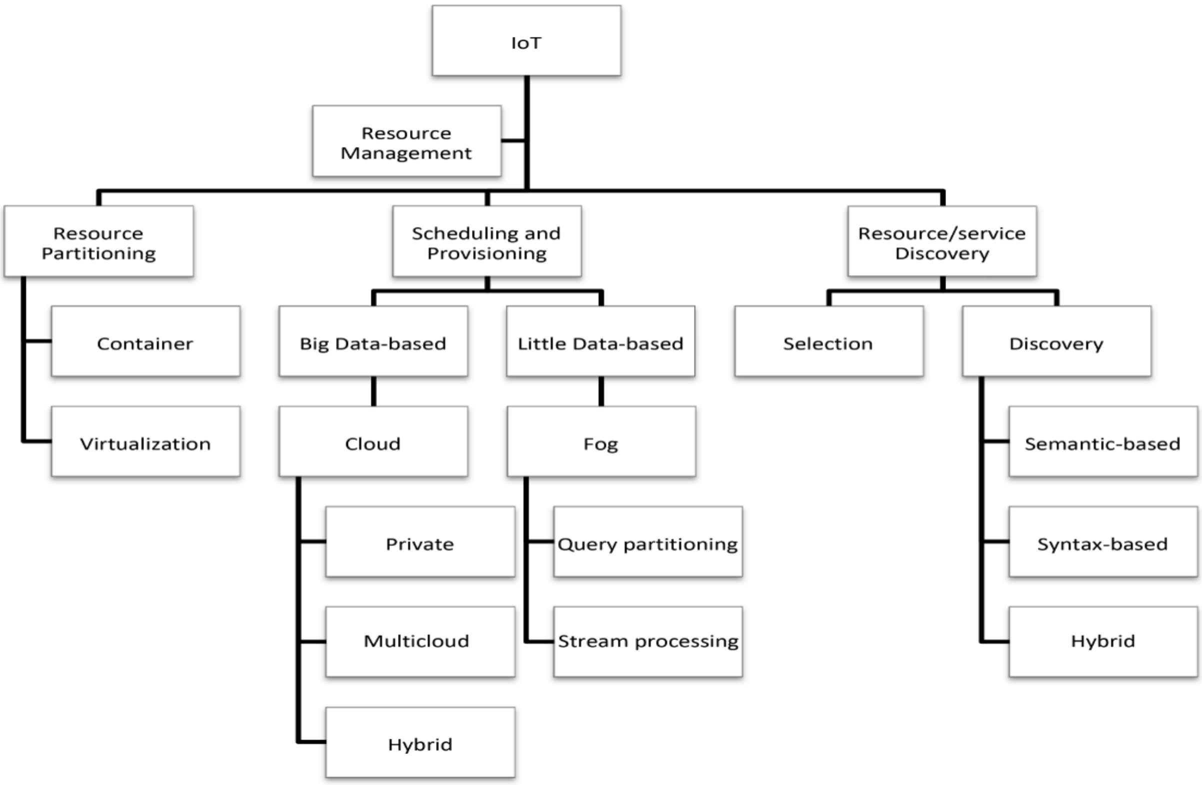
Computation Offloading

- Code offloading (computation offloading) is a solution for addressing the limitation of available resources in mobile and smart devices.

  The advantages of using code offloading are:
  - translate to more efficient power management
  - fewer storage requirements
  - higher application performance.
- The proposed combination of VMs and mobile clouds can create a powerful environment for sharing, synchronizing, and executing codes in different platforms.
- **Computation offloading** in IoT refers to transferring computational tasks from resource-constrained IoT devices (like sensors, wearables, or smart appliances) to

Resource Management in IoT.



- efficient resource management module needs considerable robustness, fault‐tolerance, scalability, energy efficiency, QoS, and SLA.
- Resource management involves discovering and identifying all available resources, partitioning them to maximize a utility function—which can be in terms of cost, energy, performance, etc.

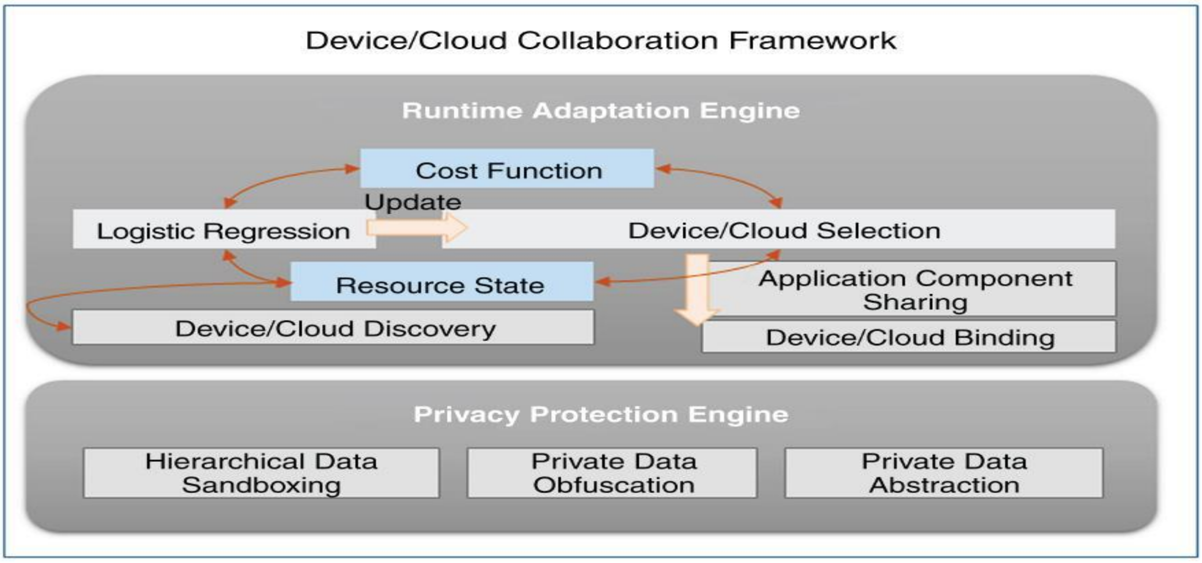
- The first step for satisfying resource provisioning requirements in IoT is resource partitioning
- The hypervisor, is responsible for managing interactions between host and guest VMs.

- To address these challenges, the concept of **Containers** has emerged as a new form of virtualization .
- Containers are able to provide portable and platform-independent environments for hosting the applications and all their dependencies, configurations, and input/output settings.
- In addition, containers are lightweight virtualization solutions.
- Hence containers, compared to VMs, are ideal for distributed applications in IoT

Caching Technique.

- **caching techniques**, which typically reduce network traffic, enhance the availability of data to the users (sink), and reduce the cost of expensive cloud-access operations (eg, I/O operations to public clouds).
- The caching concept involves maintaining sensor (data streams) data to a cache memory in order to facilitate fast and easy access to them.
- caching can play a critical role in optimizing query execution, especially for SPARQL queries.

Describe the framework that enables collaboration between smart mobile devices and cloud.



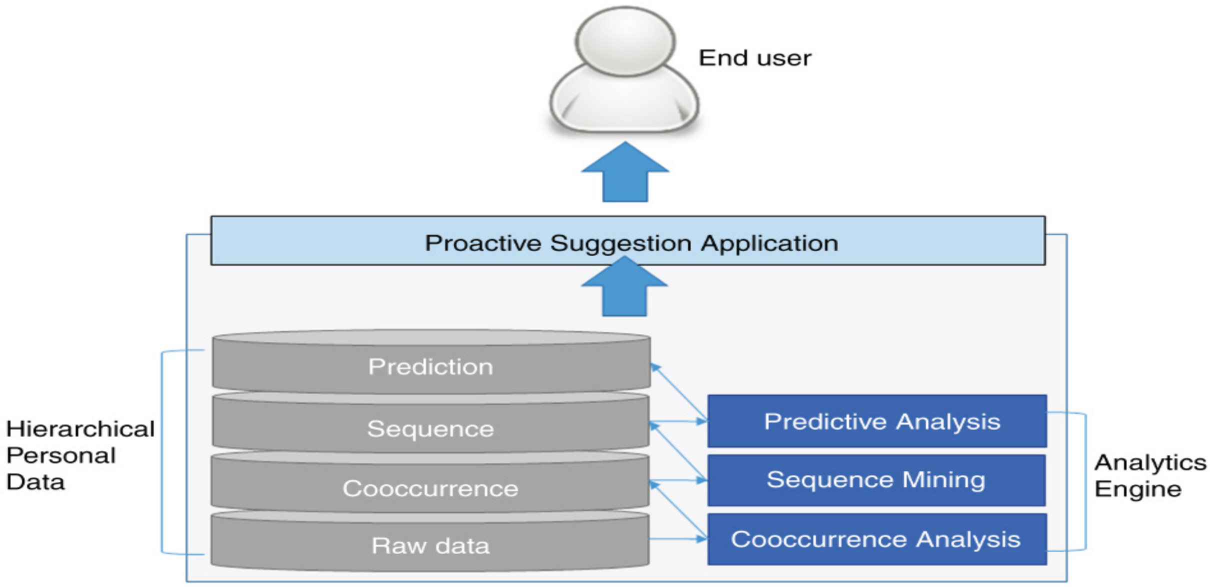
- POWERFUL SMART MOBILE DEVICES
– RUNTIME ADAPTATION ENGINE
– PRIVACY-PROTECTION SOLUTION

- Smartphones nowadays have enough computing capacity to process various computing tasks.
- The Runtime Adaptation Engine (RAE) sits at the core of framework.
- The RAE maintains a list of available devices and cloud and monitors the state of available resources.
- The RAE employs a Logistic Regression algorithm to learn the most cost-effective policy for distributing tasks among devices and cloud.
- cost function is the weighted sum of the resource state (such as battery life), network, and CPU usage.
- The mechanism of the RAE is actually an autonomous agent, which can be deployed on each device and cloud.

- The policy obtained by running the Logistic Regression is enforced by the Device/Cloud Selection module
- device-A can transfer the necessary application components to device-B through the Application Component Sharing.
- Our framework employs the technology of protecting privacy by sandboxing
- The group of data that is set to be kept only within a device will be protected by sandboxing.
- alternative solution is obfuscating (encrypting) data

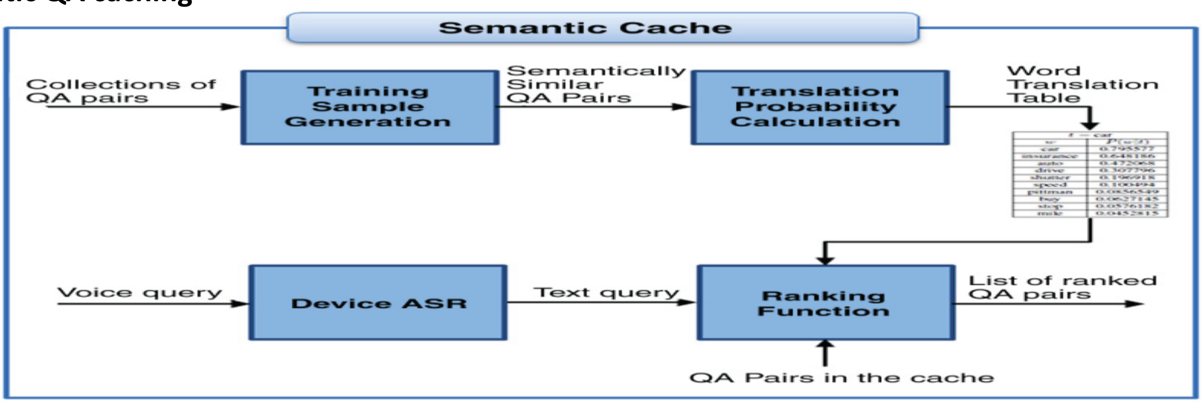
<u>applications of device/cloud collaboration</u>

– **context-aware proactive suggestion,**



- Based on the personal data collected on each mobile device,Proactive Suggestion (PS), an application that makes context-aware recommendations.
- –Analytics engines of PS produce hierarchical personal data

– **semantic QA caching**



- Semantic QA cache is a mobile application that retrieves answers to a given query from the cache filled with answers to the semantically similar queries issued in the past.
- Semantic QA cache can be useful when there is no Internet connectivity

– **automatic image/speech recognition**.

- Automatically recognizing images and speech can greatly enhance a user’s experience in using applications.

- For example, with automatic image recognition, photos taken by a user can be automatically tagged with metadata and catalogued more easily

Iot:

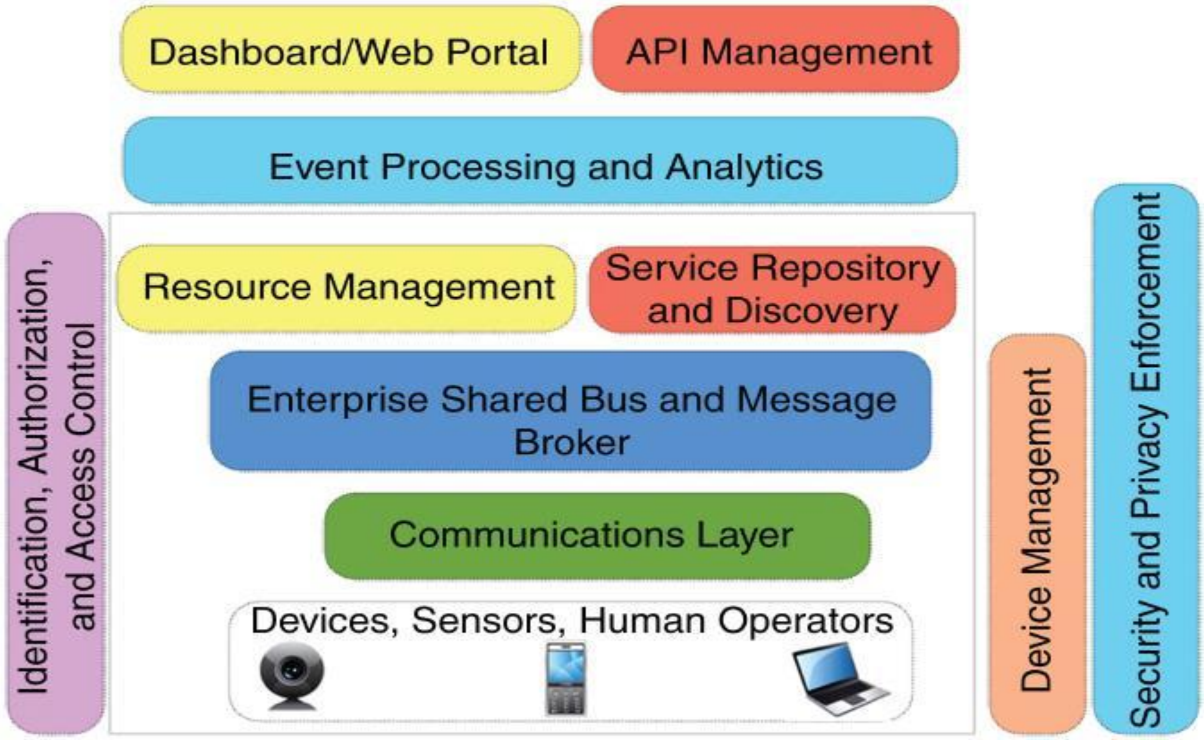
- The Internet of Things (IoT) can be defined as a heterogeneous environment that consists of a diverse range of heterogeneous devices
- Two important pillars – “Internet” and “Things”

– Every object capable of connecting to the Internet will be in “Things” category.
– Any entity able to communicate with other entities, making it accessible at anytime, anywhere.

- IoT requires integration of mobile devices, edge devices like routers and smart hubs and humans as controllers.
- IoT is an autonomous networks of objects where identification and service integration have importance
- IoT is enabler for machine-to-machine, human-to-machine and human-with environment interactions.

The building blocks of IoT are
• sensory devices
• remote service invocation
• communication networks
• and context-aware processing of events.

A Reference Architecture for IoT



- Service layers include event processing and analytics, resource management and service discovery
- API management is essential for defining and sharing system services .
- Service-Oriented Architecture (SOA)

– SOA ensures the interoperability among the heterogeneous devices.
SOA consisting of four layers

- ➢ Sensing layer is integrated with available hardware.
- ➢ Network layer is the infrastructure to support over wireless or wired connections
- ➢ Service layer is to create and manage services
- ➢ Interfaces layer consists of the interaction methods
- API-ORIENTED ARCHITECTURE

– Application Programming Interface (API)
– Building APIs helps the service provider attract more customers while focusing on the functionality of their products.
– It also provides more efficient service monitoring .

Applications of iot

- MONITORING AND ACTUATING:Monitoring devices via APIs can be helpful in multiple domains. Real-time applications can utilize these features to report current system status.. The APIs can report power usage, equipment performance, and sensor status
- BUSINESS PROCESS AND DATA ANALYSIS:

-Society level
– Industry level
– Organizational level,
– Individual level

- INFORMATION GATHERING AND COLLABORATIVE CONSUMPTION

Social Internet of Things (SIoT) is where IoT meets social networks,it link objects around us with our social media and daily interaction with other people

- SECURITY

symmetric cryptography is applied for security as it requires fewer resources

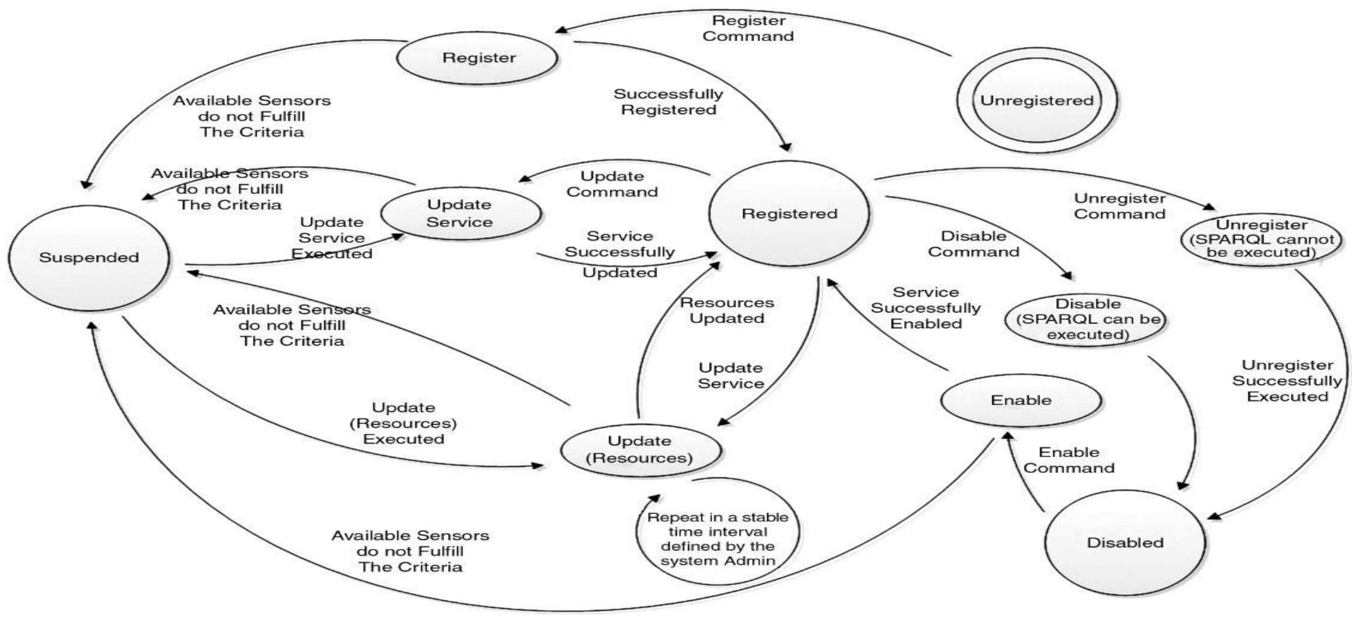
- IDENTITY MANAGEMENT AND AUTHENTICATION

Consumers, data sources, and service providers are essential parts of IoT; identity management and authentication methods applied to securely connect these entities

- PRIVACY

In distributed environments like IoT, preserving privacy can be achieved by a centralized approach .

state diagram of the open IOT services life cycle



– Resource Discovery: discovers a virtual sensor's availability. It therefore provides the resources that match the requirements of a given request for an IoT service.

– Register : a **unique identifier (ServiceID)** is assigned to the requested IoT service.

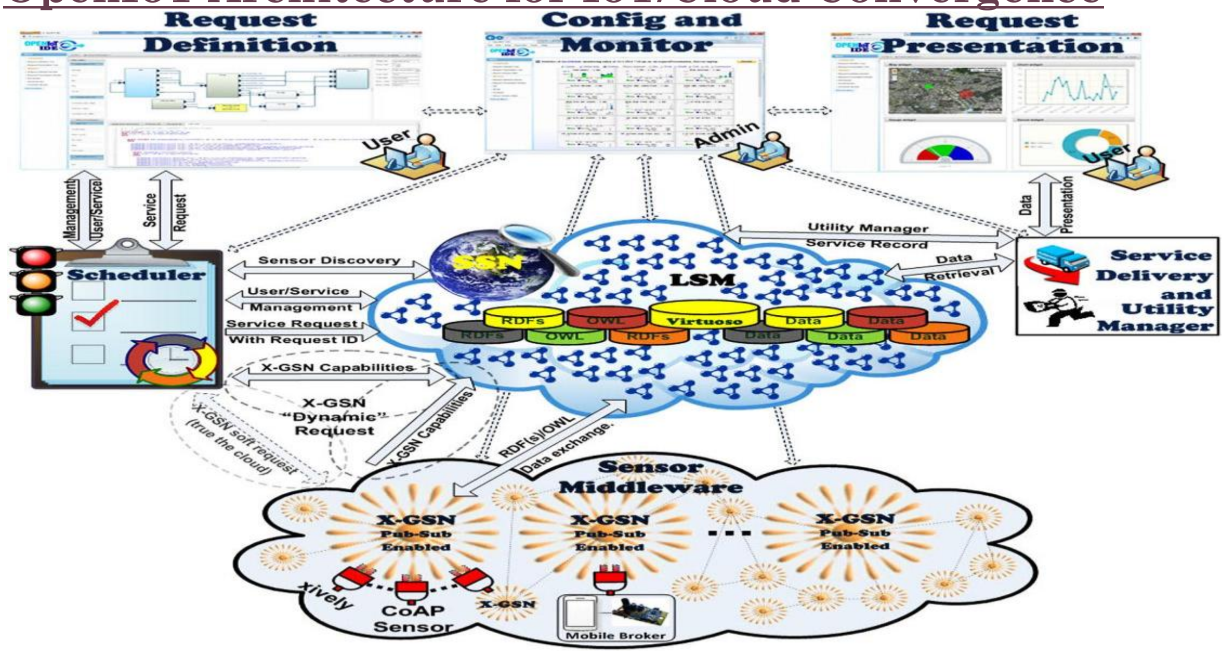
– Unregister : In the case of an active service, a deactivation process is initially applied.
– Suspend : As part of suspend functionality, the service is deactivated and therefore its operation is ceased.

– Enable from Suspension : –This functionality enables a previously suspended service.
– Enable : This service allows the enablement of an unregistered service
– Update: – This service permits changes to the IoT service.
– Registered Service Status: This service provides the lifecycle status of a given IoT service
– Service Update Resources : This service checks periodically all the enabled services,
– Get Service : This service retrieves the description of a registered service
– Get Available Services: This service returns a list of registered services that are associated with a particular user.

difference between Cloud computing and Fog computing in data

	Fog	Cloud
Response Time	Low	High
Availability	Low	High
Security Level	Medium to hard	Easy to medium
Service Focus	Edge devices	Network/enterprise core services
Cost for each device	Low	High
Dominant architecture	Distributed	Central/distributed
Main content generator - consumer	Smart devices- humans and devices	Humans-end devices

OpenIoT Architecture for IoT/Cloud Convergence



The Sensor Middleware:

- collects, filters, and combines datstreams stemming from virtual sensors
- acts as a hub between the OpenIoT platform and the physical world

The Cloud Computing Infrastructure:

- enables the storage of data streams stemming from the sensor middleware,
- act as a cloud database.
- stores metadata for the various services,

The Directory Service:

- stores information about all the sensors that are available in the OpenIoT platform.

– The Global Scheduler:

- processes all the requests.
- This component enables the scheduling of all IoT services.

– The Local Scheduler component:

- regulates the access to the resources of the OpenIoT platforms .

– The Service Delivery and Utility Manager:

a dual role.-

- it combines the data streams
- acts as a service-metering facility

– The Request Definition tool:

- enables the specification of service requests to the OpenIoT platform.

– The Request Presentation component:

- in charge of the visualization of the outputs of an IoT service.

– The Configuration and Monitoring component:

- enables management and configuration functionalities over the sensors,and the IoT services