Wireless Embedded Systems and Networking

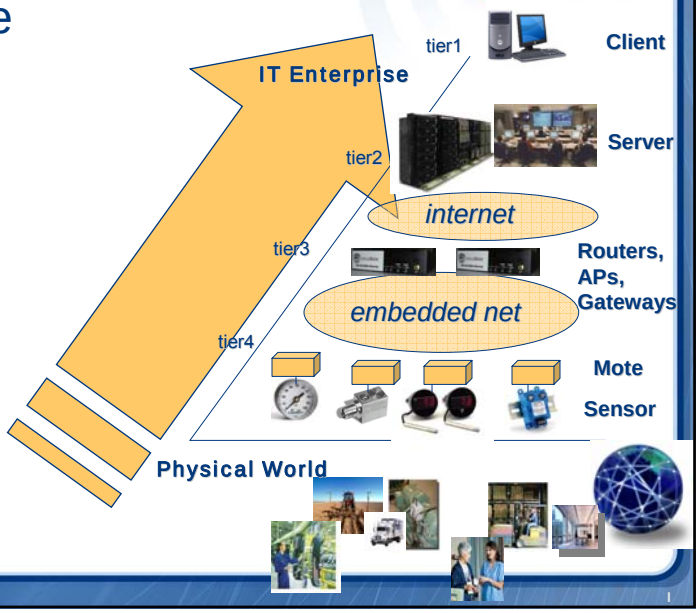
Foundations of IP-based Ubiquitous Sensor Networks

Embedded Web Services and Industrial Instrumentation Standards

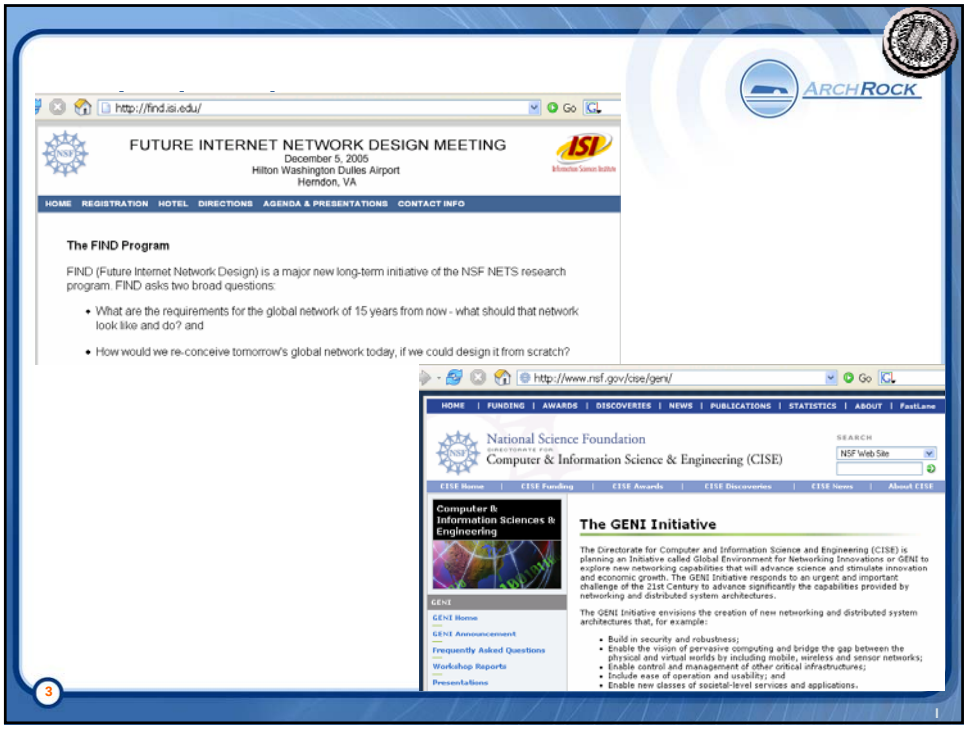
David E. Culler
University of California, Berkeley
Arch Rock Corp.
July 10, 2007

ARCHROCK

Embedded Web Services Perspective

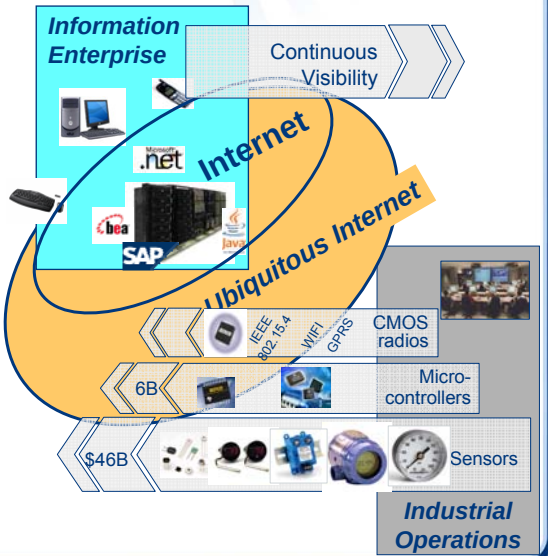


2



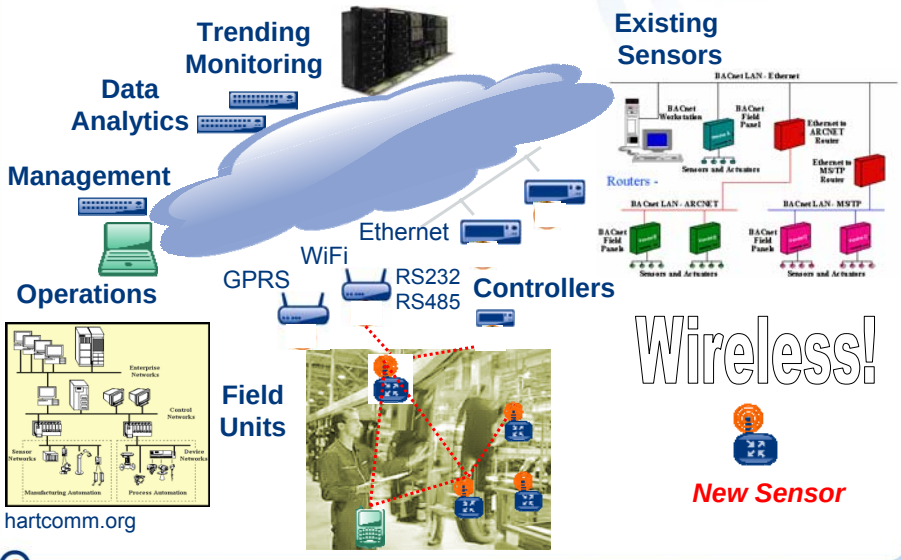
Internetworking Everything

- **Billions** of devices today sensing their environment
  - Homes, offices, factories, streets, hospitals, autos, …
  - **Data is dropped** or local
- Demand for **operational visibility** throughout the Enterprise
  - Supply chain, work flow, …
- Opportunity:
  - **Extend reach** and lower cost through wireless mesh networks
  - **Provide global visibility** by bringing sensors to the IP Network and Web



4

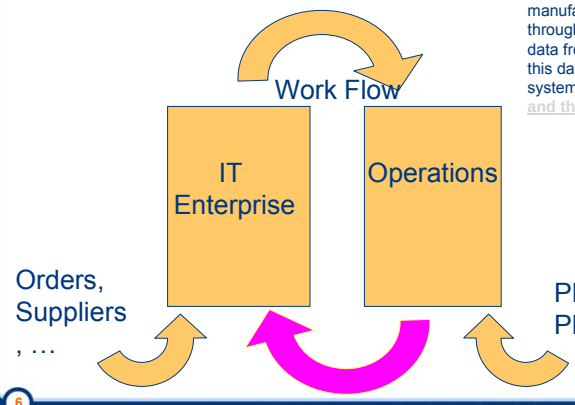
Integrating a World of Sensors



5

Closing the Loop

Today, however, corporate expectations for the manufacturing automation network landscape have changed dramatically, thanks to the rapid and ubiquitous adoption of Internet technology. Companies of all sizes, all over the world, are trying to find the best ways to connect the entire enterprise. No longer is control of the manufacturing processes enough: the new manufacturing mandate is to enable users throughout the company to access manufacturing data from any location, at any time, and to integrate this data seamlessly with business information systems. [The Common Industrial Protocol (CIP) and the Family of CIP Networks (Pub 123)]



6

Long road toward integration

- 1950: 4-20 mA “current loop”
  - Common signal wiring, ADC, and calibration
  - Vast diversity in excitation, configuration, interpretation
- 1980: HART (Highway Addressable Remote Transducer)
  - 1200 baud, half-duplex digital communication over 4-20 wiring
  - Rosemount proprietary protocol => “open” => Fieldbus
  - Fixed packet format for command / response
    - Process Variable, Host->Device Commands, Status & Diagnostic Alerts, Device Id, Calibration and Limits
- 1987: BACnet (Building Automation and Control Network)
  - RS232, RS485, ARCnet, ethernet, LONTalk, … BACnet/IP
  - Device = Collection of Objects; 23 “object types”
  - Data types, packet formats, and object defined in Abstract Syntax (ASN.1)
  - Protocol services, Data Sharing, Alarm and Events, Trending, Scheduling, Remote Device and Network Management
- 1994: CIP (Common Industrial Protocol)
  - Device Net (CAN), ControlNet, … EtherNet/IP
  - Devices as physical instances of classes.
  - Communication between objects is independent of physical links providing transport
  - Fixed binary encodings of data types, object types, classes
- 200x: Zigbee, ZWave, Wireless HART, SP100.11a, …
  - IEEE 802.15.4 radio …

7

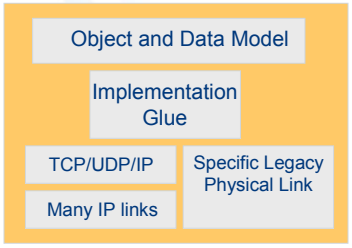
The Challenge - Diversity

- So many different kinds of sensors…
  - Different physical phenomenon
  - Different electrical connections
  - Different means of communication
  - Different Logical connections
    - Control operations, configuration, calibration
  - Translation to engineering units
  - Wide range of autonomy and intelligence
- Different data representations, encodings, …
- Different operations, capabilities, …
- Different limitations and constraints
- Different vendors, standards, interconnects, …

8

Relationship to Industrial Interconnects

- BACnet
  - RS-232 & RS-485 => IEEE 802.3 via BACnet/IP
- LONworks
  - Twisted Pair & Power Line => LonTalk/IP
- Common Industrial Protocol (CIP)
  - CAN & ControlNet => EtherNet/IP
- SCADA
  - Prop. RS-485 & Leased Line & Prop. Radios => ModBUS => Ethernet => TCP/IP
- FieldBus
  - Modbus, Profibus, Ind. Ethernet, Foundation HSE, H1, …SP100.11a?



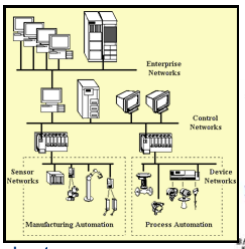
In 2000, ODVA and CI introduced another member of the CIP family: EtherNet/IP, where "IP" stands for "Industrial Protocol." In this network adaptation, CIP runs over TCP/IP and therefore can be deployed over any TCP/IP supported data link and physical layers, the most popular of which is IEEE 802.311, commonly known as Ethernet. The universal principles of CIP easily lend themselves to possible future implementations on new physical/ data link layers. [The Common Industrial Protocol (CIP) and the Family of CIP Networks (Pub 123)]

9

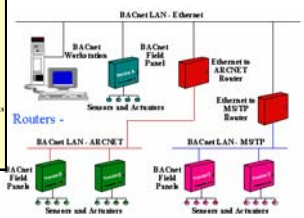
Making Sense out of Sensors

10

Technology Transformation - Bottom Line



hartcomm.org




Microcontroller


Flash Storage


Radio

- Sensors have become “physical information servers”
- Treat them as “information servers” to improve integration

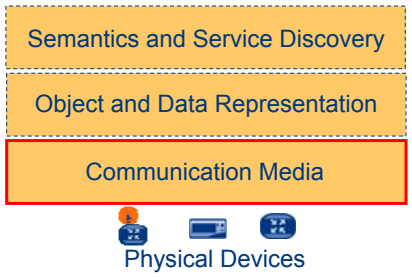
11

The Web …

Integrates diverse *Human Generated* Information

12

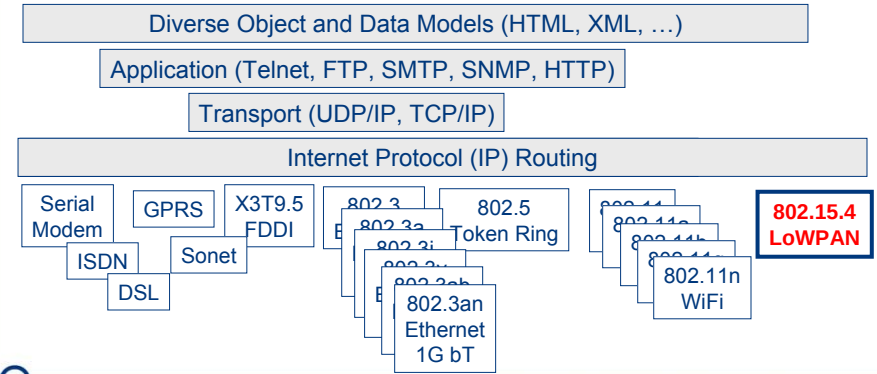
Making Sense out of Sensors



13

Lesson 1: IP

- Separate the logical communication of information from the physical links that carry the packets.
  - Naming
    - Hostname => IP address => Physical MAC
  - Routing
  - Security



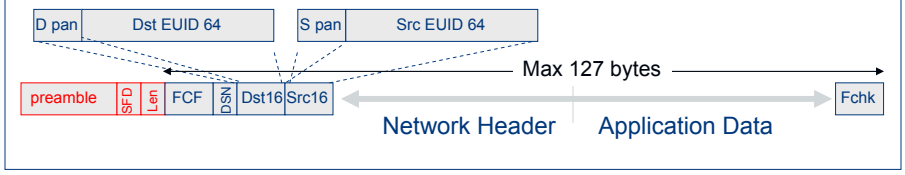
14

But, …

- isn't IP too heavyweight for low-power, wireless, microcontroller based devices?
- **No!**
- 6lowpan compression with high quality multihop routing
  - Reliability and lifetime of the best mesh
  - Interoperability of IP

15

IEEE 802.15.4



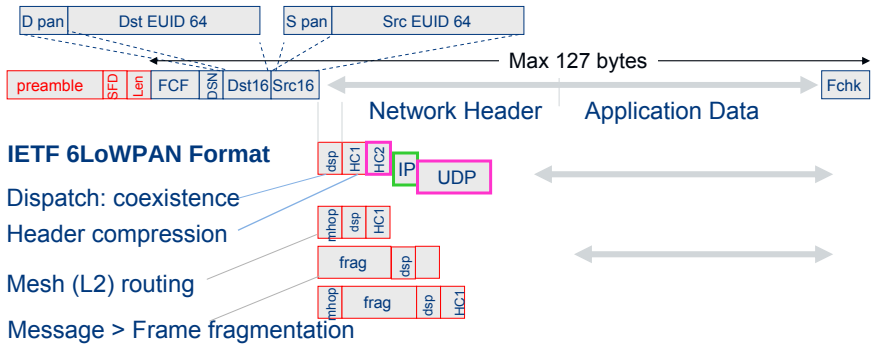
- Low Bandwidth (250 kbps), low power (1 mW) radio
- Moderately spread spectrum (QPSK) provides robustness
- Simple MAC allows for general use
  - Many TinyOS-based protocols (MintRoute, LQI, BVR, …), TinyAODV, Zigbee, SP100.11, Wireless HART, …
  - 6LoWPAN => IP
- Choice among many semiconductor suppliers
- Small Packets to keep packet error rate low and permit media sharing

16

6LoWPAN Format Design

- Orthogonal stackable header format
- Almost no overhead for the ability to interoperate and scale.
- Pay for only what you use

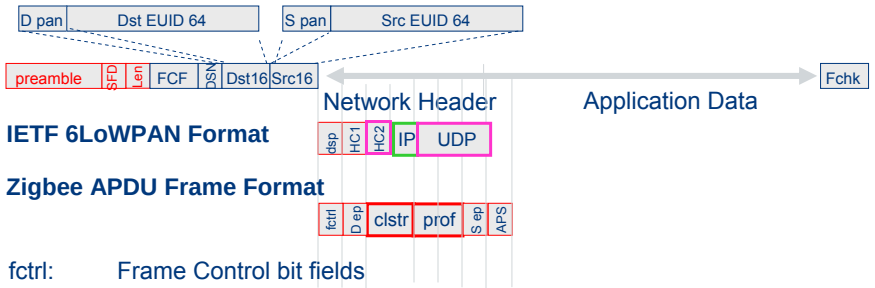
IEEE 802.15.4 Frame Format



17

6LoWPAN / Zigbee Comparison

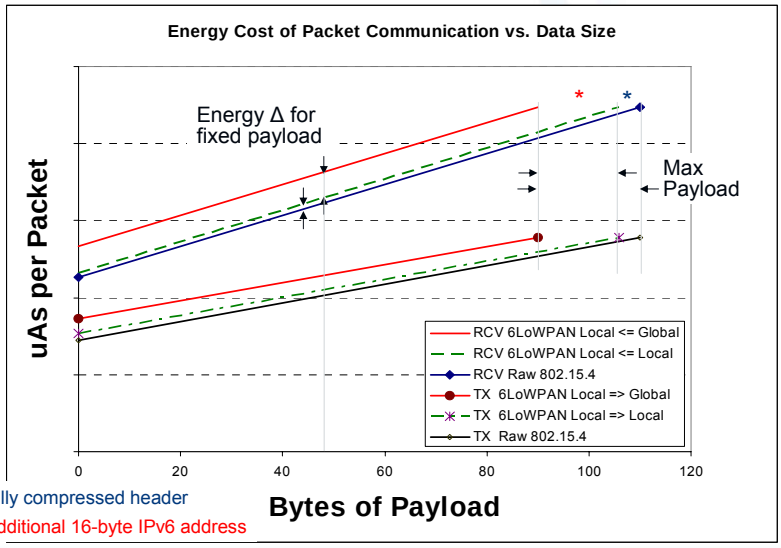
IEEE 802.15.4 Frame Format



fctrl: Frame Control bit fields
D ep: Destination Endpoint (like UDP port)
clstr: cluster identifier
prof: profile identifier
S ep: Source Endpoint
APS: APS counter (sequence to prevent duplicates)
*** Typical configuration. Larger and smaller alternative forms exist.

18

Low Impact of 6LoWPAN on Lifetime – Comparison to *Raw* 802.15.4 Frame

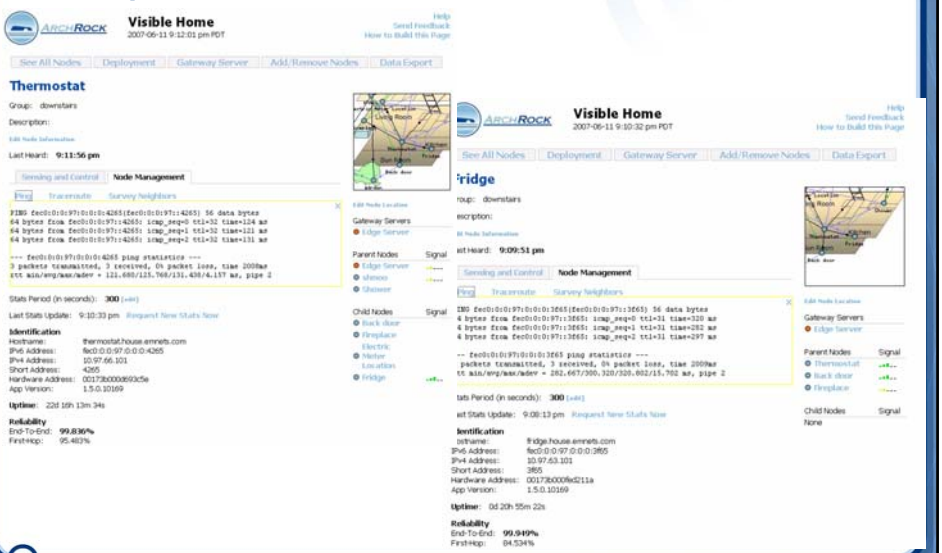


* fully compressed header

* additional 16-byte IPv6 address

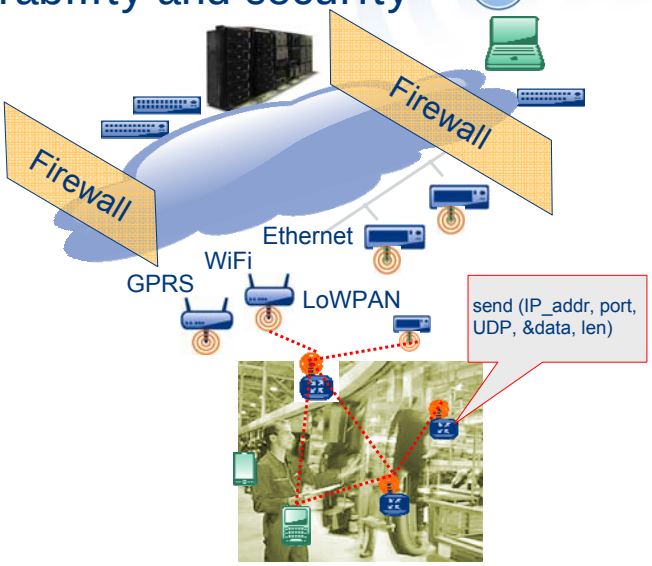
19

Examples



20

IP Interoperability and Security



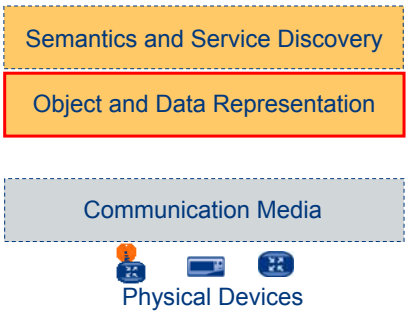
21

Many Advantages

- **Extensive Interoperability**
  - Other wireless embedded network (802.15.4) devices (goal of Zigbee, SP100.11a, …
  - Devices on any other IP network link (WiFi, Ethernet, GPRS, Serial lines, …)
- **Established security**
  - Authentication, access control, and firewall mechanisms.
  - Network design and policy determines access, not the technology
- **Established naming, addressing, translation, lookup, discovery**
- **Established proxy architectures for higher-level services**
  - NAT, Load Balancing, Caching, Mobility
- **Established application level data model and services**
  - Application profiles
- **Established network management tools**
- **Transport protocols**
  - End-to-end reliability in addition to link reliability
- **Most "industrial" standards support an IP option**

22

Making Sense out of Sensors



IP / 802.15.4

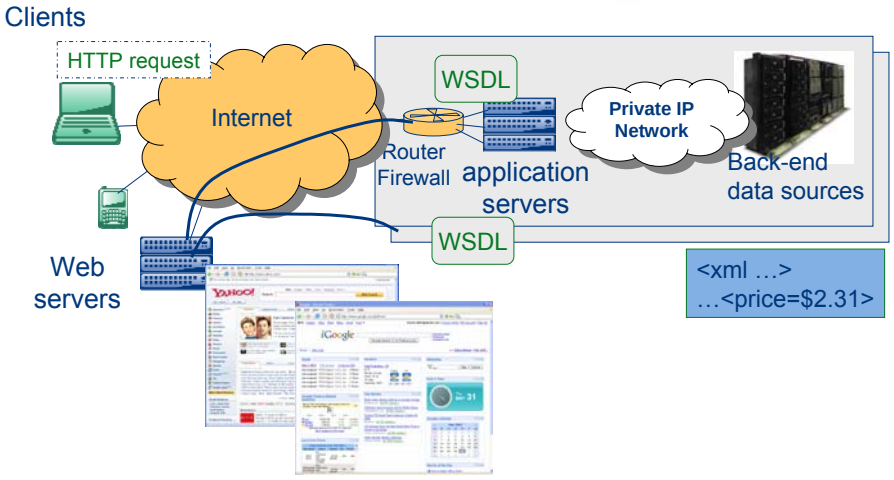
23

Lesson 2: Web, HTML, XML

- SIMPLE data formats that are easily understood
- Web Communication => transfer stream of characters
- Information Representation => nested tagged sections
  - `<tag> … </tag>`
  - Schema is machine readable and also in XML
- Behavior => GET or POST from/to a named endpoint
- Can vastly simplify the kinds of issues addressed by electronic data sheets, IEEE 1451, SP103, …
  - also much simpler than CORBA, DCOM, …

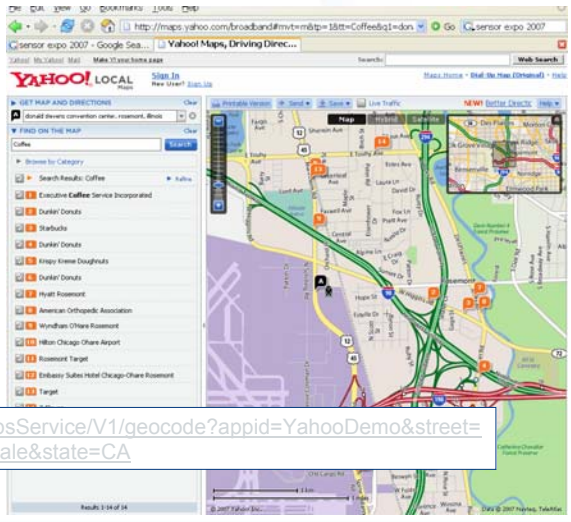
24

Internet Service Architecture



25

Example 1: geocoding



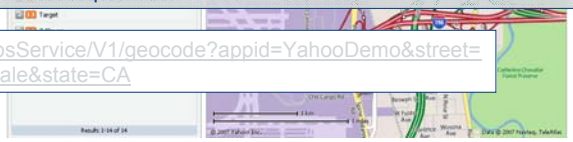
http://local.yahooapis.com/MapsService/V1/geocode?appid=YahooDemo&street=701+First+Street&city=Sunnyvale&state=CA

26

Example 1: geocoding

```
<?xml version="1.0" ?>
-<ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="urn:yahoo:maps"
   xsi:schemaLocation="urn:yahoo:maps
   http://api.local.yahoo.com/MapsService/V1/GeocodeResponse.xsd">
  <Result precision="address">
  <Latitude>37.416384</Latitude>
  <Longitude>-122.024853</Longitude>
  <Address>701 FIRST AVE</Address>
  <City>SUNNYVALE</City>
  <State>CA</State>
  <Zip>94089-1019</Zip>
  <Country>US</Country>
  </Result>
  </ResultSet>
<!-- ws05.search.scd.yahoo.com compressed/chunked Fri Jun 8 17:16:24 PDT 2007  -->
```

http://local.yahooapis.com/MapsService/V1/geocode?appid=YahooDemo&street=701+First+Street&city=Sunnyvale&state=CA



27

XML Schema for reply in XML too.

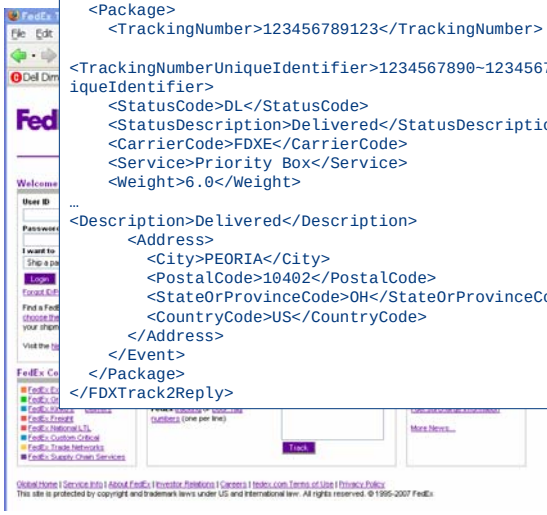
```
<xs:schema targetNamespace="urn:yahoo:maps" elementFormDefault="qualified">
  <xs:element name="ResultSet">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Result" type="ResultType" minOccurs="0" maxOccurs="50"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="ResultType">
    <xs:sequence>
      <xs:element name="Latitude" type="xs:decimal"/>
      <xs:element name="Longitude" type="xs:decimal"/>
      <xs:element name="Address" type="xs:string"/>
      <xs:element name="City" type="xs:string"/>
      <xs:element name="State" type="xs:string"/>
      <xs:element name="Zip" type="xs:string"/>
      <xs:element name="Country" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="precision" type="xs:string"/>
    <xs:attribute name="warning" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:schema>
```

Example 2: logistics

```
<?xml version="1.0" encoding="UTF-8" ?>
<FDXTrack2Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FDXTrack2Request.xsd">
  <RequestHeader>
    <CustomerTransactionIdentifier>String</CustomerTransactionIdentifier>
    <AccountNumber>123456789</AccountNumber>
    <MeterNumber>1234567</MeterNumber>
    <CarrierCode>FDXE</CarrierCode>
  </RequestHeader>
  <PackageIdentifier>
    <Value>987654321987</Value>
  </PackageIdentifier>
  <ShipDateRangeBegin>2006-01-01</ShipDateRangeBegin>
  <ShipDateRangeEnd>2006-01-23</ShipDateRangeEnd>
  <DetailScans>0</DetailScans>
</FDXTrack2Request>
```

29

Exampl

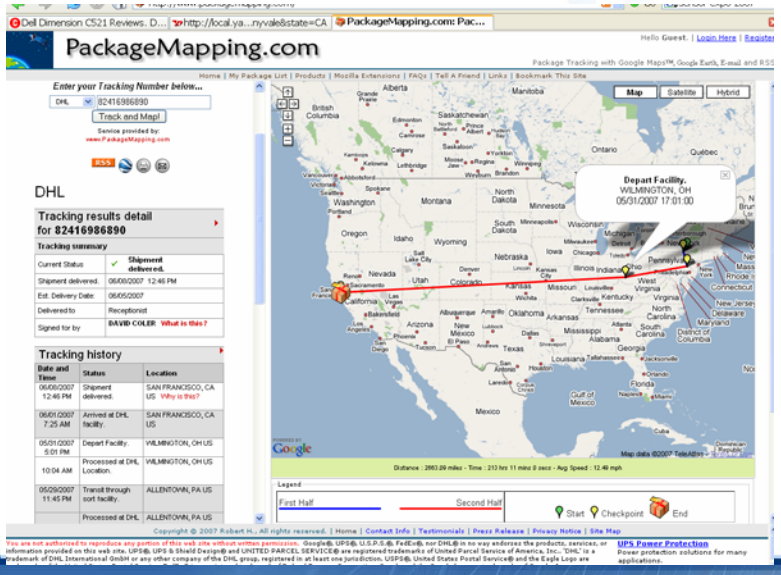


```
<?xml version="1.0" encoding="UTF-8"?>
<FDXTrack2Reply xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FDXTrack2Reply.xsd">
  <ReplyHeader>
    <CustomerTransactionIdentifier>String</CustomerTransactionIdentifier>
  </ReplyHeader>
  <Package>
    <TrackingNumber>123456789123</TrackingNumber>

<TrackingNumberUniqueIdentifier>1234567890~123456789123</TrackingNumberUn
iqueIdentifier>
    <StatusCode>DL</StatusCode>
    <StatusDescription>Delivered</StatusDescription>
    <CarrierCode>FDXE</CarrierCode>
    <Service>Priority Box</Service>
    <Weight>6.0</Weight>
…
<Description>Delivered</Description>
      <Address>
        <City>PEORIA</City>
        <PostalCode>10402</PostalCode>
        <StateOrProvinceCode>OH</StateOrProvinceCode>
        <CountryCode>US</CountryCode>
      </Address>
    </Event>
  </Package>
</FDXTrack2Reply>
```

30

Putting it together



31

Making Sense out of Sensors

Semantics and Service Discovery

Object and Data Representation

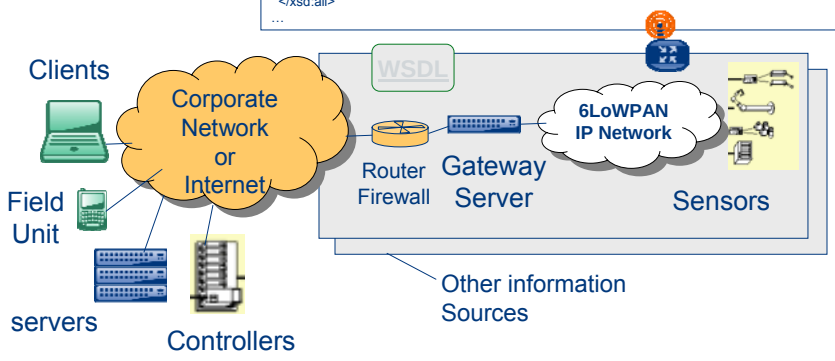
Communication Media

Physical Devices

32

Sensor Service

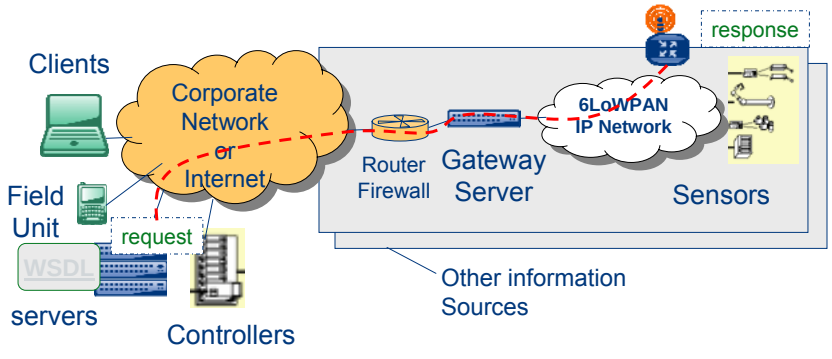
```
<?xml version="1.0" ?>
 <definitions name="GW" targetNamespace="urn:gw"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:gw="urn:gw"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <types>
 <xsd:schema xmlns="http://www.w3.org/2001/XMLSchema" targetNamespace="urn:gw">
 <xsd:complexType name="GW__attributesList_Result">
 <xsd:all>
 <xsd:complexType name="GW__eventsRead_Result">
 <xsd:all>
  <xsd:element name="offset" type="xsd:unsignedInt" />
  <xsd:element name="total" type="xsd:unsignedInt" />
  <xsd:element name="results" type="gw:GW__Event_Results" />
  </xsd:all>
...
```



33

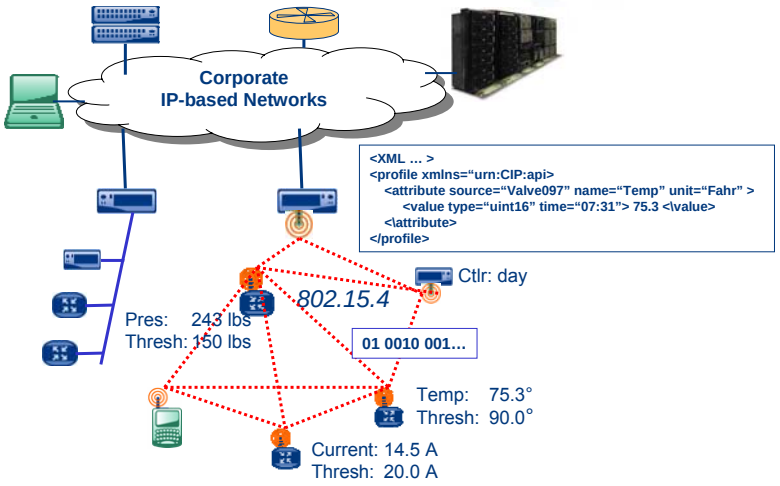
Sensor Service Architecture

```
<Results>
  <Result addr="00173b000fed211a" timestamp="1181622351.345968"
seqNo="27" name="TemperatureReadEvent">
    <Value typeName="nx_uint16_t">4240</Value>
  </Result>
</Results>
```



34

Example



35

Lesson 3: WSDL

- Machine readable description of all aspects of the services
  - Operations it performs
  - Data representation
- XML – just like any other document
- XML schema
- Programming tools do all the details

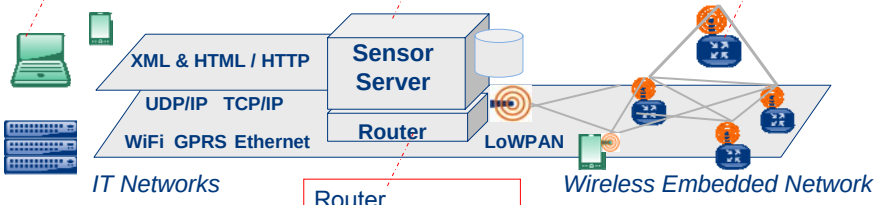
36

Sensor Web Services - Roles

Client
- Issues requests
- Consumes/Presents responses
- Receives alerts

Server
- Manages embedded network and devices
- Collects and processes readings and events
- Presents embedded services
- Services requests

Wireless Sensor Device
- Takes measurements / actions
- Appl'n-specific local processing
- Communicates over LoWPAN
- Routes (for others)
- Processes commands



IT Networks

Wireless Embedded Network

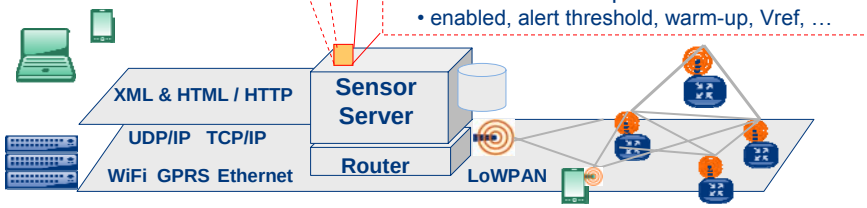
Router
- Maintains IP routes
- Forwards packets
- IPv6-v4 translation

37

Sensor Web Services

Wireless Sensor "node"
- *Named* and *Configured*
  - hostname, IP address
  - LoWPAN EUID-64, Short-16
- Provides named (get/set) node *attributes*
  - sample period, heartbeat, position, …
- Provides set of *commands*
  - send readings, activate, …
- Provides set of named *Sensors*
  - humidity, vibration, …
  - sampled periodically, on request, on interrupt
- Provides attributes per sensor
  - enabled, alert threshold, warm-up, Vref, …

Sensor Node Capabilities
- Presented as Web Services
- Described by a WSDL file
  - Web Services Description Language
- in XML



IT Networks

Wireless Embedded Network

38

Service Formation

ARCHROCK

Client

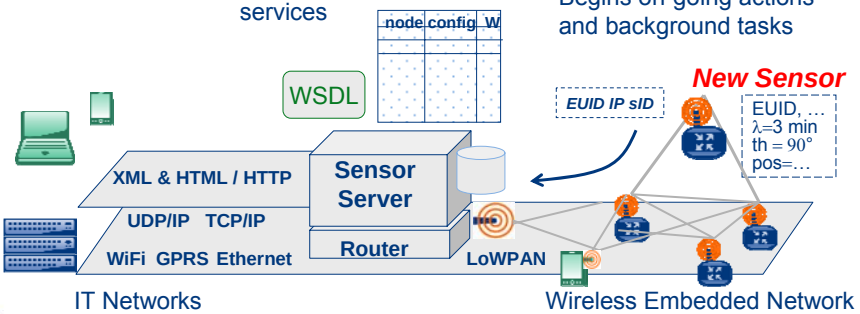
- Obtains service desc
- Utilizes services

Server

- Records presence & health of sensor nodes
- Obtains WSDL for node Schema ID
- Maintains configuration
- Provides on-going services

Sensor Node

- Powered on
- Reads local configuration
- Joins LoWPAN
- Announces presence
- Synchronizes configuration with server
- Begins on-going actions and background tasks



39

On-going Operation (data plane)

ARCHROCK

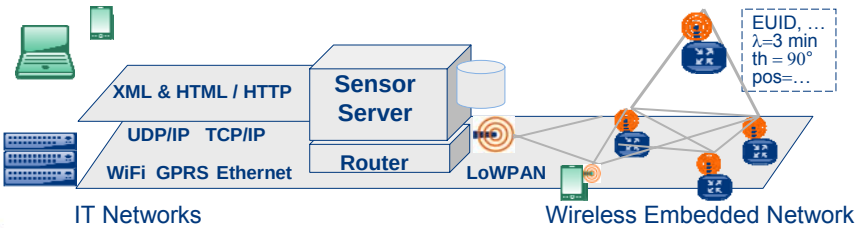
Client

- Request WSDLs, node info, mgmt, historical reading
- Update configuration, request current readings, actuate, diagnosis, mgmt
- Receive alerts

Server

Sensor Node

- Sample sensors
- Process readings
  - conversions, thresholds, smoothing, …
- Send readings to server
- Receive configuration updates
- Receive actuation requests
- Send alerts



40

On-going Operation (control plane)

Client

- Mgmt requests

Server

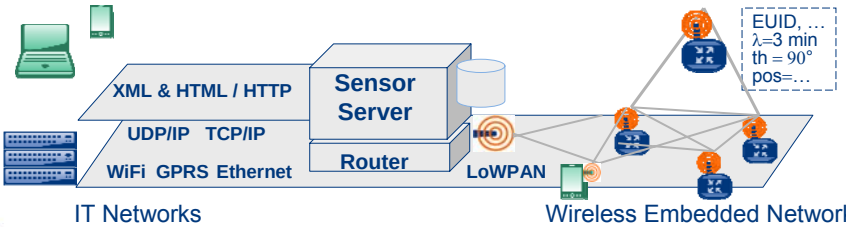
- Record and monitor embedded network health
- Process mgmt requests

Router

- Route traffic
- Maintain LoWPAN routing

Sensor Node

- Receive and process mgmt requests
- Send health mgmt info
- Update node software
- Route Traffic
- Maintain LoWPAN routing



41

Lesson 4 - Compression

- Describe sensor networks in terms of generic XML
- Use similar automated tools to compress into compact binary representations
  - Like the formats we spend months hammering out in the standards meeting

42

Embedded Web Services

Web Services

```
< get temp …
  set sample_rate
  set alarm … >
```

<request service>

www.weather.com

Service Description

XML information

```
<value>
  source=library
  time=12:53
  temp=26.7
<\value>
```

Wireless Packets

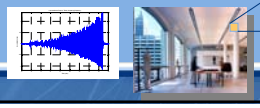
802.15.4

Sampled Value

`int temp;` 010010001

11 010110111

Physical Signal



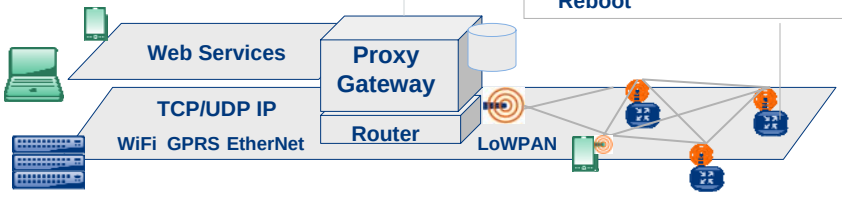
43

Example: Primer Pack / IP

Rich Web View per Node
Web Services / WSDL
SNMP, Ganglia, Email
Adapters
Data Warehouse

Sensor & Mgmt Services
HTTPm
Systat, Netstat, Echo
Ping, Traceroute, DHCP
Reboot

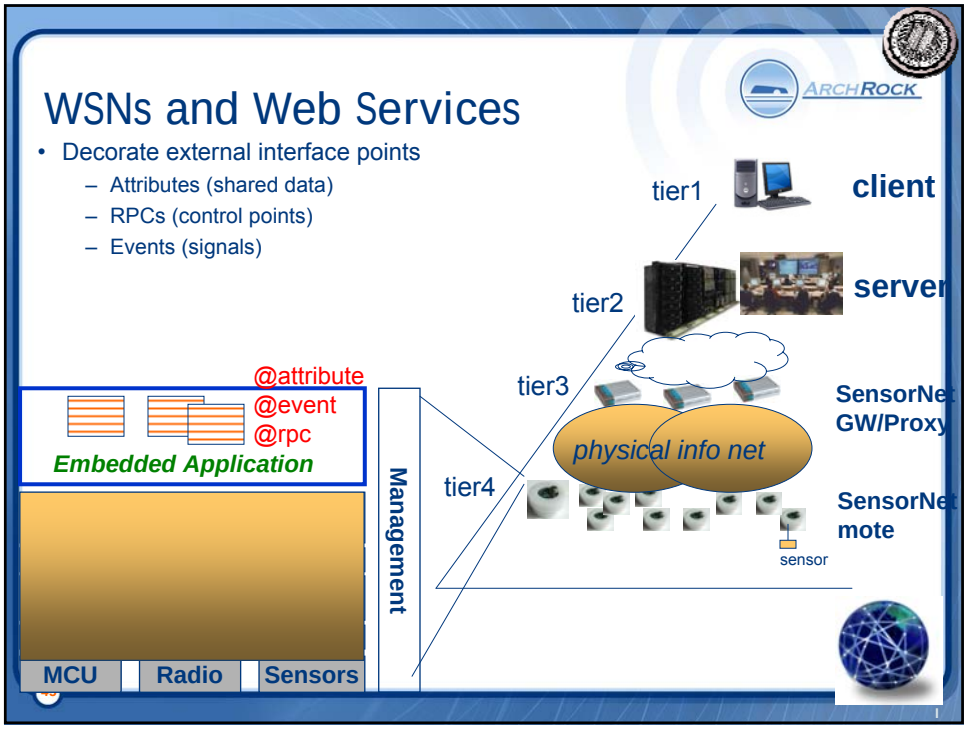
Browser,
Enterprise,
Controller

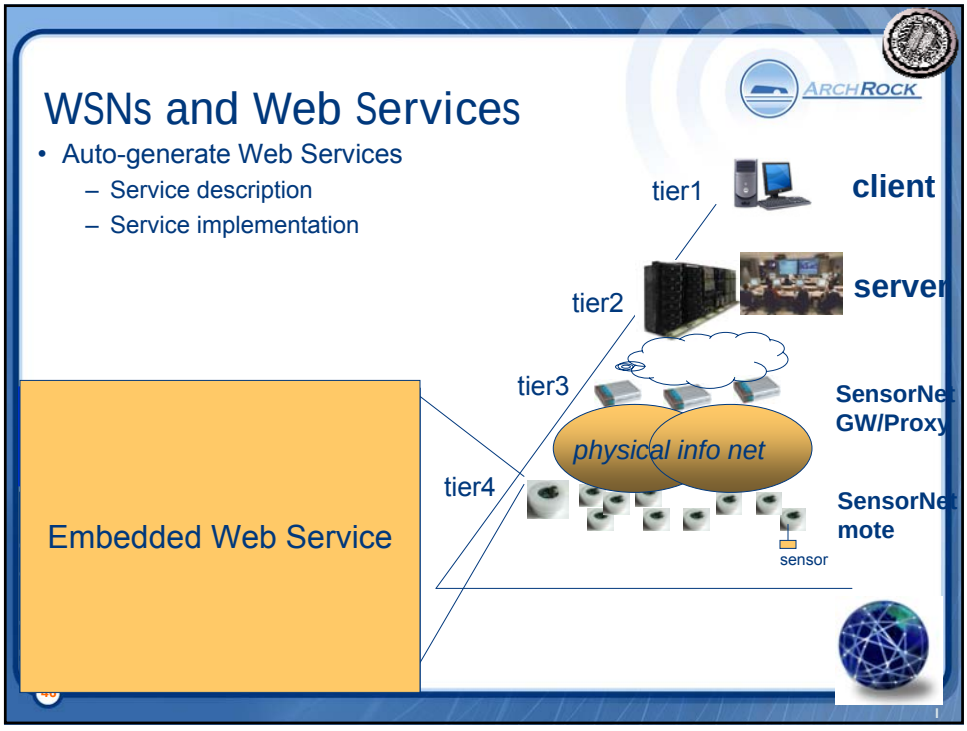


nc, telnet, ping, traceroute…

High Reliability
Triply Redundant
Ultra-low power
Highly Responsive
AES128 Secured
IP-based Mesh Network

44

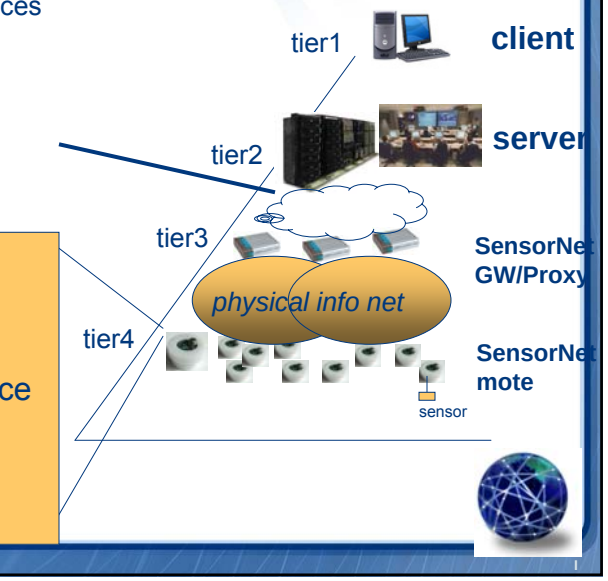

WSNs and Web Services
• Decorate external interface points
– Attributes (shared data)
– RPCs (control points)
– Events (signals)
@attribute
@event
@rpc
Embedded Application
MCU
Radio
Sensors
Management
tier1
client
tier2
server
tier3
SensorNet
GW/Proxy
physical info net
tier4
SensorNet
mote
sensor


WSNs and Web Services
• Auto-generate Web Services
– Service description
– Service implementation
Embedded Web Service
tier1
client
tier2
server
tier3
SensorNet
GW/Proxy
physical info net
tier4
SensorNet
mote
sensor

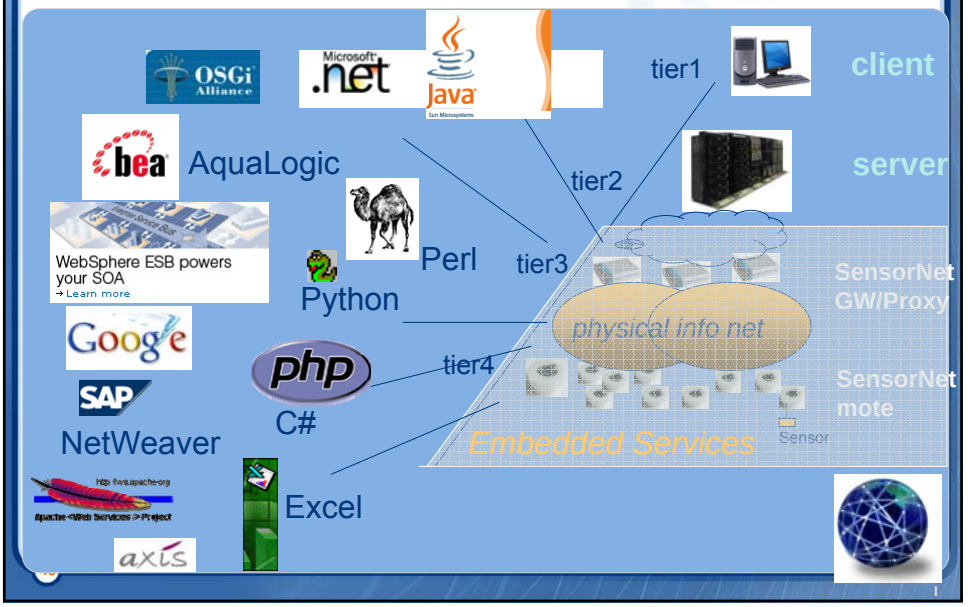
WSNs and Web Services

- Auto-generate Web Services
  - Service description
  - Service implementation

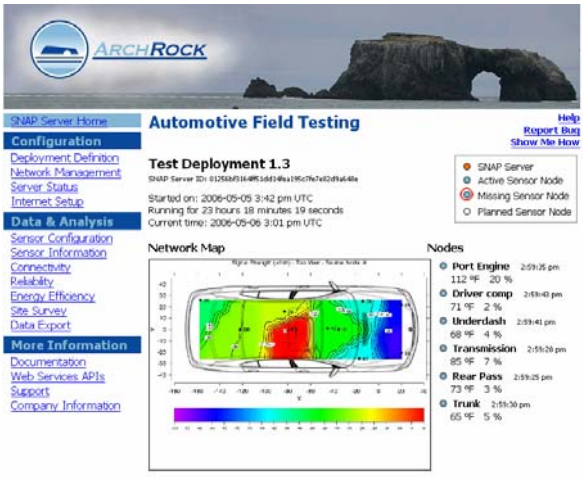
Embedded Web Service



A new world of WSN tools



Sensor Web ⇔ IT Enterprise



49

Summary: Lessons from the Web

To integrate diverse information sources:

1. IP: separate communication from physical links
   - 6LoWPAN enables efficient low-power, reliable mesh with IP
2. HTML, HTTP, XML: simple self-describing text
   - electronic data sheets that programs understand
3. WSDL: descriptions of services in XML and XML schema
   - Describe what you do so programs can understand it
   - Simple Executable specifications!
4. Compress the common case: compact instrumentation and control
   - Simple subset of XML. Automatic translation.

50

Backup

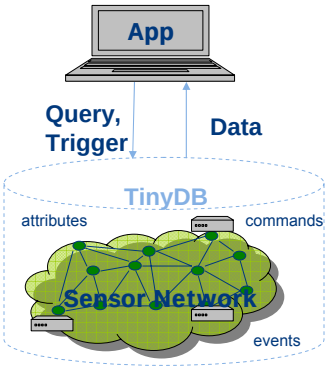
- Earlier view – Sensor Net as Distributed Data Base
- TinyDB vs Web Services

51

SensorNet as a Database

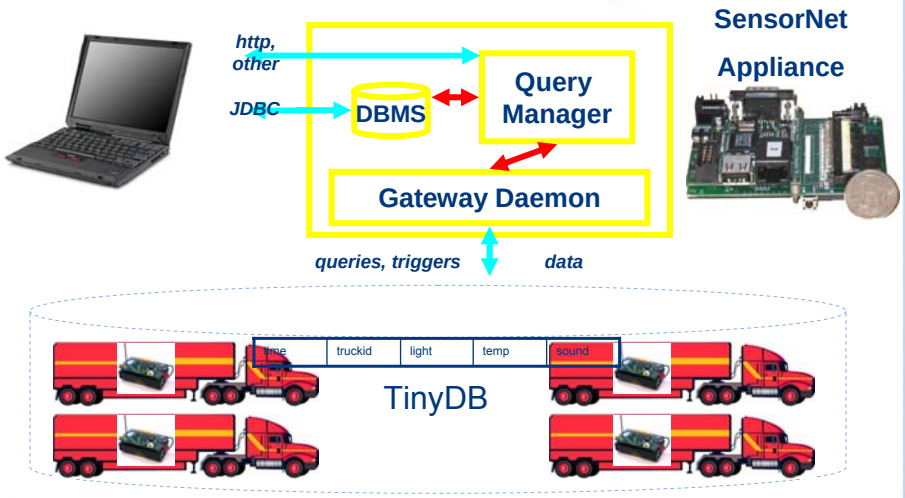
- High level abstraction:
  - Declarative programming
  - Extensible framework
    - User-defined attributes, commands and events
- Under the hood:
  - Intelligent query processing: query optimization, power efficient execution
  - Fault mitigation: automatically introduce redundancy, avoid problem areas

```
SELECT MAX(mag)
FROM sensors
WHERE mag > thresh
SAMPLE PERIOD 64ms
```



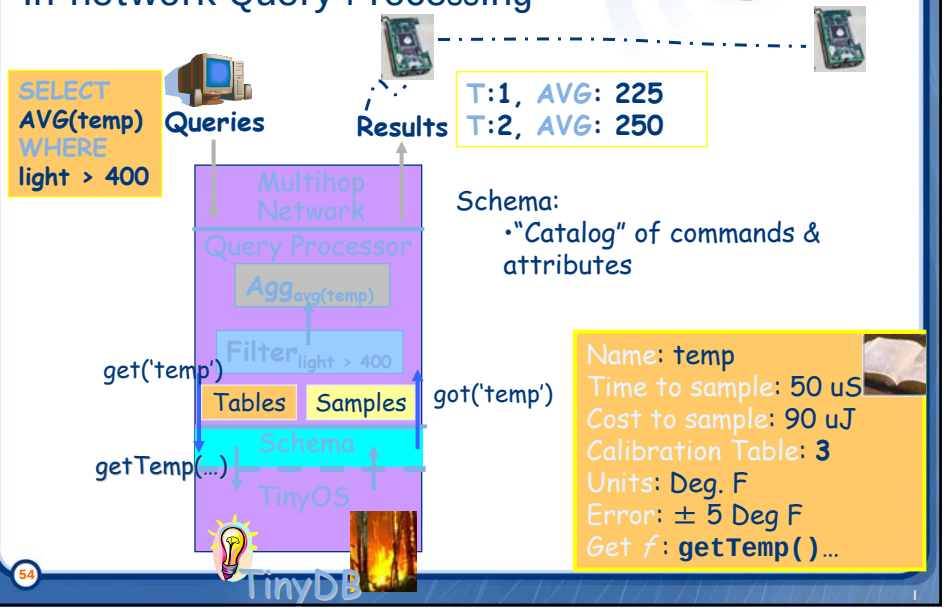
52

Container Monitoring Example



53

In-network Query Processing



54

Biz Logic as Queries/Triggers

- Threshold monitoring for light and temperature

```
SELECT nodeid, light, temp
FROM sensors
WHERE light > 400 or temp > 32
SAMPLE PERIOD 4s
```

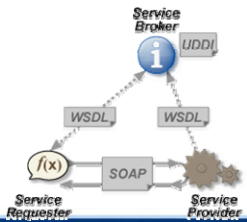
- Light exposure alert

```
SELECT nodeid, light
FROM sensors
WHERE light > 400
SAMPLE PERIOD 4s
TRIGGER ACTION sounder(500)
```

55

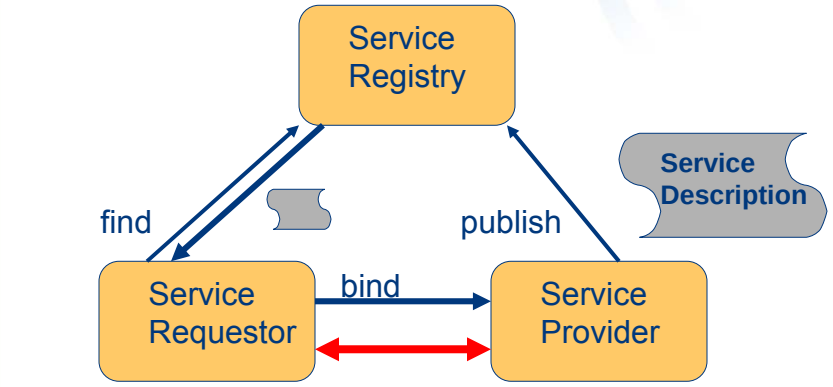
Web Services

- The ultimate interoperability
- unify within and between enterprises
  - Service Oriented Architecture
  - Widely adopted standards
    - Backed by OASIS and W3C
    - Open-source reference implementations, e.g., Apache
    - SOAP + WSDL (+ UDDI)
    - Or XML-RPC over HTTP (REST)



56

Service Oriented Architecture



- Service Description => interface & implementation
  - Operations supported, input/output objects
  - Bindings to network and data encoding schemes
  - Network address where service can be invoked
- Enough that client can generate code to access the service well

57