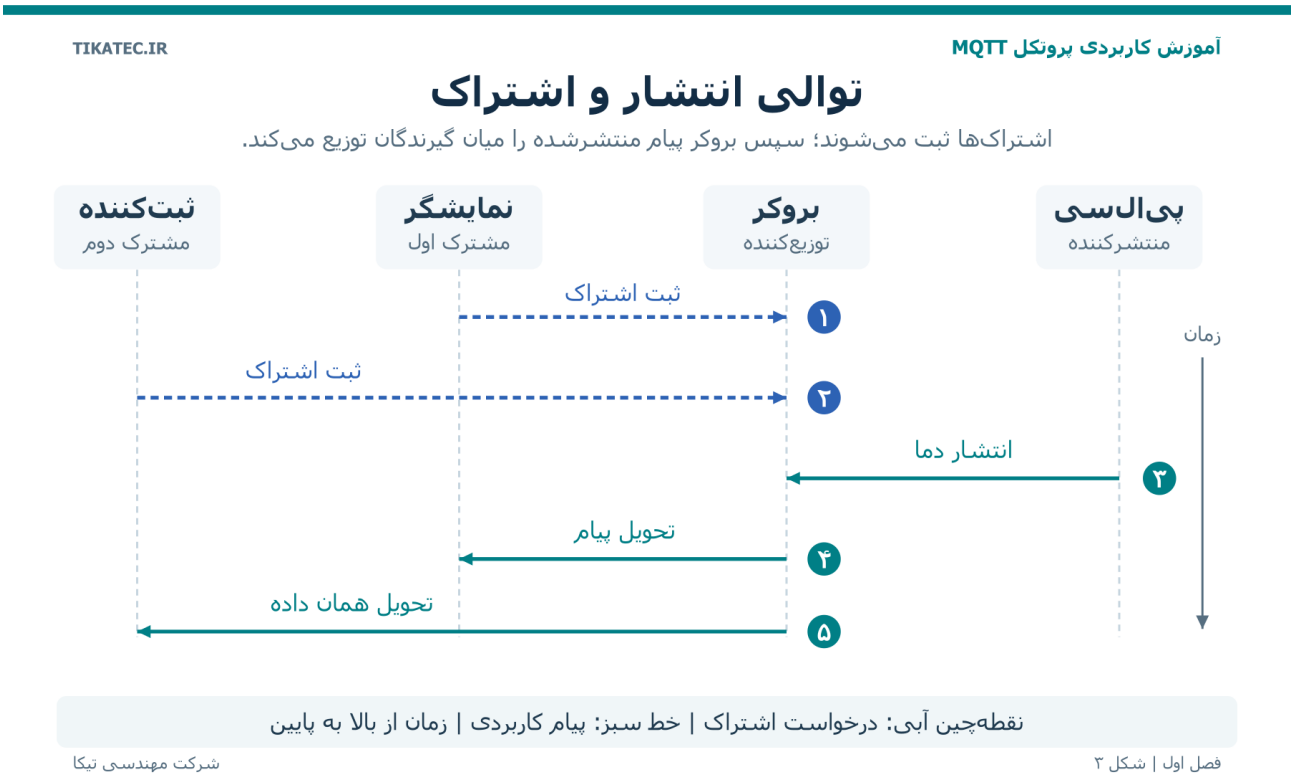
پس از دریافت درخواست روشن شدن، پی‌ال‌سی باید شرایط مجاز بودن راه‌اندازی را بررسی کند: وضعیت حفاظ‌ها، خطای درایو، انتخاب حالت کاری و سایر شرط‌های پروژه. تنها در صورت مجاز بودن، منطق کنترل فرمان مناسب را صادر می‌کند. نمایشگر نیز بهتر است روشن شدن واقعی را از بازخورد معتبر بفهمد، نه از اینکه دکمه استارت فشرده شده است.

در نتیجه، رسیدن پیام فرمان به مقصد با انجام شدن فرمان برابر نیست. پاسخ کاربردی مانند «پذیرفته شد»، «رد شد» یا «موتور روشن است» باید در طراحی برنامه مشخص شود. این تمایز، هنگام عیب‌یابی از تعویض بی‌دلیل تجهیزات شبکه جلوگیری می‌کند.

روش انتشار و اشتراک پیام

مدل اصلی ارتباط، انتشار و اشتراک یا Publish/Subscribe نام دارد. برای دیدن ترتیب کار، فرض کنیم پی‌ال‌سی می‌خواهد دمای تابلو را به نمایشگر و ثبت‌کننده اطلاعات برساند.

- ابتدا هر کلاینت اتصال خود را با بروکر برقرار می‌کند و شرایط احراز هویت و دسترسی آن بررسی می‌شود.
- نمایشگر و ثبت‌کننده، اشتراک موضوع دمای تابلو را به بروکر اعلام می‌کنند.
- برنامه پی‌ال‌سی مقدار اندازه‌گیری‌شده را آماده می‌کند و آن را همراه موضوع مشخص منتشر می‌کند.
- بروکر، پیام را برای کلاینت‌های مجاز دارای اشتراک منطبق ارسال می‌کند.
- نمایشگر مقدار را نشان می‌دهد و ثبت‌کننده همان داده را طبق برنامه خود ذخیره می‌کند.



پی‌ال‌سی برای این کار لازم نیست یک پیام را جداگانه به نشانی هر نمایشگر بفرستد. البته این استقلال به معنی نبودن تنظیمات نیست؛ نام موضوع، قالب محتوا و مجوزهای دریافت همچنان باید سازگار باشند.گیرنده‌ای که تازه وصل می‌شود الزاماً تمام پیام‌های گذشته را دریافت نمی‌کند. دریافت آخرین مقدار نگه‌داری‌شده، ادامه نشست و پیام‌های زمان قطع ارتباط، قواعد و تنظیمات جداگانه دارند. آن‌ها را در فصل مربوط به رفتار تحویل پیام بررسی می‌کنیم؛ فعلاً نباید وجود تاریخچه خودکار را فرض کرد.

تفاوت انتشار پیام با پرس‌وجوی دوره‌ای



در روش پرس‌وجوی دوره‌ای یا Polling، نرم‌افزار در فاصله‌های مشخص مقدار را درخواست می‌کند. مثلاً نمایشگر هر ثانیه رجیستر دما را می‌خواند، حتی اگر مقدار دما از دفعه قبل تغییر نکرده باشد. این روش برای بسیاری از ارتباط‌های صنعتی ساده و قابل استفاده است.

در مدل انتشار، برنامه فرستنده تصمیم می‌گیرد چه زمانی پیام بفرستد. این تصمیم می‌تواند ارسال دوره‌ای، ارسال هنگام تغییر، ارسال هنگام وقوع رویداد یا ترکیبی از این‌ها باشد. خود MQTT تغییر سنسور را تشخیص نمی‌دهد؛ این کار را منطق برنامه یا نرم‌افزار واسط انجام می‌دهد.

برای درِ یک تابلو، انتشار هنگام باز و بسته شدن می‌تواند مناسب باشد. برای دما، می‌توان شرطی مانند تغییر نیم درجه تعریف کرد و علاوه بر آن، هر ۳۰ ثانیه هم یک گزارش تازه فرستاد. این اعداد فقط نمونه آموزشی‌اند و باید با سرعت تغییر فرایند و نیاز پایش انتخاب شوند.در روش ترکیبی، پیام تکراری کمتر می‌شود و گیرنده نیز مدت طولانی بدون گزارش تازه نمی‌ماند. با این حال، کاهش ترافیک نتیجه طراحی مناسب است؛ ارسال یک پیام بزرگ برای هر تغییر بسیار کوچک، ممکن است ترافیک بیشتری از یک خواندن دوره‌ای ساده ایجاد کند.

آشنایی مقدماتی با موضوع پیام یا تاپیک

موضوع پیام یا Topic نامی است که بروکر برای تطبیق پیام با اشتراک‌ها به کار می‌برد. بهتر است این نام طوری انتخاب شود که محل و مفهوم داده برای اعضای تیم قابل فهم باشد. نمونه زیر می‌تواند موضوع دمای یک تابلو در خط اول کارخانه باشد.

factory/line1/panel01/temperature

در این مثال، بخش اول کارخانه، بخش دوم خط، بخش سوم تابلو و بخش آخر کمیت اندازه‌گیری‌شده را نشان می‌دهد. علامت / سطح‌ها را از هم جدا می‌کند. این معنا را طراح پروژه تعیین کرده است؛ خود پروتکل نمی‌داند خط اول یا تابلوی شماره یک دقیقاً چه تجهیزی است.

موضوع را با نشانی شبکه یا شماره رجیستر یکی نگیرید. نشانی بروکر می‌گوید به کدام میزبان وصل شوید؛ موضوع می‌گوید پیام به کدام دسته اطلاعات مربوط است. مثلاً رجیستر ۴۰۰۰۱ می‌تواند در برنامه واسط به همین موضوع دما نگاشت شود، اما این ارتباط باید صریحاً تعریف شده باشد.

نام موضوع به حروف بزرگ و کوچک حساس است؛ بنابراین Temperature و temperature یک نام نیستند. استفاده از نام‌گذاری ثابت و مستند، یکی از ساده‌ترین راه‌های جلوگیری از خطای «اتصال برقرار است اما داده نمی‌رسد» محسوب می‌شود.

تفاوت پیام با محتوای آن

برای فهم کاربردی، پیام را یک پاکت و محتوای آن را نوشته داخل پاکت در نظر بگیرید. موضوع می‌گوید پیام به چه چیزی مربوط است و محتوای پیام یا Payload داده واقعی را حمل می‌کند. بسته انتشار علاوه بر این‌ها، اطلاعات کنترلی مرتبط با پروتکل هم دارد.
اگر موضوع پیام «دمای تابلو» باشد، محتوا می‌تواند فقط عدد ۳۸٫۵ باشد. در این حالت، فرستنده و گیرنده باید از قبل توافق کرده باشند که این عدد برحسب درجه سلسیوس است. اگر واحد، زمان اندازه‌گیری یا کیفیت داده هم لازم باشد، می‌توان از قالب ساختاریافته استفاده کرد. برای نمونه، محتوای JSON زیر هم مقدار و هم واحد را مشخص می‌کند.

{"temperature":38.5,"unit":"C"}



این دو نمونه، دو انتخاب برای قالب محتوا هستند و الزاماً هر دو با هم ارسال نمی‌شوند. قالب ساده کم‌حجم‌تر است؛ قالب ساختاریافته خواندن و گسترش داده را آسان‌تر می‌کند. گیرنده باید دقیقاً بداند کدام قالب را دریافت خواهد کرد.

پیام‌رسانی به‌خودی‌خود عدد خام را به مقدار مهندسی تبدیل نمی‌کند. اگر کنترلر عدد ۳۸۵ را با قرارداد ضریب ۰٫۱ ارسال کند، مصرف‌کننده باید همان قرارداد را بداند تا ۳۸٫۵ درجه نمایش دهد. اشتباه در ضریب یا واحد ممکن است با وجود ارتباط کاملاً سالم، نتیجه نادرست ایجاد کند. محتوا نیز الزامی به متنی بودن ندارد؛ ارسال داده دودویی با قرارداد مشخص امکان‌پذیر است.

استقلال فرستنده و گیرنده از یکدیگر

استقلال در این معماری یعنی فرستنده معمولاً لازم نیست تعداد گیرندگان یا نشانی تک‌تک آن‌ها را بشناسد. اگر دمای موردنیاز از قبل با موضوع و قالب مشخص منتشر می‌شود، اضافه کردن یک داشبورد مجاز اغلب با تنظیم سمت داشبورد و بروکر انجام می‌شود، بدون اینکه برای آن، مقصد تازه‌ای در برنامه پی‌ال‌سی بنویسیم.با این حال، وابستگی معنایی داده همچنان وجود دارد. اگر نام موضوع عوض شود یا واحد دما از سلسیوس به فارنهایت تغییر کند، مصرف‌کنندگان باید از آن مطلع شوند. استقلال ارتباطی جای قرارداد داده را نمی‌گیرد.برای پروژه عملی، یک فهرست کوتاه شامل موضوع، معنی مقدار، واحد، نوع داده، زمان ارسال و مسئول تولید داده تهیه کنید. این فهرست باعث می‌شود تکنسین بعدی هنگام افزودن نمایشگر، مجبور نباشد معنی هر پیام را با آزمون و خطا پیدا کند.

مقایسه کاربردی پروتکل‌های صنعتی

انتخاب پروتکل باید براساس وظیفه ارتباط باشد. مدباس در بسیاری از تجهیزات برای خواندن و نوشتن بیت‌ها و رجیسترها استفاده می‌شود MQTT . برای توزیع پیام میان تولیدکنندگان و مصرف‌کنندگان مناسب است. استاندارد OPC UA علاوه بر ارتباط، امکانات گسترده‌ای برای مدل‌سازی اطلاعات و توصیف داده دارد.

در جدول، کاربردهای متداول مقایسه شده‌اند؛ این جدول بیانگر تمام قابلیت‌های هر استاندارد نیست. برای مثال، OPC UAفقط به روش درخواست و پاسخ محدود نیست و مدل انتشار و اشتراک آن نیز وجود دارد؛ حتی می‌تواند از MQTT به‌عنوان یکی از بسترهای انتقال استفاده کند.

پروتکل	شیوه رایج نمایش داده	نمونه کاربرد در کارخانه
Modbus	بیت و رجیستر؛ معنی و ضریب داده با قرارداد تجهیز مشخص می‌شود	خواندن جریان از کنتور یا تبادل داده با درایو
MQTT	پیام دارای موضوع و محتوا؛ قالب محتوا را پروژه تعریف می‌کند	رساندن داده دما به نمایشگر، ثبت‌کننده و داشبورد
OPC UA	داده دارای نوع و مدل اطلاعاتی؛ قابلیت‌های گسترده ارتباط و توصیف	تبادل اطلاعات ساختاریافته تجهیزات با نرم‌افزارهای بالادستی

یک مثال ترکیبی، قرائت جریان از کنتور مدباس است. دروازه ارتباطی رجیستر مناسب را می‌خواند، ضریب تبدیل را اعمال می‌کند و مقدار آمپر را با موضوع مشخص منتشر می‌کند. در اینجا مدباس ارتباط نزدیک به تجهیز را انجام می‌دهد و MQTT توزیع داده به نمایشگر و سامانه انرژی را بر عهده دارد. لازم نیست برای استفاده از پیام‌رسانی، کنتور سالم موجود تعویض شود.

روش اتصال تجهیزات دارای قابلیت پیام‌رسانی و تجهیزات قدیمی

در مسیر اول، کنترلر یا اچ‌ام‌آی کتابخانه یا قابلیت MQTT دارد و مستقیماً به بروکر متصل می‌شود. وجود این قابلیت باید در راهنمای همان مدل، نسخه سیستم‌عامل و محیط برنامه‌نویسی بررسی شود؛ وجود اترنت به‌تنهایی کافی نیست.

در مسیر دوم، یک دروازه ارتباطی یا Gatewayبین تجهیز قدیمی و شبکه پیام‌رسان قرار می‌گیرد. مثلاً دروازه، داده کنتور را از مدباس سریال می‌خواند و در شبکه اترنت منتشر می‌کند. نگاشت رجیسترها، ترتیب بایت‌ها، ضریب تبدیل و زمان نمونه‌برداری باید درست تنظیم شوند.



در مسیر سوم، رایانه صنعتی یا نرم‌افزار واسط با درایور مناسب، داده را از دستگاه می‌گیرد و نقش کلاینت پیام‌رسان را اجرا می‌کند. انتخاب میان این مسیرها به امکانات موجود، تعداد نقاط، نیاز نگهداری و محدودیت منابع بستگی دارد.
هنگام تست، دو ارتباط را جداگانه بررسی کنید: ارتباط تجهیز با دروازه و ارتباط دروازه با بروکر. ممکن است دروازه به بروکر وصل باشد، اما کابل کنتور قطع شده باشد. در چنین وضعیتی، نمایش آخرین عدد بدون اعلام کهنه بودن داده گمراه‌کننده است. زمان آخرین اندازه‌گیری موفق و کیفیت داده باید در طراحی لحاظ شوند.

شناخت نسخه‌های رایج پروتکل

نسخه‌های ۳٫۱٫۱ و ۵٫۰ در تجهیزات و نرم‌افزارها دیده می‌شوند. مدل اصلی اتصال، انتشار و اشتراک در هر دو وجود دارد. نسخه ۵ امکانات بیشتری مانند زمان انقضای پیام، کنترل دقیق‌تر عمر نشست و اطلاعات روشن‌تر درباره علت بعضی خطاها دارد.
انتخاب نسخه باید براساس پشتیبانی واقعی طرفین باشد. ممکن است بروکر هر دو نسخه را بپذیرد، اما کتابخانه پی‌ال‌سی فقط نسخه ۳٫۱٫۱ داشته باشد. در این حالت نباید قابلیت ویژه نسخه ۵ را برای آن کلاینت فرض کرد. تجهیزات قدیمی‌تر ممکن است نسخه ۳٫۱ را هم مطرح کنند و سازگاری آن‌ها باید جداگانه بررسی شود.
شماره نسخه، به‌تنهایی معیار امنیت نیست. رمزنگاری ارتباط، احراز هویت و مجوز انتشار و دریافت موضوع‌ها، تنظیمات و تصمیم‌های جداگانه‌ای هستند که باید متناسب با شبکه صنعتی اجرا شوند.

مزایا محدودیت‌ها و کاربردهای مناسب

مزیت مهم این پروتکل، امکان توزیع داده میان چند مصرف‌کننده با یک الگوی ارتباطی مشترک است. کاربردهای قابل فهم برای تکنسین اتوماسیون عبارت‌اند از:

- پایش دما و رطوبت تابلوها و سالن‌ها.
- نمایش وضعیت ماشین‌آلات در اتاق کنترل.
- ثبت شمارنده تولید و زمان‌های توقف.
- رساندن آلارم‌ها به سامانه اطلاع‌رسانی.
- جمع‌آوری داده مصرف انرژی برای تحلیل.
- تبادل درخواست‌های نظارتی با منطق اعتبارسنجی و بازخورد مناسب.

در مقابل، زمان رسیدن پیام در یک شبکه معمولی کاملاً ثابت و تضمین‌شده نیست. بار شبکه، وضعیت بروکر، تعداد پیام‌ها و اجرای نرم‌افزار بر تأخیر اثر می‌گذارند. بنابراین مشاهده پاسخ سریع در یک تست کوتاه، برای انتخاب این مسیر به‌عنوان حلقه کنترل حساس کافی نیست.
همچنین سطح تحویل پیام یا QoS درباره رفتار تحویل در ارتباط MQTT است؛ درست بودن مقدار اندازه‌گیری‌شده یا انجام شدن فرمان توسط ماشین را تضمین نمی‌کند. کیفیت سنسور، اعتبار محتوا، بررسی مجوزها و منطق کنترل همچنان مسئولیت بخش‌های مربوط به خود هستند.
برای توقف اضطراری، حفاظت افراد یا هماهنگی دقیق محورها، یک ارتباط معمولی MQTT را جایگزین مدار و شبکه ایمنی یا کنترل قطعیِ طراحی و اعتبارسنجی‌شده نکنید. در مقابل، ارسال دمای تابلو هر چند ثانیه به سامانه پایش، کاربردی متناسب با این معماری است. رفتار زمان قطع ارتباط، داده قدیمی و بازیابی اتصال نیز باید پیش از بهره‌برداری مشخص شود.

بررسی یک پروژه ساده از سنسور تا نمایشگر

اکنون یک پروژه کوچک را کامل دنبال کنیم. فرض کنید می‌خواهیم دمای تابلوی خط اول را در اچ‌ام‌آی و نرم‌افزار ثبت داده ببینیم و در دمای بالا هشدار داشته باشیم. این مثال برای آموزش است؛ حدود دما و زمان‌ها باید در پروژه واقعی از نیاز فرایند و مشخصات تجهیزات به دست آیند.

اندازه‌گیری و تبدیل مقدار

ترانسمیتر فرضی، بازه صفر تا صد درجه را با سیگنال ۴ تا ۲۰ میلی‌آمپر منتقل می‌کند. پی‌ال‌سی ورودی آنالوگ را می‌خواند و طبق تنظیم واقعی کارت و ترانسمیتر، به مقدار مهندسی تبدیل می‌کند. در یک لحظه، نتیجه محاسبه ۳۸٫۵ درجه سلسیوس است. تشخیص قطع سنسور یا خارج شدن سیگنال از بازه معتبر باید در همین بخش پیش‌بینی شود.

تعیین نام و قالب داده

موضوع دما همان factory/line1/panel01/temperature است. برای این مثال توافق می‌کنیم محتوا از قالب JSON معرفی‌شده استفاده کند. این قرارداد باعث می‌شود نمایشگر و ثبت‌کننده درباره معنی مقدار و واحد اختلاف نداشته باشند. اگر کیفیت یا زمان نمونه‌برداری نیز در پیام لازم باشد، نام و قالب آن‌ها از ابتدا مشخص می‌شود.

انتشار و دریافت

برنامه، برای مثال هر دو ثانیه مقدار معتبر را منتشر می‌کند. اگر پی‌ال‌سی قابلیت مستقیم نداشته باشد، دروازه این کار را انجام می‌دهد. نمایشگر و ثبت‌کننده با مجوز مناسب مشترک موضوع می‌شوند؛ یکی عدد را نشان می‌دهد و دیگری آن را برای رسم نمودار ذخیره می‌کند. زمان اندازه‌گیری با زمان دریافت پیام یکی نیست؛ برای تحلیل دقیق باید معلوم باشد کدام زمان ثبت شده است.

هشدار و کنترل محلی

فرض کنیم آستانه آموزشی هشدار ۵۰ درجه باشد. برنامه کنترل، براساس منطق تعریف‌شده، وضعیت هشدار را تعیین می‌کند و می‌تواند آن را در موضوع جداگانه زیر منتشر کند. آستانه هشدار، نحوه برگشت آن و کنترل فن باید مطابق نیاز واقعی طراحی شوند.

factory/line1/panel01/alarm/high_temperature

فن طبق برنامه محلی پی‌ال‌سی عمل می‌کند و ارسال هشدار برای اطلاع‌رسانی است. اگر یک داشبورد جدید به پروژه اضافه شود، می‌تواند با مجوز مناسب همین موضوع‌های موجود را دریافت کند؛ به شرط آنکه داده موردنیازش از قبل منتشر شود.

آزمایش قطع ارتباط

در تست راه‌اندازی، مسیر شبکه را به‌صورت کنترل‌شده و با رعایت شرایط ایمن پروژه قطع کنید. ممکن است نمایشگر هنوز عدد ۳۸٫۵ را نشان دهد، اما این عدد دیگر گزارش تازه نیست. نمایشگر باید وضعیت قطع یا کهنه بودن داده را واضح اعلام کند و برنامه کنترل نیز رفتار از پیش تعیین‌شده داشته باشد. بعد از وصل شدن، تازه شدن داده و بازگشت صحیح وضعیت بررسی می‌شود.
با دنبال کردن همین مثال، اکنون می‌توان مسیر داده را روی نقشه پروژه مشخص کرد: اندازه‌گیری در تجهیز میدان، پردازش در کنترلر، انتشار توسط کلاینت، توزیع در بروکر و استفاده در نمایشگر یا ثبت‌کننده. فصل بعد به قرارداد داده، طراحی موضوع‌ها و انتخاب قالب پیام می‌پردازد تا این مسیر در پروژه‌های بزرگ‌تر هم قابل نگهداری بماند.

۲- ارتباط MQTT در ماژول TM-2950

۲-۱-پیش نیازها



برای شروع ارتباط در ابتدا نیاز به سیمکارت (دائمی/ اعتباری) با امکان اتصال به اینترنت ، بروکر و پلتفرم یا اپلیکیشن به عنوان رابط کاربری لازم میباشد. همه پروژه‌ها به نصب و اداره مستقیم نرم‌افزار بروکر نیاز ندارند. گاهی اجاره یک سرویس آماده، زمان راه‌اندازی را کوتاه می‌کند؛ گاهی نیز نیاز اصلی پروژه، مجموعه‌ای از امکانات شامل ذخیره داده، داشبورد و هشدار است که یک پلتفرم اینترنت اشیا آن‌ها را یکجا فراهم می‌کند.در این قسمت، انتخاب میان این روش‌ها را از دید بهره‌برداری واقعی بررسی می‌کنیم: چه چیزی می‌خریم، چه مسئولیتی بر عهده ما باقی می‌ماند و هنگام قطع ارتباط یا خرابی سرویس چه باید کرد؟ محور بحث همچنان خود بروکر و جایگاه آن در سامانه است.

اجاره بروکر ابری چگونه است؟ اشتراکی/بدون سرور / نمونه اختصاصی

در اجاره بروکر ابری، به‌جای نصب نرم‌افزار روی یک سرور خام، معمولاً حسابی در سامانه ارائه‌دهنده ایجاد می‌کنیم. سپس نشانی اتصال، اطلاعات احراز هویت و امکاناتی برای تنظیم دسترسی‌ها دریافت می‌کنیم. اجرای زیرساخت و بخشی از نگهداری، مطابق قرارداد، بر عهده ارائه‌دهنده است.در سرویس اشتراکی، چند مشتری از زیرساخت یا خوشه مشترک استفاده می‌کنند، ولی حساب‌ها و مجوزهای آن‌ها باید از یکدیگر جدا باشد. اشتراکی‌بودن به معنی عمومی‌بودن پیام‌ها نیست؛ کیفیت جداسازی و محدودیت منابع، بخشی از مشخصات فنی خدمت است.در سرویس بدون‌سرور، مشتری درگیر انتخاب و اداره سرور نیست و هزینه معمولاً با مصرف ارتباط دارد. این نام به معنی نبودن سرور یا رایگان‌بودن ارتباط دائمی نیست. برای نمونه، در سرویس بدون‌سرورEMQX Cloud ، زمان اتصال نشست‌ها و ترافیک از عوامل محاسبه هزینه هستند. در نمونه اختصاصی، محیط بروکر برای مشتری جدا می‌شود؛ اما این موضوع لزوماً به معنی اختصاص تمام سخت‌افزار فیزیکی نیست.

تفاوت بروکر عمومی آزمایشی با سرویس ابری دارای حساب خصوصی درچیست؟

بروکر عمومی آزمایشی معمولاً برای یادگیری، بررسی یک کتابخانه یا آزمایش کوتاه ارتباط در دسترس است. اگر کاربران ناشناس بتوانند روی موضوع‌های مشترک منتشر کنند یا مشترک شوند، دانستن نام موضوع می‌تواند برای مشاهده یا ایجاد پیام کافی باشد. انتخاب نام طولانی و تصادفی برای موضوع، جای کنترل دسترسی را نمی‌گیرد.در مقابل، یک سرویس رایگان دارای حساب خصوصی ممکن است احراز هویت و مجوزهای جداگانه داشته باشد. بنابراین، «رایگان»، «عمومی» و «اشتراکی» سه مفهوم متفاوت‌اند. رمزگذاری ارتباط نیز به‌تنهایی تعیین نمی‌کند چه کسی حق دریافت یا ارسال پیام دارد.گردانندگان بروکر آزمایشی مسکویتو صریحاً توصیه می‌کنند داده حساس در آن منتشر نشود و برای کاربرد مهم به آن اتکا نشود. این سرویس برای آزمون مفید است، اما محل مناسبی برای فرمان واقعی پمپ، درِ ورودی یا تجهیزات تولید نیست.

محدودیت‌ها و شیوه محاسبه هزینه ها به چه صورت میباشد؟



پیش از خرید، سقف اتصال هم‌زمان، نرخ پیام، اندازه پیام، حجم ترافیک، طول صف و محدودیت نگهداری نشست بررسی می‌شود. عبور از سقف ممکن است به افزایش هزینه، محدودشدن ارسال یا رد اتصال منجر شود؛ رفتار دقیق به طرح بستگی دارد.برای یک مثال فرضی، صد دستگاه که در تمام یک ماه سی‌روزه متصل بمانند، چهار میلیون و سیصد و بیست هزار دقیقه اتصال ایجاد می‌کنند. این عدد فقط زمان اتصال را نشان می‌دهد؛ تعداد پیام‌ها و ترافیک را باید جدا حساب کرد. همچنین پیامی که به ده مشترک تحویل می‌شود، می‌تواند ترافیک خروجی بیشتری از همان پیام با یک گیرنده بسازد.طرح‌های مختلف ممکن است بر اساس زمان سرویس، ظرفیت رزروشده، تعداد پیام یا ترافیک قیمت‌گذاری شوند. سهمیه رایگان نیز معمولاً سقف دارد. برآورد هزینه باید با یک الگوی مصرف واقعی و با درنظرگرفتن رشد تعداد دستگاه‌ها انجام شود، نه صرفاً قیمت آغازین روی صفحه فروش.

معرفی سرویس‌های بروکر ابری ایرانی و خارجی

در نمونه‌های خارجی،EMQX Cloud کلاد مدل‌های بدون‌سرور و اختصاصی را عرضه می‌کند. بدون‌سرور می‌تواند برای شروع با مصرف محدود مناسب باشد؛ نمونه اختصاصی برای نیاز به جداسازی و ظرفیت مشخص قابل بررسی است. انتخاب نهایی به محدودیت‌ها، منطقه استقرار و شرایط دسترسی مشتری بستگی دارد. HiveMQ Cloud نیز سرویس مدیریت‌شده ارائه می‌کند، اما نباید معرفی‌های قدیمی طرح رایگان آن را بدون بررسی تکرار کرد. طبق اعلام رسمی در زمان نگارش، ایجاد خوشه جدید در طرح قدیمی بدون‌سرور متوقف شده و پایان فعالیت خوشه‌های موجود برای ۳۱ دسامبر ۲۰۲۶ اعلام شده است. طرح استارتر مسیر دیگری با دوره آزمایشی و شرایط پرداخت خودش دارد. این تغییر نمونه روشنی از ضرورت بازبینی خدمات ابری در طول عمر پروژه است.**در ایران، چابکان و رانفلر** خدمات میزبانی آماده مسکویتو را معرفی کرده‌اند. این‌ها نمونه‌هایی برای تهیه محیط آماده اجرای بروکر هستند؛ نباید بدون بررسی قرارداد، آن‌ها را معادل تمام امکانات یک بروکر ابری کاملاً مدیریت‌شده دانست. وضعیت داده پایدار، نسخه پشتیبان، تمدید گواهی و مسئولیت به‌روزرسانی باید در طرح انتخابی روشن شود.**برای آموزش**، یک بروکر عمومی آزمایشی با داده ساختگی می‌تواند کافی باشد. برای یک پروژه کوچک واقعی، حساب خصوصی با دسترسی محدود و امکان پشتیبان‌گیری یا انتقال، اهمیت بیشتری دارد. رایگان‌بودن مفید است، ولی جای پاسخ‌گویی در زمان خرابی را نمی‌گیرد.**در کاربرد صنعتی**، معیار انتخاب باید مدت قابل تحمل توقف، امکان ادامه کار محلی، تعهد پشتیبانی و فرایند بازیابی باشد. اختصاصی‌بودن نمونه به‌تنهایی این نیازها را برآورده نمی‌کند. یک نمونه اختصاصی تک‌گرهی نیز ممکن است با خرابی همان گره از دسترس خارج شود.

مقایسه ظرفیت، امکانات، پشتیبانی و دسترسی

نوع خدمت	کاربرد قابل بررسی	نکته تعیین‌کننده
بروکر عمومی آزمایشی	یادگیری و آزمون کوتاه	بدون داده حساس و بدون اتکای عملیاتی
بروکر ابری با حساب خصوصی	پروژه کوچک تا مقیاس بزرگ، بسته به طرح	سقف مصرف، مجوزها و تعهد خدمت
میزبانی آماده مسکویتو	استقرار سریع یک محیط بروکر	تفکیک مسئولیت میزبان و مدیر نرم‌افزار
نمونه مدیریت‌شده اختصاصی	نیاز به جداسازی یا ظرفیت مشخص	معماری پایداری، قرارداد و هزینه

استفاده از پلتفرم اینترنت اشیا به‌جای تهیه بروکر مستقل



پلتفرم چه امکاناتی فراتر از بروکر ارائه می‌دهد؟ بروکر، پیام را دریافت و بر اساس اشتراک‌ها و مجوزها توزیع می‌کند. برای تبدیل این ارتباط به محصول نهایی، معمولاً به بخش‌های دیگری نیاز داریم: ثبت تاریخچه، تعریف دستگاه، نمایش نمودار، مدیریت کاربران و اجرای قواعد هشدار. پلتفرم اینترنت اشیا مجموعه‌ای از این امکانات را کنار هم قرار می‌دهد.وقتی از پلتفرم استفاده می‌کنیم، لزوماً بروکر از معماری حذف نمی‌شود؛ ممکن است قابلیت دریافت پیام یا یک بروکر درون آن وجود داشته باشد و مدیریت مستقیمش به کاربر واگذار نشود. در برخی معماری‌ها نیز پلتفرم به بروکر مستقل پروژه متصل می‌شود. پشتیبانی از این اتصال به محصول و نسخه آن بستگی دارد.

داشبورد، ذخیره‌سازی، هشدار و مدیریت دستگاه‌ها

در یک گلخانه، بروکر می‌تواند پیام دما را به برنامه‌های مجاز برساند. برای نمایش نمودار هفتگی، باید داده در یک سامانه ذخیره شود. برای اعلام دمای غیرعادی، قاعده تشخیص و مسیر ارسال هشدار لازم است. برای محدودکردن دسترسی کاربران هر گلخانه نیز سازوکار مدیریت کاربر نیاز داریم.پلتفرمی مانند ThingsBoardقابلیت‌هایی در زمینه جمع‌آوری و نمایش داده، مدیریت دستگاه و پردازش رویداد ارائه می‌کند. البته امکانات دقیق میان نسخه‌ها و شیوه استقرار یکسان نیست. داشتن پنل مدیریتی در یک بروکر نیز به معنی داشتن داشبورد کاربردی و تاریخچه کامل داده‌های تجهیزات نیست.

تفاوت پلتفرم خودمیزبان با پلتفرم ابری

در پلتفرم خودمیزبان، نرم‌افزار روی زیرساخت در اختیار مجموعه اجرا می‌شود. کنترل بیشتری بر محل داده و تنظیمات داریم، ولی مسئولیت نگهداری پایگاه داده، فضای ذخیره‌سازی، سرویس‌های جانبی و به‌روزرسانی‌ها نیز افزایش می‌یابد. اداره یک پلتفرم کامل معمولاً از اداره یک بروکر ساده گسترده‌تر است.در پلتفرم ابری، بخش بیشتری از زیرساخت آماده است و راه‌اندازی داشبورد می‌تواند سریع‌تر باشد. در عوض، باید هزینه تعداد دستگاه‌ها، حجم داده، مدت نگهداری تاریخچه و دسترسی به رابط برنامه‌نویسی را بررسی کرد. امکان خروجی‌گرفتن از داده و انتقال تنظیمات نیز پیش از خرید اهمیت دارد.

نمونه‌های ایرانی و خارجی و محدودیت سازگاری آن‌ها

در ایران، تینگزپاد خدمات ابری و استقرار محلی را معرفی می‌کند و امکاناتی مانند داشبورد، مدیریت دستگاه و قواعد پردازشی دارد. پشتیبانی از ام‌کیوتی‌تی در آن به معنی ضرورت بررسی قالب پیام و روش شناسایی دستگاه مطابق مستندات پلتفرم است.پاپلی نیز در معرفی پلتفرم خود، اتصال تجهیزات، داشبورد، هشدار و گزینه‌های استقرار ابری، خصوصی یا داخل سازمان را توضیح می‌دهد. برای تطبیق با پروژه، باید نسخه قابل تحویل، محل میزبانی، تعرفه و مسیر اتصال سخت‌افزار از ارائه‌دهنده استعلام شود؛ صفحه معرفی به‌تنهایی جای آزمون عملی را نمی‌گیرد.در نمونه‌های خارجی، تینگزبورد برای ترکیب دریافت داده، مدیریت و داشبورد قابل بررسی است. سرویس اینترنت اشیای آمازون نیز امکانات اتصال امن، مدیریت ارتباط و هدایت داده به خدمات دیگر فراهم می‌کند؛ اما نباید آن را معادل یک داشبورد نهایی آماده برای هر کاربرد دانست. ساخت محصول کامل ممکن است به اجزای دیگری نیاز داشته باشد.

AWS IoT Core

نکته مهم این است که هر سامانه دارای پشتیبانی ام‌کیوتی‌تی، یک بروکر عمومی با آزادی کامل در موضوع‌ها نیست. برای نمونه، مستندات مایکروسافت محدودیت‌های ارتباط ام‌کیوتی‌تی در آی‌اوتی هاب و قراردادهای مشخص موضوع را توضیح می‌دهد. در پلتفرم‌ها ممکن است نام موضوع، ساختار پیام و اطلاعات ورود از پیش تعیین شده باشد. پیش از انتخاب، یک ارسال داده و یک دریافت فرمان با همان دستگاه واقعی آزمایش کنید.

Azure IoT Hub

مقایسه بروکر محلی، سرور اجاره‌ای، بروکر ابری و پلتفرم

بهترین گزینه، روشی است که نیاز ارتباطی و توان نگهداری پروژه را هم‌زمان پوشش دهد. خرید امکانات زیاد بدون فرد مسئول بهره‌برداری، لزوماً پایداری ایجاد نمی‌کند. از طرف دیگر، ساده‌ترین استقرار نیز ممکن است هزینه توسعه داشبورد و ثبت داده را به تیم تحمیل کند.

روش	مزیت اصلی	مسئولیت مهم پروژه
بروکر محلی	ارتباط نزدیک تجهیزات و امکان کار محلی	برق، سخت‌افزار، نسخه پشتیبان و دسترسی دور
بروکر روی سرور اجاره‌ای	اختیار نصب و تنظیمات	اداره سیستم‌عامل و نرم‌افزار
بروکر ابری مدیریت‌شده	کاهش امور اجرایی زیرساخت	مجوزها، مصرف و سازگاری دستگاه‌ها
پلتفرم اینترنت اشیا	امکانات آماده برای داده و کاربر	طراحی کاربرد، قواعد و امکان مهاجرت

انتخاب برای آموزش، خانه هوشمند، گلخانه و صنعت

برای آموزش، مسکویتو روی رایانه یا حساب آزمایشی خصوصی، مشاهده رفتار بروکر را آسان می‌کند. در خانه هوشمند، اگر کنترل روزمره باید هنگام قطع اینترنت ادامه یابد، معماری محلی با دسترسی ابری تکمیلی ارزش بررسی دارد.در گلخانه، تصمیم مهم این است که آبیاری یا تهویه چقدر به اینترنت وابسته باشد. پایش از راه دور می‌تواند ابری باشد، در حالی که منطق ضروری کنترل در محل اجرا شود. در صنعت، حساسیت توقف، نیاز به ثبت رویداد و الزامات شبکه سازمان تعیین می‌کند بروکر کجا قرار بگیرد و چه پشتیبانی لازم داشته باشد.این مثال‌ها نسخه ثابت برای همه پروژه‌ها نیستند. یک گلخانه کوچک با کنترل دستیِ پشتیبان با مجموعه‌ای که تجهیزات حیاتی‌اش فقط فرمان اینترنتی می‌گیرند، نیاز یکسانی ندارد.

چالش‌های ارتباطی و دسترسی

در اتصال سیم‌کارتی، تغییر کیفیت پوشش و قطع کوتاه ارتباط عادی است. دستگاه باید اتصال مجدد با فاصله‌گذاری مناسب داشته باشد تا بازگشت هم‌زمان تعداد زیادی دستگاه، بار ناگهانی ایجاد نکند. همچنین لازم است مشخص شود داده‌های تولیدشده در زمان قطعی کجا نگهداری می‌شوند. بروکر ابری نمی‌تواند داده‌ای را که هرگز به آن نرسیده بازیابی کند. اگر حفظ داده لازم است، دستگاه یا گیت‌وی به بافر با ظرفیت و سیاست حذف مشخص نیاز دارد. زمان اندازه‌گیری باید همراه داده ذخیره شود تا داده دیررس با اندازه‌گیری تازه اشتباه نشود.پیش از وابسته‌کردن محصول به یک سرویس خارجی، امکان ایجاد حساب، پرداخت، دریافت پشتیبانی و استفاده از مناطق میزبانی بررسی شود. شرایط ارائه خدمت و دسترسی شبکه می‌تواند تغییر کند. موفقیت آزمایش روی اینترنت دفتر، تضمین دسترسی تمام مشتریان از همه اپراتورها نیست.

آدرس بروکر و اطلاعات اتصال را تا حد امکان قابل تنظیم نگه دارید. اگر تغییر سرویس مستلزم مراجعه حضوری به هزار دستگاه باشد، هزینه و زمان مهاجرت بسیار بیشتر خواهد شد.برای پایش روند دما، تأخیر چندثانیه‌ای ممکن است پذیرفتنی باشد؛ برای یک حلقه کنترل سریع، همان

تأخیر مناسب نیست. معیار باید زمان رفت‌وبرگشت تا برنامه و تجهیز واقعی باشد، نه فقط پاسخ شبکه سرور.ارسال بی‌وقفه مقادیر تکراری می‌تواند مصرف سیم‌کارت و هزینه سرویس را افزایش دهد. انتخاب دوره نمونه‌برداری، ارسال هنگام تغییر معنادار و ارسال دوره‌ای وضعیت سلامت باید متناسب با نیاز کاربرد انجام شود. کاهش ترافیک نباید باعث پنهان‌ماندن قطع دستگاه یا حذف رویداد مهم شود.