National Research Council of Italy (CNR)
Institute of Science and Technologies of Information (ISTI)
Domotics Lab

Domotic evolution towards the IoT

Vittorio Miori, Dario Russo

# National Research Council of Italy

The *National Research Council of Italy* (*CNR)* is the biggest research public organization of Italy. Its duty is to carry out, promote, spread, transfer and improve research activities in the main sectors of knowledge growth and of its applications for the scientific, technological, economic and social development of the Country.

Domotics and AmI Lab

Research Lab of the "*Institute of Science and Technologies of Information*" (*ISTI*) of the *National Research Council of Italy* (*CNR*).

Domotics and AmI Lab focuses its activities on the **home** and **smart cities** environments and aims to improve the quality of everyday life contributing to achieve the **Ambient Intelligence** (*AmI*) and **Internet of Things** (*IoT*) visions.

Ambient Intelligence

Ambient Intelligence and Internet of Things are upon Domotics

Internet of Things
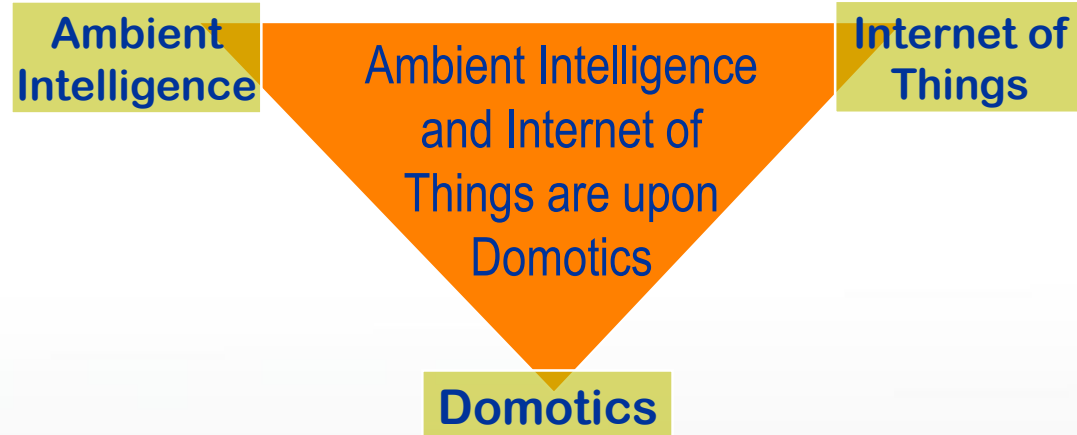
Domotics

The idea

In complaint with the *IoT* paradigm, the idea is to make available on Internet every domotic device, providing it with a *IPv6* address, **even in presence of closed and heterogeneous proprietary systems lacking any compatibility among them and with Internet Protocols**.
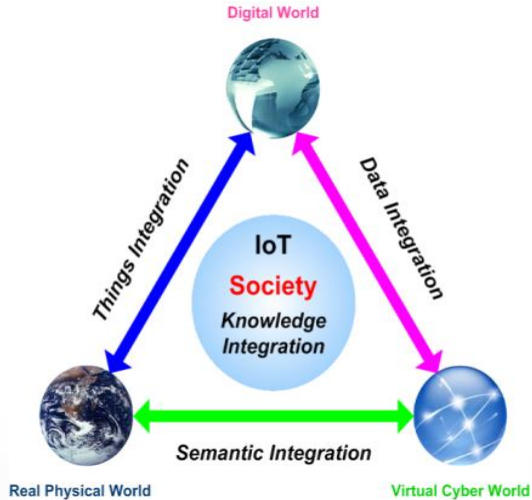
The solution should be:
- open (based on Internet standards);
- universal (independent by the technology of the domotic systems);
- scalable (with no hardware interventions).

To realize the desired objectives, **interoperability** among different domotic technologies and a high **abstraction** layer of devices are crucial.

Interoperability issue

One of the main obstacles of domotics is the presence of many non-interoperable standards such as:
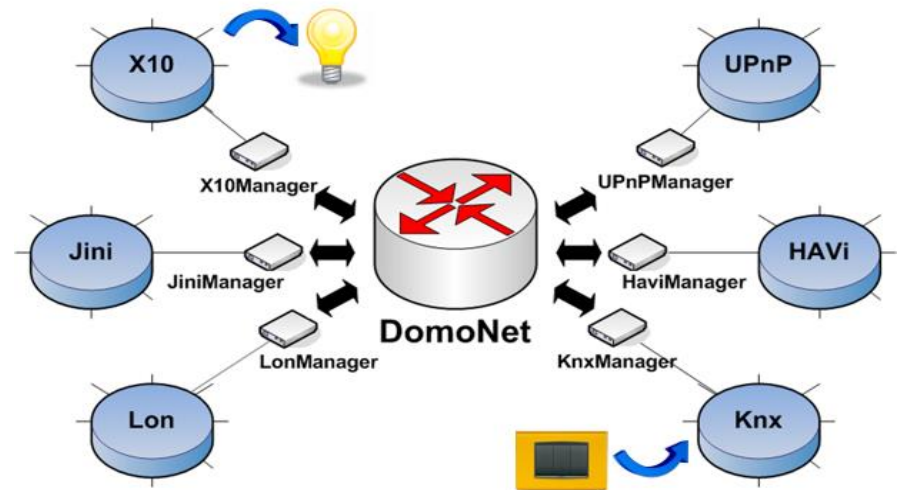
DomoNet Middleware 1 / 2

It provides integration and interoperability between the different home automation systems providing data, state, and action sharing, eliminating logically technological and protocol differences.

It is composed by:

- **server**: the core of the system;

- ***techManagers***: each of them is a special gateway able to interface:

  o a domotic bus, with its available devices;

  o the server, using the abstraction language called *DomoML.*

DomoNet Middleware 2 / 2

DomoNet is a genuine Internet node that offers *web services* facilities to control the functions of domotic devices from Internet;

**DomoML** is an *XML* universal and standardized language able to describe:

- **DomoDevice** (domotic devices and their functions);

- **DomoMessage** (commands, events and interactions);

- **DomoAddress** (to identify devices inside the *DomoNet* network).
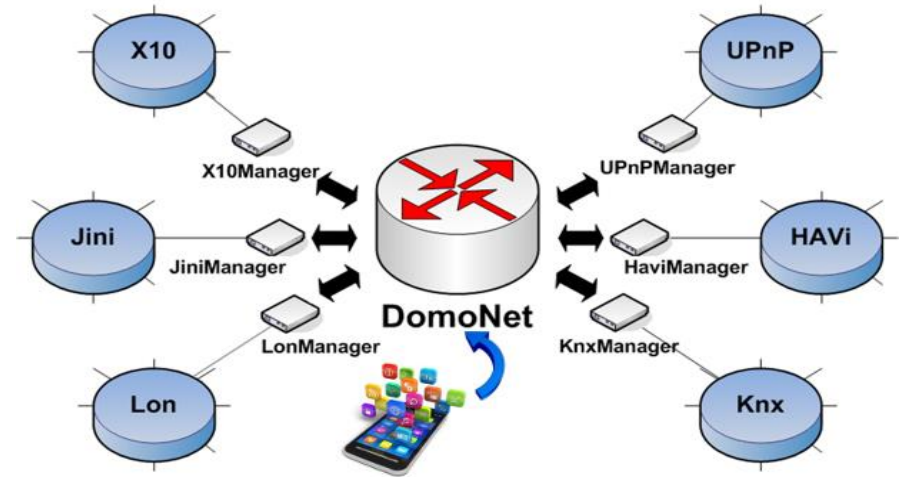
The IPv6 Gateway

The **IPv6 Gateway** is capable of making the *DomoNet* devices as true *IoT* things.

DomoNet manages the *IoT network* in the same way it treats the home automation networks.
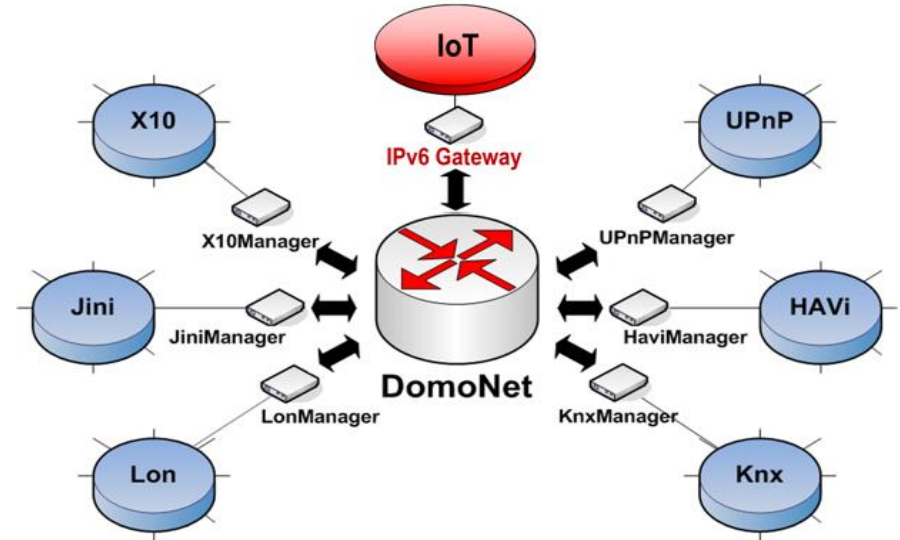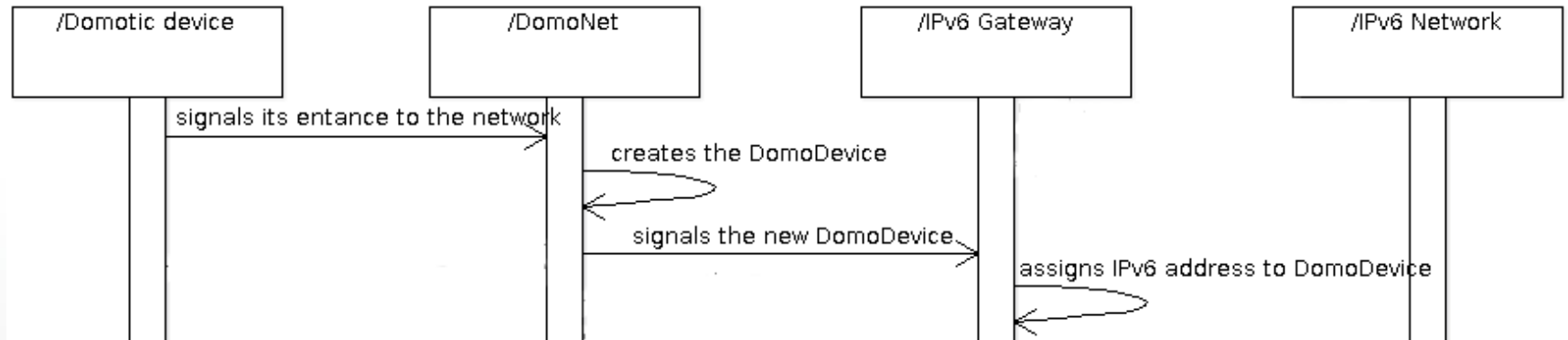
The IPv6 Gateway

The solution:

1. assigns a set of *IPv6* addresses to the machine where the *IPv6 Gateway* is running;
2. couples each *IPv6* address with each domotic device;
3. when a *IPv6* Internet request is addressed to a domotic device, the *IPv6 gateway* respond in place of this device;
4. implements a *IPv6 dynamic DNS* (*Bind9 server*) functionality to assign mnemonic names to devices.
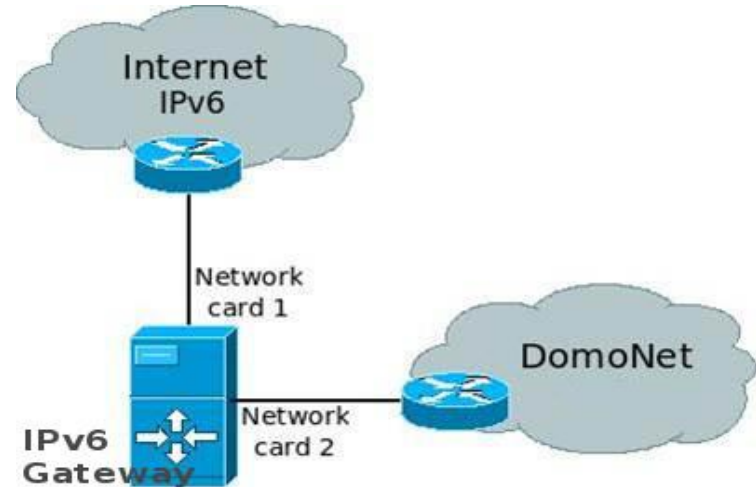
The IPv6 Gateway functioning 1 / 2

1. creation of a set of *IPv6* addresses;

2. when a new device signals its entry, *DomoNet* creates the corresponding *XML DomoDevice* and *DomoAddress;*

3. the *DomoDevice* is acquired by *IPv6 Gateway;*

4. *IPv6 Gateway* associates an *IPv6 address* to the device   *DomoAddress*, providing a dual address identification mechanism.

/Domotic device	/DomoNet	/IPv6 Gateway	/IPv6 Network

signals its entance to the network

creates the DomoDevice

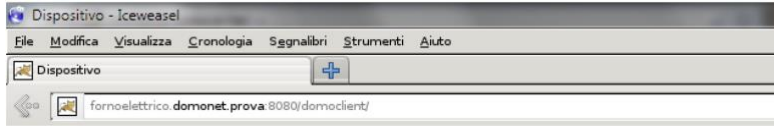signals the new DomoDevice

assigns IPv6 address to DomoDevice

The IPv6 Gateway functioning 2 / 2

1. when a device *IPv6* request is coming, the *DomoNet IPv6 gateway* responds;
2. the *IPv6 Gateway* find the corresponding *DomoNet Address* to forward the request to *DomoNet* device exploiting the dual address identification mechanism;
3. finally, the *IPv6 Gateway* delivers the device's response to the requester.
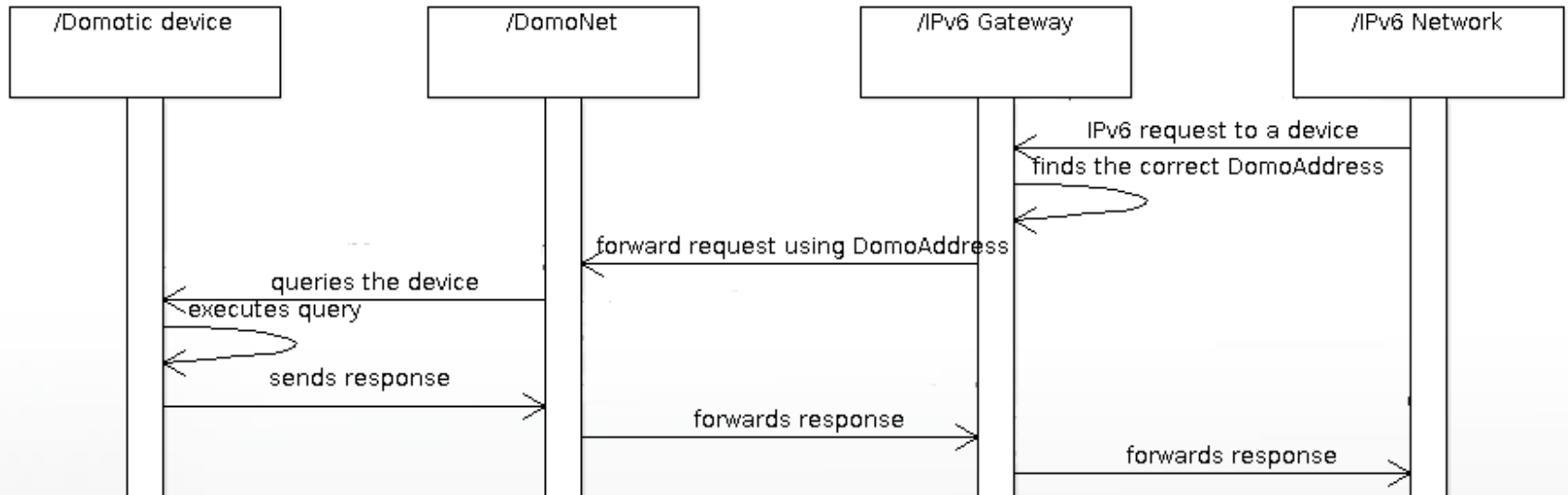
The IPv6 Gateway: a test web interface

- The *IPv6 Gateway* implements a dedicated *MVC (Model View Controller) Web interface* for each *DomoML* domotic device.

- This test interface displays the status and the available functions of each domotic *IPv6* device, in real time.

Conclusions

The solution:

- is scalable and do not requires hardware device modifications.

- is an extension of the *DomoNet* interoperability framework to add *IPv6* capabilities even to domotic systems unable to communicate natively via *IP* protocols.

- furnish a unique *IPv6* address to each domotic device belonging the *DomoNet* network, thereby making them remotely accessible directly via the Web.

- is immediately feasible helping hand to integrate in domotic systems the *IPv6*-based communications, towards the promised *Internet of Things*.

- can be applied also in other emerging fields where the necessary *IPv6* support is lacking, such as *Smart Cities*, *Smart Grid*, *Smart Lighting* and *Transportation*.

Thanks for the attention!

Vittorio Miori: vittorio.miori@isti.cnr.it

Dario Russo: dario.russo@isti.cnr.it

Domotics Lab Home Page: http://www.isti.cnr.it/research/unit.php?unit=HA

The IPv6 Gateway: IPv6 address creation

Each *IPv6* address in the set, created by the *DomoNet* server machine itself, is composed by two main fields in order to be valid:

- *a prefix* that it is used for routing purposes. This can be achieved by exploiting *OS* functions and *api*s. On a Linux machine, for instance, the *ip* command line is invoked to obtain the *network prefix field* and assign an address to the network interface;
- *an interface identifier*, which distinguishes the *host network interface*. This can be created randomly by the server machine itself.

If combining the *prefix* with *the interface identifier* leads to the generation of an already existing *IPv6* address, a *Duplicate Address Detection* error of the *NDP* protocol results.

128 bits

Network Prefix				Interface ID			
XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX
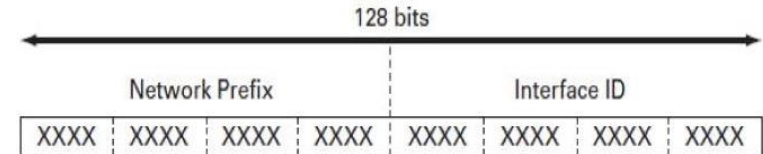
XXXX = 0000 through FFFF

3.4×10^{38} = ~340,282,366,920,938,463,374,607,432,768,211,456 IPv6 Addresses

330522