Microservices as a design choice for IoT

Mayank Joneja

Abstract—**The use of microservices as an approach to designing cloud based software applications has become increasingly popular owing to its advantages related to factors of scalability and rapid deployment. Several features of microservices make them highly suited even in the context of developing Internet of Things (IoT) applications as a design choice.**

1. Introduction

Traditional application development has always followed a monolithic approach of dividing functionalities into methods and classes but inside the same single unit which is deployed as a process. Such an architecture allows for proper testing of the application as well as usage of the deployment pipeline.

However, this approach shows severe limitations when considering scaling of systems to multiple users and rapid deployment cycles. Each and every change requires the entire application to be re-deployed. Furthermore, while scaling up, the entire application is is duplicated onto servers instead of just scaling the functions which are facing a higher load.

For this reason, companies like Netflix and Spotify popularized the use of a new style of software development based on microservices. Microservices are a way of breaking down large applications. Instead of in-memory function calls, each service runs as a separate process. Microservices are structured in a way that they are small, highly decoupled and task specific. They use lightweight mechanisms to communicate with each other.

Recent popularity of the Internet of Things (IoT) has shown another domain wherein this approach could make design of applications easier as well as more robust in dealing with large number of connected devices. This paper looks into the reasons for using microservices in IoT as well as existing frameworks which help developers use this approach.

The rest of the paper is structured as follows: Section 2 provides an overview of the advantages of a microservice based architecture. Section 3 discusses typical charecteristics of an IoT application. Section 4 presents some popular IoT platforms based on the microservices approach whose common design aspects are summarized in section 5. Section 6 provides concluding remarks.

2. Characteristics of Microservices

Microservices are characterized by the componentization of the service layer. Each functionality is mapped onto a separate microservice which requires them to have well defined context boundaries. Furthermore, each service has a separate "view" of the world in terms of the context and database that they access. Each microservice, in addition, is independently upgradable and the communication between these services is usually via HTTP-REST APIs.

Such a structure allows for the usage of different programming languages, messaging protocols inside the services themselves and thus allow for technology heterogenity. Furthermore, the separation of application features into independent processes allows for a more resilient overall application as the crashing of one microservice does not lead to the crash of the entire application.

The independence of microservices makes scaling of applications also easier as only the services facing heavier load can be replicated onto additional servers. In addition to the technological advantages provided by such an architecture, there has also been a change in the organizational alignment in general because of microservices.

Microservices have led to the popularization of the "Dev-Ops" culture wherein developers of the service are responsible for its maintenance and operation as well. This has led to organizations like Spotify and Amazon being structured in full-stack teams each responsible for a microservice instead of conventional separation between development and operations teams.

3. Characteristics of IoT

Specific features of IoT based on the various definitions of IoT and its characterestics have been well analysed by several industrial experts as well as independent agencies. [1] There is a lot of discussion about how to characterize an IoT application. Some of the common characteristics that are presented in most cases include:

- **Things**: The core component of an IoT application is the large number of connected devices or sensors which provide the first step of data aggregation. These devices might be directly connected to the internet or might require to first connect to a gateway device which in turn forwards the streams to the cloud application.
- **Data**: Raw data that is collected from the sensors is the starting point of insight generation or the additional business value that is created by the IoT application. Often, there is local processing carried out at the gateways before passing on the data to the cloud.

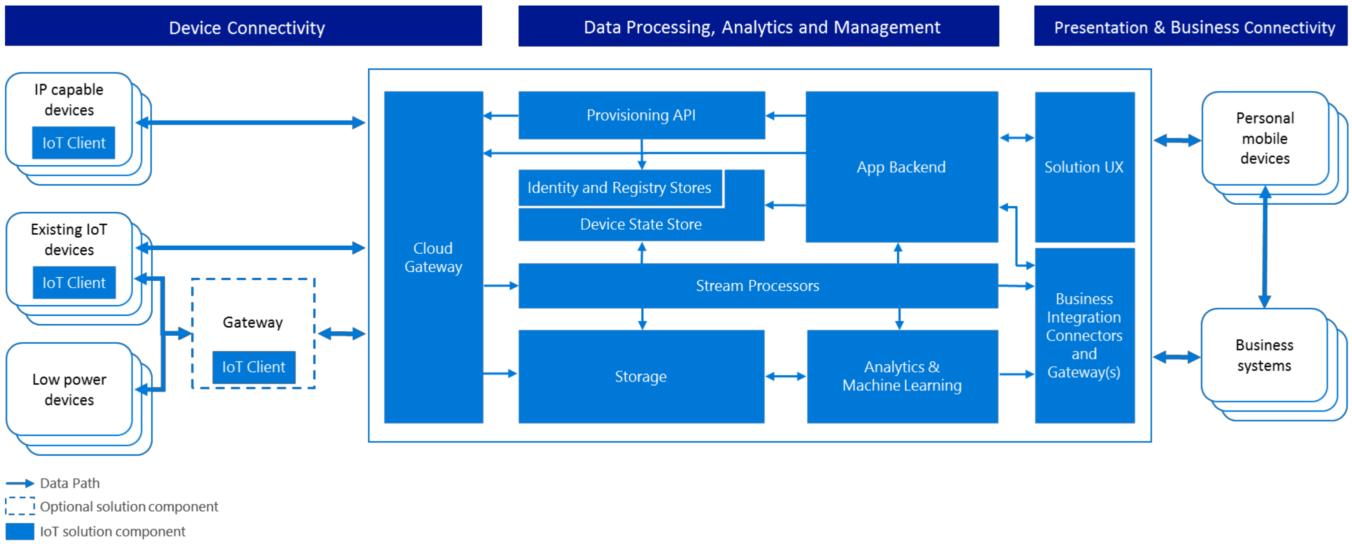

Figure 1. Overview of architecture for Microsoft Azure IoT

- **Communication**: The flow of the data via various communication protocols and mechanisms from the devices up to the format and way in which it is presented to the end application user is also an important component of an IoT application.
- **Intelligence**: The additional value that is generated from an IoT application is the result of the analysis of the aggregated data which provides key insights in the context of the specific use case for which the IoT has been deployed.
- **Action**: IoT provides the configuration of automatic actuation based on the analysed data. These actions may be the manipulation of real world actuators or automation of processes and further data flows.
- **Ecosystem**: The community in which the IoT application is operating provides the overall context for the application and is also an essential factor to be considered while gauging the characteristics of any given IoT application.
- **Connectivity**: The backbone in terms of infrastructure is the network which is used for the various stages of data flow. This might include several technologies such as 6loWPAN or Bluetooth/Zigbee working in tandem with traditional WiFi networks allowing for machine to machine as well as machine to cloud connectivity.

4. Overview of Platforms

- **Microsoft Azure IoT**: Microsoft Azure IoT [2] is one of the two most popular enterprise level cloud solutions for IoT. The architecture for Azure IoT is clearly divided into the clusters of components concerned with device connectivity, data processing and analytics, and presentation. The device connectivity part of the solution offers the end sensors to connect to the cloud either directly to the Azure IoT Hub microservice, or indirectly througha gateway.Further, there is also the option of there being an intermediate custom cloud gateway for protocol translation. The devices that directly connect to the Azure IoT gateway are required to be capable of TLS or SSL.
  The device identity store microservice is responsible for the storage of the credentials of the IoT nodes. The device registry store microservice contains the metadata including important attributes such as location or list of sensors for the device. The device state store microservice is the one which preserves the last known state, or set of readings from the device.
  In addition to the device management microservices, there are dedicated services for the processing of the data stream from these devices as well as analytics. The presentation layer of the cloud platform links to the end user interaction applications and displays.
- **Amazon Web Services (AWS) IoT**: The Amazon cloud solution for IoT [3] is similar to the Azure suite in terms of the overall functional structure. The starting point of the AWS IoT solution is the device SDK which allows for the connection of several sensors over MQTT, HTTP or WebSockets. The interaction of these sensors occurs directly with the device gateway microservice. The communication is secured by an authentication and authorization service which takes care of security and verification of the identity of the devices that connect to the

cloud platform.
Once the messages reach the broker, they can be routed to the rules engine microservice which relays the messages further to other AWS services for analytics, or they can pass through the device shadows microservice onto end user applications. The device shadows microservice is responsible for maintaining a persistent state during intermittent connections.
- **Lelylan**: Lelylan [4] is an open source microservice based cloud platform for IoT which covers most of the features included in the AWS IoT and Azure IoT solutions. It has microservices for an API proxy for dealing with the front end of any IoT application, a devices API for monitoring and control of devices as well as other microservices for communication using MQTT. Lelylan also has a microservice for OAuth 2.0 based authentication and authorization. Lelylan has a microservice analogous to the Amazon Device Shadow which is called the Physical proxy microservice too.
- **Mainflux**: Mainflux [5] is an open source cloud platform as well and supports over-the-air firmware updates. It has a set of clean APIs based on HTTP-REST and supports MQTT, WebSockets and CoAP. Mainflux also supports security via TLS and DTLS and allows for easy deployment of scalable applications via Docker images.

5. Common characteristics

Based on the features and architectural components of the cloud platforms described in the previous section, there are some key common aspects that emerge. These are also supported by the design choices made in the recent work in the domain.

- Device management and provisioning
- Support for HTTP-REST, MQTT, Websocket and CoAP
- Secure communication via TLS/OAuth
- Integration with front-end user experience via APIs or Webhooks
- Support for stateless communication via device proxy/shadow.

A. Krylovskiy et al. [6],describe a middleware architecture for smart cities which includes these same common building blocks in addition to application specific services for energy and building information related processing.

Similarly, for the use case of Smart Metering, T.Vresh et al. [7] highlight a similar middleware platform functionally divided into data transfer (from IoT devices), processing and transformation with independent microservices for each stage.

6. Conclusion

Although a microservices based design pattern is a relatively new concept, there is already a consensus emerging with regards to the implementation details of modern cloud platforms for IoT. Other than Amazon and Microsoft's platforms, others still are under active development. This indicates that the disadvantages of such a microservices based architecture, is still something that remains to be fully studied in the field of IoT.

References

[1] I-Scoop. What is the Internet of Things? actionable internet of things definitions and realities in context. [Online]. Available: http://www.i-scoop.eu/internet-of-things__trashed-2/what-is-the-internet-of-things-iot-definitions/

[2] Microsoft. Microsoft azure iot reference architecture available. [Online]. Available: https://azure.microsoft.com/en-us/updates/microsoft-azure-iot-reference-architecture-available/

[3] Amazon. How the aws iot platform works. [Online]. Available: https://aws.amazon.com/iot-platform/how-it-works/

[4] Lelylan. Architecture microservices explained. [Online]. Available: http://dev.lelylan.com/architecture

[5] Mainflux. Mainflux internet of things solutions. [Online]. Available: http://mainflux.com/

[6] A. Krylovskiy, M. Jahn, and E. Patti, "Designing a smart city internet of things platform with microservice architecture," in *2015 3rd International Conference on Future Internet of Things and Cloud*, Aug 2015, pp. 25–30.

[7] T. Vresk and I. avrak, "Architecture of an interoperable iot platform based on microservices," in *2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, May 2016, pp. 1196–1201.