In this paper, we present a system architecture called Sindrion which allows to create a cheap, energy-efficient, wireless control network to integrate small embedded sensors and actuators into one of the most established middleware platforms for distributed semantic services, namely Universal Plug and Play (UPnP).

To meet rigid constraints regarding cost and power consumption, complex data processing is sourced out from the sensor or actuator nodes to dedicated computing terminals, which establish a proxy in the UPnP network.

I. I NTRODUCTION

In many future scenarios, the networked communication in different life spheres plays a critical role. Mainly, home automation systems are addressed, but also health care applications, information spreading beacons as well as maintenance and control of white goods are in the focus of such scenarios. For all these applications, wireless networked sensors and actuators which provide certain information and execute controlled actions are the central point.

Around those networks, several standards have been formed during the past years. Basically, in the home automation domain these are the Local Operating Network (LON) and the European Installation Bus (EIB). Unfortunately, these two standards are not based on wireless technology and only aim at the connection of larger devices, such as washing machines, HVAC, or door controlling systems. Furthermore, the integration cost of new devices to existing LON or EIB networks are too high for a wide-spread use of these systems.

In the area of wireless networking of devices, several standards have been developed, such as Bluetooth, Wireless Local Area Networks (WLAN) IEEE 802.11x, Wireless Personal Area Networks (WPAN) IEEE 802.15.4, or Short Range Wireless (SRW) systems. These standards primarily address the physical and the lower networking layers of the wireless connection. Flexible application-layer support for autonomous device interaction is just beginning to be supported on PCs and high-tech appliances. One of the most established middleware platforms for this purpose is Universal Plug and Play (UPnP): UPnP defines a set of universal, open, XML-based protocols which allow the semantic description and control of various devices. Thereby, it supports the flexible co-operation of the network nodes.

Unfortunately, the technical realization of the UPnP protocols is complex. Transceivers that support the entire UPnP protocol stack are thus power demanding and expensive, so that in particular many small and cheap peripheral devices will be excluded from the integration into such networks.

To this end, we have developed a distributed system architecture called Sindrion which features an effective solution for these small appliances. The basic idea is that complex operations are sourced out from the peripherals to dedicated computing terminals, which we call terminals. Sindrion supports the wireless communication and the autonomous inter-operation of these nodes. The system is aimed at wireless sensors, for which the power consumption and a low price of the nodes are as important as the capability to embed the devices into a well-established IT infrastructure.

II. B ASIC P RINCIPLE

The basis of the Sindrion system is to set up a wireless link between peripheral devices and dedicated computing terminals. The objective of this connection is to source out complex data processing from the peripherals to the terminals. To this end, the peripheral devices contain small smart transceivers\(^1\), the so-called Sindrion Transceivers, which are attached to embedded

\(^1\)The term “smart transceiver” refers to an RF transceiver with integrated microcontroller.
sensors or actuators (see Fig. 1). Typical peripheral devices are environmental sensors, small actuators like switches, or home appliances. They bear very limited or no computing power, and the embedded sensors and actuators can be controlled by simple proprietary analog or digital control lines. These are connected to the input- and output ports (I/O ports) of the Sindrion Transceivers. The terminal is equipped with an RF transceiver which is compatible with that included in the Sindrion Transceiver. Data and protocol processing are done in the terminal, which features a virtually unlimited amount of processing power and memory compared to the Sindrion Transceiver.

Fig. 1 shows the fundamental structure establishing the communication between the terminal and a previously unknown Sindrion Transceiver. The procedure is as follows:

1) **Discovery**: The two end devices find each other in the discovery phase. To this end, the UPnP discovery protocol is used.

2) **Code Download**: If the terminal does not yet contain the control application for the Sindrion Transceiver, the transceiver’s application code is downloaded by the terminal. Preferably, this code is written for a middleware platform such as the Java Virtual Machine. This guarantees platform independence and allows for the seamless integration into various terminals. Further, high-level programming languages facilitate application development.

3) **Application-Specific Communication**: The following communication between the downloaded service application on the terminal and its counterpart - the transceiver control - on the Sindrion Transceiver may be completely application-specific and does not have to be defined by any standard.

III. **THE UPNP PLATFORM FOR SENSOR NODES**

A middleware platform for the seamless interaction of end devices has to support spontaneous ad-hoc networking of components without manual installation and has to be flexible enough to integrate low-cost smart sensors and actuators.

The Universal Plug and Play (UPnP) Architecture is one of the best established middleware platforms for Ubiquitous Computing. UPnP uses open and standardized protocols based on XML which allow to describe and control various devices. All communication is transferred over TCP/IP resp. UDP/IP. The most important advantages of this open concept are semantic interoperability of devices from different vendors and the simple extensibility to future devices. On the odd side, there is the intense requirement for memory and computing power for UPnP Devices since variable XML-based protocols have to be processed. On first sight, this excludes cheap and low-power sensors and actuator nodes from participating in a UPnP environment.

On second sight, UPnP bridges resp. gateways may be used to connect a non-UPnP network to a UPnP environment. In this approach, sensor nodes have to be configured as nodes of some specific network and additionally as UPnP Devices inside the gateway. Therefore, seamless integration without user interaction and complex configuration as envisaged for ambient intelligence applications cannot be accomplished. Furthermore, if just one or few sensor nodes are added to an existing UPnP environment, an additional gateway device has to be installed as expensive overhead.
IV. THE SENSOR NODE AS UPnP DEVICE

With Sindrion, we have found a third way to integrate devices into a UPnP environment which is compatible with UPnP and neither requires any gateway nor gateway configuration. For a better understanding, we take a closer look at the steps of the UPnP Device Architecture:

- **Addressing**: A UPnP Device requires a network address using a DHCP server or, if none is available, using Auto-IP.
- **Discovery**: UPnP Devices advertise themselves by multicast messages and react according to search messages.
- **Description**: UPnP Devices and their embedded services are defined by XML description files which have to be provided for HTTP download by the device. Note that these files are static and will not change except for few lines of code.
- **Control**: UPnP Devices are controlled by the XML-based SOAP protocol and therefore they process XML messages and they answer with dynamically created XML messages.
- **Eventing**: UPnP Devices maintain event subscriber lists, accept subscriptions and keep track of subscription durations. When an event occurs, XML-based event messages with dynamic content are sent to each subscriber.
- **Presentation**: The optional presentation page exposes an HTML-based user interface which may be used for controlling the device by embedded elements like Java applets.

The complexity of UPnP control and eventing prevents a cheap and low-power device to implement the complete UPnP Device Architecture. Thus, every Sindrion Transceiver will act as so-called UPnP Basic Device, which is a special device type without embedded UPnP services. The Basic Device supports UPnP discovery, description and presentation, which need much less computational effort than UPnP control and eventing. Without prerequisites, a Sindrion Transceiver can therefore be connected directly to a UPnP network. Following the Sindrion approach, the downloaded presentation page contains a Java applet, by which the Sindrion Transceiver and its attached peripheral device can be controlled from inside a Web browser.

Complex data processing in UPnP control and eventing can, however, also be exported to a terminal following the Sindrion approach. Thereto, a Sindrion Application Manager demon running on the terminal is used to discover a Sindrion Transceiver by standard UPnP mechanisms and to download the service application code from the transceiver. Next, the service application is started as Java application on the terminal and acts as a UPnP Proxy for the connected transceiver. This UPnP Proxy translates the application-specific communication protocol into full-fledged XML-based UPnP control and eventing messages.

With the given software architecture, a Sindrion Transceiver is seamlessly embedded into the UPnP environment. The underlying scheme is opaque to the other members of the UPnP network - the sensor node just appears as any fully enabled UPnP Device as shown in Fig. 2. Further, different nodes can interact by the UPnP interface to form complex services. For example, a thermometer device may provide temperature data to a small LCD display in another room, at the same time to a PC to store a temperature timeline, and to an HVAC system for ventilation control.

V. CONCLUSION

In this paper, we have introduced the Sindrion system which provides a wireless connection between so-called Sindrion Transceivers attached to peripheral devices and dedicated computing terminals. By establishing a proxy on the terminal, also small embedded systems can be integrated into the Universal Plug and Play (UPnP) environment, which is the predominant middleware platform for semantic device interoperability. The Sindrion system itself operates in accordance to the UPnP framework and handles the integration of the transceivers into the UPnP network in an autonomous and transparent way.