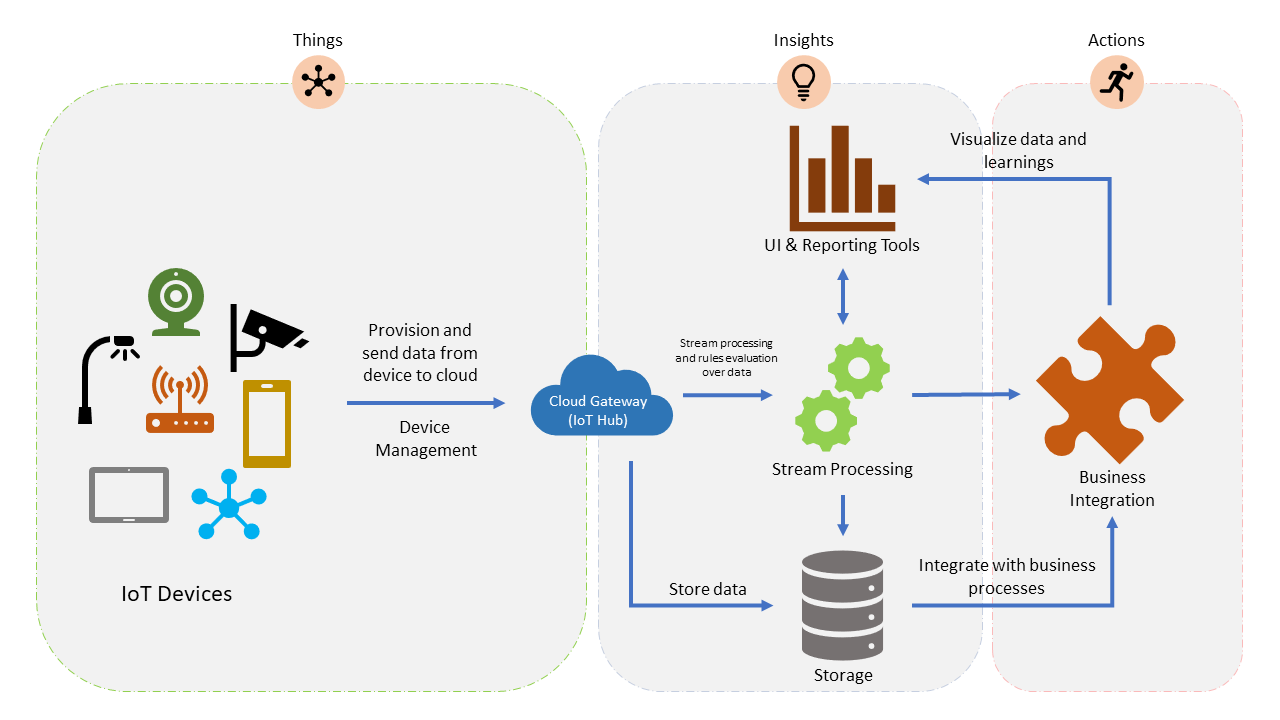
**Core Subsystems of an IoT Architecture**

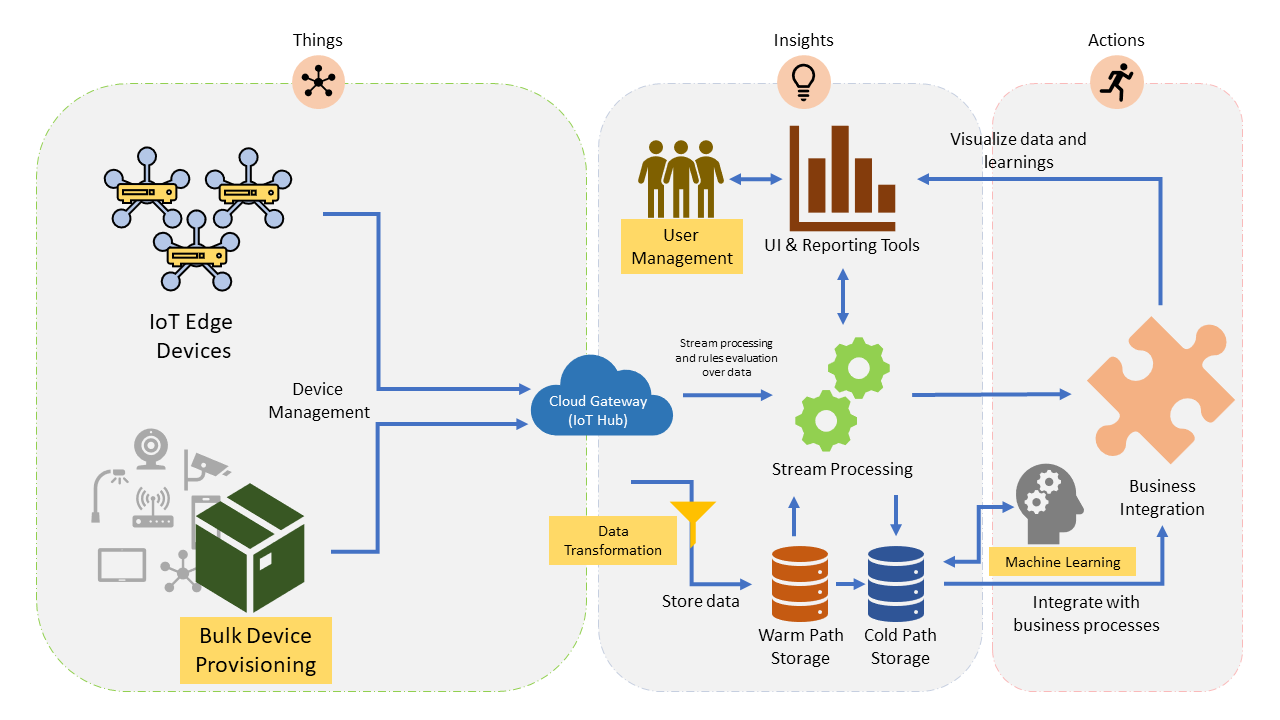
At the core an IoT application consists of the following subsystems: 1) devices (and/or on premise edge gateways) that have the ability to securely register with the cloud, and connectivity options for sending and receiving data with the cloud, 2) a cloud gateway service, or hub, to securely accept that data and provide device management capabilities, 3) stream processors that consume that data, integrate with business processes, and place the data into storage, and 4) a user interface to visualize telemetry data and facilitate device management. Following, these subsystems are briefly described with prescriptive technology recommendations.

1. **Cloud Gateway**: The Cloud Gateway provides a cloud hub for secure connectivity, telemetry and event ingestion and device management (including command and control) capabilities.
2. **Stream Processing**: Processes large streams of data records and evaluates rules for those streams.
3. **Business Process Integration**: Facilitates executing actions based on insights garnered from device telemetry data during stream processing. Integration could include storage of informational messages, alarms, sending email or SMS, integration with CRM, and more.
4. **Storage**: Storage can be divided into warm path (data that is required to be available for reporting and visualization immediately from devices), and cold path (data that is stored longer term and used for batch processing).
5. **User Interface**: The user interface for an IoT application can be delivered on a wide array of device types, in native applications, and browsers.



In addition to the core subsystems many IoT applications will include subsystems for: 5) telemetry data transformation which allows restructuring, combination, or transformation of telemetry data sent from devices, 6) machine learning which allows predictive algorithms to be executed over historical telemetry data, enabling scenarios such as predictive maintenance, and 7) user management which allows splitting of functionality amongst different roles and users.

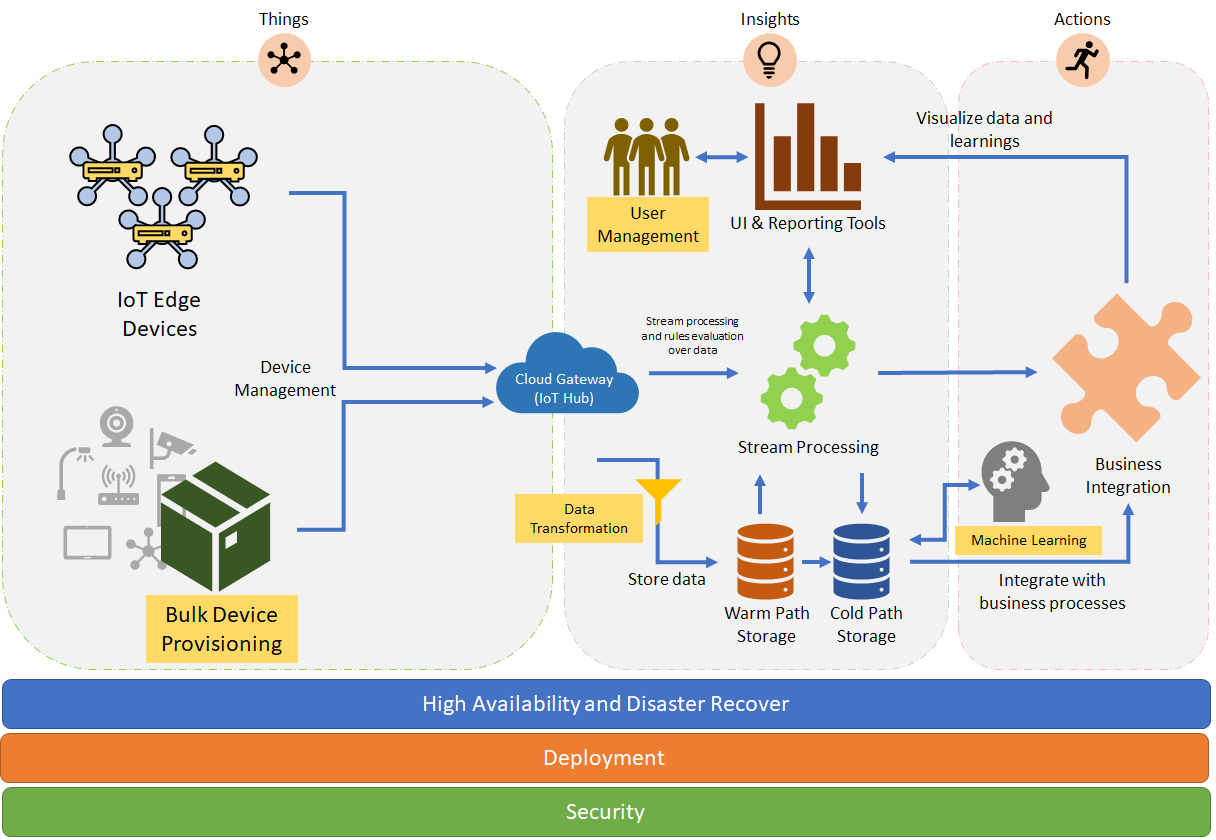
1. **Data transformation**: The manipulation or aggregation of the telemetry stream either before or after it is received by the cloud gateway service (the IoT Hub). Manipulation can include protocol transformation (e.g. converting binary streamed data to JSON), combining data points, and more.
2. **Machine Learning (ML) Subsystem**: Enables systems to learn from data and experiences and to act without being explicitly programmed. Scenarios such as predictive maintenance are enabled through ML.
3. **User Management Subsystem**: Allows specification of different capabilities for users and groups to perform actions on devices (e.g. command and control such as upgrading firmware for a device) and capabilities for users in applications.



## Cross-Cutting Architectural Concerns

There are multiple cross-cutting needs for IoT applications that are critical for success including: 8) security requirements; including user management and auditing, device connectivity, in-transit telemetry, and at rest security, 9) logging and monitoring for an IoT cloud application is critical for determining health and for troubleshooting failures both for individual subsystems and the application as a whole, and 10) high availability and disaster recovery which is used to rapidly recover from systemic failures.

1. **Security**: Security is a critical consideration in each of the subsystems. Protecting IoT solutions requires secure provisioning of devices, secure connectivity between devices, edge devices, and the cloud, secure access to the backend solutions, and secure data protection in the cloud during processing and storage (encryption at rest).
2. **Logging and monitoring**: Logging actions and monitoring activity of IoT application is critical for determining system uptime and troubleshooting failures.
3. **High availability and disaster recovery (HA/DR)**: This focuses on ensuring an IoT system is always available, including from failures resulting from disasters. The technology used in IoT subsystems have different failover and cross-region support characteristics. For IoT applications this can result in requiring hosting of duplicate services and duplicating application data across regions depending on acceptable failover downtime and data loss.



## Data Flow and Processing

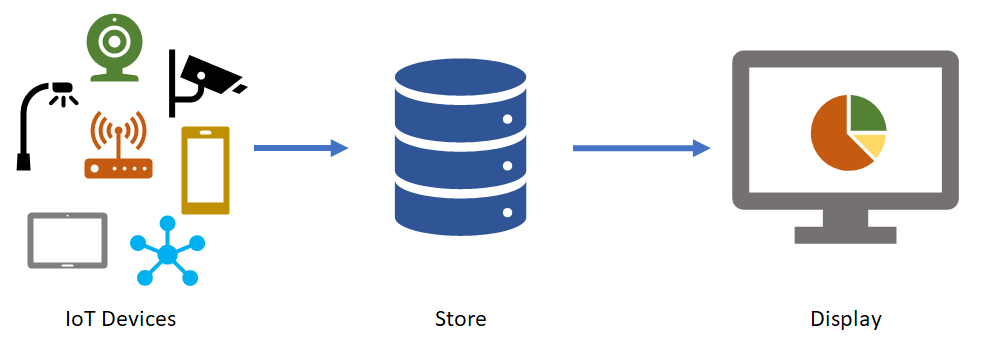
As data is delivered to the IoT backend, it is important to understand how the flow of data processing may vary. Depending on scenarios and applications, data records can flow through different stages, combined in different order, and often processed by concurrent, parallel tasks.

These stages can be classified in four categories - storage, routing, analysis and action/display:

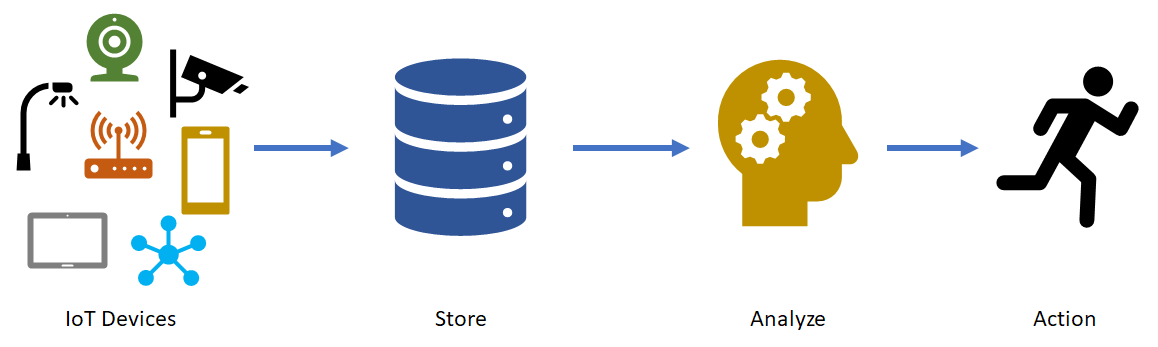
* Storage includes [in-memory caches](https://en.wikipedia.org/wiki/CPU_cache), temporary queues and permanent archives (e.g. a database).
* Routing allows sending data records to one or more storage endpoints, analysis processes, and actions. Routing makes decisions on what data should go which target and when.
* Analysis is used to run data records through a set of conditions and can produce different output data records. For instance, input telemetry data encoded in one format may return output telemetry [encoded](https://en.wikipedia.org/wiki/Code) in another format.
* Original input data records and analysis output records are typically stored and available to display, and may trigger actions such as emails, instant messages, incident tickets, CRM tasks, device commands, etc.

These processes can be combined in simple graphs, for instance to display raw telemetry received in real time, or more complex graphs executing multiple and advanced tasks, for example updating dashboards, triggering alarms, and starting business integration processes, etc.

For example, the following graph represents a simple scenario in which devices send telemetry records which are temporarily stored in Azure IoT Hub, and then are immediately displayed on graph on screen for visualization:



The following graph represents another common scenario, in which devices send telemetry, store it short term in Azure IoT Hub, shortly after analyzing the data to detect anomalies, then trigger actions such as an email, SMS text, instant message, etc.:



IoT architectures can also support multiple systems that can accept and do something with data. For instance, some telemetry storage and/or analysis may occur on premise, within devices and field/edge gateways. In other scenarios, [protocol translations](https://en.wikipedia.org/wiki/Protocol_converter) may be required to connect constrained devices to the cloud. While the resulting graph is more complex, the logical building blocks are the same:

