THE 9TH ANNUAL INTERNATIONAL SYMPOSIUM ON APPLICATIONS AND THE INTERNET (SAINT 2009)

DEVICE INTEGRATION IN SODA USING THE DEVICE DESCRIPTION LANGUAGE

Chao Chen and Abdelsalam (Sumi) Helal

UF UNIVERSITY of FLORIDA
The Foundation for The Gator Nation

INTRODUCTION

- The Pervasive Internet → a wild jungle of devices and gadgets.
- Heterogeneity of
  - Devices:
    - From pin-head sensors to complex devices & appliances
  - Networks:
    - Ethernet, WiFi, Bluetooth, ZigBee, …
  - Running Environments:
    - None, OSGi, Jini, UPnP…
  - Device standards:
    - SensorML, IEEE 1451, ECHONET, Device Kit…

The Hard-wiring Problem

- The typical ad-hoc strategy to integrate a device
  - A steep learning curve and laborious hacking experience
    - Needs to examine its interface and study the device protocol to establish a connection.
    - May not produce re-usable code or know-how.
  - The lack of standard on both hardware and software ends
    - No automatic way to link the numerous and heterogeneous device I/O
    - Programmers have to manually write code to associate devices with specific tasks.
- The problem becomes even more acute when
  - the number of device types continues to grow → the number of interfaces, connections, protocols multiplies;
  - One considers how systems need to evolve throughout their life-cycles.

Our Response and Approach

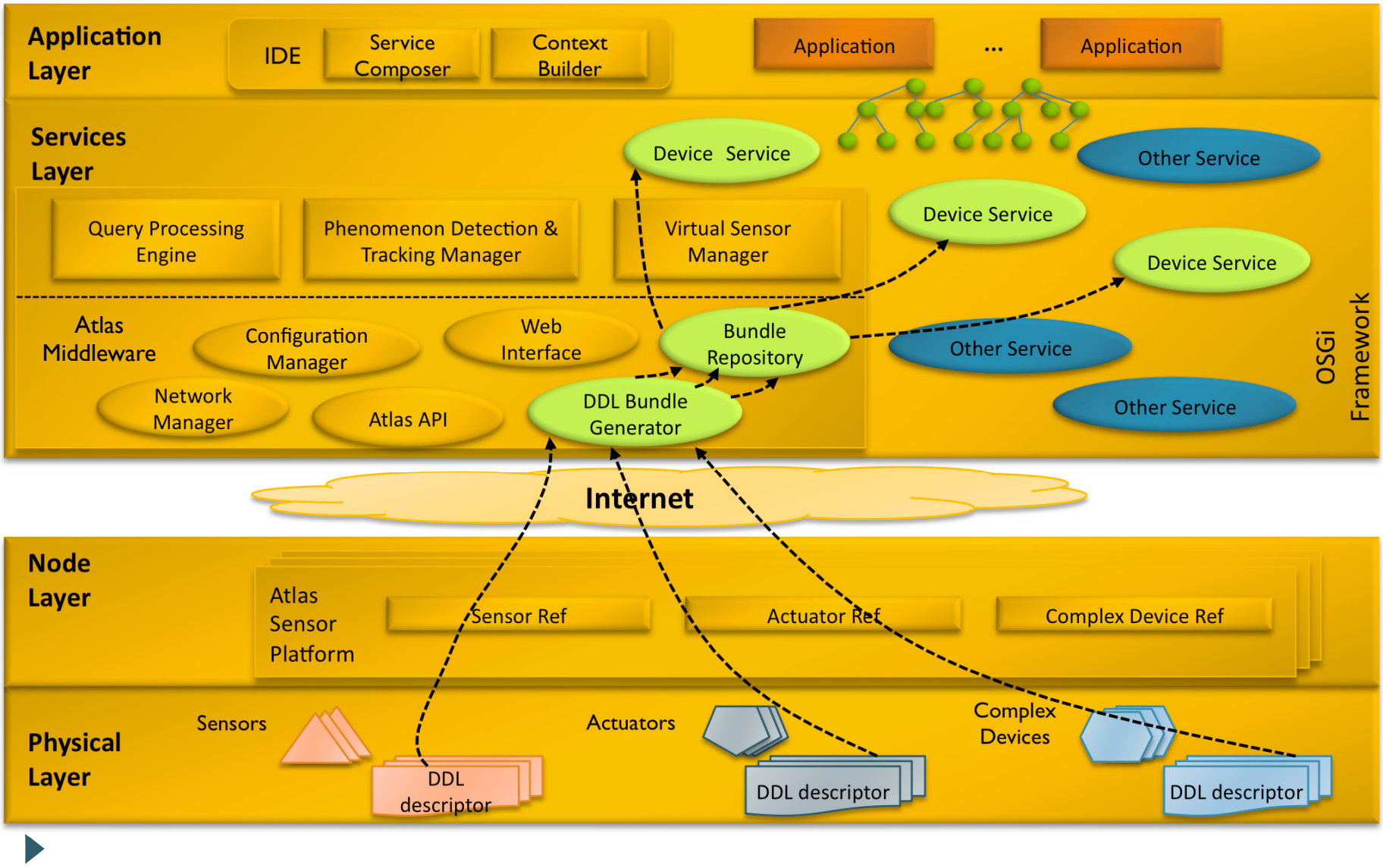
- Automate the process of device integration, so it would require ***0-effort*** from system integrators and pervasive system builders.
- Create an integration technology and associated proposed open standard (SODA) to allow device and sensor vendors to introduce such self-integrating products into the market place.
  - DDL - A descriptive language that describes a broad range of devices.
  - ATLAS - a service-oriented reference architecture for device integrations.
  - DDL / ATLAS - A proposed implementation of the SODA standard.

Talk Outline

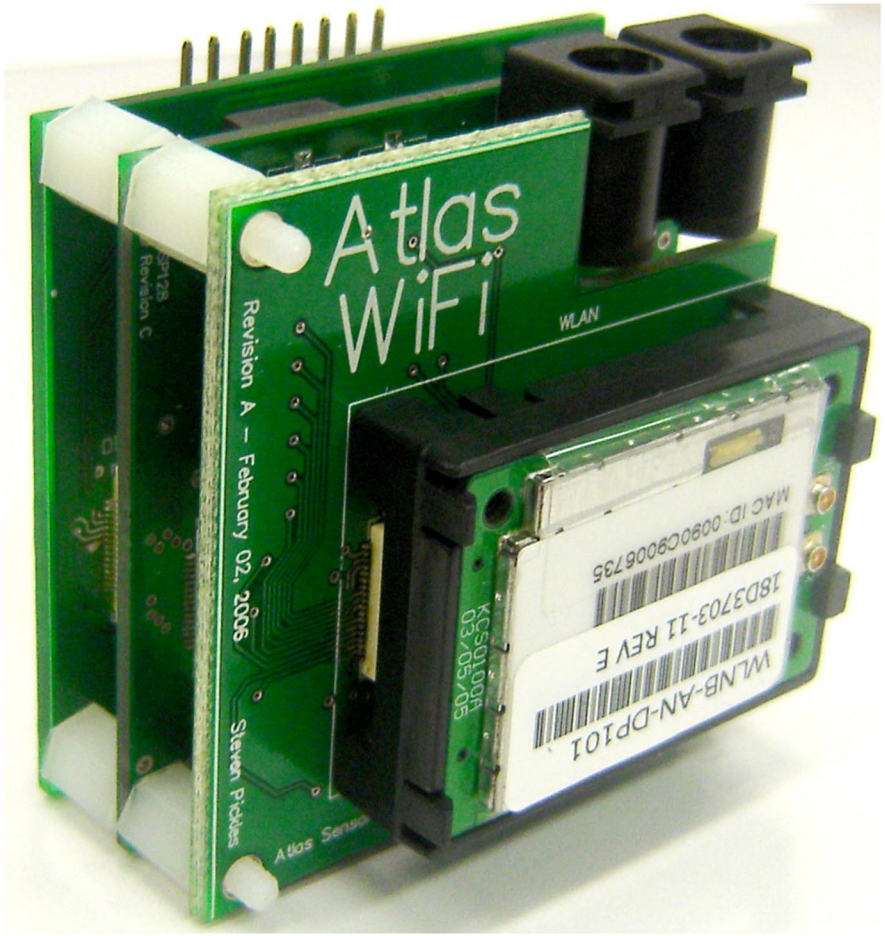
- ATLAS: a service-oriented reference architecture for sensor and device integration
- DDL: Device Description Language - schema and its reference implementation
- Case Study: constructing a self-sensing space in the Gator Tech Smart House (GTSH)
- Standardization through SODA
- Related work
- Conclusion

ATLAS:
A Reference Architecture for Service Oriented Sensor Platforms

The Atlas Architecture



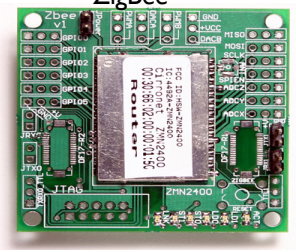
Atlas Platform



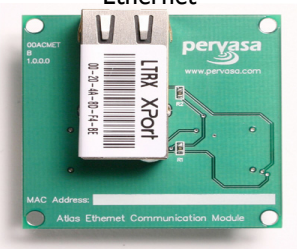
Atlas Platform

Atlas Communication Layer

ZigBee

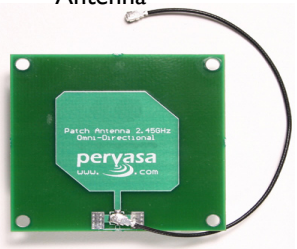


Ethernet



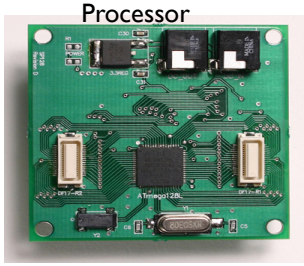
Wi-Fi

Antenna



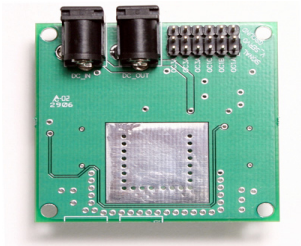
Atlas Processing Layer

Processor

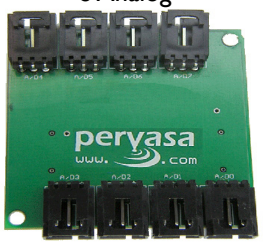


Atlas Device Interface Layer

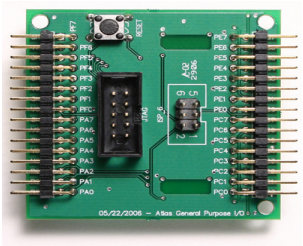
Servo



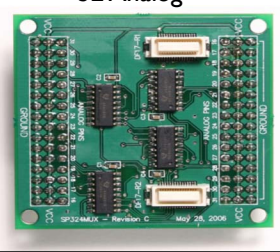
8 Analog



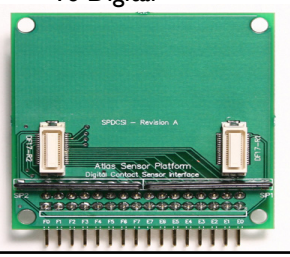
GPIO



32 Analog



16 Digital



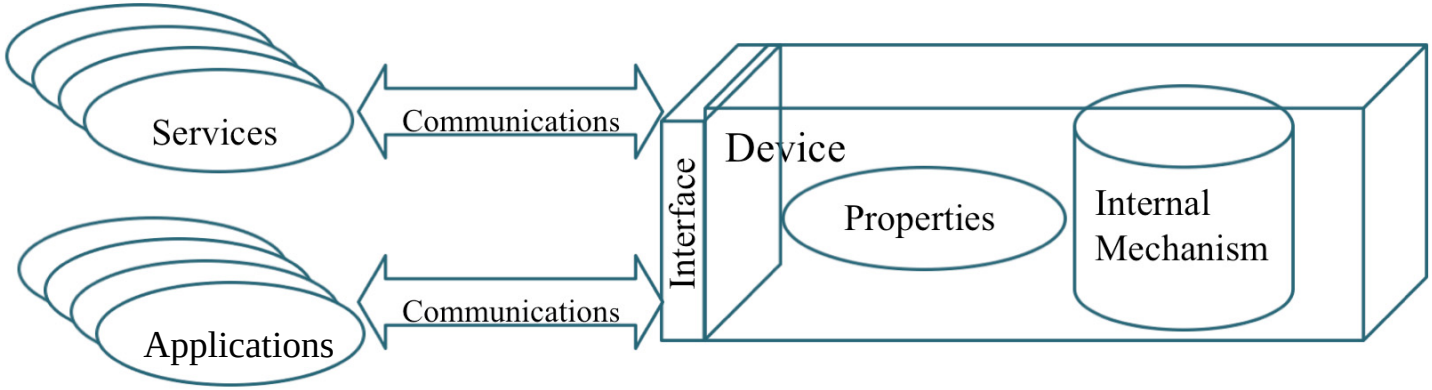
peryasa

Device Description Language (DDL)

Language Schema and its Implementation

Modeling Devices in DDL

- A DDL device model



- DDL classifies devices into 3 categories:
  - Sensors
  - Actuators
  - Complex Devices

The DDL Language Schema

- DDL uses XML encodings.
- Readable to both human and machine.
- A DDL schema defines the constraints on the structure and the content of a DDL document.
- The schema will be enforced by the DDL validity checker, a component of the DDL language processor.

```
<xsd:complexType name="Device">
<xsd:sequence>
 <xsd:element name="Description" type="DescriptionType"
minOccurs = "1" maxOccurs="1" />
  <xsd:element name="Interface" type="InterfaceType"
minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
```

What's inside a DDL Descriptor File?

- Each DDL descriptor file describes a single type of device.
- It contains:
  - Information for service registration and discovery
    - e.g., device name, model, function description, etc.
  - Description of device operations
    - each operation is a collection of *input/processing/output* function chains
    - the low-level communication between a device and its service are represented as '**Signals**'
    - the high level semantics of signals are '**Readings**'

An Example: TMP36 Analog Temperature Sensor

- An analog sensor:
  - "**Signal**": a converted value output from the ADC port on the sensor platform.
  - "**Reading**": the temperature value in centigrade.
  - DDL defines
    - the semantics of a temperature reading
    - the process of the **signal to reading** conversion

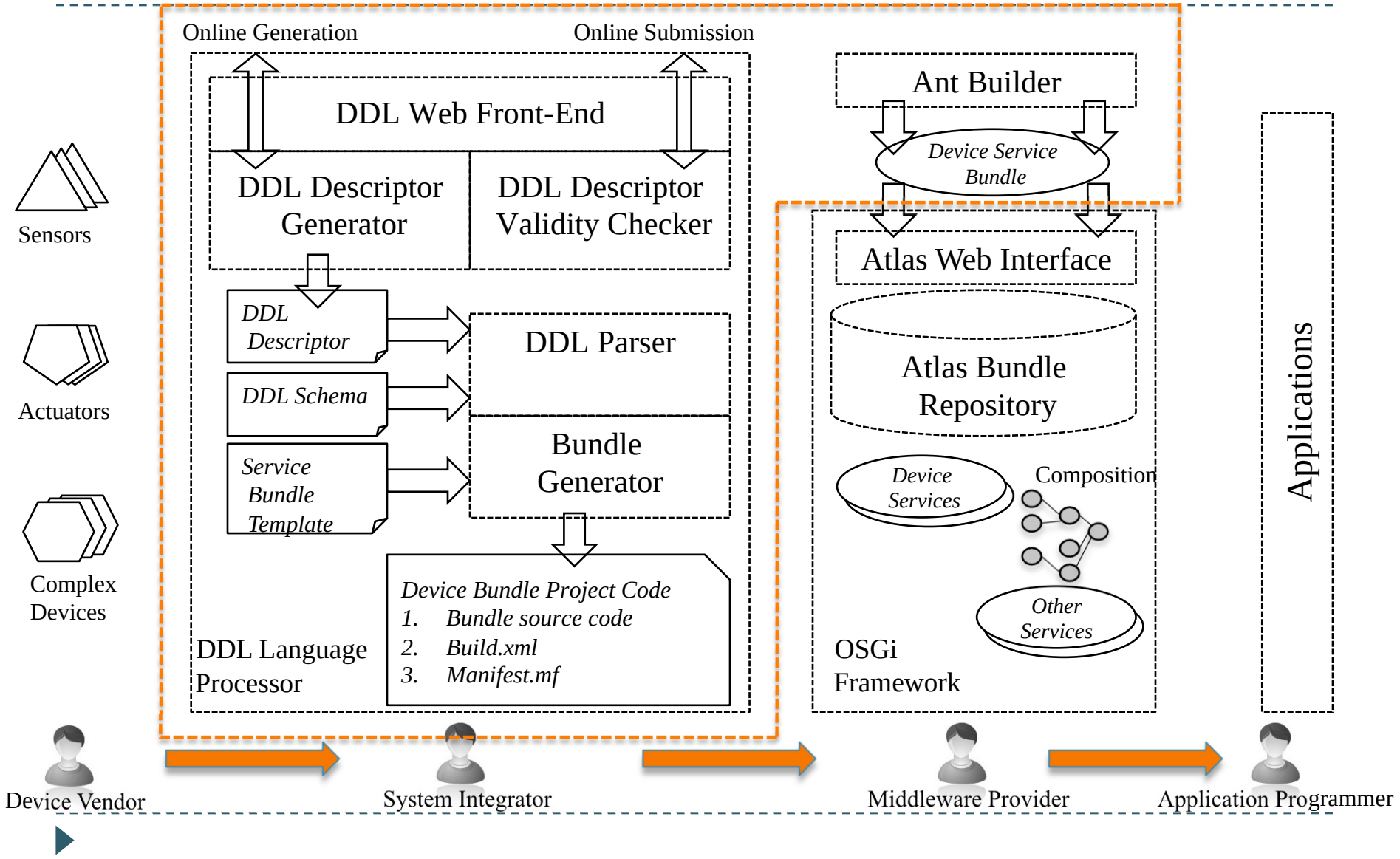
```
<Sensor>
<Description>…</Description>
 <Interface>
  <Signal id="ADC1">…</Signal>
  <Reading id="Temp1">
   <Type>Physical</Type>
   <Measurement>Temperature </
Measurement>
   <Unit>Centigrade</Unit>
   <Computation>
  <Type>Formula</Type>
  <Expression> Temp1 =
(((ADC1/1023) * 3.3)-0.5)*
(1000/10)</Expression>
   </Computation>
  </Reading>
 </Interface>
</Sensor>
```

Another Example: UA-767PC Digital Blood Pressure Monitor

- A complex device:
  - "**Signal**": the byte stream output from the serial port.
  - "**Reading**": the blood pressure measurement converted from the byte stream.

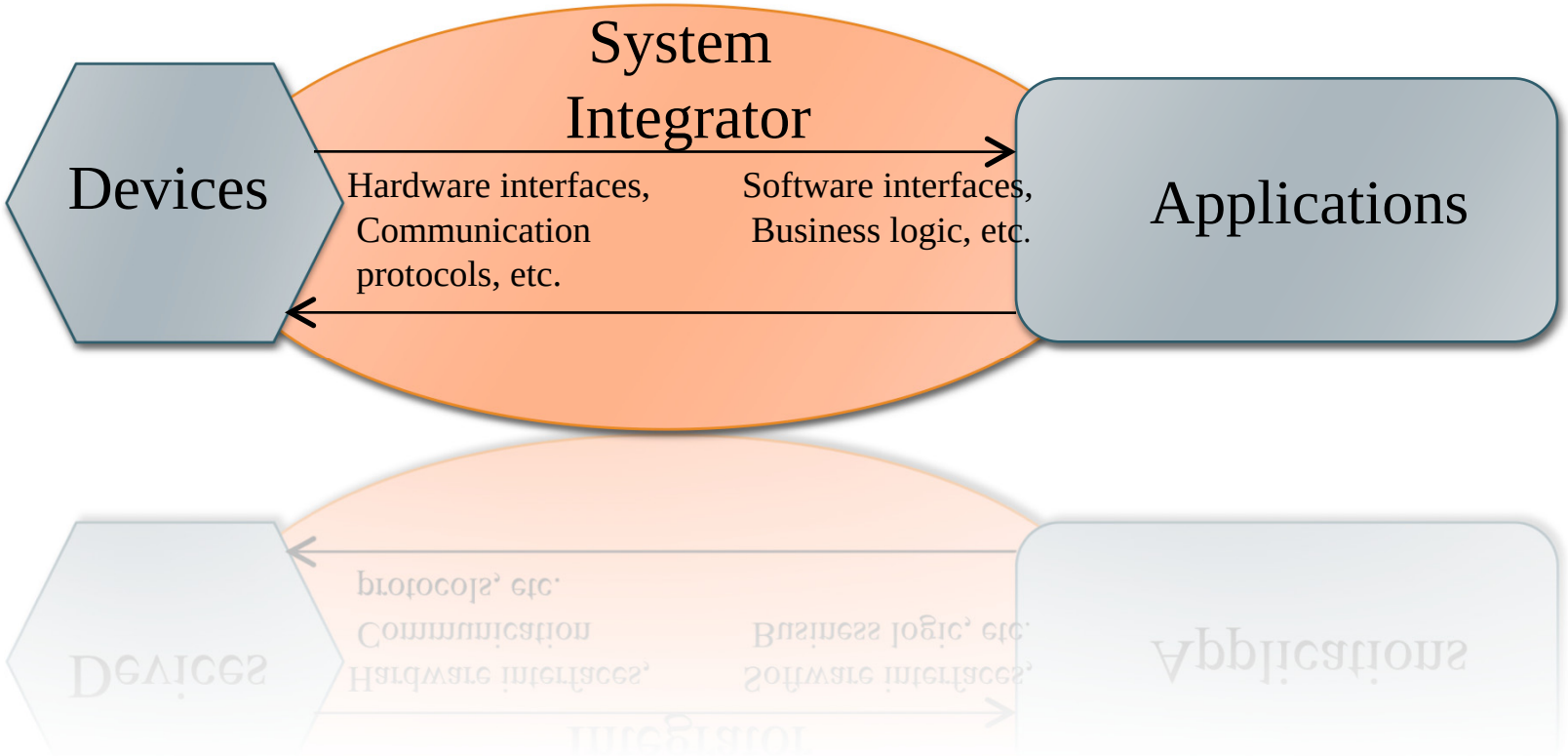
```
<ComplexDevice>
<Description>…</Description>
 <Interface>
  <Signal id=”signal1">…</Signal>
  <Reading id=”diastolic">
   <Type>Physical</Type>
   <Measurement>Diastolic Pressure</
Measurement>
   <Unit>mmHg</Unit>
   <Computation>
  <Type>ByteStreamFilter</Type>
  <Expression> diastolic =
signal1.substring(4,6)</Expression>
   </Computation>
  </Reading>
 </Interface>
</ComplexDevice>
```

The Integration Process



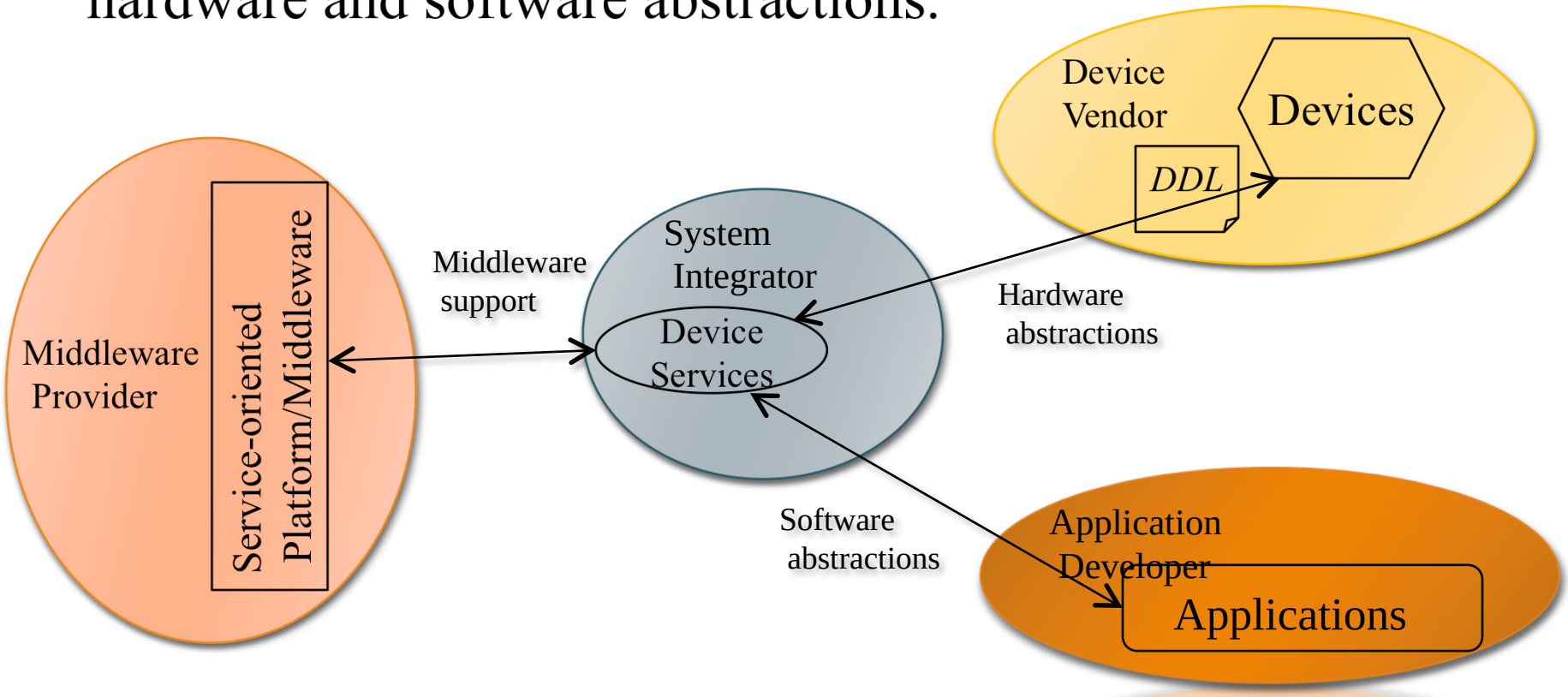
Role Implications (without ATLAS/DDL)

- The system integrator is the sole player in the field and has to deal with both hardware and software complications.



Role Implications (with ATLAS/DDL)

- Multiple roles are engaged;
- Their responsibilities are clearly separated by both hardware and software abstractions.

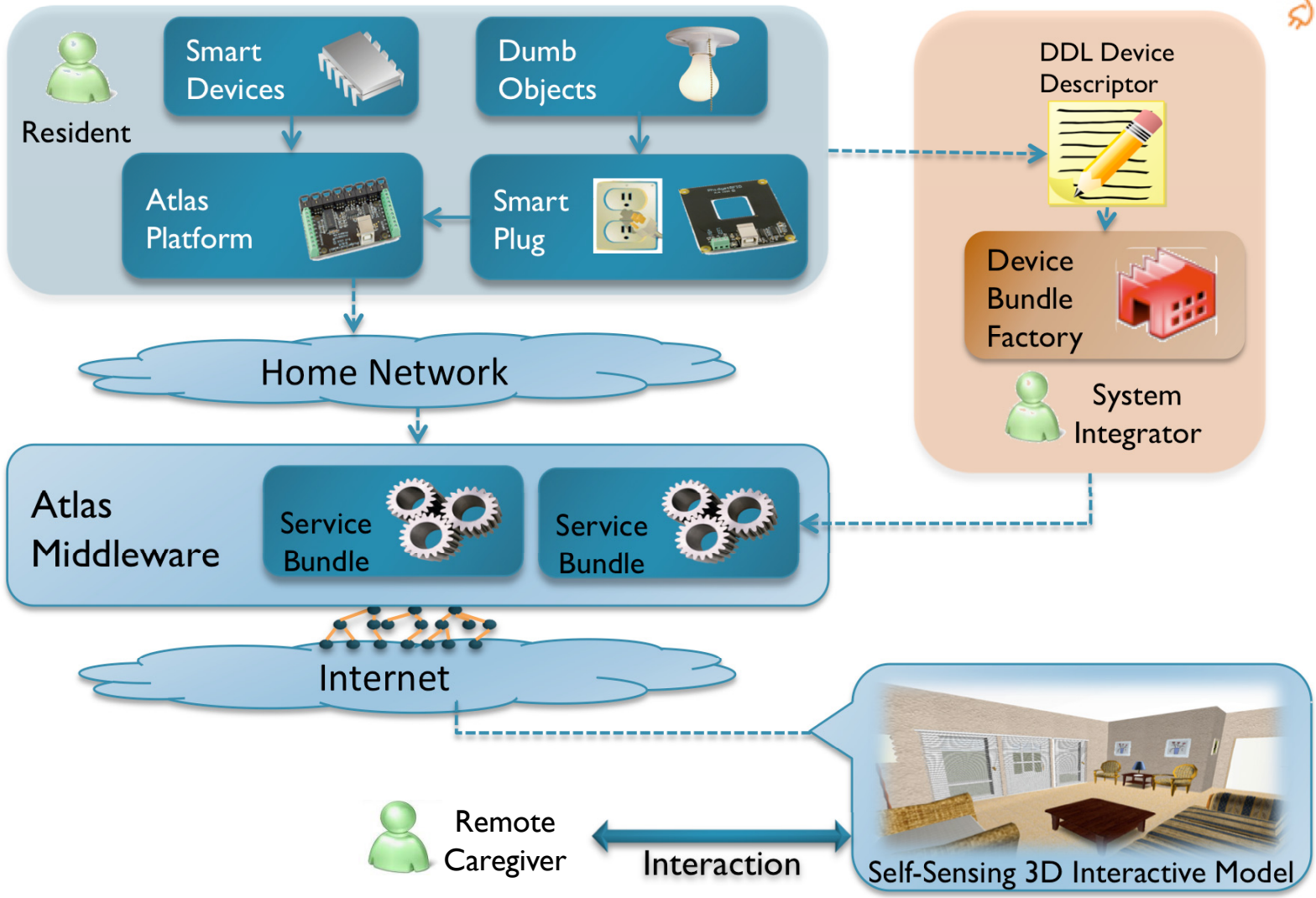


Case Study:
Constructing a Self-Sensing Space in the Gator Tech Smart House

CASE STUDY

- Self-Sensing Space:
  - An intelligent environment that recognizes its devices and services, interpret their status, and generate a model of the space.
- The challenges:
  - Integration of dumb objects: a self-sensing space should not ignore everyday objects such as furniture and electric appliances.
  - End-to-end self-integration: both smart devices and dumb objects should be seamlessly self-integrated into the space.

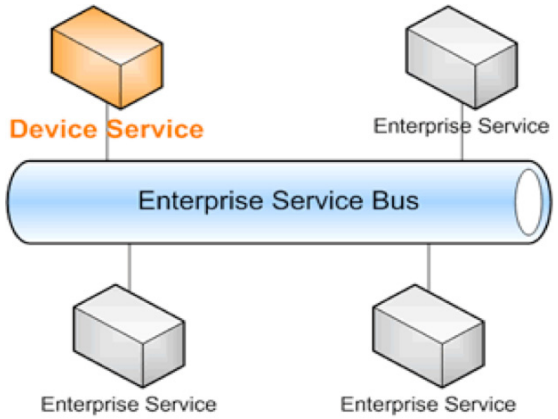
Application Scenario



Standardization through SODA

STANDARDIZATION

- DDL is a proposed implementation of the *Service-Oriented Device Architecture (SODA)* standard framework.
- SODA is
  - an emerging standard alliance,
  - an extension to *SOA* to incorporate devices in distributed enterprise systems.
- When modeled as a service, device access and control can be made available to a wide range of enterprise applications using SOA mechanisms.



STANDARDIZATION (II)

- The DDL language specification and its software are available online at
  - http://www.icta.ufl.edu/atlas/ddl/

Related Work

Related Work

- There have been a number of standards proposed:
  - ECHONET
    - The Energy Conservation and Homecare Network standard, initiated in Japan.
  - IEEE 1451
    - The IEEE standard for smart transducer interfaces.
  - SensorML
    - The Sensor Model Language, initiated in the geospatial community.
  - Device Kit
    - An IBM implementation of the SODA architecture

Comparison of Standards

Key Comparisons	ECHONET	IEEE 1451	SensorML	Device Kit	DDL
Encoding	Class specification in plaintext	Interface Definition Language	XML	XML	XML
Design perspective	Object-oriented	Modular	Data-oriented	Modular	Data-oriented
Device Model	Single object	Multiple blocks	Process chain	Multiple layers	Single device, cross layer
Other Comparisons	ECHONET	IEEE 1451	SensorML	Device Kit	DDL
Basic component	Device	Block	Process	Device function layer	Device
Composite component	NA	Device	Process Chain	Device	Derived virtual sensor
Measurement modeling	Primitive data types	Complex data types	Complex data types	Primitive data types	Primitive data types with aggregation
Protocol modeling	Inexplicit	Explicit	Inexplicit	Explicit	Explicit
Software support	NA	NA	NA	DKML parser and Eclipse plug-in	DDL Language processor
Specification	Published	Published	Published	Only schema available	Published online

Conclusion

CONCLUSION

- The scale and pace by which the Pervasive Internet is evolving today demand a new breed of integration technology that is scalable and automatic.
- The Device Description Language within the Atlas sensor platform and middleware is capable for describing and integrating a great variety of devices ranging from a pinhead sensor to a complex device.
- Currently sensors and complex device integration are supported.
- We are now working on an improved design of the Atlas firmware to better support actuator integrations.

Thank you!