What is Web of Things (WoT)?

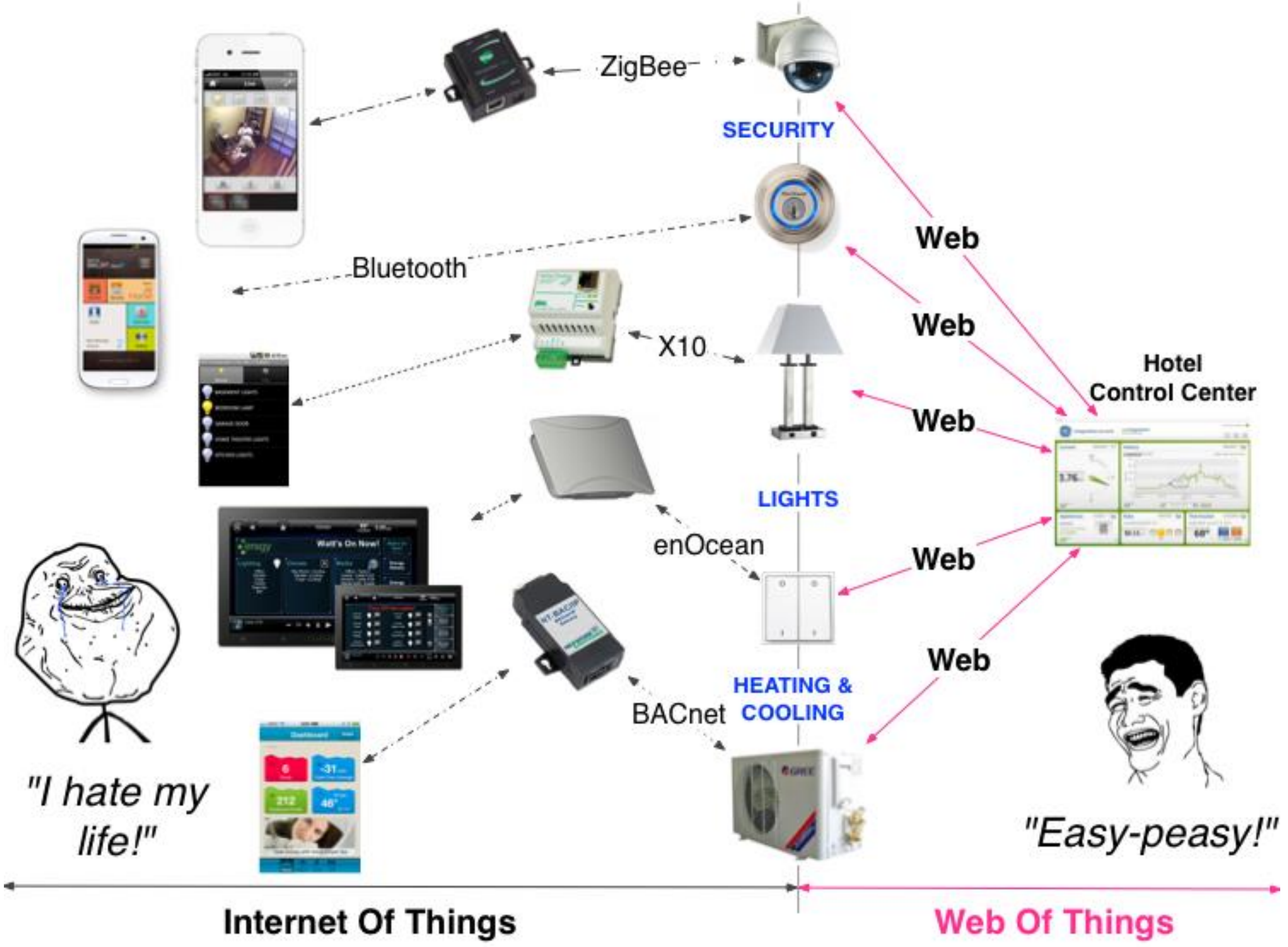
- Connecting every Thing to the Internet and giving them an IP addresses is only the first step towards the Internet of Things.
- Things could then easily exchange data with each other, but not necessarily understand what that data means.
- This is what Web protocols like HTTP brought to the Internet: a universal way to describe images, text, and other media elements so that machines could “understand” each.
- The Web of Things – or WoT – is simply the next stage in this evolution: using and adapting Web protocols to connect anything in the physical world and give it a presence on the World Wide Web!

The Web of Things is a refinement of the Internet of Things by integrating smart things not only into the Internet (network), but into the Web Architecture (application).

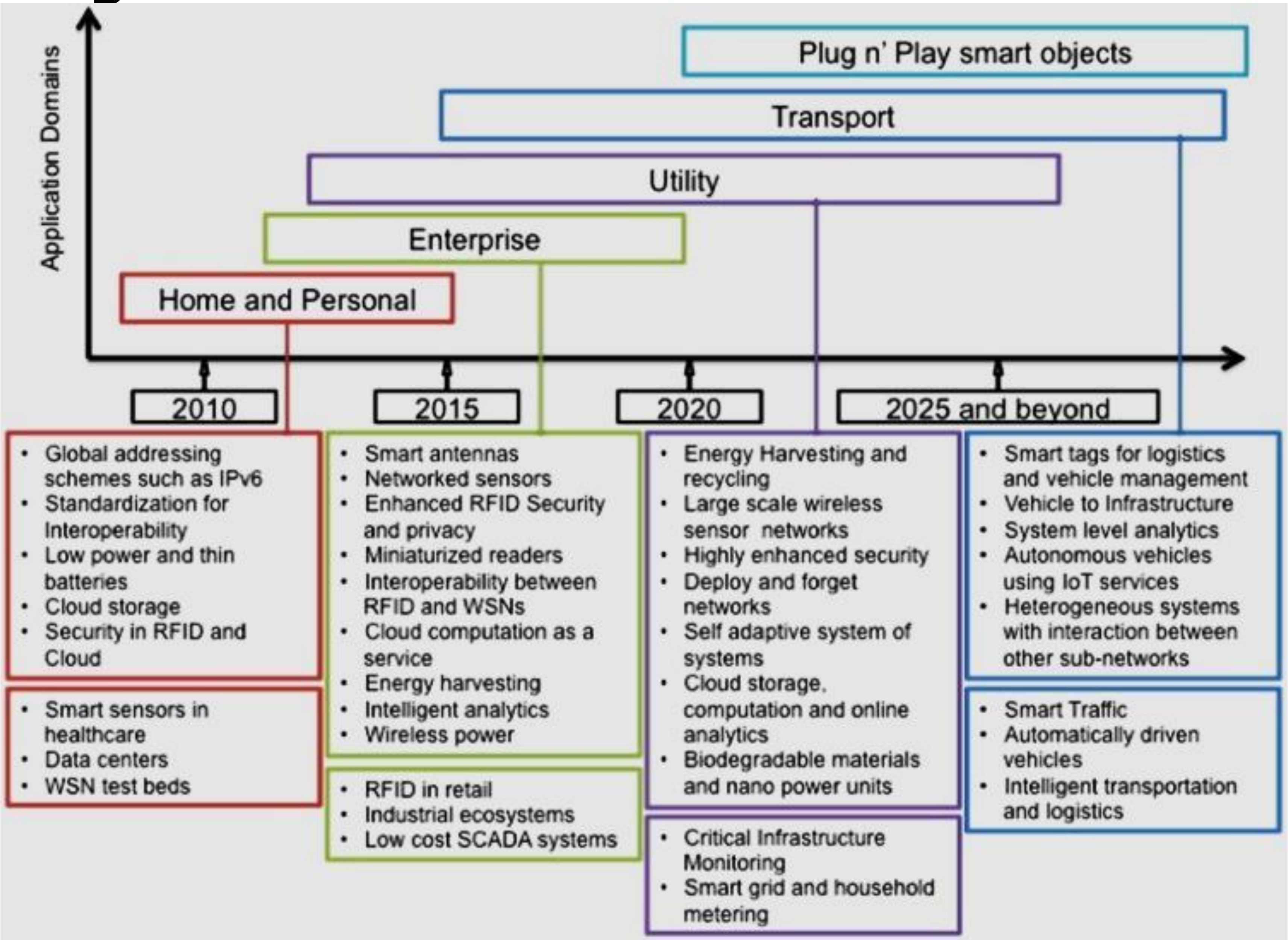
What is Web of Things (WoT)?

- From Wikipedia
  - The Web of Things is a vision inspired from the Internet of Things where everyday devices and objects, i.e. objects that contain an Embedded devices or computer, are connected by fully integrating them to the Web. Examples of smart devices and objects are Wireless Sensor Networks, Ambient devices, household appliances, etc.
  - Unlike in the many systems that exist for the Internet of things, the Web of Things is about re-using the Web standards to connect the quickly expending eco-system of Embedded devices built into everyday smart objects. Well-accepted and understood standards and blueprints (such as URI, HTTP, REST, RSS, etc.) are used to access the functionality of the smart objects.

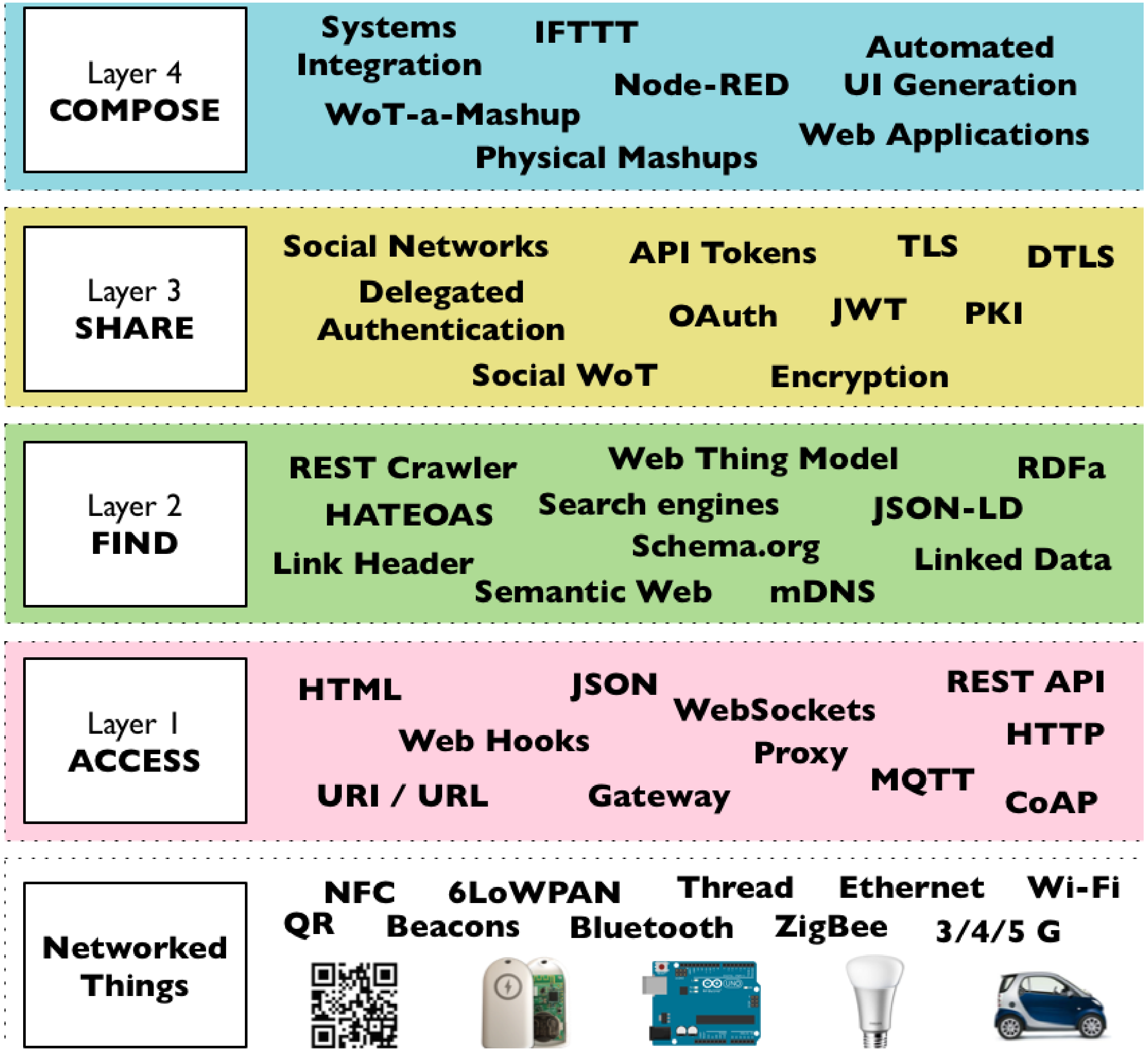
IoT v/s WoT


ZigBee
SECURITY
Bluetooth
X10
Web
Web
Web
Hotel Control Center
LIGHTS
enOcean
Web
Web
HEATING & COOLING
BACnet
"I hate my life!"
"Easy-peasy!"
Internet Of Things
Web Of Things

Internet of Things vision



Web of Things (WoT) Architecture



Web of Things (WoT) Architecture

- Just like the OSI layered architecture organises the many protocols and standards of the Internet, the WoT architecture is an attempt to structure the galaxy of Web protocols and tools into a useful framework for connecting any device or object to the Web.

- The WoT architecture stack is not composed of layers in the strict sense, but rather of levels that add extra functionality, as shown in the figure below. **Each layer helps to integrate** Things to the **Web** even more intimately and hence **making** those **devices** more **accessible for applications and humans!**

Web of Things (WoT) Architecture: Example of Human Things

Layer-1: Access

- This layer is responsible for turning any Thing into a Web Thing that can be interacted with using HTTP requests just like any other resource on the Web. In other words, a Web Thing is a REST API that allows to interact with something in the real world, like opening a door or reading a temperature sensor located across the planet.
- To illustrate this, the sensors of our Pi can be accessed via a simple HTTP request on the following URL: http://devices.webofthings.io/pi/sensors/
- What you are doing here is navigating the RESTful API of our Pi, just like you would be browsing a Web page.

HTML is great for humans, but not always for machines who prefer the JSON notation – Our Pi provides both. Run the following command in your terminal using cURL, a tool for communicating with HTTP APIs:

```
curl -X GET -H "Accept: application/json" "http://devices.webofthings.io/pi/sensors/humidity/"
```

You will see the humidity level in our London office in JSON in your terminal. This is the ideal first step to build your first application that expands the Web into the real world!

- Many IoT scenarios are real-time and/or event-driven. Instead of your application continuously asking for data from Pi, you want it to get notified when something happens in the real world, for example humidity reaches a certain threshold or noise gets detected during the night.
- This is where another Web protocol can help: WebSocket.

- This Javascript code below is enough for a Web page to automatically get temperature updates from the WoT Pi. You can paste it in the console of your Web browser and you will see Pi pushing the temperature every second to your browser.

```
var socket = new WebSocket('ws://devices.webofthings.io/pi/sensors/temperature/');
socket.onmessage = function (event) { //Called when a message is received
var result = JSON.parse(event.data);
console.log(result);
};
```

Layer-2: Find

- Marking things accessible via an HTTP and WebSocket API is great but it doesn’t mean applications can really “understand” what the Thing is, what data or services it offers, and so on.
- This is where the second layer – Find – becomes interesting. This layer ensures that your Thing can not only be easily used by other HTTP clients but can also be findable and automatically usable by other WoT applications.
- The approach here is to reuse web semantic standards to describe things and their services. This enables searching for things through search engines and other web indexes as well as the automatic generation of user interfaces or tools to interact with Things. At this level technologies such as JSON-LD are in use: a language for semantically annotating JSON. This is also where standards such as the Web Things Model help: they define an abstract set of REST resources that Things should offer.

Layer-3: Share

- The Internet of Things will only blossom if Things have a way to securely share data across services. This is the responsibility of the Share layer, which specifies how the data generated by Things can be shared in an efficient and secure manner over the web.

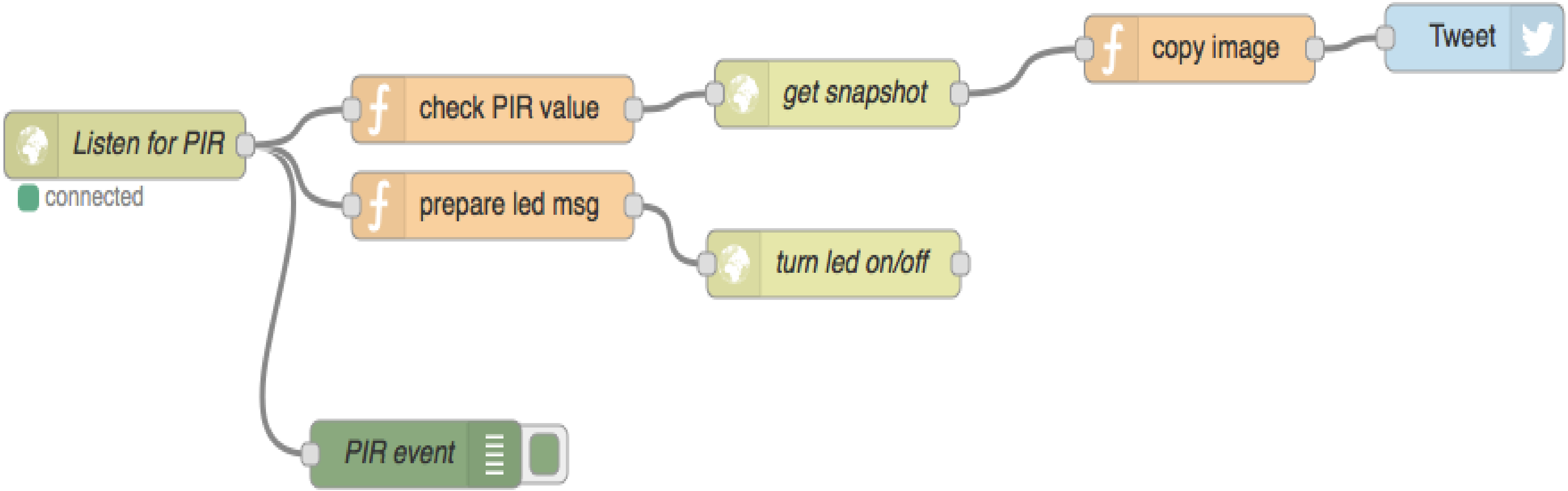
- At this level, another batch of Web protocols help. First, TLS, the protocol that makes transactions on the Web secure. Then, techniques such as delegated web authentication mechanisms like OAuth which can be integrated to our Things’ APIs.

- Finally, we can also use social networks to share Things and their resources to create a Social Web of Things!

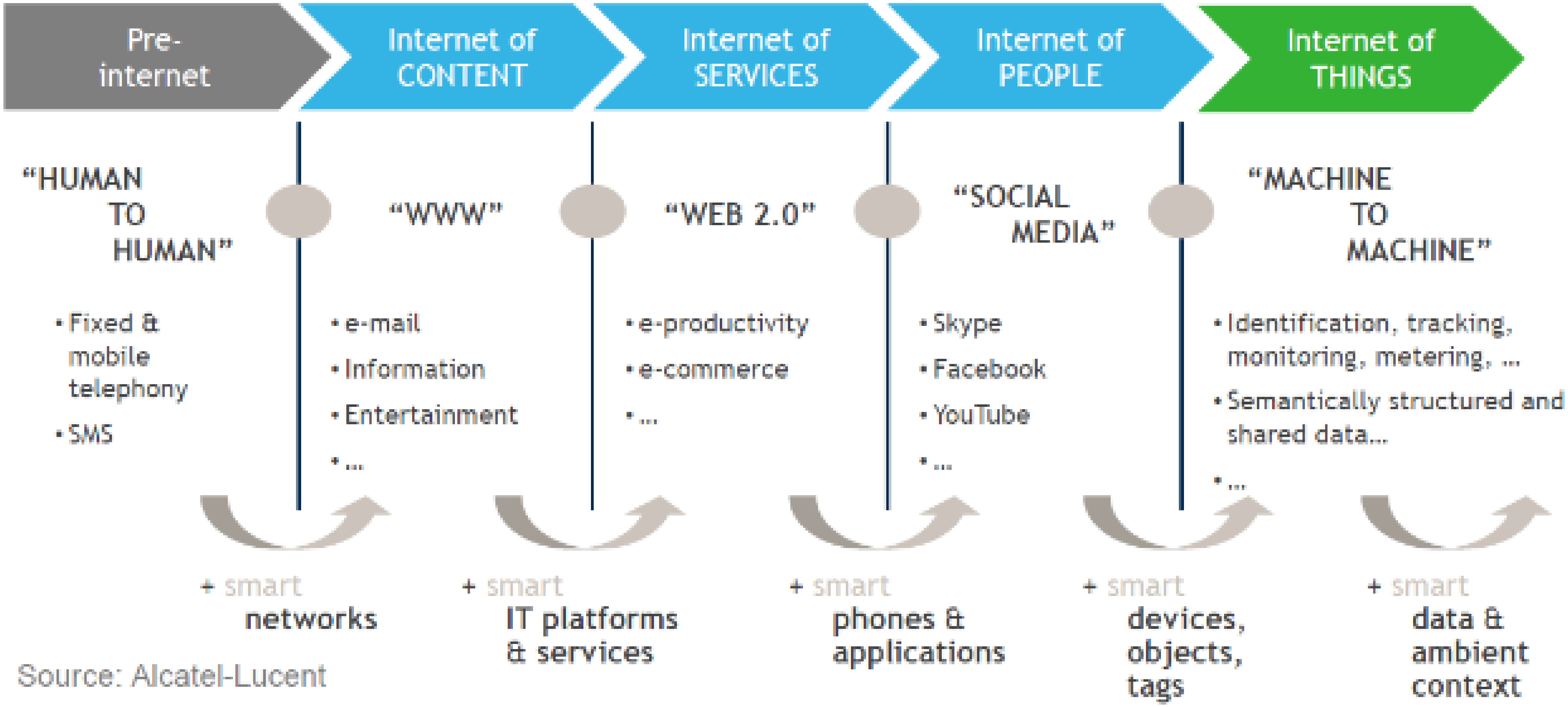
Layer-4: Compose

- Once Things are on the Web (layer 1) where they can be found by humans and machines (layer 2) and their resources can be shared securely with others (layer 3), it's time to look at how to build large-scale, meaningful applications for the Web of Things.
- In other words, we need to understand the integration of data and services from heterogeneous Things into an immense ecosystem of web tools such as analytics software and mashup platforms. Web tools at the Compose layer range from web toolkits—for example, JavaScript SDKs offering higher-level abstractions—to dashboards with programmable widgets, and finally to physical mashup tools such as Node-RED as shown below.
- Inspired by Web 2.0 participatory services and in particular web mashups, physical mashups offer a unified view of the classical web and Web of Things and empower people to build applications using data and services from Web Things without requiring programming skills.

Layer-4: Compose

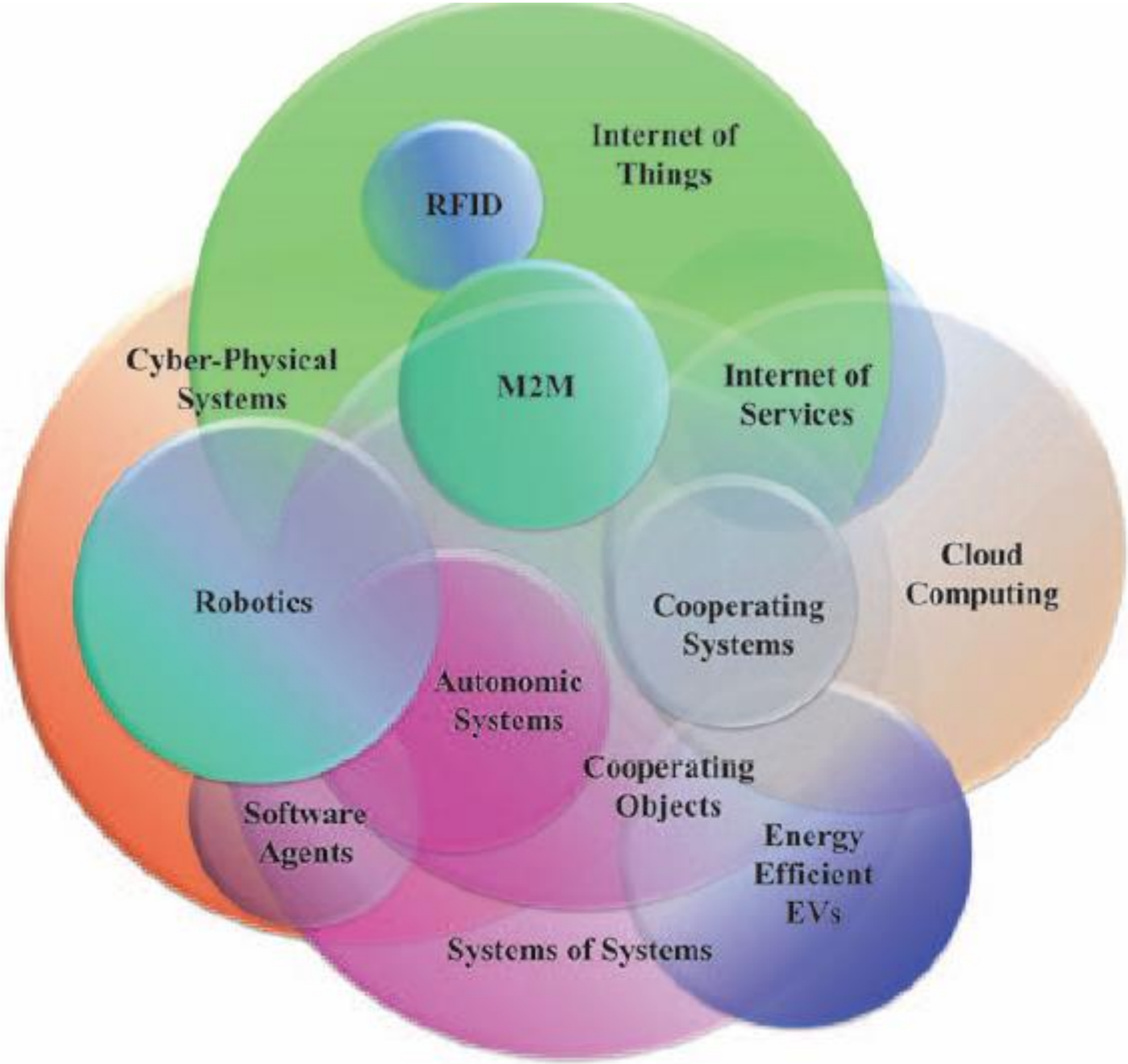


Next Step for the internet Evolution



Source: Alcatel-Lucent

Technology convergence.



Future Internet Technologies

Cloud Computing

- cloud computing has been established as one of the major building blocks of the Future Internet.
- New technology enablers have progressively fostered virtualization at different levels and have allowed the various paradigms known as “Applications as a Service”, “Platforms as a Service” and “Infrastructure and Networks as a Service”.
- Such trends have greatly helped to reduce cost of ownership and management of associated virtualized resources, lowering the market entry threshold to new players and enabling provisioning of new services.
- the convergence of cloud computing and Internet of Things will enable unprecedented opportunities in the IoT services arena.
- As part of this convergence, IoT applications (such as sensor-based services) will be delivered on-demand through a cloud environment.

Future Internet Technologies

IoT and Semantic Technologies

- During the past years, semantic web technologies have also proven their ability to link related data (web-of-data concept), while relevant tools and techniques have just emerged.
- Semantic technologies will also have a key role in enabling sharing and re-use of virtual objects as a service through the cloud.
- The semantic enrichment of virtual object descriptions will realize for IoT what semantic annotation of web pages has enabled in the Semantic Web.

Future Internet Technologies

Autonomy

- Spectacular advances in technology have introduced increasingly complex and large scale computer and communication systems.
- Autonomic computing , inspired by biological systems, has been proposed as a grand challenge that will allow the systems to **self-manage** this complexity, using high-level objectives and policies defined by humans.
- The objective is to provide some self-x properties to the system, where x can be adaptation, organization, optimization, configuration, protection, healing, discovery, description, etc.
- **Autonomy** is thus an imperative property for IoT systems to have. However, there is still a lack of research on how to adapt and tailor existing research on autonomic computing to the specific characteristics of IoT, such as high dynamicity and distribution, real-time nature, resource constraints, and lossy environments.

Future Internet Technologies

- **Properties of Autonomic IoT Systems**
  - The following properties are particularly important for IoT systems and need further research:

Self-adaptation

- In the very dynamic context of the IoT, from the physical to the application layer, self-adaptation is an essential property that allows the communicating nodes, as well as services using them.

- IoT systems should be able to reason autonomously and give self-adapting decisions.

- Cognitive radios at physical and link layers, self-organizing network protocols, automatic service discovery and (re-)bindings at the application layer are important enablers for the self-adapting IoT.

Future Internet Technologies

Self-organization

- In IoT systems, it is very common to have nodes that join and leave the network spontaneously.
- The network should therefore be able to re-organize itself against this evolving topology.
- Self-organizing, energy efficient routing protocols have a considerable importance in the IoT applications in order to provide seamless data exchange throughout the highly heterogeneous networks.
- When working on self-organization, it is also very crucial to consider the energy consumption of nodes and to come up with solutions that maximize the IoT system lifespan and the communication efficiency within that system.

Future Internet Technologies

Self-optimization

- Optimal usage of the constrained resources (such as memory, bandwidth, processor, and most importantly, power) of IoT devices is necessary for sustainable and long-living IoT deployments.

Self-configuration

- IoT systems are potentially made of thousands of nodes and devices such as sensors and actuators.
- Configuration of the system is therefore very complex and difficult to handle by hand.
- The IoT system should provide remote configuration facilities so that self-management applications automatically configure necessary parameters based on the needs of the applications and users.
- It consists of configuring for instance device and network parameters, installing/uninstalling/upgrading software, or tuning performance parameters.

Future Internet Technologies

Self-protection

- Due to its wireless and ubiquitous nature, IoT will be vulnerable to numerous malicious attacks.
- As IoT is closely related to the physical world, the attacks will for instance aim at controlling the physical environments or obtaining private data.
- The IoT should autonomously tune itself to different levels of security and privacy, while not affecting the quality of service and quality of experience.

Future Internet Technologies

Self-healing

- The objective of this property is to detect and diagnose problems as they occur and to immediately attempt to fix them in an autonomous way.
- IoT systems should monitor continuously the state of its different nodes and detect whenever they behave differently than expected.
- It can then perform actions to fix the problems encountered.
- Encounters could include re-configuration parameters or installing a software update.

Future Internet Technologies

Self-description

- Things and resources (sensors and actuators) should be able to describe their characteristics and capabilities in an expressive manner in order to allow other communicating objects to interact with them.
- Adequate device and service description formats and languages should be defined, possibly at the semantic level.
- Self-description is a fundamental property for implementing plug and play resources and devices.

Future Internet Technologies

- **Self-discovery**
- Together with the self-description, the self-discovery feature plays an essential role for successful IoT deployments.
- IoT devices/services should be dynamically discovered and used by the others in a seamless and transparent way.
- Only powerful and expressive device and service discovery protocols (together with description protocols) would allow an IoT system to be fully dynamic (topology-wise).

Future Internet Technologies

Self-matchmaking

- IOT objects, also have to cope with their unreliable nature and be able to find suitable "equivalent object" alternatives in case of failure, unreachability etc.
- Such environments will require self-matchmaking features (between services and objects and vice versa) that will prevent users of IoT future services from having to (re-)configure objects themselves.

Self-energy-supplying

- And finally, self-energy-supplying is a tremendously important (and very IoT specific) feature to realize and deploy sustainable IoT solutions.
- Energy harvesting techniques (solar, thermal, vibration, etc.) should be preferred as a main power supply, rather than batteries that need to be replaced regularly, and that have a negative effect on the environment.

Infrastructure

- The Internet of Things will become part of the fabric of everyday life.
- It will become part of our overall infrastructure just like water, electricity, telephone, TV and most recently the Internet.
- the Internet of Things (as part of the Future Internet) will connect everyday objects with a strong integration into the physical world.
  - **Plug and Play Integration**
  - **Infrastructure Functionality**
  - **Semantic Modeling of Things**
  - **Physical Location and Position**
  - **Security and Privacy**

Networks and Communication

- Networking Technology
  - Complexity of the Networks of the Future
  - Growth of Wireless Networks
  - Mobile Networks
  - Expanding Current Networks to Future Networks
  - Overlay Networks
  - Network Self-organization
  - IPv6, IoT and Scalability
  - Green Networking Technology
- Communication Technology

Self Study

- Networks and Communication
  - Communication Technology
- Processes

Data Management

- Data management is a crucial aspect in the Internet of Things. When considering a world of objects interconnected and constantly exchanging all types of information, the volume of the generated data and the processes involved in the handling of those data become critical.
- There are many technologies and factors involved in the “data management” within the IoT context. Some of the most relevant concepts which enable us to understand the challenges and opportunities of data management are:
  - Data Collection and Analysis (DCA)
  - Big Data
  - Semantic Sensor Networking
  - Virtual Sensors
  - Complex Event Processing.

Data Management (Data Collection and Analysis (DCA))

- Data Collection and Analysis modules or capabilities are the essential components of any IoT platform or system, and they are constantly evolving in order to support more features and provide more capacity to external components
- The DCA module is part of the core layer of any IoT platform. Some of the main functions of a DCA module are:
  - **User/customer data storing:**
  - Provides storage of the customer's information collected by sensors
  - **User data & operation modeling**:
  - Allows the customer to create new sensor data models to accommodate
  - collected information and the modeling of the supported operations
  - **On demand data access:**
  - Provides APIs to access the collected data

Data Management (Data Collection and Analysis (DCA))

- **Device event publish/subscribe/forwarding/notification:**
  - Provides APIs to access the collected data in real time conditions
- **Customer rules/filtering:**
  - Allows the customer to establish its own filters and rules to correlate events
- **Customer task automation:**
  - Provides the customer with the ability to manage his automatic processes.
    - Example: scheduled platform originated data collection,…
- **Customer workflows:**
  - Allows the customer to create his ownworkflowto process the incoming events from a device

Data Management (Data Collection and Analysis (DCA))

- **Multitenant structure:**
- Provides the structure to support multiple organizations. In the coming years, the main research efforts should be targeted to some features that should be included in any Data Collection and Analysis platform:
  - **Multi-protocol**. DCA platforms should be capable of handling or understanding different input (and output) protocols and formats.
  - Different standards and wrappings for the submission of observations should be supported.
- **De-centralisation**.
  - Sensors and measurements/observations captured by them should be stored in systems that can be de-centralised from a single platform.
  - It is essential that different components, geographically distributed in different locations may cooperate and exchange data.

Data Management (Data Collection and Analysis (DCA))

- **Security**.
  - DCA platforms should increase the level of data protection and security, from the transmission of messages from devices (sensors, actuators, etc.) to the data stored in the platform.
- **Data mining features**.
  - Ideally, DCA systems should also integrate capacities for the processing of the stored info, making it easier to extract useful data from the huge amount of contents that may be recorded.

Data Management (Big Data)

- Big data is about the processing and analysis of large data repositories so it is impossible to treat them with the conventional tools of analytical databases.
- the majority of data will be stamped out by machines. These machines generate data a lot faster than people can, and their production rates will grow exponentially with Moore's Law. Storing this data is cheap, and it can be mined for valuable information.
- Examples of this tendency include:
  - Web logs, RFID, Sensor networks, Social networks, Social data (due to the Social data revolution), Internet text and documents, Internet search indexing, Call detail records,
  - Astronomy, atmospheric science, genomics, biogeochemical, biological, and other complex and/or interdisciplinary scientific research;
  - Military surveillance, Medical records, Photography archives, Video archives, Large scale e-commerce.

Data Management (Big Data)

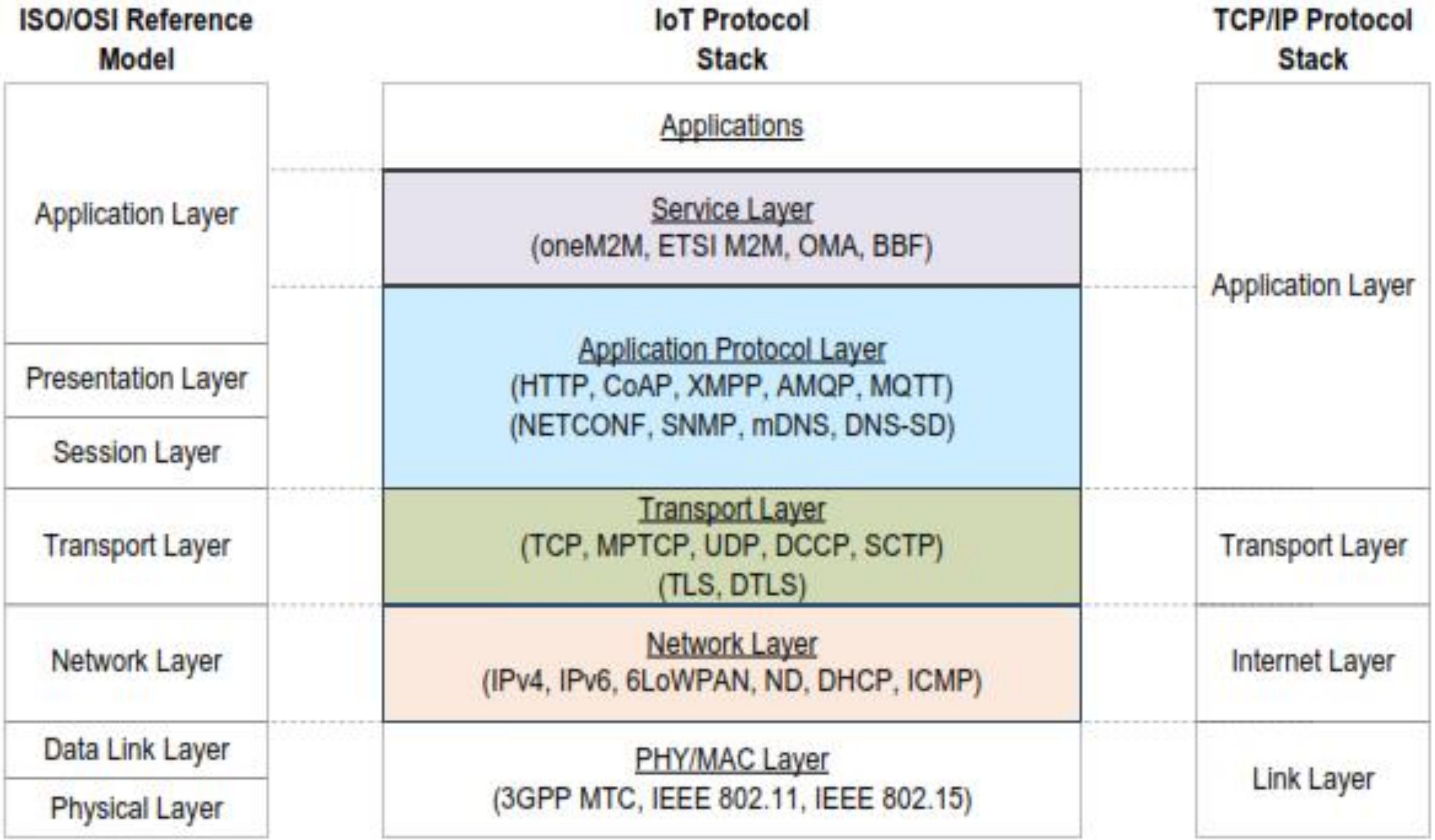
- according to IBM, more than 2.5 quintillion bytes per day, to the extent that 90% of the world’s data have been created over the past two years.
- Technologies being applied to big data include massively parallel processing (MPP) databases, data-mining grids, distributed file systems, distributed databases, cloud computing platforms, the Internet, and scalable storage systems.

Data Management (Semantic Sensor Networks)

- Millions of sensors around the globe currently collect avalanches of data about our environment.
- Millions of sensor has their own characteristics and they collect the huge number of data without enough knowledge.
- Annotating and interpreting the data, and also the network resources, enables management of the large scale distributed networks that are often resource and energy constrained, and provides means that allow software agents and intelligent mechanisms to process and reason the acquired data.
- Ontologies and other semantic technologies can be key enabling technologies for sensor networks because they will improve semantic interoperability and intergration, as well as facilitate reasoning, classification and other types of assurance and automation.
- The Semantic Sensor Network (SSN) ontology enables expressive representation of sensors, sensor observations, and knowledge of the environment.

- Semantics allow machines to interpret links and relations between different attributes of a sensor description and also other resources. Utilizing and reasoning this information enables the integration of the data as networked knowledge

IoT Protocol Stack



IoT Protocol Stack Vs Web Stack

TCP/IP Model	IoT Stack		Web Stack
TCP/IP Model	IoT Applications	Device Management	Web Applications
Data Format	Binary, JSON, CBOR		HTML, XML, JSON
Application Layer	CoAP, MQTT, XMPP, AMQP		HTTP, DHCP, DNS, TLS/SSL
Transport Layer	UDP, DTLS		TCP, UDP
Internet Layer	IPv6/IP Routing		IPv6, IPv4, IPSec
	6LoWPAN		
Network/Link Layer	IEEE 802.15.4 MAC		Ethernet (IEEE 802.3), DSL, ISDN, Wireless LAN (IEEE 802.11), Wi-Fi
	IEEE 802.15.4 PHY / Physical Radio		