

دانشگاه دامغان

دانشکده فنی و مهندسی

پایان‌نامه دوره کارشناسی مهندسی کامپیوتر

**طراحی و توسعه یک دروازه اینترنت اشیاء (Gateway IoT)**

**علی سپهرنیا**

استاد راهنما:

**دکتر محمود معلم**

بهمن ۱۴۰۰



تشکر و قدردانی

بدین‌وسیله از خانواده عزیزم به‌ویژه پدر و مادرم بابت تمام زحمات ایشان در طول این سالیان و از همة اساتید گرامی به‌ویژه آقای دکتر معلم، آقای دکتر مرتضوی، آقای دکتر متقی، خانم دکتر کریمی، خانم خاتمی و خانم عرفانی که بسیار از آنها آموختم کمال تشکر و قدردانی را دارم. امیدوارم با گام برداشتن در مسیر دانش و کسب مهارت بتوانم اندکی از زحمات آنها را جبران نمایم.

چکیده فارسی

هدف از این پروژه طراحی و برنامه‌نویسی یک نرم‌افزار واسط در حوزه اینترنت اشیاء است که در قالب یک دروازه عمل می‌کند. این نرم‌افزار در پروژه‌های اینترنت اشیاء به‌منظور ایجاد یک شبکه غیرمتمرکز (Decentralized) برای تقسیم بار شبکه (Load Balancing)، پردازش در لبه (Edge Computing)، اتصال دستگاه‌هایی که توانایی اتصال مستقیم به سرور مرکزی را ندارند (Legacy Devices) و ... مورداستفاده قرار می‌گیرد. دستگاه‌های مختلف موجود در شبکه از جمله سرورها، مودم‌ها، سوئیچ‌ها، دستگاه‌های ارتباطی، دوربین‌های نظارتی، لوازم‌خانگی متصل به اینترنت و ... با استفاده از پروتکل‌های مختلف ارتباطی از جمله TCP، UDP، SNMP، Mqtt و ... به دروازه متصل می‌شوند. دروازه در سمت دیگر خود به سرور مرکزی متصل می‌شود و این اطلاعات را به سرور مرکزی منتقل می‌کند. در این نرم‌افزار دو کلاس کمکی برای برقراری ارتباط با سرور مرکزی بر مبنای پروتکل Mqtt تعبیه شده است. یکی از کلاس‌ها وظیفه راه‌اندازی و مدیریت یک سرورِ Mqtt برای دریافت اطلاعات از سرور مرکزی را دارد و کلاس دیگر وظیفه راه‌اندازی یک Mqtt API برای ارسال اطلاعات به سرور مرکزی. علاوه بر این برای مدیریت بهتر نرم‌افزار کلاس‌های دیگر از جمله کلاس پیکربندی (Config) برای پیکربندی بخش‌های مختلف دروازه و کلاس ثبت وقایع (Log) برای ثبت وقایع که در بخش‌های مختلف دروازه صورت‌گرفته است به کار می‌روند. این نرم‌افزار به زبان C# و تحت چهارچوب ASP.Net Core توسعه می‌یابد. برای ارتباطات Mqtt از کتابخانه MqttNet، برای پیکربندی از کتابخانه Microsoft.Extensions.Configuration، برای ثبت وقایع از کتابخانه Microsoft.Extensions.Logging و برای پایگاه‌داده از پایگاه‌داده غیررابطه‌ای MongoDB و کتابخانه MongoDB Driver استفاده شده است.

لازم به ذکر است به جهت کاهش هزینه‌های توسعه آتی و با درنظرداشتن استفاده از نرم‌افزار در راه‌اندازی یک شبکه اینترنت اشیاء برای خانه هوشمند، این نرم‌افزار با تمرکز بیشتر بر روی معماری نرم‌افزار، ماژولار بودن و قابلیت گسترش توسعه‌یافته است. در واقع به‌جای اضافه‌کردن امکانات (Feature) متعدد، سعی بر این بوده است که با استفاده از رویکرد‌های استاندارد مهندسی نرم‌افزار و استاندارهای توسعه ASP.Net، یک اکوسیستم اختصاصی برای این منظور خلق شود که قابلیت شخصی‌سازی و تغییر و گسترش با هزینه نسبتاً پایین را داشته باشد.

واژه‌های کلیدی: دروازه - شبکه - سرور - اینترنت اشیاء - IoT - TCP - UDP - Mqtt - Gateway - MongoDB - ASP.Net Core - Web API

فهرست مطالب

[فصل 1: اینترنت اشیاء 1](#_Toc95934143)

[1-1 مقدمه 1](#_Toc95934144)

[2-1 تاریخچه 2](#_Toc95934145)

[3-1 تعریف 2](#_Toc95934146)

[4-1 فناوری‌های اساسی مورداستفاده 3](#_Toc95934147)

[5-1 قابلیت‌ها 3](#_Toc95934148)

[6-1 حوزه‌های استفاده 4](#_Toc95934149)

[7-1 ویژگی‌ها 4](#_Toc95934150)

[8-1 چالش‌ها 5](#_Toc95934151)

[9-1 معماری 6](#_Toc95934152)

[1-9-1 لایه A: لایه دستگاه‌ها (سنسور‌ها) 6](#_Toc95934153)

[2-9-1 لایه B: لایه دروازه و شبکه 6](#_Toc95934154)

[3-9-1 لایه C: لایه مدیریت و سرویس 7](#_Toc95934155)

[4-9-1 لایه D: لایه اپلیکیشن 7](#_Toc95934156)

[10-1 نتیجه‌گیری 8](#_Toc95934157)

[1-10-1 مقدمه 8](#_Toc95934158)

[2-10-1 تاریخچه 9](#_Toc95934159)

[3-10-1 تعریف 9](#_Toc95934160)

[4-10-1 فناوری‌های اساسی مورداستفاده 9](#_Toc95934161)

[5-10-1 قابلیت‌ها 9](#_Toc95934162)

[6-10-1 حوزه‌های استفاده 10](#_Toc95934163)

[7-10-1 ویژگی‌ها 10](#_Toc95934164)

[8-10-1 چالش‌ها 10](#_Toc95934165)

[9-10-1 معماری 11](#_Toc95934166)

[فصل 2: دروازه اینترنت اشیاء 12](#_Toc95934167)

[1-2 مقدمه 12](#_Toc95934168)

[2-2 شبکه‌های حوزه حسگر 13](#_Toc95934169)

[1-2-2 شبکه شخصی (PAN) 13](#_Toc95934170)

[2-2-2 شبکه وسیله نقلیه 14](#_Toc95934171)

[3-2-2 شبکه خانگی 14](#_Toc95934172)

[3-2 شبکه‌های حوزه اپلیکیشن 14](#_Toc95934173)

[4-2 ویژگی‌ها 14](#_Toc95934174)

[1-4-2 رابط‌های چندگانه 15](#_Toc95934175)

[2-4-2 مبدل پروتکل 15](#_Toc95934176)

[3-4-2 مدیریت پذیری 15](#_Toc95934177)

[5-2 معماری 16](#_Toc95934178)

[1-5-2 ماژول انتزاع سرویس 17](#_Toc95934179)

[2-5-2 ماژول سازگار کننده 17](#_Toc95934180)

[3-5-2 ماژول مبدل پروتکل و ارسال داده 18](#_Toc95934181)

[4-5-2 ماژول نمایه اپلیکیشن و ماژول نمایه شبکه 18](#_Toc95934182)

[5-5-2 ماژول امنیت و مدیریت 18](#_Toc95934183)

[6-2 نتیجه‌گیری 18](#_Toc95934184)

[1-6-2 مقدمه 18](#_Toc95934185)

[2-6-2 شبکه‌های حوزه حسگر و حوزه اپلیکیشن 19](#_Toc95934186)

[3-6-2 ویژگی‌ها 19](#_Toc95934187)

[4-6-2 معماری 19](#_Toc95934188)

[فصل 3: اصطلاحات و مفاهیم 21](#_Toc95934189)

[1-3 پروتکل Mqtt : استاندارد پیام‌رسانی اینترنت اشیا 21](#_Toc95934190)

[فصل 4: تحلیل 23](#_Toc95934191)

[1-4 مقدمه 23](#_Toc95934192)

[2-4 مدل توسعه سریع اپلیکیشن (RAD) 23](#_Toc95934193)

[1-2-4 مقدمه 23](#_Toc95934194)

[2-2-4 مراحل 23](#_Toc95934195)

[3-4 معماری (Context Diagram) 24](#_Toc95934196)

[1-3-4 مقدمه 24](#_Toc95934197)

[2-3-4 سطح 0 25](#_Toc95934198)

[3-3-4 سطح 1 26](#_Toc95934199)

[4-3-4 سطح 2 27](#_Toc95934200)

[5-3-4 خلاصه 27](#_Toc95934201)

[4-4 نمودار جریان داده (DFD) 28](#_Toc95934202)

[1-4-4 مقدمه 28](#_Toc95934203)

[2-4-4 سطح 0 28](#_Toc95934204)

[3-4-4 سطح 1 29](#_Toc95934205)

[4-4-4 سطح 2 30](#_Toc95934206)

[5-4-4 سطح 2 (نهایی) 31](#_Toc95934207)

[5-4 نمودار Activity 31](#_Toc95934208)

[1-5-4 خلاصه 31](#_Toc95934209)

[2-5-4 نمودار اولیه 32](#_Toc95934210)

[3-5-4 نمودار نهایی 33](#_Toc95934211)

[6-4 نمودار کلاس (مدل داده) 34](#_Toc95934212)

[7-4 نتیجه‌گیری 34](#_Toc95934213)

[فصل 5: پیاده‌سازی 35](#_Toc95934214)

[1-5 مقدمه 35](#_Toc95934215)

[1-1-5 فاز 1 35](#_Toc95934216)

[2-1-5 فاز 2 36](#_Toc95934217)

[3-1-5 فاز 3 37](#_Toc95934218)

[2-5 افزودن پایگاه‌داده MongoDB 38](#_Toc95934219)

[3-5 توسعه سرویس ثبت وقایع و سرویس پیکر‌بندی 39](#_Toc95934220)

فهرست نمودارها، عکس‌ها و نقشه‌ها

[شکل ‏1‑1 1](#_Toc95934243)

[شکل ‏1‑2 8](#_Toc95934244)

[شکل ‏2‑1 12](#_Toc95934245)

[شکل ‏2‑2 15](#_Toc95934246)

[شکل ‏2‑3 16](#_Toc95934247)

[شکل ‏2‑4 17](#_Toc95934248)

[شکل ‏3‑1 22](#_Toc95934249)

[شکل ‏4‑1 26](#_Toc95934250)

[شکل ‏4‑2 26](#_Toc95934251)

[شکل ‏4‑3 27](#_Toc95934252)

[شکل ‏4‑4 28](#_Toc95934253)

[شکل ‏4‑5 29](#_Toc95934254)

[شکل ‏4‑6 30](#_Toc95934255)

[شکل ‏4‑7 31](#_Toc95934256)

[شکل ‏4‑8 32](#_Toc95934257)

[شکل ‏4‑9 33](#_Toc95934258)

[شکل ‏4‑10 34](#_Toc95934259)

[شکل ‏5‑1 36](#_Toc95934260)

[شکل ‏5‑2 37](#_Toc95934261)

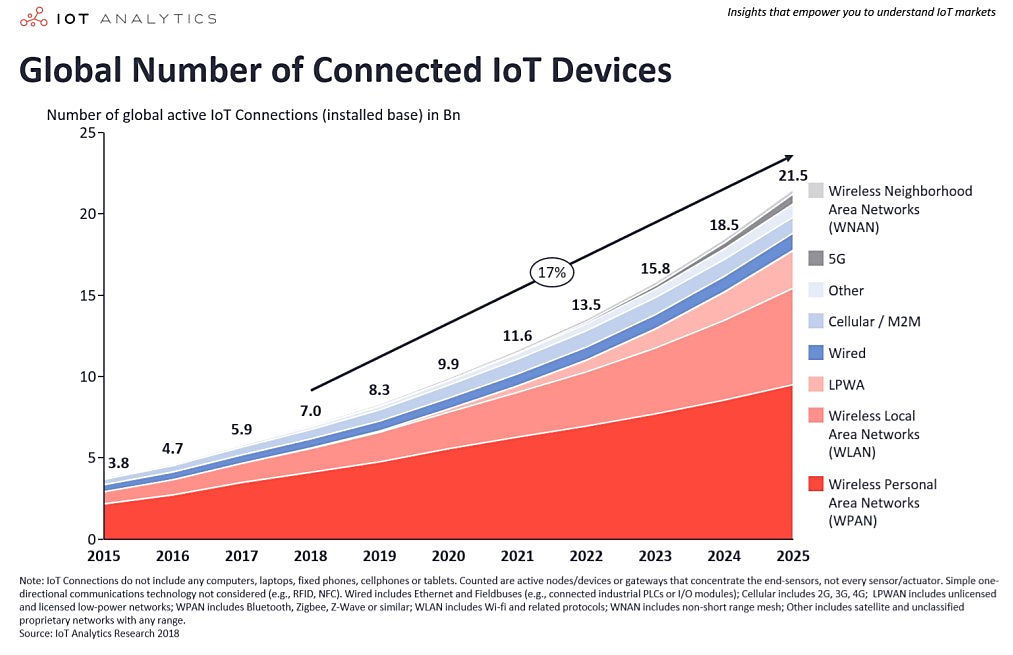
# اینترنت اشیاء

## مقدمه

امروزه اصرار برای اتصال همه چیز به اینترنت در حال افزایش است. این کار نه‌تنها برای ارسال اطلاعات به سرورها و ذخیره‌سازی و پردازش آنها بلکه برای کنترل کامل دستگاه‌های فیزیکی بر روی وب است. [1]

دنیایی را تصور کنید که در آن میلیاردها شیء بتوانند حس کنند، ارتباط برقرار کنند و اطلاعات به اشتراک بگذارند. این اشیای متصل‌به‌هم به طور منظم داده‌ها را جمع‌آوری کرده، آنالیز می‌کنند و از آنها برای انجام رخدادهایی استفاده می‌کنند که باعث می‌شود سرمایه‌ای هوشمند برای برنامه‌ریزی، مدیریت و تصمیم‌گیری در اختیار داشته باشیم. [1]

طبق آمار منتشر شده در سال ۲۰۱۸ تعداد دستگاه‌های متصل به اینترنت حدود ۷ میلیارد دستگاه بوده است و پیش‌بینی می‌شود تا سال ۲۰۲۵ این رقم به ۲۱٫۵ میلیارد برسد که این یعنی حدود 200% افزایش در تعداد دستگاه‌های شبکه اینترنت اشیاء. علاوه بر این در نمودار ذکر شده که تعداد دستگاه‌ها را از سال ۲۰۱۵ تا ۲۰۲۵ ذکر کرده است، رشد نمایی این تعداد مشهود است. [2]



شکل ‏1‑1

بنابراین آمار نشان می‌دهد که حوزه‌ی اینترنت اشیاء روز‌به‌روز بیشتر موردتوجه قرار خواهد گرفت و همچنین بازار قابل‌توجهی از صنعت فناوری اطلاعات و ارتباطات را به خود اختصاص می‌دهد. در نتیجه لازم است متخصصین، پژوهشگران و سیاست‌گذاران این حوزه علاوه بر پایش روند رشد این شبکه عظیم، به معایب، مشکلات و مسائل این رشد روزافزون نیز توجه کنند تا مزایای استفاده از این پتانسیل ارزشمند افزایش یابد و مخاطرات آن به حداقل برسد.

## تاریخچه

اولین دورسنجی در سال ۱۹۱۲ و در شیکاگو صورت گرفت. گفته می‌شود که در آن از خطوط تلفن برای پایش داده‌های نیروگاه استفاده شده است. دورسنجی به پایش آب‌وهوا در سال‌های ۱۹۳۰ نیز گسترش یافت و در آن از یک دستگاه به نام رادیوسوند به طور گسترده برای پایش وضعیت آب‌وهوا از بالون‌های هواشناسی استفاده می‌شد. [3]

استفاده گسترده از اینترنت اشیاء و در واقع ارتباطات ماشین با ماشین در دهه ۸۰ میلادی با فناوری موسوم به SCADA در سامانه‌های امنیتی کارخانه‌ها، خانه‌ها و دفاتر تجاری رقم خورد. بعدها در دهه ۹۰ استفاده از فناوری‌های بی‌سیم برای ارتباطات اشیاء استفاده شد که به دلیل گران بودن ارتباطات سلولی از شبکه‌های رادیویی استفاده می‌شد. در نهایت شروع گسترش شدید اینترنت اشیاء با معرفی اولین ماژول شبکه سلولی توسط شرکت زیمنس در سال ۱۹۹۵ آغاز شد. [3]

## تعریف

اینترنت اشیاء در توصیه‌نامه ITU-T Y.2060 (06/2012) توسط اتحادیه جهانی ارتباطات به‌صورت زیر تعریف شده است:

«یک زیرساخت جهانی برای جامعه اطلاعاتی است که با به اتصال اشیاء (فیزیکی و مجازی) به هم و بر اساس فناوری‌های اطلاعاتی و ارتباطیِ متقابلِ که یا موجودند و یا درحال‌توسعه، خدمات پیشرفته‌ای را امکان‌پذیر می‌کند.» [4]

اینترنت اشیاء در ویکی‌پدیا به‌صورت زیر تعریف شده است:

«اشیای فیزیکی (گروهی از آنها) که شامل حسگرهای تعبیه شده، قابلیت پردازش، نرم‌افزار و دیگر فناوری‌هایی هستند که آنها را قادر می‌سازد تا از طریق اینترنت یا شبکه‌های ارتباطی دیگر به دستگاه‌ها و سامانه‌های دیگر وصل شوند و به تبادل داده بپردازند.» [5]

## فناوری‌های اساسی مورداستفاده

در ابتدایی‌ترین سطح، فناوری‌هایی که اینترنت اشیاء را ممکن می‌سازد، دستگاه‌ها و فناوری‌های ارتباطی است. دستگاه‌ها می‌توانند ساده‌ترین تا پیچیده‌ترین اشیاء مورداستفاده ما باشد که معمولاً در قلب آنها از فناوری‌های توکار نظیر میکروکنترلرها و ریزپردازنده‌ها استفاده می‌شود. فناوری‌های ارتباطی مورداستفاده می‌تواند شامل ارتباطات با سیم، بی‌سیم یا ترکیبی از این دو باشد. فناوری‌های با سیم مورداستفاده شامل ارتباط سریال، اترنت و ... و فناوری‌های بی‌سیم شامل مادون‌قرمز، بلوتوث، وای‌فای، شبکه سیم‌کارت و ... می‌شود. [1]

در سطوح بالاتر اشیای فیزیکی ساده معمولاً با یا بدون واسط‌های سخت‌افزاری، به سرورهای محلی متصل شده و سرورهای محلی نیز به سرور مرکزی متصل می‌شوند. بنا بر نیازمندی‌ها سرورهای محلی می‌توانند تجهیزات ساده‌ای شامل میکروکنترلرها باشند که صرفاً پیام‌ها را، بین سرور مرکزی و اشیاء پایین‌دست ردوبدل می‌کنند یا اینکه سرورهای حتی قدرتمند از سرور مرکزی باشند که توانایی مدیریت و کنترل کامل کل یا بخشی از شبکه را به عهده بگیرند. در ادبیات اینترنت اشیاء معمولاً به این سرورهای میانی، دروازه (Gateway) گفته می‌شود که در فصل سوم مفصل در مورد آن صحبت خواهد شد.

به‌طورکلی فناوری‌های اساسی مورداستفاده را می‌توان به ۳ دسته زیر تقسیم نمود:

* فناوری‌هایی که اشیاء را قادر می‌سازد تا اطلاعات پیرامون را به دست آورند. (عملکردی)
* فناوری‌هایی که اشیاء را قادر می‌سازد تا اطلاعات زمینه‌ای را پردازش کنند. (عملکردی)
* فناوری‌هایی که امنیت و حریم شخصی را بهبود می‌دهند. (ایجابی)

## قابلیت‌ها

اینترنت اشیاء قابلیت این را دارد تا یک دستگاه ساده فیزیکی را با استفاده از فناوری توکار و قدرت پردازشی به یک دستگاه هوشمند تبدیل کند. با استفاده از حسگرها و عملگرهای موجود برای تضمین قابلیت‌های دستگاه، می‌توان اطلاعات موجود را، بین دستگاه‌ها به اشتراک گذاشت تا تجربه کاربری بهبود یابد. [6]

نمونه‌هایی از جمله خانه‌های هوشمند که کاربران می‌توانند ترموستات یا چراغ‌های خانه را با گوشی هوشمند خود، کنترل کنند، نمونه‌هایی اساسی از اینترنت اشیاء هستند. اینترنت اشیاء با حضور در تجربیات روزمره ما از جمله در حمل‌ونقل، سلامت و اتوماسیون صنعتی می‌تواند نقش بسزایی را در بهبود کیفیت زندگی ما بازی کند. [6]

پروژه‌های اینترنت اشیاء می‌تواند قابلیت‌هایی فراتر از اتصال یک دستگاه به اینترنت را فراهم آورد. آنها می‌توانند با اضافه‌کردن قابلیت‌های جدیدی نظیر هوش مصنوعی به همه‌ی اشیای متداول، افق جدیدی را آشکار سازند. [6]

## حوزه‌های استفاده

با وجود پیشرفت روزافزون فناوری و در نتیجه اینترنت اشیاء، دامنه کاربرد آن نیز وسیع‌تر خواهد شد. گرچه امروزه هم نام‌بردن تمام حوزه‌هایی که اینترنت اشیاء در آن مورداستفاده است کاری بسیار وقت‌گیر است. بااین‌وجود برای کامل شدن این فصل به چندی از حوزه‌های پرکاربرد که اینترنت اشیاء در آن نفوذ کرده است اشاره خواهد شد:

* **خانه هوشمند:** کنترل از راه دور و خاموش و روشن‌کردن لوازم‌خانگی نظیر چراغ‌ها، سامانه‌های سرمایشی و گرمایشی و ... / پایش متغیرهای محیطی نظیر دما، رطوبت، فشار هوا و ... / مدیریت تجهیزات امنیتی نظیر دوربین و آژیر / تشخیص باز و بسته‌شدن درب و پنجره‌ها / پایش میزان مصرف انرژی و آب و گاز و مواردی نظیر اینها
* **شهر هوشمند:** پایش سلامتی سازه‌ها و ساختمان‌ها / کنترل چراغ‌ها باتوجه‌به میزان روشنایی و آب‌وهوا / پایش ترافیک از طریق دوربین‌های نظارتی، پارک هوشمند / مدیریت پسماند و سطل‌های زباله
* **محیط‌زیست هوشمند:** پایش آلودگی هوا و میزان آلاینده‌ها / تشخیص حوادث زیست‌محیطی نظیر زلزله، سیل، آتشفشان، آتش‌سوزی و ... / پایش کیفیت آب / پایش آب‌وهوا و متغیرهای محیطی نظیر دما، رطوبت و فشار / ردیابی و محافظت از گونه‌های جانوری
* **صنایع هوشمند:** پایش و مدیریت گلخانه‌ها از راه دور / آبیاری خودکار باغ‌ها و مزارع / پایش کیفیت خاک / ردیابی حیوانات مزارع / تشخیص نشت گاز یا مواد سمی در کارخانه‌ها / حفاظت از کارگران با پایش متغیرهای محیطی نظیر میزان اکسیژن هوا / پایش و مدیریت خطوط تولید از راه دور [1]

و بسیاری موارد دیگر!

## ویژگی‌ها

* **اتصال:** یکی از ویژگی‌های کلیدی که پتانسیل زیادی از اینترنت اشیاء مربوط به آن است، قابلیت اتصال است. اشیاء موجود در این شبکه بسته به ماهیت مسئله و نیازمندی‌ها می‌توانند از طیف گوناگونی از روش‌های ارتباطی بهره‌مند باشند. اتصال در شبکه‌های اینترنت اشیاء می‌تواند از نوع محلی و بین دستگاه‌های واقع در یک شبکه صورت بگیرد یا به‌صورت فراگیر و بین دستگاه و شبکه خارجی و ... باشد.
* **ناهمگونی:** دستگاه‌های موجود در اینترنت اشیاء از نظر سخت‌افزار، سکو، سیستم‌عامل و شبکه‌های ارتباطی قابل‌استفاده از هم متفاوت هستند.
* **بزرگ مقیاسی:** اگرچه در خیلی از موارد شبکه‌های کوچک و محلی از اشیای متصل به اینترنت نیز مورداستفاده قرار می‌گیرد (مثلاً در خانه هوشمند) اما با اتصال همه‌ی این شبکه‌ها به هم شبکه‌های بسیار بزرگی تشکیل می‌شود که برقراری امنیت و کنترل و پردازش داده‌های آنها به مسئله‌ای جدی تبدیل می‌شود. علاوه بر این امروزه مشاهده می‌کنیم که تعداد اشیای متصل به کل شبکه اینترنت از تعداد انسان‌ها نیز فراتر رفته است!
* **خدمات مرتبط با اشیاء:** می‌توان با توسعه سرویس، ویژگی‌های مختص به هر شیء را مورداستفاده قرار داد.
* **پویایی:** دستگاه‌های موجود می‌تواند در وضعیت‌های متفاوتی نسبت به هم باشند. مثلاً متصل، قطع و ... . علاوه بر این تعداد دستگاه‌های متصل و همچنین ویژگی‌های تعریف شده برای دستگاه می‌تواند در هر لحظه تغییر کند.
* **امنیت:** بحث امنیت همواره در شبکه‌های ارتباطی یکی از موضوعات مهم و بحث‌برانگیز بوده است. اینترنت اشیاء با وجود رشد روزافزون و اتصال دستگاه‌های بیشتر و بیشتر به آن نیز نه‌تنها از این قاعده مستثنی نیست بلکه یکی از مسائل چالش‌برانگیز موجود در این اکوسیستم است.

## چالش‌ها

* **حریم شخصی و امنیت:** فراهم‌آوردن اعتماد در راستای به‌اشتراک‌گذاری محتوا / تبادل امن داده‌ها / امنیت ارتباطات / حفاظت از دستگاه‌های آسیب‌پذیر / حفظ اصل محرمانگی اطلاعات / احراز هویت / تعیین سطوح دسترسی و ...
* **هزینه در برابر قابلیت استفاده:** هزینه‌ی معقول جهت راه‌اندازی یک سامانه اینترنت اشیاء با قابلیت پایش محیط، رهگیری و کنترل
* **قابلیت سازگاری:** پشتیبانی از سکوهای مختلف / پشتیبانی از پروتکل‌های ارتباطی مختلف / قابلیت ارتباط عناصر ناهمگون با یکدیگر / پشتیبانی و انعطاف‌پذیری سامانه در برخورد با ناهمگونی‌های سخت‌افزاری، ارتباطی و اطلاعاتی (قابلیت‌های ارتباطی، نوع شبکه، فرمت داده و ...)
* **مدیریت داده:** روش‌های ذخیره‌سازی داده / فیلتر داده‌ها / سازگاری داده‌ها / تجزیه‌وتحلیل / بایگانی داده‌ها / تمیزکردن داده‌ها / پردازش بلادرنگ داده‌ها / یکپارچه‌سازی داده‌ها و ...
* **مصرف انرژی:** اتصال اشیاء به هم با درنظرگرفتن محدودیت‌های انرژی دستگاه‌ها / بهینه‌سازی پروتکل‌های ارتباطی / توسعه فناوری‌های ارتباطی کم‌مصرف‌تر و ... [1]

## معماری

اینترنت اشیاء به‌عنوان یک زیرساخت و اکوسیستم بزرگ، از لایه‌های مختلفی از فناوری‌ها تشکیل شده است. شکل زیر نشان می‌دهد که چگونه عناصر این اکوسیستم در کنار هم اینترنت اشیاء را تشکیل می‌دهند و هرکدام از آنها چه ابعادی از نظر ارتباط، مقیاس، ماژولار بودن و قابل تنظیم بودن را دارا هستند. [1]

### ****لایه A:**** ****لایه دستگاه‌ها (سنسور‌ها)****

لایه دستگاه/سنسور هوشمند به‌عنوان پایین‌ترین لایه، از اشیای هوشمندی تشکیل شده است که با حسگرها یکپارچه شده‌اند. حسگرها با جمع‌آوری و پردازش اطلاعات به‌صورت بلادرنگ، ارتباط بین دنیای فیزیکی و دنیای دیجیتال را فراهم می‌کنند. انواع مختلفی از حسگرها برای مقاصد گوناگون وجود دارند. به‌عنوان نمونه برخی از آنها قابلیت اندازه‌گیری مواردی نظیر دما، کیفیت هوا، سرعت، رطوبت، فشار هوا و ... را دارند. در برخی موارد بعضی از آنها به سطحی از حافظه مجهزند که آنها را قادر می‌سازد تا حد معینی از آخرین داده‌های اندازه‌گیری شده را درون خود نگهداری کنند. [1]

بسیاری از حسگرها نیازمند اتصال به دروازه هستند. این اتصال می‌تواند در اشکال مختلفی از شبکه‌های محلی مانند اترنت، وای‌فای، زیگبی، بلوتوث و ... برقرار شوند. برای حسگرهایی که نیازمند به تجمیع کننده‌های حسگر نیستند، این ارتباط می‌تواند یک ارتباط مستقیم با سرور از طریق شبکه‌های گسترده مانند GSM، GPRS و LTE باشد. حسگرهایی که قدرت کمتری دارند و نرخ انتقال داده آنها کم است معمولاً نوعی از شبکه‌های موسوم به شبکه‌های بی‌سیم حسگر را تشکیل می‌دهند. این نوع حسگرها معمولاً در تعداد فراوانی با فواصل فیزیکی نسبتاً محدود از هم قرار می‌گیرند و در کنار هم به تبادل حجم عظیمی از داده‌ها بین خودشان و سرور می‌پردازند. [1]

### ****لایه B: لایه دروازه و شبکه****

عناصر لایه A در کنار هم علاوه بر تعدد در شیوه‌های ارتباطی، با تولید مستمر داده و انتقال بلادرنگ آنها به سمت سرور مرکزی، نیازمندی‌هایی را رقم بزنند که وظیفه رفع آن به عهده یک‌لایه بالاتر از آن یعنی لایه دروازه و شبکه است. با وجود نیاز به سرویس‌دهی به طیف وسیعی از سرویس‌ها و اپلیکیشن‌های اینترنت اشیاء نظیر سرویس‌های تراکنش پرسرعت، برنامه‌های کاربردی آگاه از متن و غیره نیاز به شبکه‌های با فناوری‌ها و پروتکل‌های دسترسی متفاوت را رقم می‌زنند. یکی از چالش‌های مهم در این لایه، پیکربندی این شبکه ناهمگون است. از جمله ابعاد مهم دیگر در این شبکه‌ها، توانایی پشتیبانی آنها از نیازمندی‌هایی نظیر نرخ تأخیر، پهنای باند، امنیت و دردسترس‌بودن است. [1]

از جمله راهکارهای متداول برای پیکربندی این لایه، استفاده از دروازه‌ها است که بنا بر موضوعیت آن در این پروژه، به طور مفصل در فصل‌های بعد مورد بررسی قرار خواهد گرفت.

### ****لایه C: لایه مدیریت و سرویس****

این لایه با امکاناتی نظیر تجزیه‌وتحلیل، کنترل‌های امنیتی، مدل‌سازی و مدیریت دستگاه‌ها، پردازش داده‌ها را ممکن می‌سازد. [1]

یکی از مهم‌ترین ویژگی‌های لایه سرویس مدیریت، موتور پردازش قواعد است. اینترنت اشیاء با فراهم‌آوردن اتصال و ارتباط بین اشیاء و سامانه‌ها باعث ایجاد اطلاعات در شکل رویدادها و داده‌های زمینه محور می‌شود. دمای مطلوب و نامطلوب، موقعیت فیزیکی فعلی، داده‌های ترافیکی و ... همه مثال‌هایی از این داده هستند که نیازمند هستیم تا آنها را فیلتر کرده و برحسب مورد به سمت سامانه‌های دیگر رهنمون کنیم. همچنین برخی از این رویدادها که معمولاً با امنیت و جان افراد مرتبط هستند، نیازمند پاسخ فوری سامانه هستند. موتور قواعد با ضابطه‌مند کردن منطق تصمیم‌گیری‌ها و پاسخ به نیازمندی‌ها، یکی از کلیدی‌ترین نقش‌های موجود در این لایه، یعنی پردازش و تصمیم‌گیری را بر عهده دارد. [1]

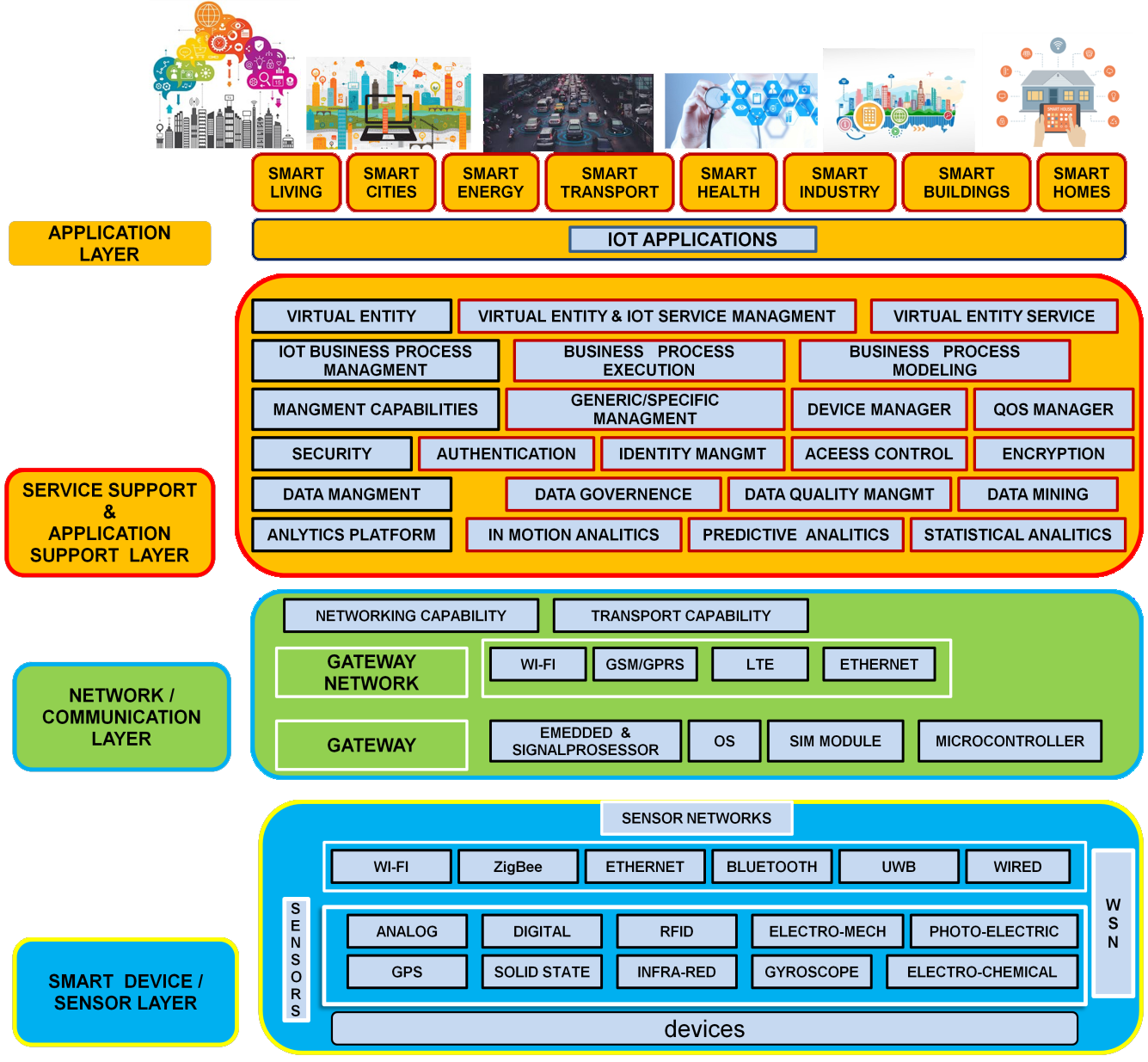
تجزیه‌وتحلیل داده‌ها یکی دیگر از ابعاد مهم این لایه است. معمولاً تجزیه‌وتحلیل در سطوح مختلفی از انتزاع در سامانه‌های اینترنت اشیاء ظاهر می‌شود. در پایه‌ای‌ترین سطح تجزیه‌وتحلیل داده‌ها به‌صورت بلادرنگ و در حافظه موقت صورت می‌پذیرد که به دلیل سادگی و سرعت آن، یکی از روش‌های پرطرف‌دار در میان توسعه دهندگان است و همچنین قابلیت تصمیم‌گیری بلادرنگ را نیز فراهم می‌کند. در مقابل و در سطوح بالاتر تجزیه‌وتحلیل می‌تواند با استفاده از الگوریتم‌های داده‌کاوی نظیر دسته‌بندی و خوشه‌بندی به کاوش الگوها و استخراج دانش ازروی داده‌های ذخیره شده در پایگاه‌داده سرور بپردازد. [1]

مدیریت داده به‌عنوان یکی دیگر از عناصر این لایه از اهمیت کلیدی برخوردار است. با استفاده از این عنصر می‌توان به دسترسی، یکپارچه‌سازی و کنترل داده‌ها اهتمام ورزید. یکی از مزایای مهم مدیریت داده، محافظت از نیاز به پردازش داده‌های نامرتبط یا کم‌اهمیت در سطوح بالاتر سامانه است. روش‌های گوناگون فیلتر داده‌ها نظیر بی‌نام سازی، همگام کردن و یکپارچه کردن از جمله روش‌هایی است که علاوه بر خلاصه‌سازی داده‌ها باعث می‌شود که تنها اطلاعات موردنیاز هر سرویس در دسترس آن باشد. [1]

امنیت به‌عنوان یکی دیگر از وجه‌های مهم معماری اینترنت اشیاء مطرح است که باید در تک‌تک لایه‌ها از لایه حسگرها و دستگاه‌ها گرفته تا لایه اپلیکیشن اعمال شود. [1]

### ****لایه D: لایه اپلیکیشن****

این لایه فضایی که را هوشمند تلقی می‌شود در حوزه‌های گوناگون اعم از: حمل‌ونقل، ساختمان، شهر، سبک زندگی، فروشگاه، کشاورزی، کارخانه، زنجیره تأمین، اورژانس، سلامت، فرهنگ، گردشگری، محیط‌زیست و انرژی خلق می‌کند. به عبارتی دیگر این لایه ارائه‌دهنده سرویس‌ها به کاربران اینترنت اشیا است که با استفاده از داده‌ها و امکانات فراهم شده در لایه C، ارتباط کاربران را با اشیاء میسر می‌سازد. [1]



شکل ‏1‑2

## نتیجه‌گیری

### مقدمه

اینترنت اشیاء دنیایی از اشیای هوشمند است. اشیایی که با بهره‌گیری از حسگرها، محرک‌‌ها و قابلیت اتصال توانایی جمع‌آوری داده‌های محیطی را در یک سامانه نرم‌افزاری فراهم می‌آورد. این سامانه نرم‌افزاری که تحت عنوان سرور مرکزی یا Backend از آن یاد می‌شود، می‌تواند قابلیت‌های فراوانی از جمله نگهداری، تجزیه‌وتحلیل، مدیریت، تصمیم‌گیری و ... را فراهم کند.

### تاریخچه

اگرچه تاریخچه اینترنت اشیاء به سال ۱۹۱۲ در پی اقداماتی نظیر دورسنجی نیروگاه‌ها از طریق خطوط تلفن برمی‌گردد اما شروع‌کننده موج استفاده از اینترنت اشیاء به شکل صنعتی را می‌توان شرکت زیمنس دانست که با آغاز معرفی اولین ماژول شبکه سلولی در سال 1995، هزینه‌های پیاده‌سازی را به شکل قابل‌ملاحظه‌ای کاهش داد.

### تعریف

اگرچه در طول تاریخ تعاریف متعددی برای اینترنت اشیاء صورت پذیرفته است، اما یکی از تعریف‌های به‌روز و جامع تعریفی است که در دانشنامه ویکی‌پدیا ذکر شده است:

«اشیای فیزیکی (گروهی از آنها) که شامل حسگرهای تعبیه شده، قابلیت پردازش، نرم‌افزار و دیگر فناوری‌هایی هستند که آنها را قادر می‌سازد تا از طریق اینترنت یا شبکه‌های ارتباطی دیگر به دستگاه‌ها و سامانه‌های دیگر وصل شوند و به تبادل داده بپردازند.» [5]

### فناوری‌های اساسی مورداستفاده

در قلب صنعت اینترنت اشیاء فناوری‌هایی وجود دارد که در کنار یکدیگر باعث پدیدآمدن این شبکه می‌شوند. این فناوری‌ها در سطح پایین شامل سخت‌افزارهایی نظیر حسگرها، محرک‌ها، میکروکنترلرها، ریزپردازنده‌ها برای جمع‌آوری و پردازش اولیه داده‌ها و ماژول‌های ارتباطی مثل اترنت، USB، مادون‌قرمز، امواج رادیویی، وای‌فای، بلوتوث، شبکه سلولی و ... برای تبادل اطلاعات است. در سطوح بالاتر شبکه‌ها، دروازه‌ها، سرورهای میانی و در نهایت سرور مرکزی یا Backend قرار دارد که وظیفه برقراری ارتباطات گسترده‌تر و همچنین پردازش ثانویه اطلاعات و ارسال دستورات مدیریتی و کنترلی را بر عهده دارند.

### قابلیت‌ها

با پیشرفت روزافزون فناوری، بر قابلیت‌های اینترنت اشیاء نیز افزوده می‌شود. برخی از قابلیت‌های کلیدی که در پروژه‌های اینترنت اشیاء به‌وضوح به چشم می‌خورد عبارت‌اند از:

* **دورسنجی**: اندازه‌گیری و مشاهده مقادیر حسگرها (دما، رطوبت، فشار، سرعت، میزان گازها و ...) از راه دور
* **کنترل از راه دور**: مشاهده وضعیت عنصر مختلف نظیر لوازم روشنایی، لوازم‌خانگی، سامانه‌های امنیتی و ... از راه دور و صدور فرمان‌های کنترلی برای خاموش و روشن‌کردن یا دیگر تغییر وضعیت‌ها.
* **تصمیم‌گیری خودکار**: اضافه‌کردن قابلیت تصمیم‌گیری خودکار بر مبنای داده‌های جمع‌آوری شده (مثلاً خاموش‌کردن سیستم گرمایشی در هنگام بالارفتن دما بالاتر از حد معینی)

### حوزه‌های استفاده

دامنه کاربرد اینترنت اشیاء بسیار وسیع است و با پیشرفت فناوری، هر چه بیشتر نیز وسیع خواهد شد. برخی از حوزه‌هایی که اینترنت اشیاء امروزه در آن نفوذ زیادی کرده است عبارت‌اند از: خانه هوشمند، شهر هوشمند، محیط‌زیست هوشمند و صنایع هوشمند. اندازه‌گیری، پایش و کنترل متغیرهای محیطی، اشیاء و دستگاه‌ها، انرژی، آب و گاز از جمله موارد استفاده اینترنت اشیاء در هر حوزه است.

### ویژگی‌ها

اینترنت اشیاء دارای ویژگی‌های مهمی است از جمله:

* **اتصال**: اتصال دستگاه‌ها به یکدیگر (ارتباطات درون‌شبکه‌ای) و اتصال دستگاه‌ها به دستگاه‌ها در شبکه‌های دیگر (ارتباطات برون شبکه‌ای) باعث شکل‌گیری شبکه‌ای پویا از فناوری‌های سخت‌افزاری و نرم‌افزاری می‌شود که از آن به‌عنوان اینترنت اشیاء یاد می‌کنیم.
* ناهمگونی: دستگاه‌های موجود در اینترنت اشیاء از نظر سخت‌افزار، سکو، سیستم‌عامل و شبکه‌های ارتباطی قابل‌استفاده از هم متفاوت هستند.
* **بزرگ مقیاسی**: اگرچه شبکه‌های کوچک اینترنت اشیاء بسیار مورداستفاده است اما اتصال همین شبکه‌های کوچک به یکدیگر باعث به‌وجودآمدن شبکه‌ای بزرگ‌مقیاس می‌شود که مدیریت آن را به مسئله‌ای فراگیر تبدیل می‌کند.
* **پویایی**: امکان تعریف وضعیت‌های مختلف برای هر دستگاه، کم و اضافه‌شدن تعداد و انواع دستگاه‌های متصل به شبکه و مواردی ازاین‌قبیل مواردی است که این شبکه پویا را شکل می‌دهد.
* **امنیت**: وجود فناوری‌ها و پروتکل‌های مختلف ارتباطی، محدودیت‌های پردازشی و انرژی در دستگاه‌های کم‌مصرف و رشد روزافزون تعداد اتصالات در اینترنت اشیاء مسئله امنیت را به یک مسئله جدی در اینترنت اشیاء مبدل می‌سازد.

### چالش‌ها

اینترنت اشیاء نیز مانند هر فناوری دیگر در کنار ارائه مزایا و پتانسیل‌های منحصر‌به‌فرد، چالش‌هایی را نیز فراهم می‌آورد. رعایت حریم شخصی و برقراری امنیت، هزینه بالای پیاده‌سازی، برقراری سازگاری میان دستگاه‌ها، سکوها و پروتکل‌های متفاوت، مدیریت داده‌ها (جمع‌آوری، فیلتر، یکپارچه‌سازی، نگهداری و ...) و مدیریت مصرف انرژی از جمله مسائل و چالش‌های نیازمند رسیدگی در این حوزه است.

### معماری

با وجود درنظرگرفتن اشکال مختلف برای معماری اینترنت اشیاء، می‌توان یک معماری چهار لایه‌ای برای همه‌ی سامانه‌های اینترنت اشیاء متصور شد. این ساختار به درک بهتر اینترنت اشیاء و اجزای سازنده آن کمک بسزایی می‌کند.

1. **لایه دستگاه‌ها (سنسور‌ها)**

این لایه به‌عنوان پایین‌ترین سطح لایه (از نظر نزدیکی به سخت‌افزار) از اشیاء، دستگاه‌ها‌، حسگرها و محرک‌ها - که از آنها به‌عنوان اشیای هوشمند یاد می‌شود - تشکیل شده است. این اشیای هوشمند قادر به اندازه‌گیری متغیرهای محیطی (مثل دما، رطوبت، فشار، سرعت و ...) هستند. می‌توانند با دریافت دستورات، اقدام به فعال و غیرفعال‌کردن محرک‌ها (مثل خاموش و روشن‌کردن) کنند و از طریق شبکه‌های ارتباطی متنوع بی‌سیم و با سیم (مثل USB، بلوتوث، وای‌فای، زیگبی، LTE و ...) به تبادل اطلاعات و دستورات با لایه‌های بالاتر بپردازند.

1. **لایه شبکه (دروازه)**

این لایه به‌عنوان لایه میانی نقشی واسط بین لایه A و C را بازی می‌کند. قابلیت‌های پردازشی و ارتباطی متنوع برای تبادل داده با نرخ تأخیر مناسب، پهنای باند کافی، امن و دسترسی‌پذیری با وجود ناهمگونی از ویژگی‌های لازمه این لایه است.

1. **لایه مدیریت و سرویس**

این لایه که یک سطح انتزاعی بالاتر از لایه قبل را فراهم می‌کند، وظایف بزرگی نظیر پردازش نهایی داده‌ها، انتقال پیام‌ها به بخش‌های مربوطه، تصمیم‌گیری خودکار، پاسخ به نیازمندی‌ها، تجزیه‌وتحلیل داده‌ها، یکپارچه‌سازی، کنترل و ذخیره‌سازی داده‌ها و دیگر موارد مدیریتی نظیر این را بر عهده دارد.

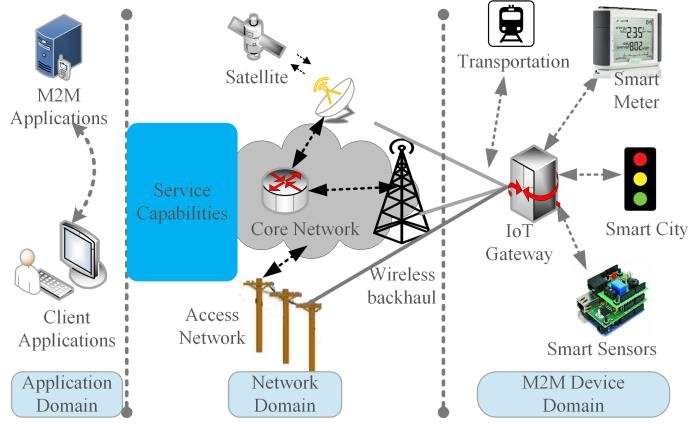
1. **لایه اپلیکیشن**

این لایه به‌عنوان نزدیک‌ترین لایه به کاربر، وظیفه ارائه سرویس‌های کاربردی مدنظر کاربر با استفاده از داده‌ها و امکانات لایه C بر عهده دارد.

# دروازه اینترنت اشیاء

## مقدمه

باتوجه‌به آموخته‌های فصل قبل در مورد معماری اینترنت اشیاء، با یکی کردن لایه مدیریت و لایه اپلیکیشن می‌توان معماری اینترنت اشیاء را در ۳ حوزه، حسگر، شبکه و اپلیکیشن خلاصه کرد که در شکل زیر نمایش‌داده‌شده است. [7]



شکل ‏2‑1

در فصل قبل آموختیم، لایه شبکه (دروازه) به‌عنوان یک لایه میانی نقش واسط را بین لایه‌های دیگر بازی می‌کند. این لایه میانی با ارائه قابلیت‌های پردازشی و ارتباطی متنوع تلاش می‌کند تا ارتباطی دوطرفه با نرخ تأخیر پایین، پهنای باند کافی، حفظ امنیت و محرمانگی اطلاعات، دسترسی‌پذیری و نرخ آماده‌به‌کار بودن بالا فراهم کند.

در این فصل می‌خواهیم معماری دروازه‌های اینترنت اشیاء را به‌عنوان یکی از اجزای کلیدی و پلی رابط میان لایه حسگر و لایه مدیریت را که توانایی رسیدگی به بسیاری از چالش‌های مطرح شده در فصل قبل دارد، به‌تفصیل بررسی کنیم. برای این کار ابتدا لازم است ابتدا اشاره‌ای به شبکه‌های درگیر در دو سوی دروازه که متعلق به لایه‌های حسگر و مدیریت است، داشته باشیم.

همان‌طور که گفته شد اجزای لایه حسگر از گوناگونی خاصی از نظر امکانات و پروتکل‌های ارتباطی برقرارند. لایه مدیریت نیز در سمت دیگر اگرچه شاید به‌اندازه لایه حسگر تنوع نداشته باشد، اما بازهم ممکن است از شبکه‌های مختلف ارتباطی پشتیبانی کند. انواع ارتباطات باسیم و بی‌سیم قدیمی و مدرن ممکن است در دستگاه‌ها تعبیه شده باشد و دروازه باید بتواند علاوه بر دریافت پیام‌ها از تمامی این دستگاه‌ها، بتواند آنها را تجزیه کرده، پردازش کند و برای ارسال به شبکه یا شبکه‌های مربوطه در لایه مدیریت، تبدیل پروتکل و بسته‌بندی مجدد انجام دهد؛ بنابراین حداقل عناصر لازم برای طراحی و توسعه دروازه، ماژول‌های تجزیه و تبدیل داده و پروتکل‌ها هستند. علاوه بر این نیازهای مدیریتی و اصول امنیتی و محرمانگی اطلاعات ایجاب می‌کند تا اجزای دیگری نظیر ماژول نمایه اپلیکیشن، ماژول نمایه شبکه، ماژول هدایت پیام‌ها و امکانات کاربردی نظیر رمزنگاری، پیکربندی، ثبت وقایع و ... را نیز در طراحی دروازه مدنظر قرار دهیم.

در این فصل ابتدا سه شبکه متداول که در حوزه حسگر مورداستفاده قرار می‌گیرد، معرفی می‌شود. سپس ویژگی‌های معمول یک دروازه اینترنت اشیاء - به‌عنوان جزء اصلی متصل‌کننده حوزه حسگر و حوزه شبکه - در سه بخش خلاصه می‌شود. در پایان فصل نیز یک مدل مرجع برای دروازه اینترنت اشیاء توصیف می‌شود. [7]

## شبکه‌های حوزه حسگر

اشیای هوشمند ممکن است شبکه‌هایی محلی شکل دهند که از طریق دروازه‌های اینترنت اشیاء به زیرساخت ارتباطات عمومی متصل می‌شود. در این بخش سه نوع از معماری‌های شبکه که در حوزه حسگر مطرح است مورد بررسی قرار گرفته است: [7]

### **شبکه شخصی (PAN)**

این شبکه اجازه شکل‌گیری ارتباطات شفاف، امن و قابل‌اطمینان را میان اشیای مجهز به حسگر که در اطراف یک شخص قرار دارند، می‌دهد. شبکه‌های شخصی می‌توانند از طریق با‌سیم مثل USB به گذرگاه کامپیوتر متصل شوند. شبکه‌های بی‌سیم شخصی (WPAN) نیز با فناوری‌هایی نظیر بلوتوث، USB بی‌سیم، زیگبی و ... مورداستفاده قرار می‌گیرند. در شبکه‌های شخصی معمولاً یک تلفن هوشمند یا لپ‌تاپ و یا یک سخت‌افزار اختصاصی به‌عنوان دروازه عمل می‌کند و کنترل سیگنال‌ها و داده‌هایی که از شبکه شخصی می‌آیند یا به شبکه شخصی می‌روند را به عهده دارد. [7]

### **شبکه وسیله نقلیه**

شبکه‌هایی هستند که درون وسایل متحرک نظیر اتومبیل و قطار قرار دارند و به تبادل داده‌های وسیله نقلیه و داده‌های اشیای هوشمند داخل آن می‌پردازند. در این شبکه‌ها معمولاً از فناوری RFID و فناوری‌های تشخیص هویت بی‌سیم برای ارتقای لجستیک حمل‌ونقل، کنترل کیفیت، رهگیری و مدیریت استفاده می‌شود. [7]

### **شبکه خانگی**

شبکه خانگی از حسگرها، محرک‌ها، لوازم‌خانگی هوشمند، تجهیزات اتوماسیون، ارتباطات شبکه‌ای و مانند آن تشکیل می‌شود. [7]

در این شبکه‌ها، دروازه اینترنت اشیاء دو نقش مهم را ایفا می‌کند:

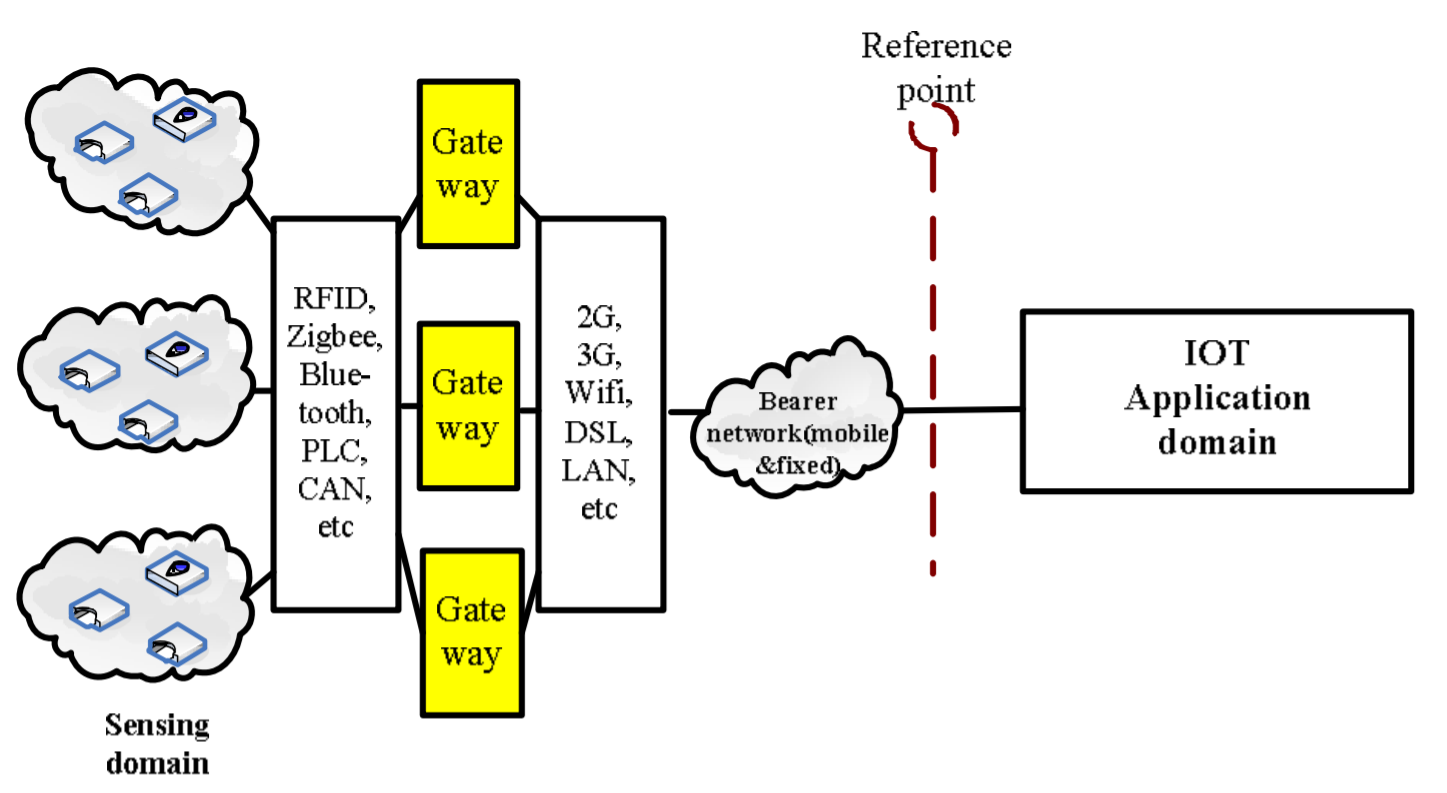
* اتصال چندین دستگاه هوشمند به یکدیگر برای تشکیل یک شبکه خانگی داخلی و اشتراک منابع و اطلاعات بین آنها.
* اتصال شبکه خانگی داخلی به شبکه‌های خارجی و فراهم‌آوردن امکان دسترسی به اجزای یکدیگر [7]

## **شبکه‌های حوزه اپلیکیشن**

به طور معمول در سمت این حوزه از شبکه‌های گسترده محلی، شهری یا جهانی استفاده می‌شود. اینترنت نیز به‌عنوان فراگیرترین شبکه گسترده، یکی از انتخاب‌های فراگیر برای دسترسی به خدمات این لایه است؛ بنابراین جای تعجب ندارد که شبکه‌های سلولی یا همان شبکه‌های داده تلفن همراه (GSM/GPRS/2G/3G/4G/5G)، شبکه انتقال عمومی تلفن (PSTN) و اینترنت ماهواره‌ای از روش‌های متداول دسترسی به اینترنت و همچنین اینترنت اشیاء باشد.

## ویژگی‌ها

شکل زیر موقعیت یک دروازه را در زیرساخت اینترنت اشیاء نشان می‌دهد.



شکل ‏2‑2

دروازه‌ها از اپلیکیشنی به اپلیکیشن دیگر و برای نیازمندی‌های مختلف، تفاوت‌های بسیاری با هم دارند. بااین‌وجود ویژگی‌هایی که به طور گسترده در مورد آن بحث شده است، به شرح زیر است: [7]

### رابط‌های چندگانه

اشیای هوشمند می‌توانند از طریق فناوری‌های مختلفی مثل زیگبی، بلوتوث، وای‌فای و ... به دروازه متصل شوند. در سمت دیگر دروازه نیز انتخاب‌های متفاوتی نظیر PSTN، 2G،3G،LTE،LAN،DSL و ... برای اتصال به شبکه عمومی را دارد. اینکه یک دروازه چه تعداد و چه انواعی از این رابط‌ها را پشتیبانی کند بسته به نیازمندی‌های اپلیکیشن مربوطه، استراتژی‌های عملیاتی و راهکارهای پیاده‌سازی متفاوت است. [7]

### مبدل پروتکل

دروازه‌ها در دو موقعیت مختلف نیاز به تبدیل پروتکل دارند. یکی در زمان ارتباط بین حسگرهای مختلف مثل زیگبی با بلوتوث و دیگری هنگام ارتباط بین حوزه حسگر و حوزه شبکه مثل ارتباط بین زیگبی و 3G. [7]

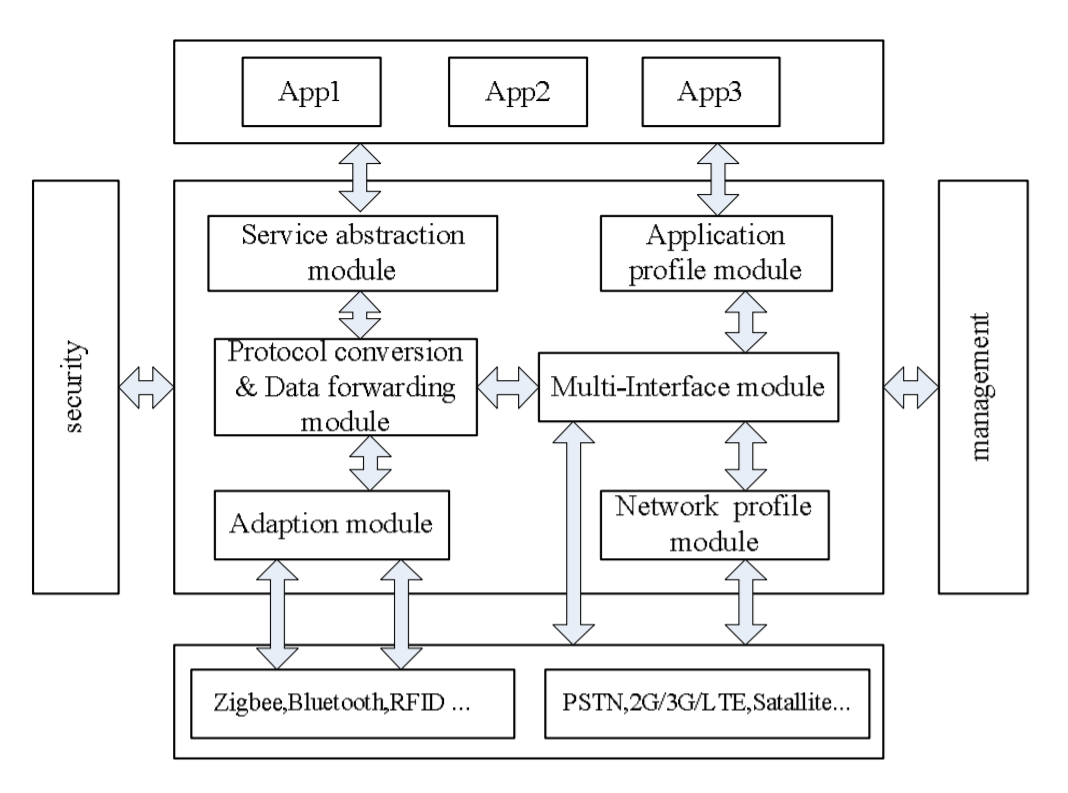
### مدیریت پذیری

اول‌ازهمه نیاز است تا خود دروازه از طریق یک سرور بالادستی مدیریت شود. از جمله موارد نیازمند مدیریت در دروازه می‌توان به اشتراک، احراز هویت، مدیریت وضعیت، مدیریت اتصالات و ... اشاره کرد. علاوه بر این باید بتوان اشیای متصل به دروازه را از طریق آن دروازه مدیریت کرد. دروازه می‌تواند قابلیت تشخیص، کنترل، عیب‌یابی، پیکربندی و نگهداری از این اشیای هوشمند را فراهم کند. [7]

برای جمع‌بندی می‌توان گفت هدف دروازه اینترنت اشیاء ایجاد یک پل بین شبکه‌های متنوع حوزه حسگر (شخصی، وسیله نقلیه و خانگی) با شبکه‌های ارتباطات عمومی (ثابت و متحرک) یا اینترنت با درنظرگرفتن ناهمگونی بین این شبکه‌های مختلف و فراهم‌آوری مدیریت خود دروازه و گره‌های متصل به آن است. [7]

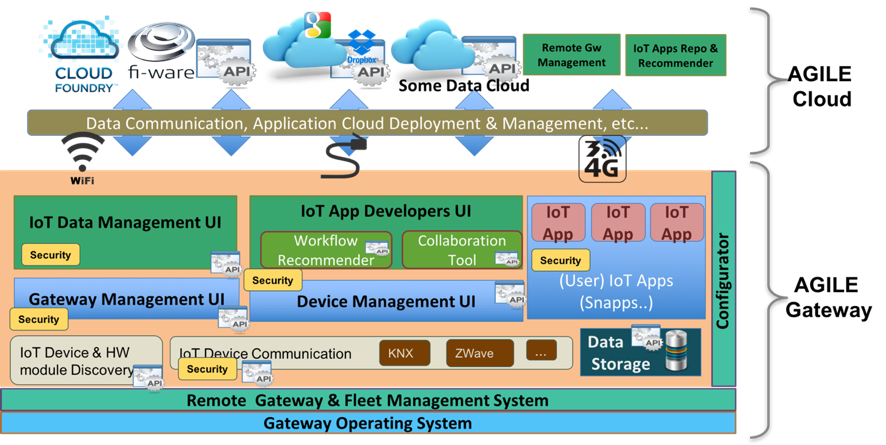
## معماری

اگرچه کاربردهای اینترنت اشیاء بسیار متنوع است و باعث می‌شود تا اشکال گوناگونی از معماری‌ها برای یک دروازه اینترنت اشیاء مورداستفاده قرار گیرد اما شکل زیر می‌تواند به‌عنوان یک معماری مرجع مدنظر قرار گیرد. [7]



شکل ‏2‑3

همچنین شکل زیر معماری یکی از دروازه‌های اینترنت اشیاء که به‌صورت متن‌باز است را نشان می‌دهد.



شکل ‏2‑4

### ماژول انتزاع سرویس

این ماژول برای رسیدگی به قطعه‌قطعه بودن اپلیکیشن‌های اینترنت اشیاء مهم است. این ماژول وظایف متعددی را داراست. در مرحله اول یک رابط یکپارچه را برای ارائه‌دهندگان سرویس فراهم می‌آورد تا روند توسعه اپلیکیشن را ساده کند. در مرحله میانی، این موضوع مطرح است که توسعه‌دهندگان اپلیکیشن‌های اینترنت اشیاء نیاز به فهم عمیق از فناوری‌های سطح پایین (سخت‌افزاری) ندارند و معمولاً سرویس‌های مختلفی را در یک سطح انتزاعی معین توسعه می‌دهند. نهایتاً این ماژول این امکان را فراهم می‌آورد تا شبکه‌های زیرین از بهینه‌سازی‌های فراهم شده در زیرساخت اینترنت اشیاء بهره‌مند شوند. وب‌سرویس، XML، SOA، JSON و Protobuf از فناوری‌های مطرح مورداستفاده در این ماژول است. [7]

### ماژول سازگار کننده

به‌طورکلی، یک دروازه اینترنت اشیاء از پروتکل‌های ارتباطیِ بی‌سیمِ کوتاه‌برد برای دریافت بسته‌های پیام از حسگرها استفاده می‌کند. در سوی دیگر از شبکه‌های سلولی (2G/3G/LTE)، ماهواره‌ای و دیگر شبکه‌ها برای ارتباط با شبکه‌های خارجی یا اینترنت بهره می‌برد. [7]

به هر جهت در مقایسه با ارتباطات سنتی و ارتباطات اینترنتی در می‌یابیم پروتکل‌ها و دستگاه‌های مورداستفاده در حوزه حسگر، به شکلی متفاوت و با دغدغه‌هایی نظیر کاهش هزینه، کاهش انرژی مصرفی و انعطاف‌پذیری در نصب طراحی شده‌اند؛ بنابراین قبل از انتقال پیام‌های دریافتی از حوزه حسگر به شبکه‌های ارتباطی وسیع و اینترنت، نیازمند به سازگار کردن هستیم. ماژول سازگار کننده، بر روی سازگار کردن اندازه پیام‌های شبکه‌های مختلف، تفکیک نشانی، قطعه‌بندی، سرهم‌بندی مجدد و امثال آن تمرکز دارد. [7]

### ماژول مبدل پروتکل و ارسال داده

این ماژول با تعامل با ماژول انتزاع سرویس، ماژول سازگار کننده و ماژول ارتباط چندگانه، به تجزیه‌وتحلیل و بسته‌بندی مجدد داده‌های حسگر بعد از دریافت آن از شبکه‌های کوتاه‌برد می‌پردازند و سپس بعد از کپسوله‌کردن، باتوجه‌به پروتکل ارتباطی، داده با دسته‌بندی مجدد را ارسال می‌کنند. به‌طورکلی این ماژول برای انتقال داده از یک رابط شبکه به یک رابط شبکه دیگر به‌صورت شفاف و درست است. [7]

### ماژول نمایه اپلیکیشن و ماژول نمایه شبکه

این ماژول اطلاعات و فراداده موردنیاز برای کمک به تصمیم‌گیری ماژول ارتباطات چندگانه را فراهم می‌کنند. ماژول نمایه اپلیکیشن، اطلاعات در سطح اپلیکیشن - که نوع و تعداد اپلیکیشن‌هایی را که در حال حاضر در دروازه در حال اجرا هستند - را ارائه می‌دهند. ماژول نمایه شبکه، اطلاعات در سطح رابط شبکه نظیر نشانگر قدرت سیگنال دریافتی، بازه پوشش ایستگاه پایه‌ای و پهنای باند شبکه را به ماژول رابط‌های چندگانه ارائه می‌دهد. [7]

### ماژول امنیت و مدیریت

امنیت و مدیریت همواره دو جزء لازم برای سامانه‌های ICT (فناوری اطلاعات و ارتباطات) هستند. ماژول امنیت، کنترل دسترسی، تعیین هویت، یکپارچگی اطلاعات و حفاظت از حریم شخصی را فراهم می‌کنند. ماژول مدیریت نیز شامل مدیریت شبکه، مدیریت انرژی، مدیریت خطا، مدیریت سطوح دسترسی، مدیریت وضعیت و مدیریت تحرک می‌شود. [7]

## نتیجه‌گیری

### مقدمه

ارتباطات اینترنت اشیاء بسیار از ارتباطات سنتی انسان با انسان متفاوت است و چالش‌های بزرگی را به زیرساخت‌های ارتباطی و اینترنتی موجود می‌آورد. اما مسئله اصلی، یکپارچه کردن، انواع شبکه‌های حسگر و شبکه‌های ارتباط از راه دور و اینترنت است. دروازه اینترنت اشیاء به‌عنوان یک نماینده برای شبکه‌های مختلف می‌تواند کلیدی برای حل این مسئله باشد. با استانداردسازی کامل دروازه اینترنت اشیاء، اپلیکیشن‌های این صنعت می‌تواند سریع‌تر و گسترده‌تر توسعه یابد. [7]

دروازه اینترنت اشیاء به‌عنوان یکی از اجزای کلیدی و پلی رابط میان لایه حسگر و لایه مدیریت مطرح است. اجزای لایه حسگر و لایه مدیریت هر دو از شبکه‌ها و پروتکل‌های مختلف قدیمی و مدرن، باسیم یا بی‌سیم بهره می‌برند. دروازه برای برقراری ارتباط دوطرفه بین این دو به شکلی امن و قابل مدیریت باید از اجزا و ماژول‌های مختلفی از جمله ماژول تجزیه و تبدیل داده و پروتکل‌ها، ماژول نمایه شبکه، ماژول نمایه اپلیکیشن، ماژول هدایت پیام‌ها و دیگر ماژول‌های کاربردی نظیر رمزنگاری، پیکربندی و ثبت وقایع برخوردار باشد.

در این فصل شبکه‌های دو سوی دروازه مورد بررسی قرار گرفت که به طور خلاصه اینجا بررسی می‌شود.

### شبکه‌های حوزه حسگر و حوزه اپلیکیشن

در حوزه حسگر از انواع مختلف شبکه‌های شخصی باسیم (USB) و بی‌سیم (USB Wireless، زیگبی، بلوتوث). گفته شد شبکه‌های وسیله نقلیه (متحرک) نیز که برای مدیریت وسیله نقلیه و اجزای داخل آن استفاده می‌شود نیز نوعی از شبکه‌های مطرح در این حوزه است و از فناوری‌های RFID و دیگر فناوری‌های بی‌سیم برای ارتباطات استفاده می‌کند. شبکه خانگی که با اتصال حسگرها، محرک‌ها، لوازم‌خانگی هوشمند، تجهیزات اتوماسیون و ... به وجود می‌آید نیز از شبکه‌های فراگیر مورداستفاده است که هم برای ارتباطات درون‌شبکه‌ای (با دستگاه‌های متصل دیگر) و هم برای ارتباطات برون شبکه‌ای (شبکه‌های خارجی و گسترده) مورداستفاده است.

از سوی دیگر آموختیم شبکه‌های حوزه اپلیکیشن که نسبتاً تنوع کمتری از حوزه حسگر دارند، از طریق شبکه‌های گسترده به خصوص اینترنت قابلیت دسترسی دارند و همان‌طور که می‌دانیم شبکه‌های سلولی (مانند GSM/2G/.../5G)، شبکه تلفنی (مانند DSL) و اینترنت ماهواره‌ای از جمله روش‌های متداول برای دسترسی به شبکه جهانی اینترنت و اینترنت اشیاء است.

### ویژگی‌ها

اگرچه دروازه‌ها در موقعیت‌ها و نیازمندی‌های مختلف، تفاوت بسیاری با هم دارند، اما برخی ویژگی‌های مشترک را می‌توان در بین همه‌ی آنها دید. وجود رابط‌های چندگانه برای اتصال از طریق فناوری‌های مختلف، مبدل پروتکل برای قابل‌فهم کردن پیام‌ها در دو سوی دروازه و قابلیت‌های مدیریتی برای تشخیص، کنترل، عیب‌یابی، پیکربندی و نگهداری از اجزای شبکه، ازاین‌دست ویژگی‌ها است.

برای جمع‌بندی می‌توان گفت هدف دروازه اینترنت اشیاء ایجاد یک پل بین شبکه‌های متنوع حوزه حسگر (شخصی، وسیله نقلیه و خانگی) با شبکه‌های ارتباطات عمومی (ثابت و متحرک) یا اینترنت با درنظرگرفتن ناهمگونی بین این شبکه‌های مختلف و فراهم‌آوری مدیریت خود دروازه و گره‌های متصل به آن است. [7]

### معماری

اگرچه کاربردهای اینترنت اشیاء بسیار متنوع است و باعث می‌شود تا اشکال گوناگونی از معماری‌ها برای یک دروازه اینترنت اشیاء مورداستفاده قرار گیرد اما شکل 2-3 می‌تواند به‌عنوان یک معماری مرجع مدنظر قرار گیرد.

در این شکل ماژول‌های مختلفی برای دروازه تصور شده است:

* **ماژول سازگار کننده:** ازآنجایی‌که فناوری‌ها و پروتکل‌های ارتباطی حوزه حسگر به‌گونه‌ای متفاوت و با دغدغه‌هایی متنوع طراحی شده‌اند، نیاز است تا برای فهم ساده‌تر این پیام‌ها توسط ماژول‌های دیگر، آن را از نظر اندازه پیام‌، تفکیک نشانی، قطعه‌بندی، سرهم‌بندی و امثال آن آماده کنیم که این وظایف به عهده ماژول سازگار کننده است.
* **ماژول مبدل پروتکل و هدایت داده:** این ماژول وظیفه دارد با کمک بقیه ماژول‌های ارتباطی، پیام‌ها را باتوجه‌به مقاصد گوناگون، با پروتکل ارتباطی هدف، بسته‌بندی مجدد کند و به مقصد مورد‌نظر هدایت کند.
* **ماژول نمایه اپلیکیشن و نمایه شبکه:** این ماژول‌ها اطلاعات و فراداده مربوط به اپلیکیشن و شبکه (مثل نوع اپلیکیشن، تعداد اپلیکیشن‌های در حال اجرا، قدرت سیگنال، پهنای باند، نرخ تأخیر و ...) را جهت فهم بیشتر پیام به ماژول رابط‌های چندگانه ارائه می‌دهد.
* **ماژول انتزاع سرویس**: یک کلاس انتزاعی مختص ارتباط با لایه مدیریت و سرویس ایجاد می‌کند تا نیاز به فهم فناوری‌های سطح پایین‌تر را توسط لایه بالا منتفی کند. معمولاً از وب‌سرویس‌های متداولی همچون SOAP، XML، REST و ... در این ماژول استفاده می‌شود.
* **ماژول امنیت و مدیریت:** ماژول امنیت، کنترل دسترسی، تعیین هویت، یکپارچگی اطلاعات و حفاظت از حریم شخصی را فراهم می‌کند. ماژول مدیریت نیز شامل مدیریت شبکه، مدیریت انرژی، مدیریت خطا، مدیریت سطوح دسترسی، مدیریت وضعیت و مدیریت تحرک می‌شود.

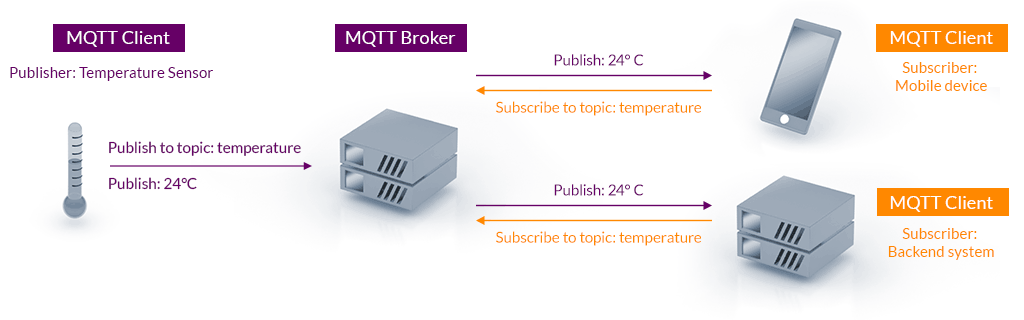
# اصطلاحات و مفاهیم

## پروتکل Mqtt : استاندارد پیام‌رسانی اینترنت اشیا

Mqtt یک پروتکل پیام‌رسانی استاندارد OASIS برای اینترنت اشیا است. این پروتکل به‌عنوان یک روش بسیار سبک[[1]](#footnote-2) پیام‌رسانی بر مبنای انتشار/اشتراک[[2]](#footnote-3) طراحی شده است. Mqtt برای اتصال دستگاه‌های از راه دور ایدئال است چرا که با یک ردپای[[3]](#footnote-4) کوچک از کد و مصرف حداقلی پهنای باند[[4]](#footnote-5) شبکه کار می‌کند. امروزه این پروتکل در صنایع مختلفی از جمله خودرو، تولید، مخابرات، نفت و گاز و ... استفاده می‌شود.

از مزایای استفاده از Mqtt می‌توان به سبکی و کارآمدی، ارتباطات دوطرفه، قابلیت اتصال به میلیون‌ها دستگاه، تحویل قابل‌اطمینان پیام[[5]](#footnote-6)، پشتیبانی از شبکه‌های غیرقابل‌اطمینان، رمزنگاری پیام و ... اشاره کرد.

پروتکل Mqtt از مدل انتشار/اشتراک استفاده می‌کند و در سطح پایین‌تر بر بستر TCP/IP سوار می‌شود. مدل انتشار/اشتراک این امکان را فراهم می‌کند تا بر روی یک بستر ارتباطی چندین کانال مختلف ارتباطی ایجاد کرد. با وجود این قابلیت می‌توان مانند یک ایستگاه پستی با تعریف آدرس‌های مختلف که اصطلاحاً Topic گفته می‌شود، به دسته‌بندی پیام‌ها پرداخت. شکل ۱ یک نمونه از این مدل ارتباطی را نشان می‌دهد.



شکل ‏3‑1

در اینجا یک سنسور دما به‌عنوان ناشر[[6]](#footnote-7) شروع به برقراری ارتباط با سرور مرکزی می‌کند و برای این کار از آدرسی به نام “temperature” استفاده می‌کند. در طرف دیگر هر کاربری می‌تواند با اتصال به سرور و قراردادن این آدرس در پیام خود به‌عنوان یک مشترک[[7]](#footnote-8) اقدام به دریافت پیام‌های منتشر شده از این کانال کند. تا زمانی که این ارتباط غیرمستقیم بین ناشر و مشترک برقرار باشد کلیه پیام‌های منتشرشده توسط ناشر به کلیه مشترکان ارسال می‌شود.

# تحلیل

## مقدمه

ازآنجایی‌که خیلی از موارد مربوط به فاز تحلیل نرم‌افزار در فصل‌های قبل به طور مفصل توضیح داده شد. در این فصل تنها به ذکر نکات مهم این فرایند می‌پردازیم.

توسعه این پروژه بر اساس ترکیبی از مدل RAD همراه با تکنیک‌های پروتوتایپ به‌خصوص MVP و بر مبنای سناریو خانه هوشمند صورت‌گرفته است.

## مدل توسعه سریع اپلیکیشن (RAD)

### مقدمه

RAD یک مدل فرایند توسعه نرم‌افزار متوالی خطی است که بر چرخه توسعه مختصر با استفاده از رویکرد ساختِ مبتنی بر عنصر تأکید دارد. اگر الزامات به‌خوبی درک و توصیف شده باشند و حوزه پروژه محدود باشد، فرایند RAD یک تیم توسعه را قادر می‌سازد تا یک سیستم کاملاً کاربردی را در یک دوره زمانی مختصر ایجاد کند. [8]

### مراحل

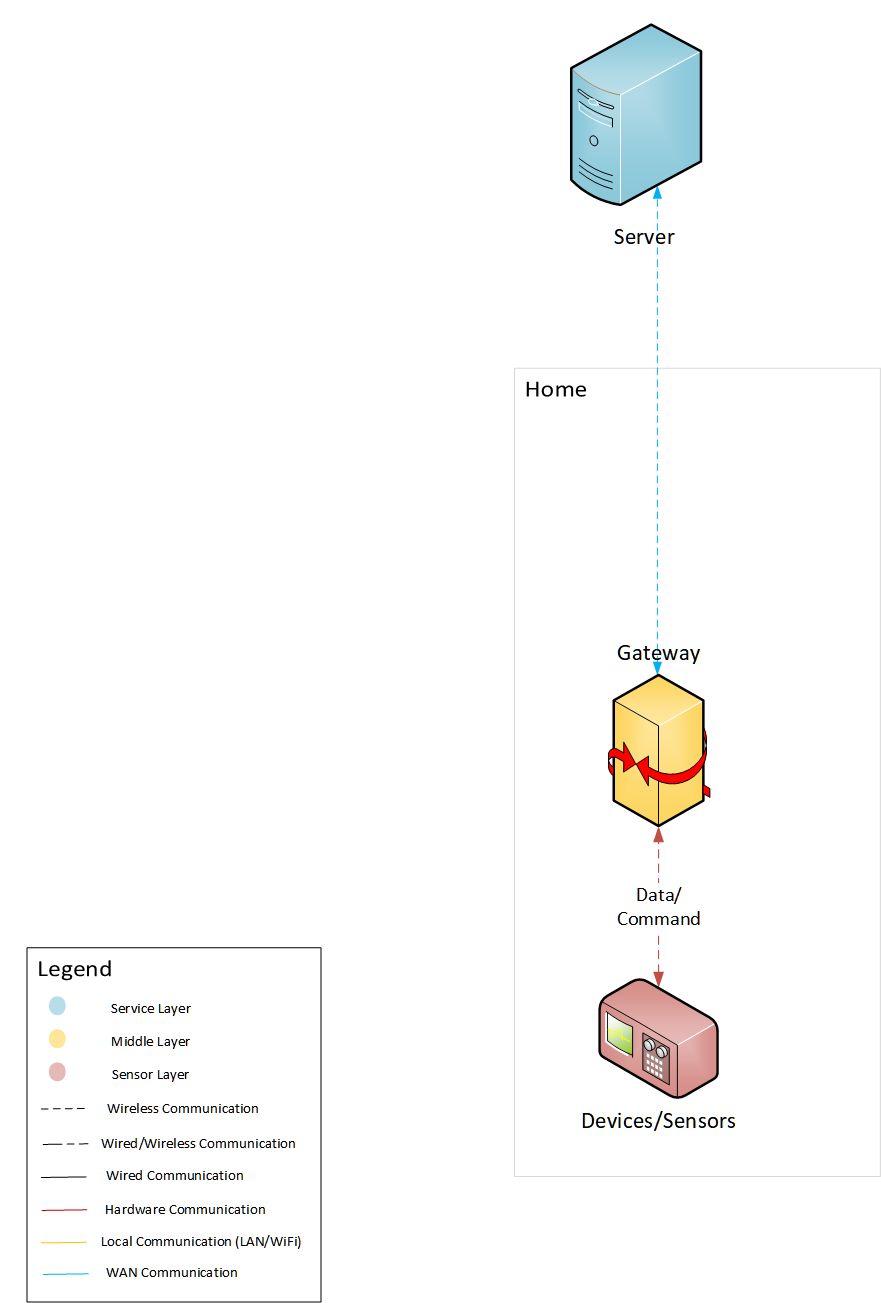
* 1. **مدل‌سازی کسب‌وکار**: جریان اطلاعات در میان کارکردهای کسب‌وکار با پاسخ به سؤالاتی مانند اینکه چه داده‌هایی فرایند کسب‌وکار را هدایت می‌کنند، چه داده‌هایی تولید می‌شوند، چه کسی آن را تولید می‌کند، اطلاعات به کجا می‌رود، چه کسی آن را پردازش می‌کند و غیره تعریف می‌شود.
  2. **مدل‌سازی داده‌ها**: داده‌های جمع‌آوری‌شده از مدل‌سازی کسب‌وکار به مجموعه‌ای از اشیاء داده (موجودات) که برای پشتیبانی از کسب‌وکار موردنیاز هستند، پالایش می‌شوند. صفات (شخصیت هر موجودیت) شناسایی شده و رابطه بین این اشیاء داده (موجودات) تعریف می‌شود.
  3. **مدل‌سازی فرایند**: شیء اطلاعاتی تعریف‌شده در فاز مدل‌سازی داده‌ها برای دستیابی به جریان داده‌های لازم برای اجرای یک عملکرد تجاری تغییر شکل می‌دهند. توضیحات پردازش برای افزودن، اصلاح، حذف یا بازیابی یک شی داده ایجاد می‌شود.
  4. **تولید برنامه**: از ابزارهای خودکار برای تسهیل ساخت نرم‌افزار استفاده می‌شود. برای نمونه می‌توان نمودارهای مدل‌سازی شده را با ابزارهای تولید کد (Code Generation Tools) به برنامه تبدیل کرد.
  5. **تست و گردش**: از آنجایی که RAD تاکید بر استفاده مجدد دارد، بسیاری از اجزای برنامه آزمایش شده‌اند. این باعث کاهش زمان کلی تست می‌شود. اما قسمت جدید باید تست شود و تمام رابط‌ها باید به طور کامل اجرا شوند.

## معماری (Context Diagram)

### مقدمه

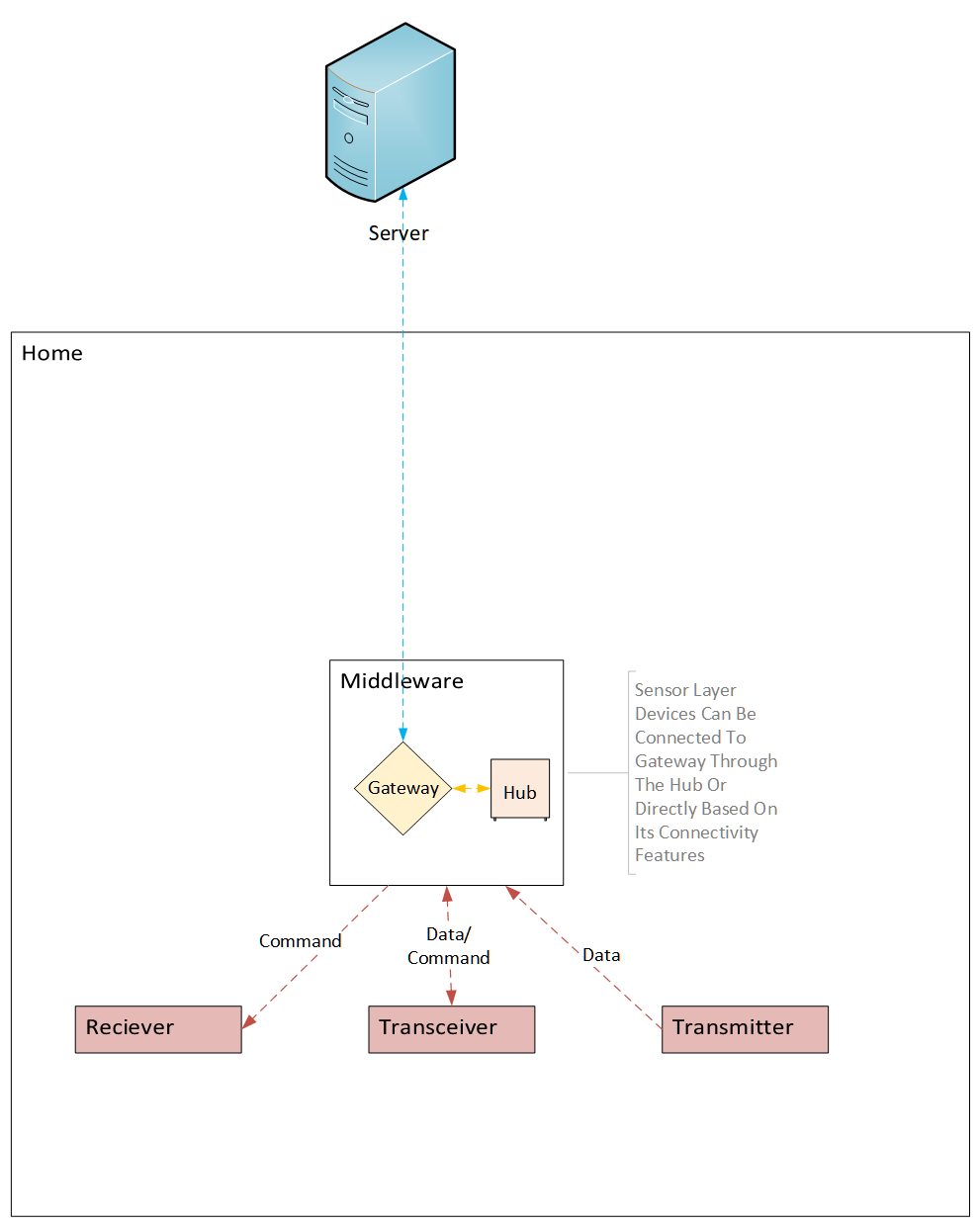
معماری کامل اینترنت اشیاء در یک خانه هوشمند در پیوست معماری خانه هوشمند (arch.pdf) آمده است. یک معماری خلاصه شده که تنها شامل ارتباطات کلیدی با دروازه است در سه سطح مختلف انتزاعی رسم شده است. این نمودار که در پیوست معماری خانه هوشمند (دروازه) (arch-g.pdf) آمده است در این فصل مورد بررسی قرار گرفته است. در این نمودار اشکال آبی نشان‌دهنده لایه سرویس، اشکال زرد نشان‌دهنده لایه میانی (دروازه) و اشکال قرمز نشان‌دهنده لایه حسگر هستند. خطوط قرمز نشان‌دهنده ارتباطات سخت‌افزاری، خطوط زرد نشان‌دهنده ارتباطات محلی و خطوط آبی نشان‌دهنده ارتباطات شبکه گسترده است. همچنین خطوط خط‌چین نشان از ارتباطات بی‌سیم و خطوط ساده نشان‌دهنده ارتباطات باسیم است.

### سطح 0



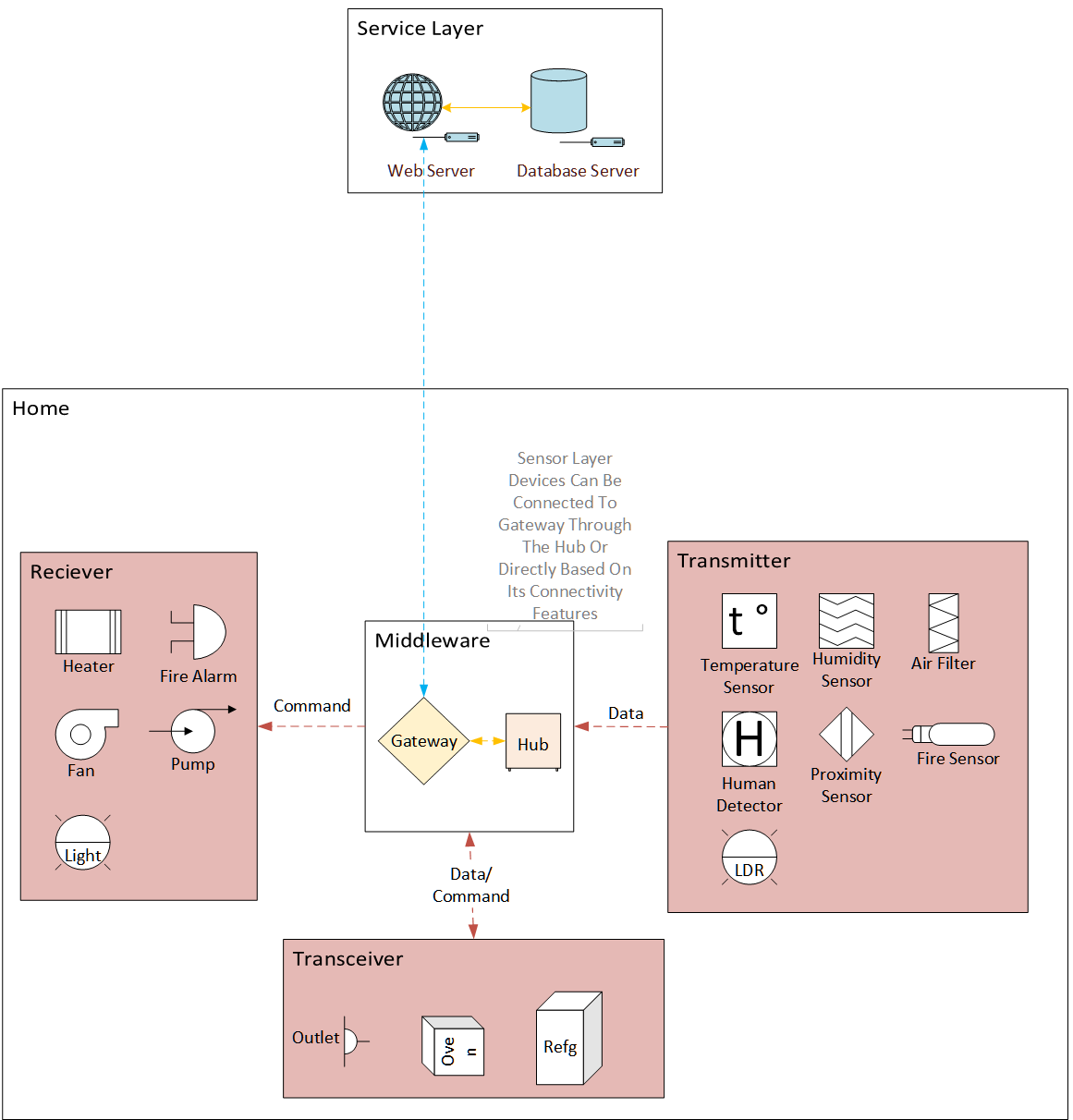
شکل ‏4‑1

### سطح 1



شکل ‏4‑2

### سطح 2



شکل ‏4‑3

### خلاصه

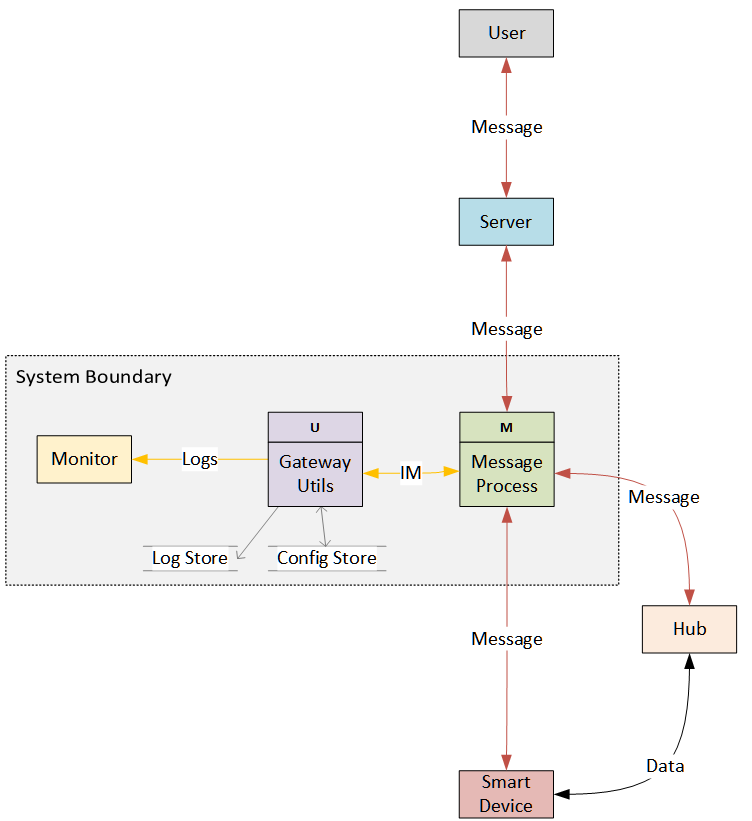
دستگاه‌های موجود در لایه حسگر از نظر ارتباطاتی سه نوع‌اند: فرستنده‌ها، گیرنده‌ها و فرستنده‌/گیرنده‌ها. فرستنده‌ها اطلاعات حسگرها را به سمت دروازه و سرور می‌فرستند. گیرنده‌ها دستورات سرور را دریافت و اجرا می‌کنند و دسته سوم ترکیبی از این دو نوع هستند. همه‌ی این اطلاعات یا به‌صورت مستقیم یا از طریق واسطه (Hub) با لایه میانی و از آن طرف با لایه سرور در ارتباط‌اند.

## نمودار جریان داده (DFD)

### مقدمه

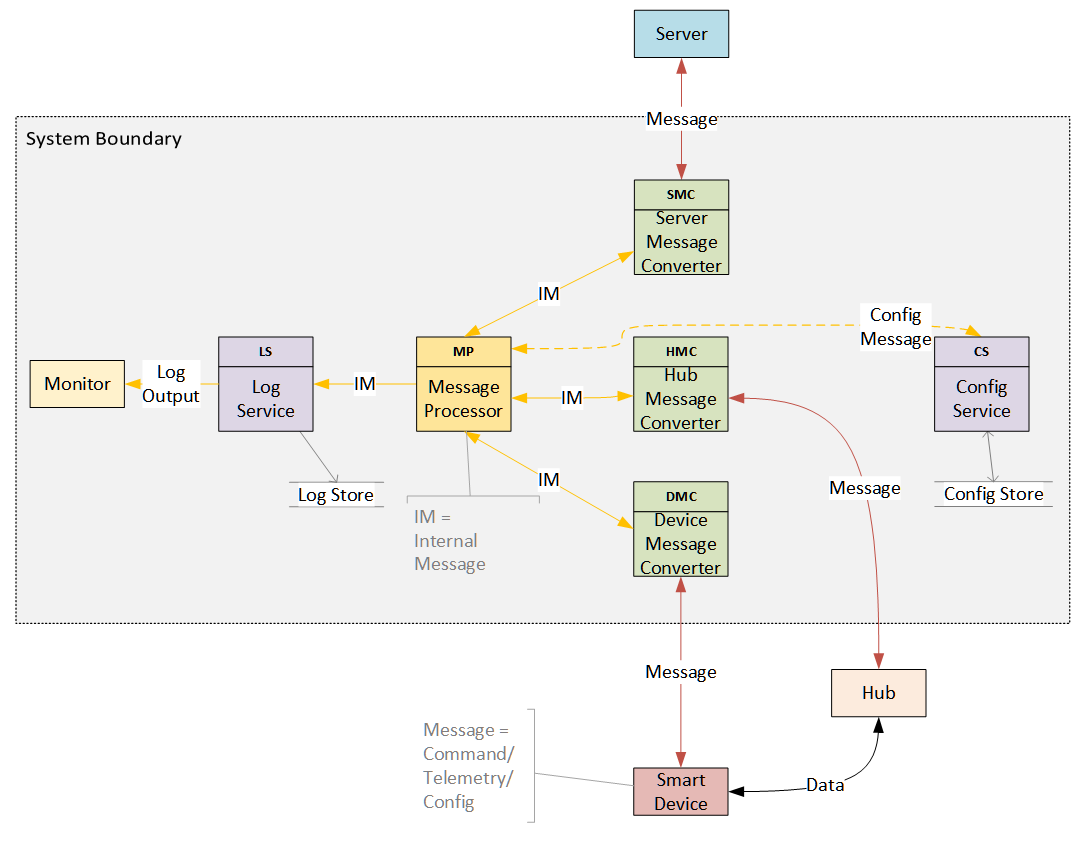
این نمودار به خوبی مسیر حرکت انواع مختلف از داده‌ها و دستورات را در سامانه نمایش می‌دهد. تا حد امکان سعی شده رنگ‌ها و نوع خطوط معنایی مطابق با همان معنای تشریح شده در نمودار معماری داشته باشد. علاوه بر این در حوزه داخلی سامانه از رنگ‌های جدیدی استفاده شده است که نشان‌دهنده بخش‌بندی‌های صورت گرفته در ماژول‌های داخلی دروازه است. لازم به ذکر است بعد از پیاده‌سازی و در هنگام بهینه‌سازی کدها نمودار جریان داده کمی دستخوش تغییر شد. این تغییر در نمودار آخر این بخش نمایش داده شده است.

### سطح 0



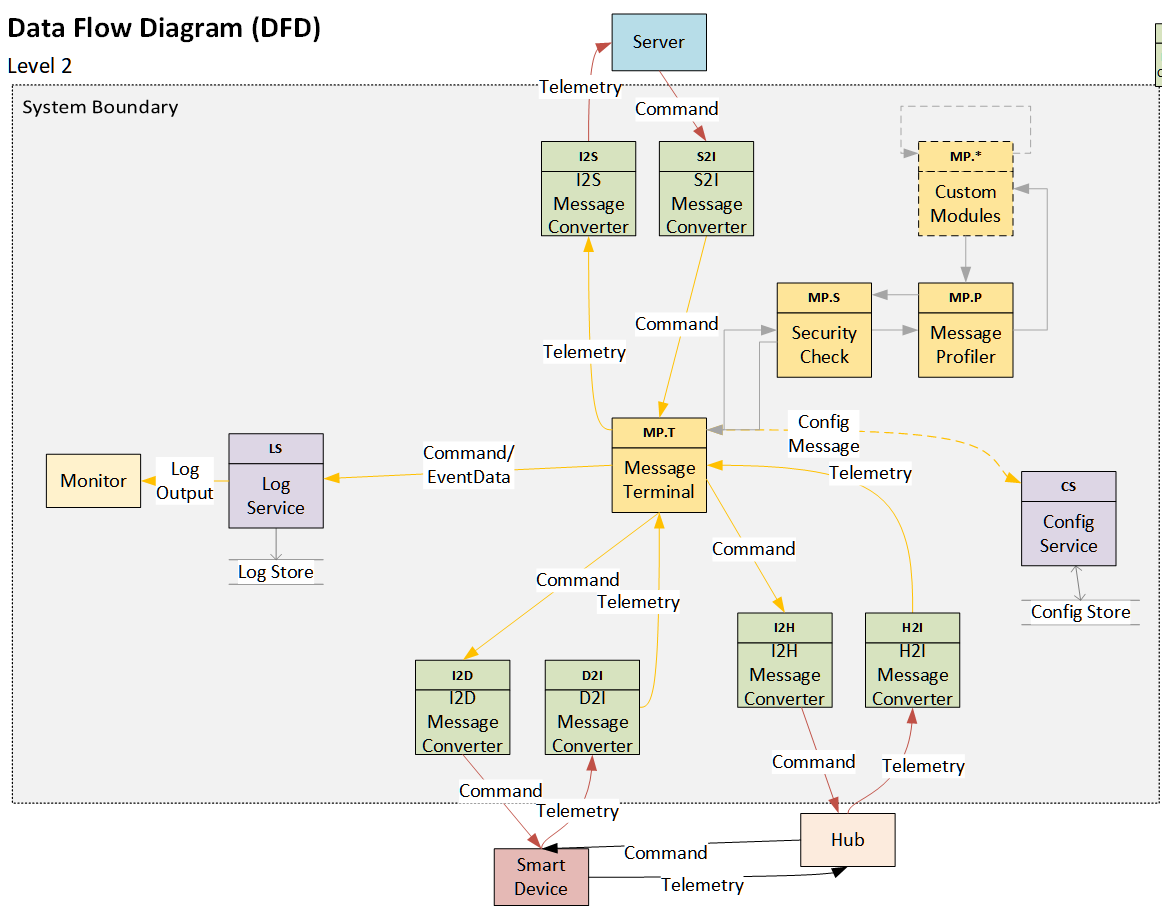
شکل ‏4‑4

### سطح 1



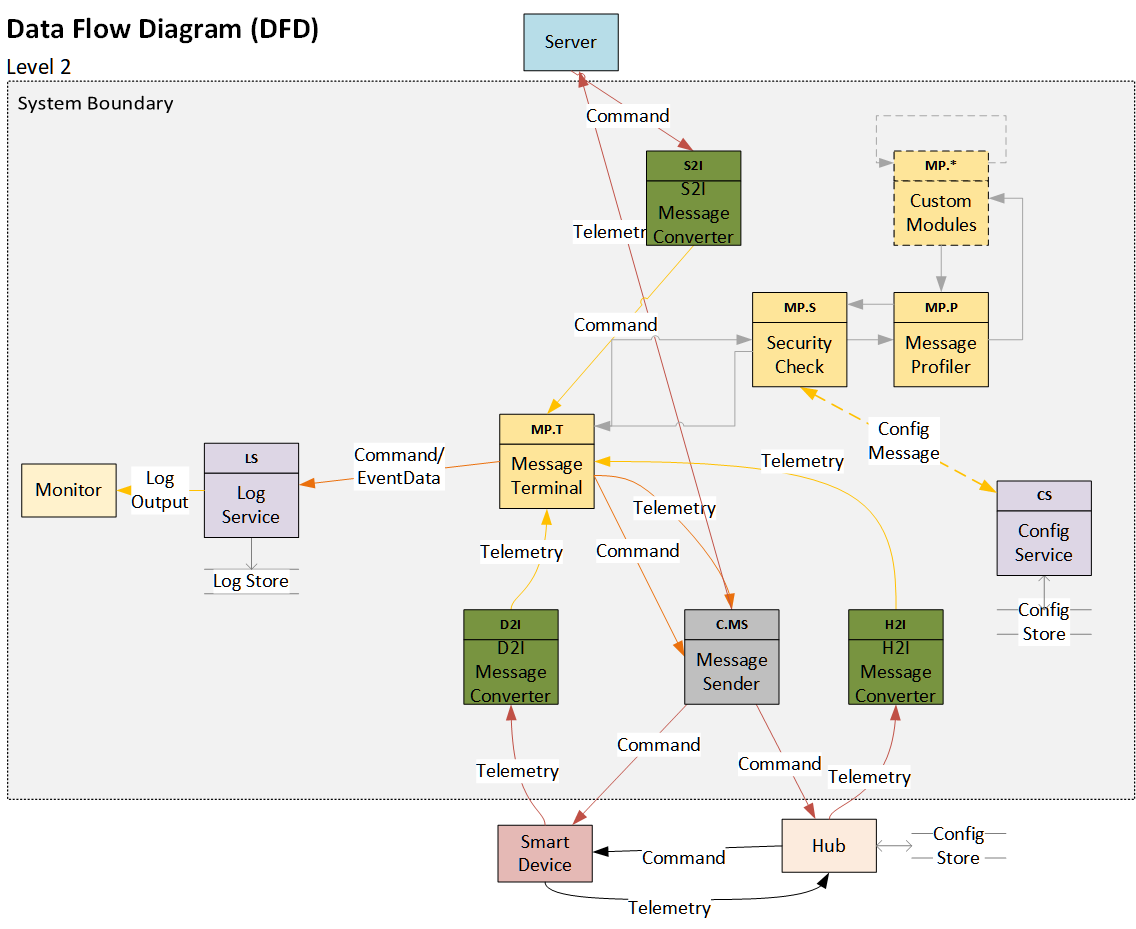
شکل ‏4‑5

### سطح 2



شکل ‏4‑6

### سطح 2 (نهایی)



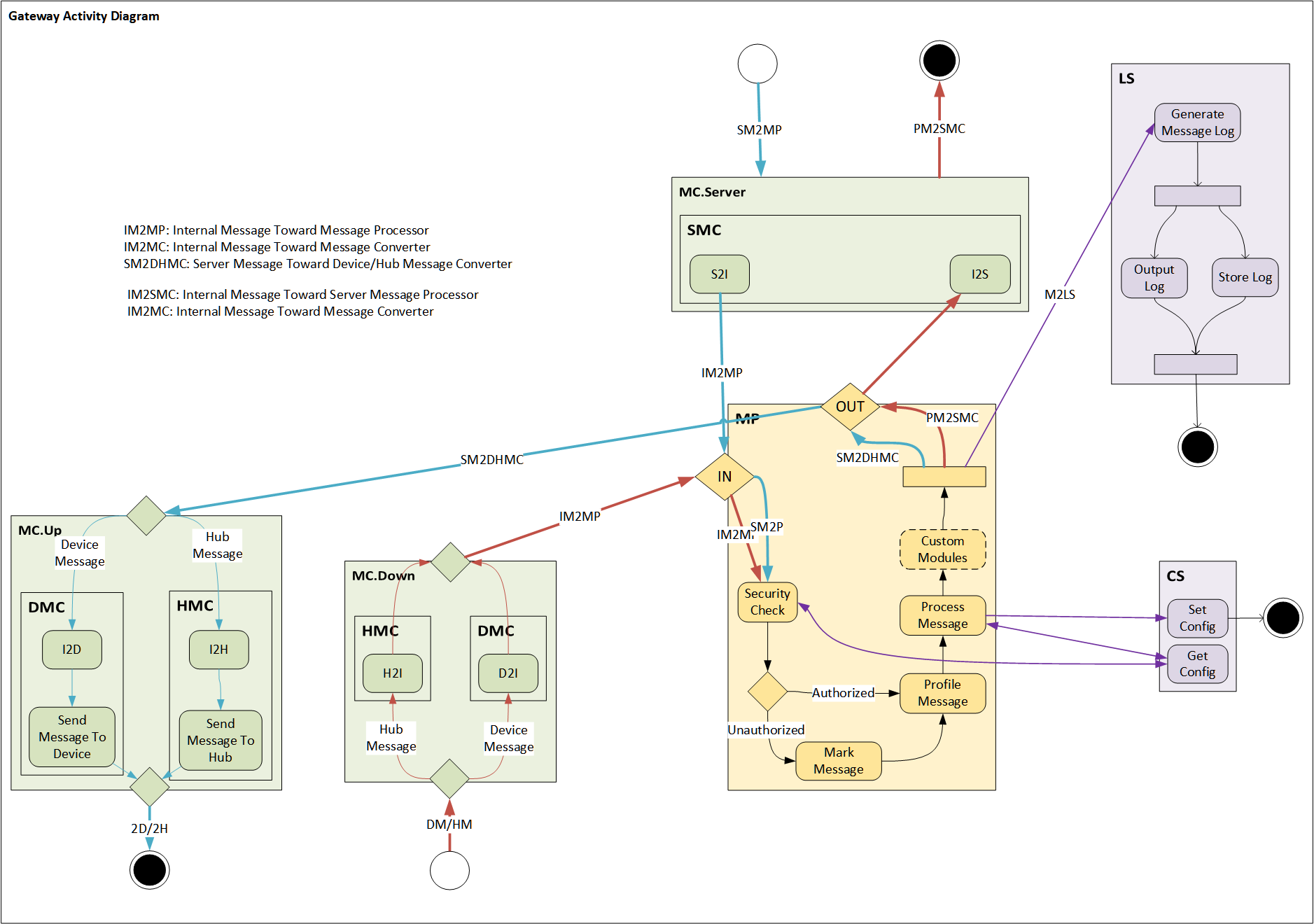
شکل ‏4‑7

## نمودار Activity

### خلاصه

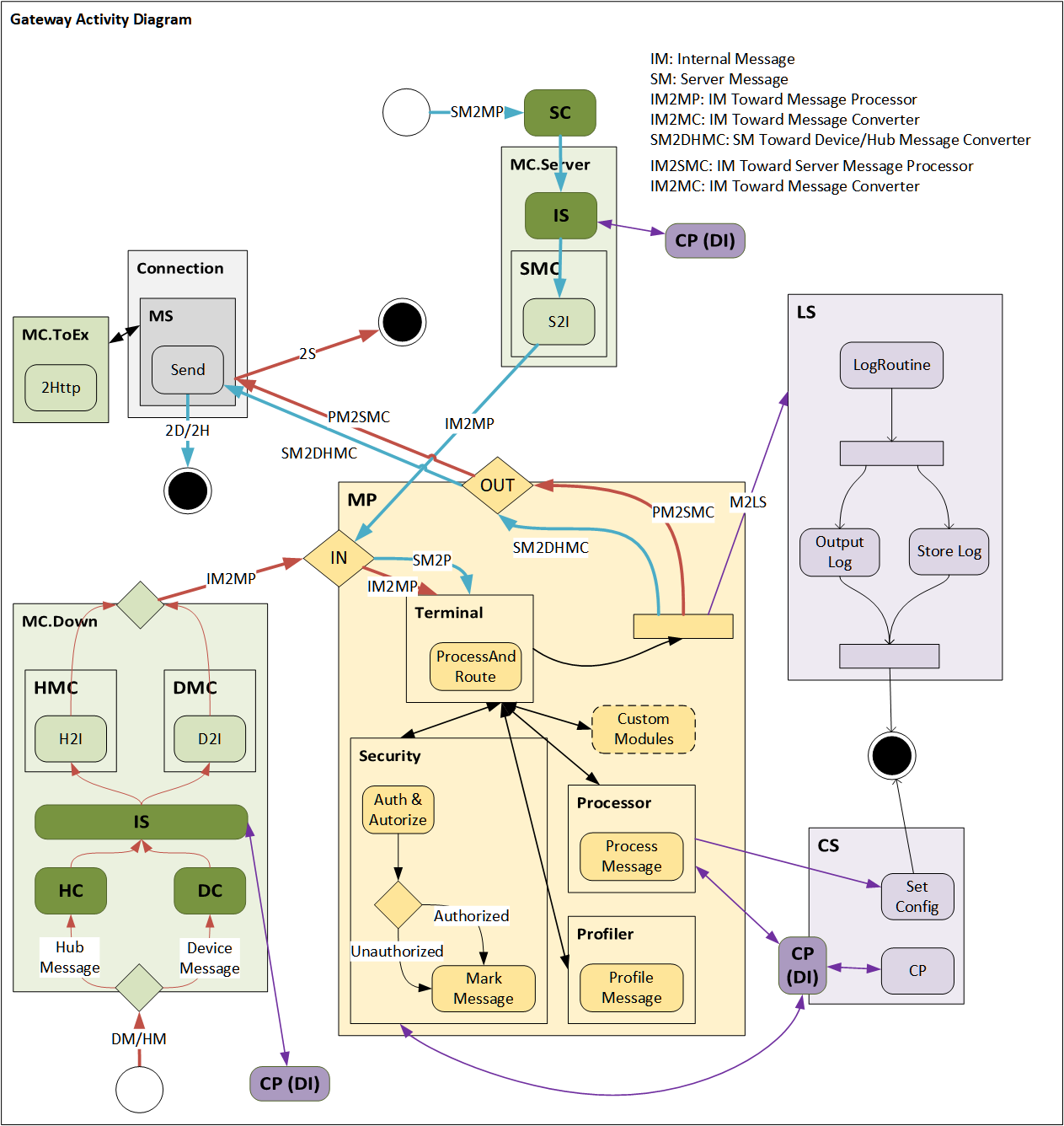
در نمودار Activity اجزای داخلی دروازه بیشتر از بیش شکسته شده اند و فعالیت‌های کلیدی که باید بین اجزای مختلف صورت بگیرد نمایان شده است. این نمودار که به صورت کامل‌تر در پیوست ضمیمه شده است،‌ در دو سطح اولیه و نهایی آورده می‌شود. سطح نهایی در واقع نمودار فعالیت‌ها بعد از اعمال بهینه‌سازی در کدها است.

### نمودار اولیه

****

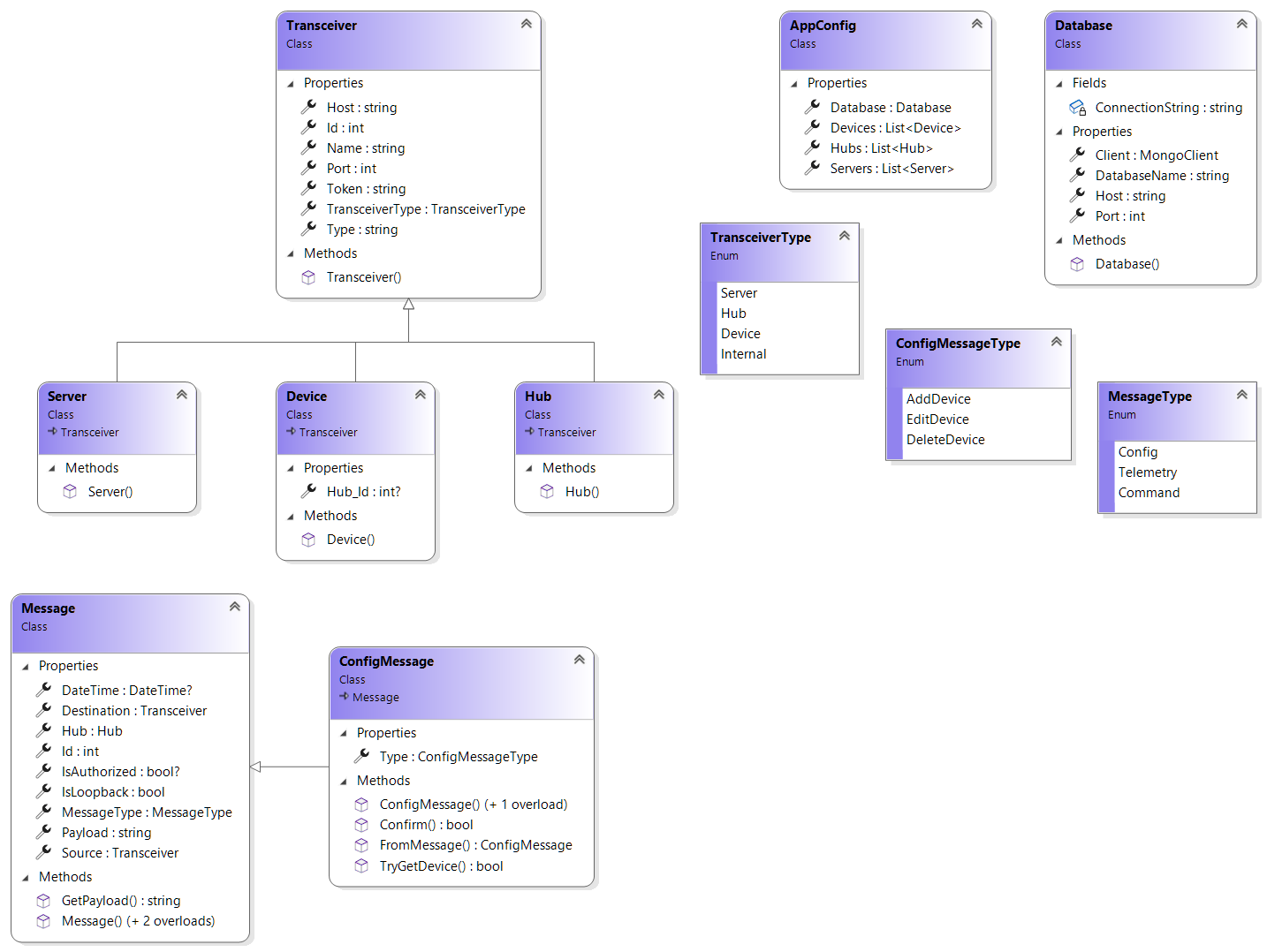
شکل ‏4‑8

### نمودار نهایی



شکل ‏4‑9

## نمودار کلاس (مدل داده)



شکل ‏4‑10

## نتیجه‌گیری

از این نمودارها در کنار هم می‌توان ساختار کلی نرم‌افزار دروازه را متصور شد و حتی بسیاری از قسمت‌های در نظر گرفته شده را مستقیماً به کد تبدیل نمود. نحوه انجام این فرآیند به تفصیل در فصل بعد مورد بررسی قرار گرفته است.

# پیاده‌سازی

## مقدمه

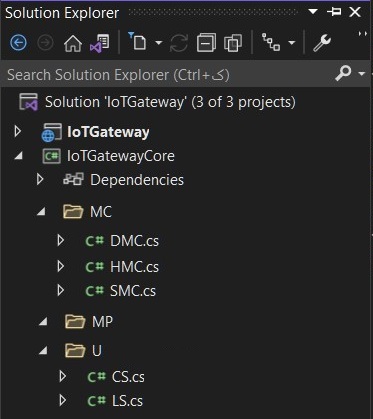
پیاده‌سازی دروازه تماماً در محیط توسعه یکپارچه Microsoft Visual Studio 2022 انجام شده است. هسته اصلی نرم‌افزار در قالب یک کتابخانه بر پایه فریم‌ورک ASP.Net Core 5 و با نام IoTGatewayCore برنامه‌نویسی شده است که توسط رابط برنامه که یک ASP.Net Core 5 Web API است مورد استفاده قرار می‌گیرد. نام پروژه رابط IoTGateway است. از این پس پروژه Web API را با نام رابط و پروژه هسته را با نام کتابخانه می‌شناسیم. کتابخانه بیشتر شامل کدهای خامC# است اما رابط از توابع کمکی گوناگونی استفاده می‌کند که موارد مهم آن در این فصل بررسی می‌شود.

لازم به ذکر است کدنویسی برنامه با یک رویکرد بالا به پایین، مشابه آنچه در فصل قبل انجام شد صورت گرفته است. یعنی ابتدا نمودارهای سطوح پایین کدنویسی و سپس مرحله به مرحله به سطوح بالاتر رفته‌ایم. علاوه بر این مدل‌های داده مورد استفاده نظیر Message و Device و ... به مرور زمان در فازهای مختلف و با کمک نمودار کلاس کدنویسی شده است.

### فاز 1

در این سطح مشاهده می‌کنیم ماژول‌های سطح 0 به چند ماژول کوچکتر تقسیم شده‌اند. ما مطابق با ماژول‌های اصلی سطح 1 نمودار جریان داده، شروع به ایجاد ساختار پروژه می‌کنیم. برای این کار در پروژه کتابخانه چند پوشه (Namespace) به نام‌ های MP، MC و U می‌سازیم. ماژول MP پردازنده پیام، MC تبدیل‌کننده پیام و U ماژول‌های کمکی دروازه است. ماژول U در همین سطح به دو ماژول CS (سرویس پیکربندی) و LS (سرویس وقایع) شکسته شده است. ما نیز داخل فولدر U دو کلاس با نام‌های CS و LS می‌سازیم. به همین شکل ماژول‌های MC یعنی SMC، HMC و DMC را نیز ایجاد می‌کنیم.

ساختار پروژه در این مرحله به شکل زیر است:



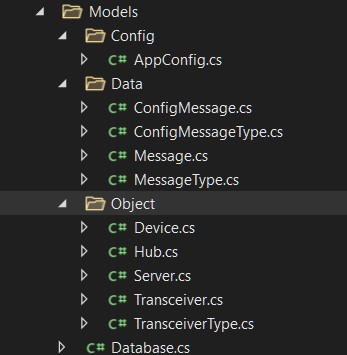
شکل ‏5‑1

### فاز 2

در این سطح نیز مطابق با سطح قبل عمل می‌کنیم. با این تفاوت که در این مرحله کلاس‌ها را نمی‌شکنیم بلکه زیرمجموعه‌های تعریف شده آن را به عنوان یک تابع تعریف می‌کنیم. بنابراین توابع I2S و S2I در SMC، توابع I2H و H2I در HMC و توابع I2D و D2I در DMC تعریف می‌شوند. علاوه بر این در پوشه MP کلاس‌های Processor، Security، Terminal، Profiler و ModuleA (به عنوان یک ماژول سفارشی) را می‌سازیم.

یک فایل کد دیگر به نام IModule می‌سازیم و در آن یک interface تعریف می‌کنیم که شامل تابع Run است. این تابع یک Message به عنوان ورودی دریافت و یک شیء (object) باز می‌گرداند. با بهره‌گیری هر کلاس از این interface و تعریف تابع Run می‌تواند یک ماژول پردازشی جدید به دروازه اضافه نمود.

مدل‌های تعریف شده در نمودار کلاس را در داخل پوشه Models ایجاد می‌کنیم. برای دسته‌بندی بهتر، مدل AppConfig در مسیر Models/Config، مدل‌های Message و ConfigMessage در Models/Data، مدل‌های مربوط به اشیاء در Models/Object و مدل Database در ریشه پوشه Models تعریف شده است. ساختار درختی پوشه Models در شکل زیر نمایش داده شده است:



شکل ‏5‑2

### فاز 3

در این مرحله که سطوح انتزاعی نمودار جریان داده یک به یک پیمایش شد و به انتها رسید، به نمودار فعالیت مراجعه می‌کنیم و سعی می‌کنیم با تعریف کلاس‌های کمکی و توابع مناسب مسیر‌های مختلف نشان داده شده در این نمودار را ساخته و به یکدیگر وصل کنیم.

### حسگر به سرور (D2S/H2S)

ابتدا از نقطه شروع پایین صفحه آغاز می‌کنیم. این مسیر قرمز رنگ نشان دهنده پیام از سمت لایه حسگر (دستگاه/هاب) است. در اولین قدم با یک بررسی شرطی برای تشخیص منبع پیام مواجه هستیم که بعد از تشخیص نوع فرستنده (دستگاه/هاب) پیام را مطابق آن به توابع تبدیل D2I و یا H2I می‌فرستد. برای پیاده‌سازی این فرآیند، ابتدا یک پوشه جدید به نام IS در کتابخانه ایجاد می‌کنیم. سپس دو کلاس مجزا به نام‌های DeviceIS و HubIS داخل این پوشه می‌سازیم تا به ما در تشخیص انواع پیام‌ها کمک کند. یک کلاس به نام IS نیز در این پوشه ایجاد می‌کنیم تا به عنوان کلاس تجمیع‌کننده، سرویس‌های کلاس DeviceIS و HubIS را یکجا ارائه کند. مهمترین تابع IS تابع ProcessByType<T> است که با توجه به نوع داده T (Device/Hub/Server) تصمیم می‌گیرد متن پیام را به یکی از سرویس‌های DeviceIS، HubIS و ... بفرستد. این تابع بعدها در رابط استفاده می‌شود.

ProcessByType<T>(payload){

if (t is Device)

DeviceIS.ProcessDeviceMessage(payload);

else if (t is Hub)

HubIS.ProcessDeviceMessage(payload);

else

LS.Log(“Unknown Message”);

}

توابع اصلی دو سرویس، یک تابع پردازشی است که با دریافت متن پیام و پارامترهای امنیتی، ضمن خواندن پیکربندی، پیام را بررسی می‌کند و پس از تبدیل به یه یک پیام داخلی، آن را به ورودی ماژول MP برای پردازش نهایی می‌فرستد.

ProcessDeviceMessage(payload){

server = GetConfig();

device = GetConfig();

LS.LogDetails();

message = DMC.D2I(payload,server,device);

terminal.ProcessAndRoute(message);

}

ProcessHubMessage(payload){

server = GetConfig();

device = GetConfig();

hub = GetConfig();

LS.LogDetails();

message = HMC.H2I(payload,server,device,hub);

terminal.ProcessAndRoute(message);

}

تبدیل پیام به یک پیام داخلی (IM) توسط دو تابع D2I و H2I انجام می‌گیرد. این تبدیل یک تبدیل ساده خطی است که با دریافت متن پیام و مشخصات دستگاه‌های گیرنده، فرستنده و یا واسط(Hub) آن را درون یک شیء Message می‌ریزد.

D2I(payload,device,server){

message = new Message();

message.Id = Random();

message.DateTime = Time.Now();

message.Source = device;

message.Destination = server;

message.MessageType = Telemetry;

}

H2I(payload,device,server,hub){

message = new Message();

message.Id = Random();

message.DateTime = Time.Now();

message.Source = device;

message.Destination = server;

message.Hub = hub;

message.MessageType = Telemetry;

}

نکته) از آنجایی که Hub تنهای یک واسطه است، نمی‌تواند به عنوان یک موجود مستقل اطلاعات خودش را برای سرور بفرستد و تنها این امکان را دارد که از طرف یک دستگاه نماینده شود تا پیامی را به سرور برساند. بنابراین تابع H2I گیرنده را نه Hub بلکه Device در نظر می‌گیرد و Hub را در پارامتر دیگری به همین نام ذخیره می‌کند.

تا به این جا بخش MC.Down پیاده سازی شد.

مرحله بعدی ورود به ماژول پردازشی MP است. برای تفکیک بهتر مسئولیت‌ها ورودی این ماژول که در نمودار تحت عنوان IN نمایش داده شده است را در کتابخانه با کلاس Terminal به عنوان بخشی از ماژول MP تعریف می‌کنیم. Terminal خود به عنوان یک ماژول نقش تجمیعی دارد و با دریافت Message و ارسال آنها به توابع Run از ماژول‌های پردازشی، آن را برای ارسال به بخش بعدی آماده می‌کند.

ProcessAndRoute(Message message){

TerminalRoutine(message);

MessageRouter(message);

}

TerminalRoutine(Message message){

Security.Run(message);

Profiler.Run(message);

Processor.Run(message);

ModuleA.Run(message);

}

MessageRouter(Message message){

if (message.MessageType == MessageType.Config)

return ProcessConfigMessage(message).Result;

else

return \_ms.Send(message).Result;

}

در صورتی که پیام از نوع پیکربندی باشد (Config Message) این بخش با کمک یک تابع داخلی به نام ProcessConfigMessage آن را پردازش و اجرا می‌کند.

ProcessConfigMessage(Message message){

cm = ConfigMessage.ConvertFromMessage(message);

device = cm.TryGetDevice();

if (!success)

{

LS.LogError();

return;

}

if (!device.IsNew)

{

if (cm.Destination.Port == -1)

CS.DeleteDevice(device);

else

CS.UpdateDevice(device, oldDevice);

}

else

CS.AddDevice(device);

}

در غیر این صورت پیام را به سمت خروجی این ماژول (OUT) هدایت می‌کند. این بخش تحت کلاس MS (Message Sender) در ذیل پوشه جدیدی به نام Connection و در کتابخانه کدنویسی شده است. MS علاوه بر ثبت وقایع، وظیفه فراخوانی توابع تبدیل پیام داخلی به پیام خروجی و ارسال آن از طریق پروتکل مد نظر را به عهده دارد.

Send(Message message){

if(message.Destination is null)

LS.LogError();

return SendHttp(message).Response;

}

SendHttp(Message message)

{

request = ToExternal.ToHttp(message);

response = HttpProvider.SafePostAsync(request);

return response;

}

MS این کار را با کمک ماژول تبدیل داده‌ای به نام ToExternal و ماژول ارسال کننده ای به نام HttpProvider انجام می‌دهد. این تقسیم بندی، توسعه پروتکل‌های ارتباطی و اضافه نمودن آن به دروازه را در آینده آسان تر می‌سازد. ToExternal از آنجایی که یک تبدیل کننده داده است در MC و HttpProvider که وظیفه ارتباطی دارد در Connection تعریف شده است.

دیدیم با وجود ماژول MS و وابستگان آن دیگر نیازی به توابع I2H/D/S که در نمودار قدیمی‌تر آورده شده است نیست. این تغییرات در نمودار شکل 4-9 لحاظ شده است.

## سرور به حسگر یا دروازه (S2H/S2D/S2G)

در ادامه مسیر نقطه شروع بالایی (سرور) را در نظر می‌گیریم. این مسیر نیز در طراحی جدید باید از ماژول IS بگذرد. مانند قبل کلاس ServerIS را ایجاد می‌کنیم. این کلاس وظیفه بررسی پیام،‌ ثبت وقایع مربوطه، دریافت پیکربندی دستگاه‌ها و از همه مهمتر تبدیل به پیام داخلی و ارسال به ترمینال را بر عهده دارد. ضمناً باید توجه کرد پیام های از مبداء سرور می توانند از نوع دستور یا از نوع پیکربندی باشند. توجه به این نکته در طراحی توابع این کلاس ضروری است. در بقیه موارد کدهای این کلاس نیز مشابه کلاس های DeviceIS و HubIS است.

تا به این نقطه بخش MC.Server به پایان رسید و مشاهده می کنیم پیام وارد ترمینال پردازش می‌شود.

## افزودن پایگاه‌داده MongoDB

پس از موفقیت در تست اولیه دروازه به این نتیجه رسیدیم که برای مدیریت راحت‌تر دستگاه‌ها نیاز به اضافه‌کردن یک پایگاه‌داده به دروازه داریم. بعد از کمی تحقیق و بررسی، باتوجه‌به این که اطلاعات موجود برای ذخیره‌سازی[[8]](#footnote-9) در پایگاه‌داده نیمه‌ساختاریافته[[9]](#footnote-10) بود و نیاز به تغییر و توسعه در آینده داشت من پیشنهاد استفاده از پایگاه‌داده‌های NoSQL را دادم و MongoDB را برای این منظور انتخاب کردیم.

خوشبختانه MongoDB برای زبان‌های مختلف برنامه‌نویسی از جمله C# کتابخانه مخصوصی جهت تسهیل ارتباط با سرور پایگاه‌داده را فراهم آورده است. با استفاده از این کتابخانه کمکی، کار برای استفاده از پایگاه‌داده در دروازه بسیار راحت‌تر شد. با این حال من یک کلاس کمکی جداگانه بر روی کتابخانه MongoDB Driver نوشتم تا کار برای ارتباط با پایگاه‌داده در سرویس راحت‌تر شود. از آنجایی که بروز خطا و عدم رسیدگی به آن در زمان اجرا[[10]](#footnote-11) می‌تواند منجر به توقف سرویس و از دسترس خارج شدن آن شود لازم است تا رسیدگی به استثنائات[[11]](#footnote-12) و خطاها به بهترین شکل انجام شود تا علاوه بر جلوگیری از توقف سرویس، با ثبت وقایع بتوان بعدها به علل بروز خطا در نرم افزار رسیدگی کرد. یکی از این اقدامات مهم نوشتن نسخه امن[[12]](#footnote-13) برای متدهای مختلف تعبیه[[13]](#footnote-14) شده در کتابخانه بود. علاوه بر این توابع کمکی تعدادی مدل داده[[14]](#footnote-15) اختصاصی در قالب [[15]](#footnote-16)POCO برای تسهیل کار با داده‌های پایگاه‌داده نیز توسعه داده شد.

## توسعه سرویس ثبت وقایع و سرویس پیکر‌بندی

دو سرویس پرکاربرد دیگر که نیاز به آن دیده شد تا توسعه داده شود عبارت بودند از: سرویس ثبت وقایع و سرویس پیکربندی. سرویس ثبت وقایع بر پایه کتابخانه معروف Serilog و سرویس پیکربندی بر پایه کتابخانه Microsoft.Extensions.Configuration توسعه داده شد. در سرویس ثبت وقایع علاوه بر تعریف تعدادی متد و یک کلاس برای تسهیل استفاده از آن، سازوکاری[[16]](#footnote-17) تعبیه شد تا در صورت خطا در فراخوانی[[17]](#footnote-18) و شروع به کار سرویس، وقایع مربوطه بدون استفاده از توابع کتابخانه‌ای و تنها با توابع داخلی[[18]](#footnote-19) C# در فایل‌های Log ذخیره شود.

در سرویس پیکربندی با راهبردی[[19]](#footnote-20) مشابه علاوه بر تعریف متد و کلاس کمکی، سازوکاری مشابه سرویس ثبت وقایع تعبیه شد تا در صورت بروز خطا در فراخوانی سرویس یا تنظیمات[[20]](#footnote-21) آن، از مقادیر پیش‌فرض برای پیکربندی استفاده شود تا نرم‌افزار بتواند به کار خود ادامه دهد و متوقف نشود. علاوه بر این به درخواست تیم فنی توابعی نوشته شد تا انعطاف‌پذیری گسترده‌تری به سرویس پیکربندی داده شود. سازوکار درخواستی به این صورت بود که ابتدا سرویس پیکربندی تنظیمات پیش‌فرض خود را از یک فایل JSON[[21]](#footnote-22) داخلی فراخوانی کند. سپس در صورت موجود بودن فایل پیکربندی سفارشی[[22]](#footnote-23) در مسیر اجرایی[[23]](#footnote-24) برنامه، اطلاعات آن را نیز فراخوانی کند و با دادن اولویت بالاتر به فایل سفارشی، اطلاعات دو فایل پیکربندی را با یکدیگر ادغام[[24]](#footnote-25) ک

واژه‌نامه

ضمائم

معماری خانه هوشمند (arch.pdf) 24

معماری خانه هوشمند (دروازه) (arch-g.pdf) 24

منابع و مراجع

|  |  |
| --- | --- |
| [1] | K. Patel and S. Patel, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *IJESC,* 2016. |
| [2] | L. Belli and A. Cilfone, "IoT-Enabled Smart Sustainable Cities: Challengesand Approaches," *MDPI,* p. 1042, 2020. |
| [3] | M. Zennaro, *Introduction to the Internet of Things,* The Abdus Salam International Centre for Theoretical Physics. |
| [4] | "Internet of Things Global Standards Initiative," [Online]. Available: https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx. |
| [5] | "Internet of things - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Internet\_of\_things. |
| [6] | A. Gloria and F. Cercasa, "Design and implementation of an IoT gateway," in *The 8th International Conference on Ambient Systems, Networks and Technologies*, Lisbon, 2017. |
| [7] | H. Chen, X. Jia and H. Li, "A Brief Introduction To IoT Gateway," in *ICCTA*, Beijing, 2011. |
| [8] | "RAD (Rapid Application Development) Model - javatpoint," javatpoint, [Online]. Available: https://www.javatpoint.com/software-engineering-rapid-application-development-model. |

1. Lightweight [↑](#footnote-ref-2)
2. Publish/Subscribe [↑](#footnote-ref-3)
3. Footprint [↑](#footnote-ref-4)
4. Bandwidth [↑](#footnote-ref-5)
5. Reliable Message Delivery [↑](#footnote-ref-6)
6. Publisher [↑](#footnote-ref-7)
7. Subscriber [↑](#footnote-ref-8)
8. Storage [↑](#footnote-ref-9)
9. Semi-Structured [↑](#footnote-ref-10)
10. Runtime [↑](#footnote-ref-11)
11. Exception Handling [↑](#footnote-ref-12)
12. Safe Version [↑](#footnote-ref-13)
13. Implement [↑](#footnote-ref-14)
14. Data Model [↑](#footnote-ref-15)
15. Plain Old CLR Objects [↑](#footnote-ref-16)
16. Mechanism [↑](#footnote-ref-17)
17. Call [↑](#footnote-ref-18)
18. Built-in [↑](#footnote-ref-19)
19. Strategy [↑](#footnote-ref-20)
20. Settings [↑](#footnote-ref-21)
21. JavaScript Simple Object Notaion [↑](#footnote-ref-22)
22. Custom Config File [↑](#footnote-ref-23)
23. Execution Path [↑](#footnote-ref-24)
24. Merge [↑](#footnote-ref-25)