IoT architecture in respect to:

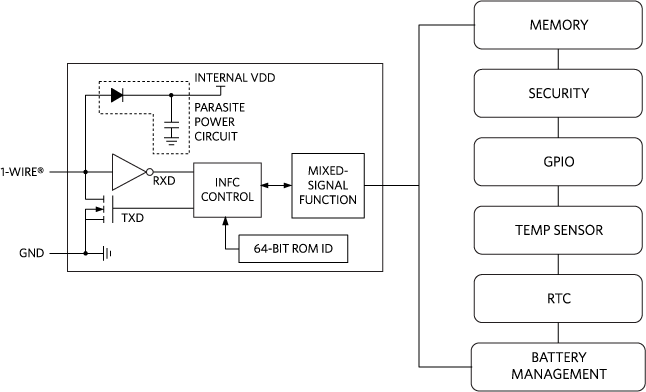
**Cloud:**   
Stack that allows back-end components to communicate with each other in order to support Big Data analytics, certain interfaces and device management; more efficient, also more cost efficient; centralized computing  
Microsoft Azure, Google Cloud, IBM Watson

**Edge:**   
distributed computing; data is processed where it was generated, meaning the periphery of the network as close to the originating source, which is preferable if real time relevancy is a priority, because it is faster; issues of latency and bandwidth

alternative names: router, gateway

**Fog:**“mix” of cloud and edge, direct connection between edge devices and the cloud / the internet; name stems from reference to the expression of “cloud” 🡪 both consist of water particles, except fog is more permeable, while a cloud is more dense

**Research on Onewire**

* aka “1-Wire”, is a communication bus system
* made by Maxim (formerly Dallas Semiconductor Corporation)
* low-speed data with 16.3 kbps
* single contact, easy to use, not very complex
* very energy-efficient and cost-efficient
* everything derives power from signal bus (parasite mode)
* only one master needed to control slaves
  + 1-Wire busmaster
  + connected to sensors 🡪 busmaster connects to first actor, that one connects to the next, etc. (multidrop, supports multiple devices on single line)
  + server controls busmaster via driver and communicates with the sensors
  + all software runs on server
* master sends impulses 🡪 under 15ms equals 1, over 15ms equals 0
* **Example:** Raspberry Pi: 1-Wire Temperatursensor DS1820
* Block diagram:

<https://www.maximintegrated.com/en/products/interface/one-wire.html>

**RS232**

* standard for serial communication
* connects two devices
* connection between terminal and modem
* has been replaced by now
* lower transmission speed, short maximum cable length, large voltage swing, large standard connectors, no multipoint capability and limited multidrop capability

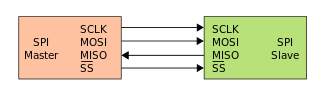
**RS485**

* industrial standard for asynchronous serial communication
* used in automation and industry
* half-duplex
* high-speed
* max. distance 1200 meters

**I²C**

* serial databus
* developed by Phillips
* five speed levels
* 10-400 kbps
* short-distance
* master-slave relationship
* slaves have fixed address
* low-active

**SPI**

* serial peripheral interface
* developed by Motorola
* “loose” standard for synchronous serial data bus
* data communication between micro controller
* settings variable: clock edge, word length
* short distance (1-10 meters)
* clock frequency up to MHz-range, also variable