Standards for the Future

of Java Embedded

Werner Keil

JavaOne Embedded

1st October 2012

CATMedia

Overview

- Introduction
- Sensors
  - Historic IT Errors and Bugs
  - UOMo, Unit-API, UCUM
  - Sensor Web, SensorML
- M2M
- NFC
  - eNFC, Use Cases
- Security
  - TPM, TEE, Secure Element
  - JavaCard
- Q&A

Who am I?

Werner Keil

- Consultant – Coach
- Creative Cosmopolitan
- Open Source Evangelist
- Software Architect
- Java Godfather
- JCP Executive Committee Member
- Eclipse UOMo Project Lead
- …

Twitter @wernerkeil

CATMedia

Java Godfather?

Type-Safety

- Java does not have strongly typed primitive types (like e.g. Ada or Smalltalk).
  - This is likely to change around Java 9 or 10 (based on Oracle Road Map and statements)
- For performance reasons most developer prefer primitive types over objects in their interface.
- Primitives type arguments can more easily lead to name clashes (methods with the same signature)

What do these disasters have in common?

- Patriot Missile

  The cause was an inaccurate calculation of the time since boot due to a computer arithmetic error.

- Ariane 5 Explosion

  Floating point number which a value was converted from had a value greater than what would be represented by a 16 bit signed integer.

- Gimli Glider (near disaster)

  Fuel loading was miscalculated through misunderstanding of the recently adopted Metric System, replacing the Imperial System

What do these disasters have in common?

CNN.com

MAIN PAGE
WORLD
U.S.
LOCAL
POLITICS
WEATHER
BUSINESS
SPORTS
TECHNOLOGY
SPACE
HEALTH
ENTERTAINMENT
BOOKS
TRAVEL
FOOD
ARTS & STYLE
NATURE
IN-DEPTH
ANALYSIS
myCNN

Headline News brief
news quiz
daily almanac

MULTIMEDIA:
video
video archive
audio
multimedia showcase
more services

E-MAIL:
Subscribe to one of our news e-mail lists.
Enter your address:

sci-tech> space> story page

exploringmars in-depth specials

Metric mishap caused loss of NASA orbiter

September 30, 1999
Web posted at: 4:21 p.m. EDT (2021 GMT)

In this story:

Metric system used by NASA for many years

Error points to nation's conversion lag

RELATED STORIES, SITES

NASA's Climate Orbiter was lost September 23, 1999

By Robin Lloyd
CNN Interactive Senior Writer

(CNN) -- NASA lost a $125 million Mars orbiter because a Lockheed Martin engineering team used English units of measurement while the agency's team used the more conventional metric system for a key spacecraft operation, according to a review finding released Thursday.

The units mismatch prevented navigation information from transferring between the Mars Climate Orbiter spacecraft team in at Lockheed Martin in Denver and the flight team at NASA's Jet Propulsion Laboratory in Pasadena, California.

CATMedia

What do these disasters have in common?

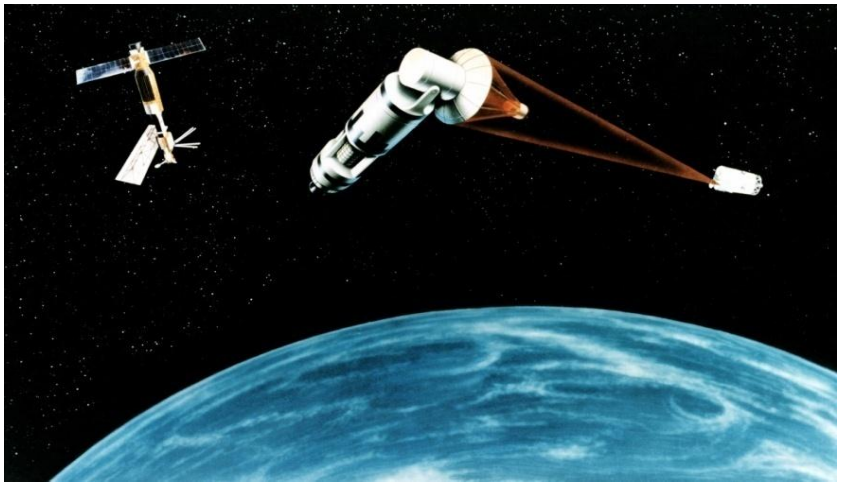
- Mars Orbiter

  Preliminary findings indicate that one team used US/English units (e.g. inches, feet and pounds) while the other used metric units for a key spacecraft operation.

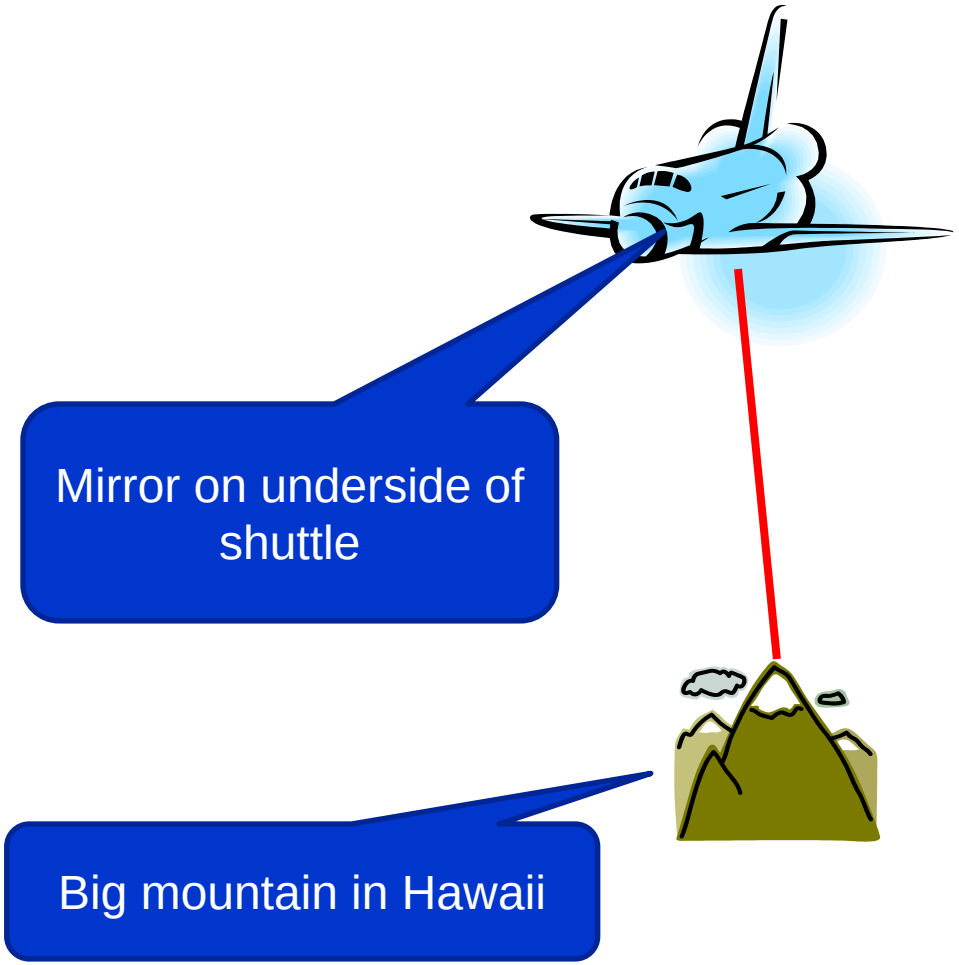
  - NASA lost a $125 million Mars orbiter because a Lockheed Martin engineering team used English units of measurement while the agency's team used the more conventional metric system for a key spacecraft operation
    - **A credible source disclosed, there was a manual step with an outsourced person to convert these calculations between the different teams, and NASA budget cuts caused them to fire him and have the wrong, unpatched data transmitted!!!**
  - This also underlines the added risk when 3rd party contractors are involved or projects are developed Offshore

NASA “Star Wars” Initiative, 1983

23rd March 1983. Ronald Reagan announces SDI (or “Star Wars”): ground-based and space-based systems to protect the US from attack by strategic nuclear ballistic missiles.

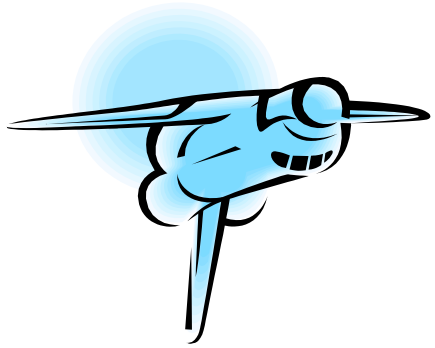


1985

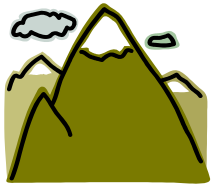


SDI Experiment: The Plan

1985



SDI Experiment: What really happened



CATMedia

1985: What happened?

ACM SIGSOFT SOFTWARE ENGINEERING NOTES vol 10 no 3 Jul 1985 page 10

Attention All Units, Especially Miles and Feet!

Much to the surprise of Mission Control, the space shuttle Discovery flew upside-down over Maui on 19 June 1985 during an attempted test of a Star-Wars-type laser-beam missile defense experiment. The astronauts reported seeing the bright-blue low-power laser beam emanating from the top of Mona Kea, but the experiment failed because the shuttle's reflecting mirror was oriented upward! A statement issued by NASA said that the shuttle was to be repositioned so that the mirror was pointing (downward) at a spot *10,023 feet* above sea level on Mona Kea; that number was supplied to the crew in units of feet, and was correctly fed into the onboard guidance system -- which unfortunately was expecting units in nautical miles, not feet. Thus the mirror wound up being pointed (upward) to a spot *10,023 nautical miles* above sea level. The San Francisco Chronicle article noted that "the laser experiment was designed to see if a low-energy laser could be used to track a high-speed target about 200 miles above the earth. By its failure yesterday, NASA unwittingly proved what the Air Force already knew -- that the laser would work only on a 'cooperative target' -- and is not likely to be useful as a tracking device for enemy missiles." [This statement appeared in the S.F. Chronicle on 20 June, excerpted from the L.A. Times; the NY Times article on that date provided some controversy on the interpretation of the significance of the problem.] The experiment was then repeated successfully on 21 June (using nautical miles). The important point is not whether this experiment proves or disproves the viability of Star Wars, but rather that here is just one more example of an unanticipated problem in a human-computer interface that had not been detected prior to its first attempted actual use.

CATMedia

What do these disasters have in common?

- Patriot Missile
  The cause was an inaccurate calculation of the time since boot due to a computer arithmetic error.
- Ariane 5 Explosion
  The floating point number which a value was converted from had a value greater than what would be represented by a 16 bit signed integer.

Unit Tests wouldn‘t find these…

Despite their name 

- All previous example illustrate three categories of errors difficult to find through Unit Testing:
  - Interface Errors (e.g. millisecond/second, radian/degree, meters/feet).
  - Arithmetic Errors (e.g. overflow).
  - Conversion Errors.

Causes of Conversion Errors

- Ambiguity on the unit
  - Gallon Dry / Gallon Liquid
  - Gallon US / Gallon UK
  - Day Sidereal / Day Calendar
  - Degree Celsius / Degree Fahrenheit
    - Did you know that Gabriel Fahrenheit was born in Gdansk (Danzig) in northern Poland?
  - ...
- Wrong conversion factors:

```
static final double PIXEL_TO_INCH = 1 / 72;
double pixels = inches * PIXEL_TO_INCH
```

What else do they have in common?

ALL OF THEM HAPPENED IN MOBILE, REAL TIME OR EMBEDDED SYSTEMS!

Measurement Package

- Namespace: org.osgi.util.measurement
- SI only Unit API “in the closet”
  - Unit
    Essentially an SI singleton holding relevant unit constants, too.
  - Measurement
    Represents a value with an error, a unit and a time-stamp.
  - State
    Groups a state name, value and timestamp.
- Some usage, especially in Automotive

► no further development by OSGi

JSR-256

Mobile Sensor API

- Namespace: javax.microediton.sensor*
- Focusing on Sensors, but it got a minimalistic Unit API "in the closet"
  - Unit
    Essentially an SI singleton holding relevant unit constants, too.
  - ChannelInfo
    Holding name, accuracy, data type,measurement ranges, scale and unit
  - MeasurementRange
    Range of possible values from minimum to maximum
- ► Dead on Arrival (no actual handsets or vendors using it today)

JSR-275

Base Classes and Packages

- Namespace: javax.measure.*
- Only one interface and one abstract class
  - Measurable<Q extends Quantity> (interface)
  - Measure<V, Q extends Quantity> (abstract class)
- Three sub-packages
  - unit (holds the SI and NonSI units)
  - quantity (holds dimensions mass, length)
  - converter (holds unit converters)

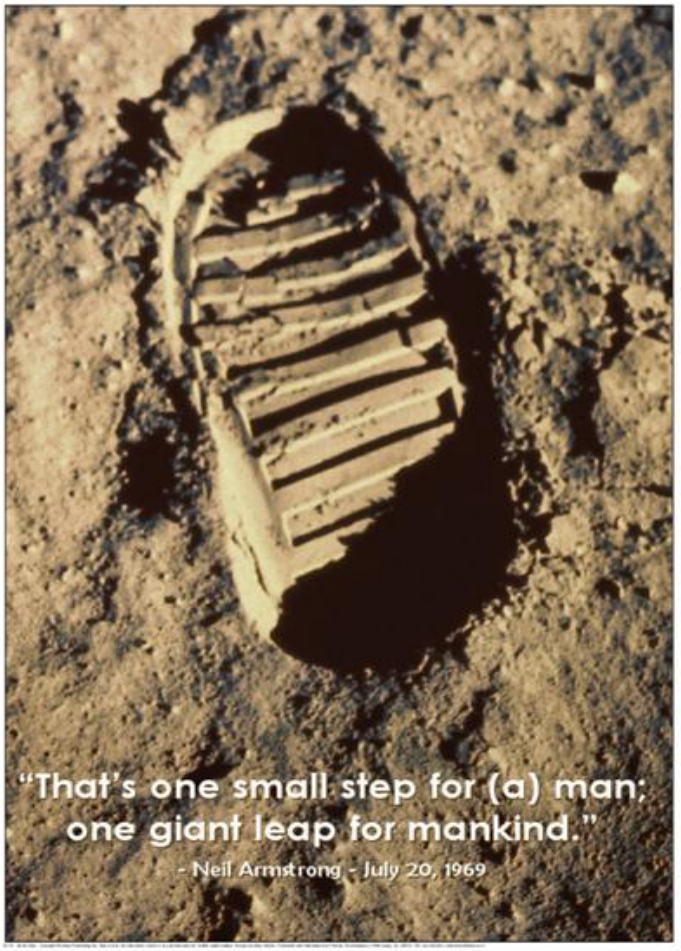
The King is Dead…

Units of Measurement API

- Namespace: org.unitsofmeasurement.*
- Only interfaces (and exception classes)
  - public interface Quantity<Q extends Quantity<Q>>
  - public interface Unit<Q extends Quantity<Q>>
- Three sub-packages
  - quantity (holds dimensions mass, length)
  - unit(holds units)
  - service (OSGi services)

Eclipse UOMo

One Small Step…



CATMedia

Eclipse UOMo

One Unit Framework to Measure them All

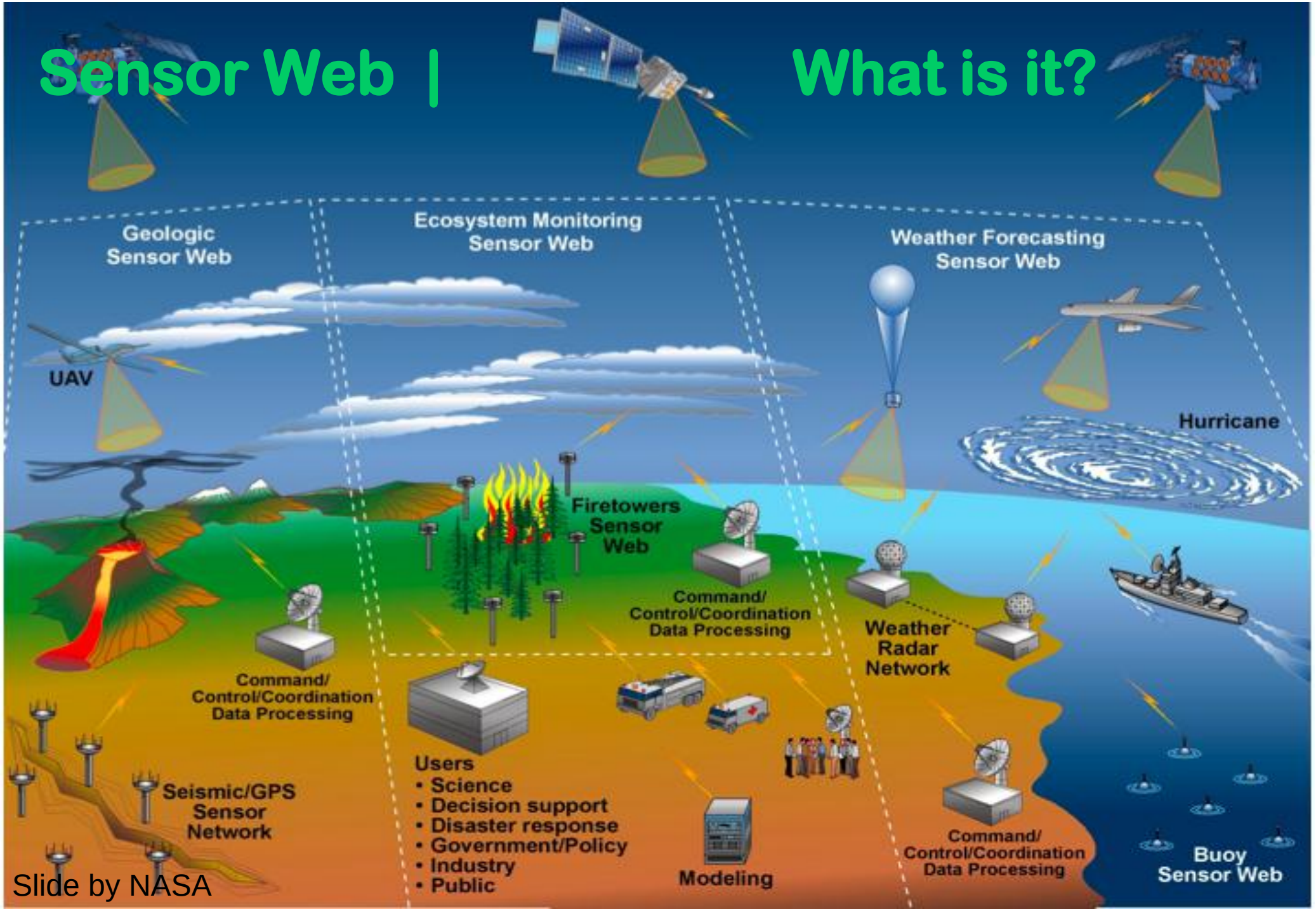
- Namespace: org.eclipse.uomo.*
- Two main areas
  - Static Type Safe Units of Measure Support
    - Based on Units of Measurement API
    - On top of ICU4J, the Globalization standard at Eclipse and others (Android, GWT, Google Financial, etc.)
  - Prime UCUM Implementation
    - Successor to Eclipse OHF UCUM Bundle

UOMo UCUM

Unified Code for Units of Measure

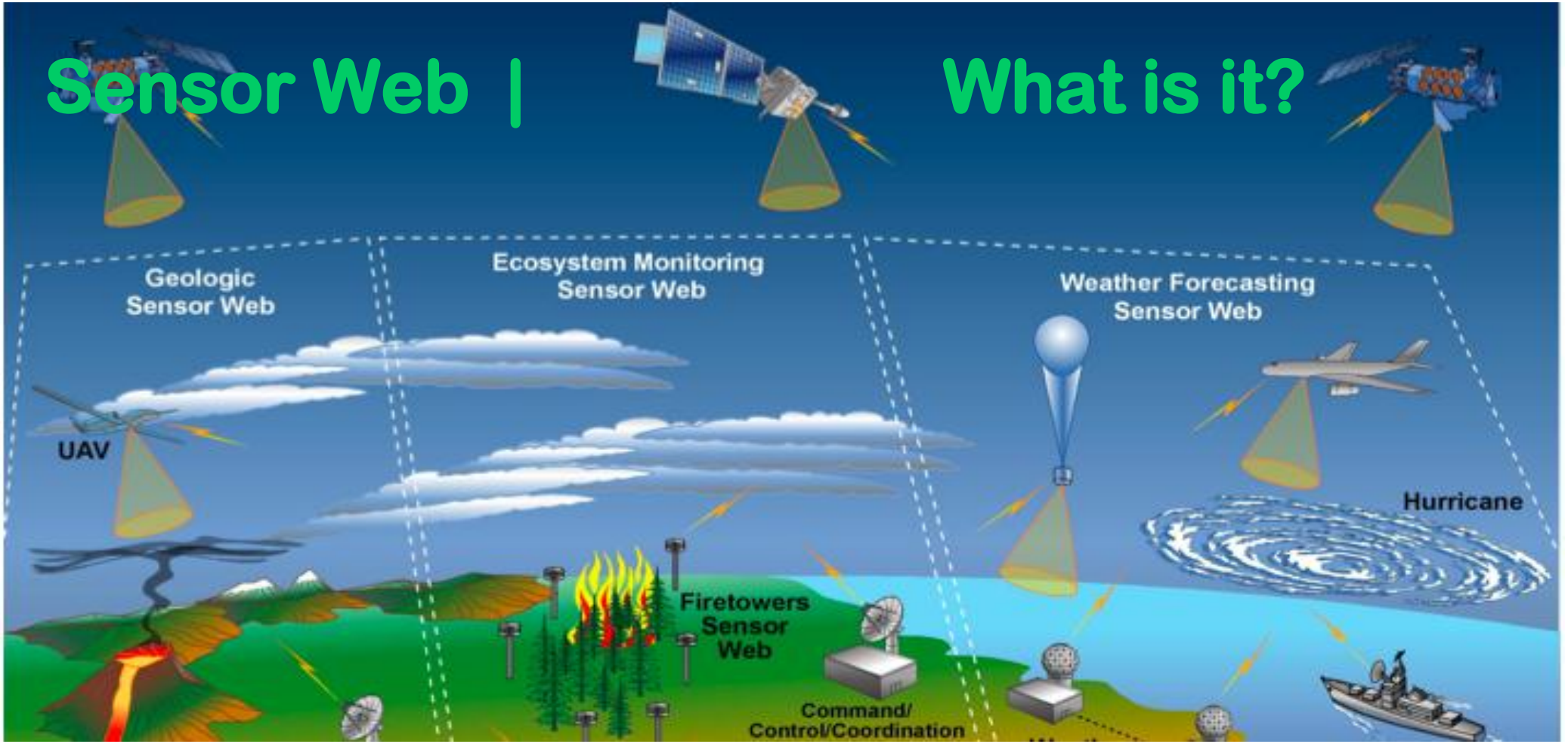
The Unified Code for Units of Measure is inspired by and heavily based on

- ISO 2955-1983
- ANSI X3.50-1986
- HL7's extensions called ISO+



Slide by NASA

Sensor Web | What is it?



"A coordinated observation infrastructure composed of a distributed collection of resources that can collectively behave as a single, autonomous, task-able, dynamically adaptive and reconfigurable observing system that provides raw and processed data, along with associated meta-data, via a set of standards-based service-oriented interfaces." (Glenn, 2007)

Sensor Web | OpenGIS Standards

- SW Enablement working group at OGC have developed a number of standards governing different aspects of Sensor Web

OGC O&M	Observations & Measurements	Approved
SensorML	Sensor Model Language	Approved
TransducerML	Transducer Model Language	Approved
OGC SOS	Sensor Observations Service	Approved
OGC SPS	Sensor Planning Service	Approved
OGC SAS	Sensor Alert Service	In progress
OGC WNS	Web Notification Services	In progress

Sensor Web | What is the OGC?

- Not-for-profit
- International industry consortium
- Founded 1994, currently 340+ members
- **Open Standards development by consensus process**

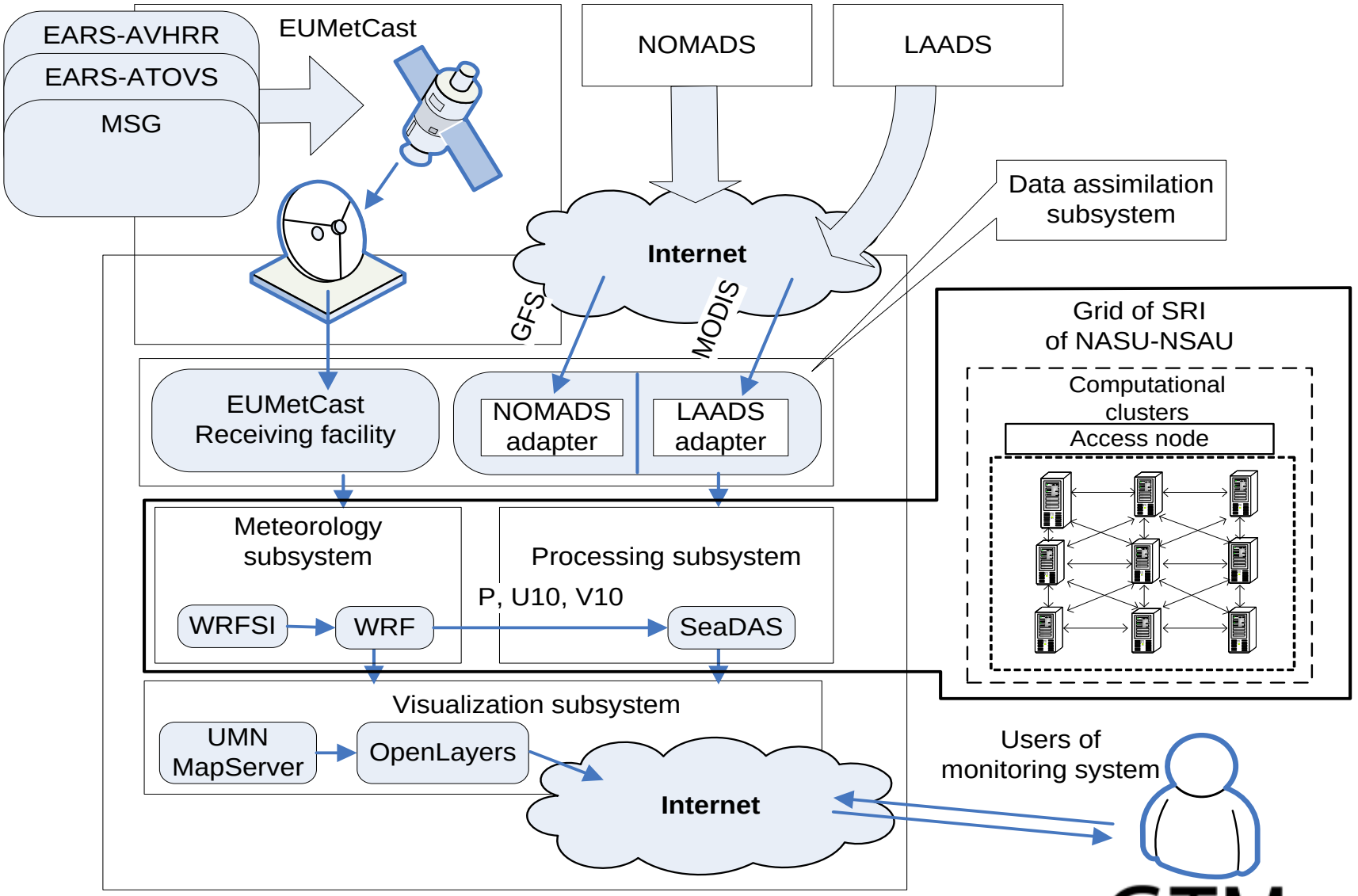
OGC Mission

To lead in the development, promotion and harmonization of open spatial standards ...

Sensor Web | Mozambique floods

- The task under study is floods in different parts of the world
- Particular test case was flooding of Mozambique

Sensor Web | Weather Prediction data



SensorML

- Sensor modeling language is the cornerstone of all SW services
- It provides comprehensive description of sensor parameters and capabilities
- It can be used for describing different kind of sensors:
  - Stationary or dynamic
  - Remote or in-situ
  - Physical measurements or simulations

SensorML | Example

```
..............
<inputs>
 <InputList>
  <input name="ambiantTemperature">
   <swe:Quantity definition=
    "urn:ogc:def:phenomenon:temperature"/>
  </input>
  <input name="atmosphericPressure">
   <swe:Quantity definition=
    "urn:ogc:def:phenomenon:pressure"/>
  </input>
  <input name="windSpeed">
   <swe:Quantity definition=
    "urn:ogc:def:phenomenon:windSpeed"/>
  </input>
</InputList>
</inputs>
..............
```

```
.............
<outputs>
 <OutputList>
  <output name="weatherMeasurements">
   <swe:DataGroup>
    <swe:component name="time">
     <swe:Time
      definition="urn:ogc:def:phenomenon:time"
      uom="urn:ogc:def:unit:iso8601"/>
    </swe:component>
    <swe:component name="temperature">
     <swe:Quantity
definition="urn:ogc:def:phenomenon:temperature
      uom="urn:ogc:def:unit:celsius"/>
    </swe:component>
    <swe:component name="barometricPressure">
     <swe:Quantity
   definition="urn:ogc:def:phenomenon:pressure"
      uom="urn:ogc:def:unit:bar" scale="1e-3"/>
    </swe:component>
    <swe:component name="windSpeed">
     <swe:Quantity
  definition="urn:ogc:def:phenomenon:windSpeed"
      uom="urn:ogc:def:unit:meterPerSecond"/>
    </swe:component>
   </swe:DataGroup>
  </output>
 </OutputList>
</outputs>
.............
```

Sensor Examples

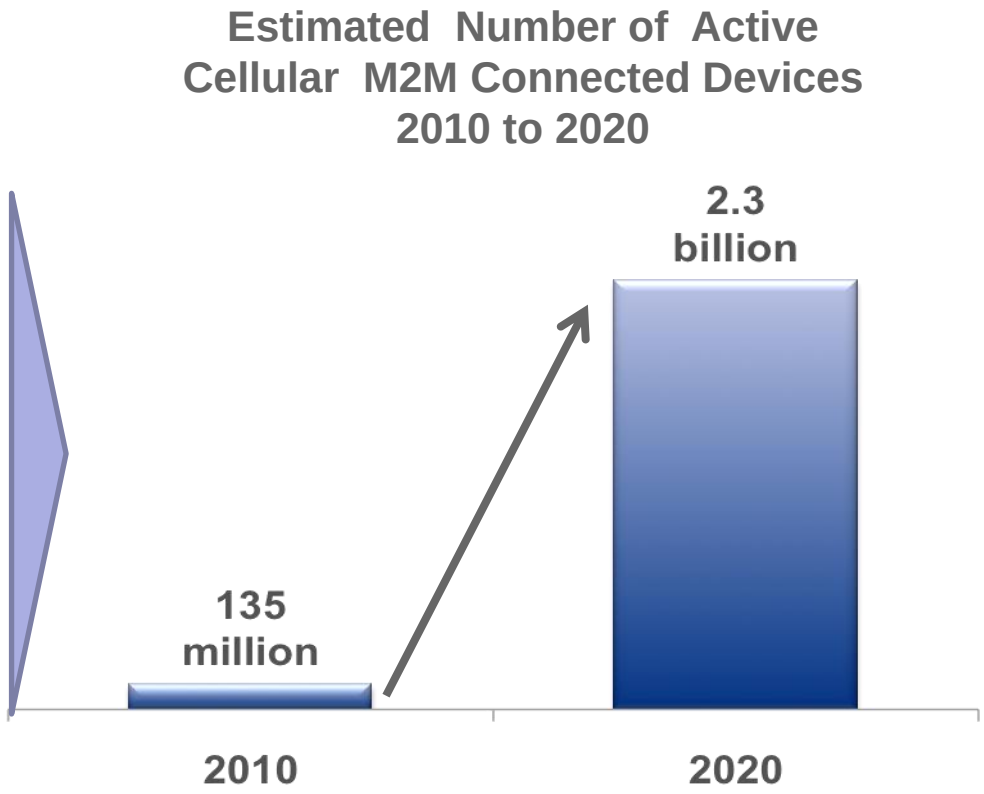
DEMO

CATMedia

M2M | Outlook

Key Trends

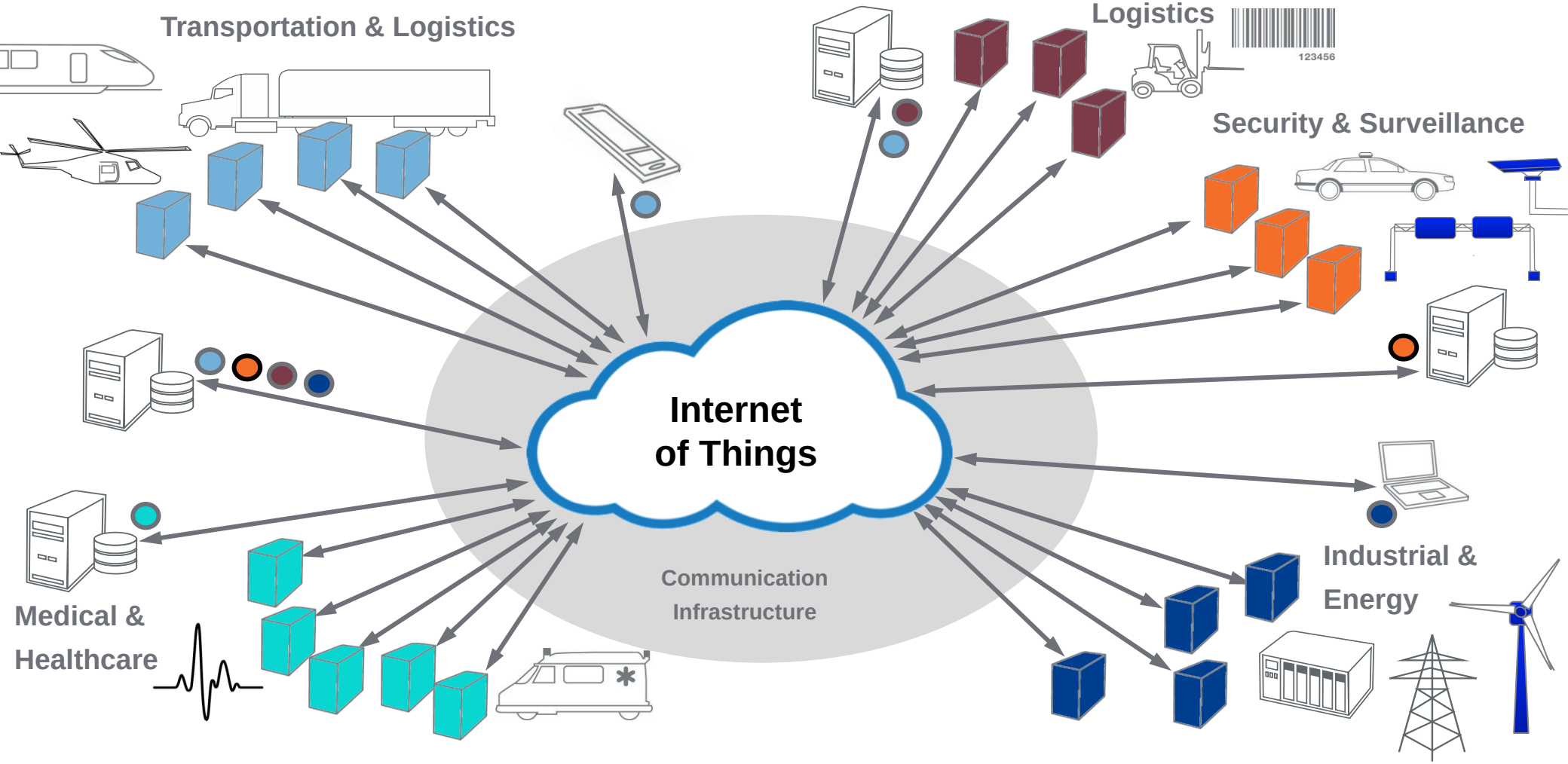
1. New connected devices, applications and services
2. Lower system costs
3. Simplified development
4. Network operator focus and investment



Source: Machina Research, July 2011

M2M | Integrated Processes

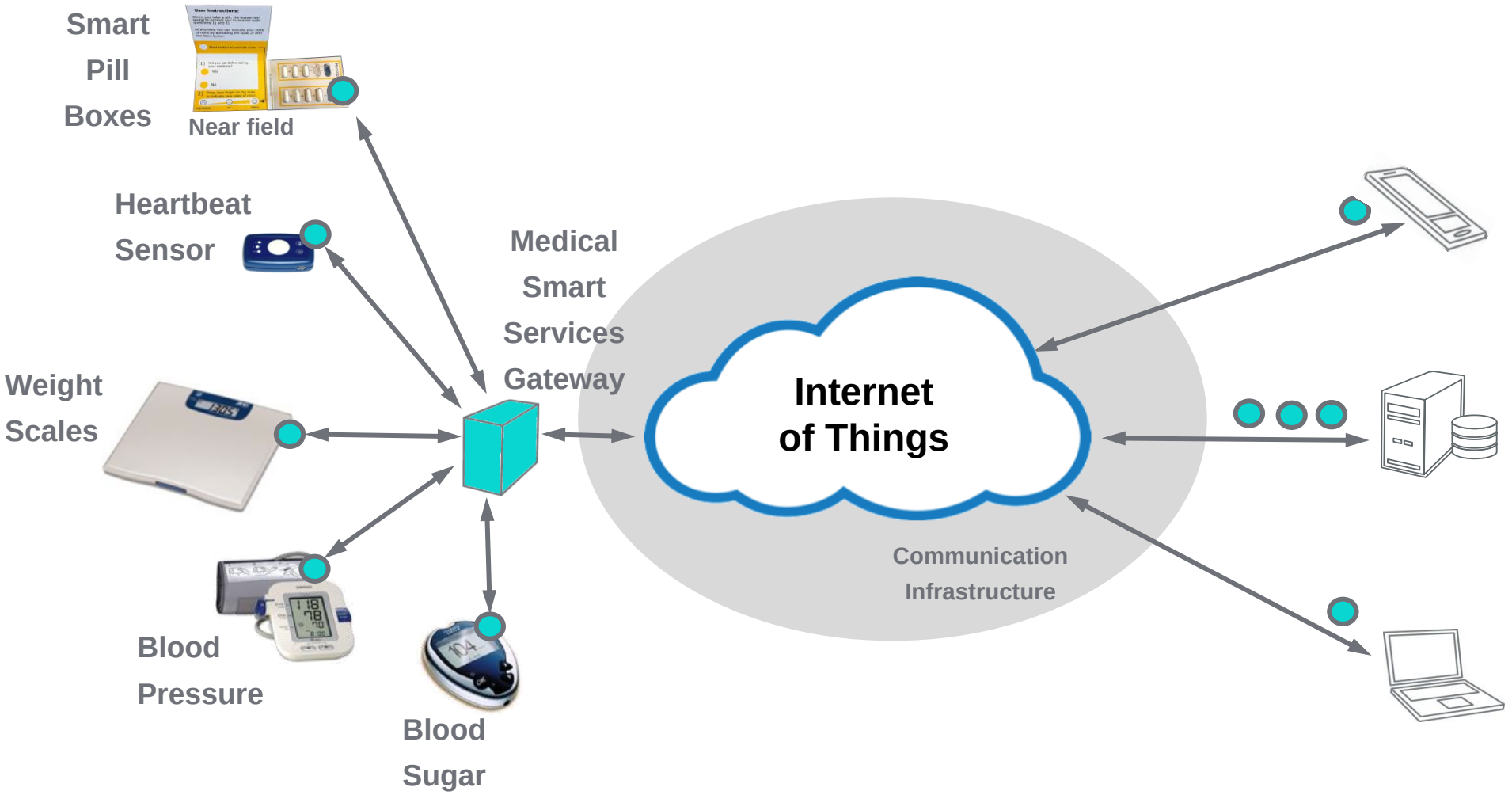
Public/Private Cloud Deployment Infrastructures



CATMedia

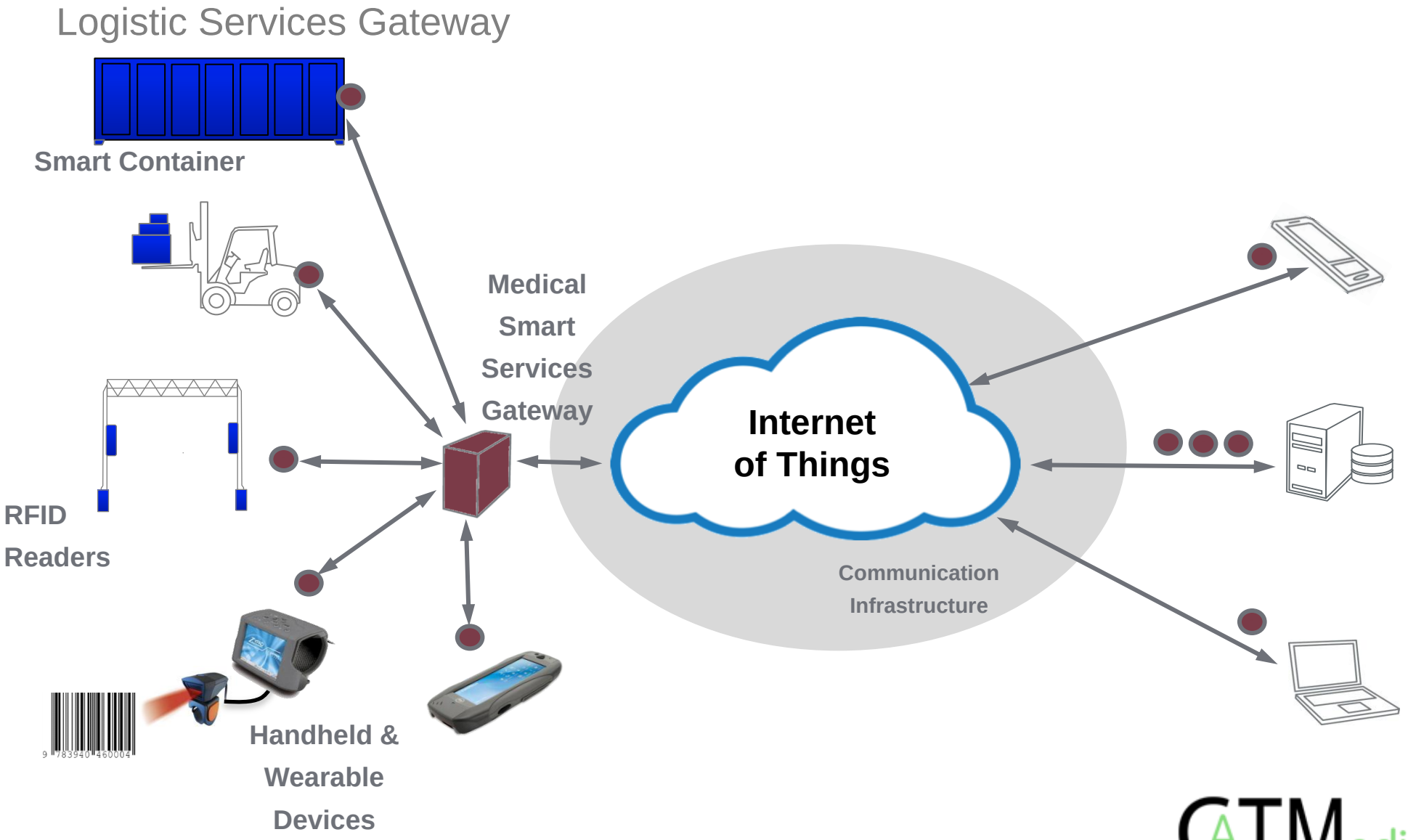
M2M | Vertical Market Scenarios

Medical Services Gateway

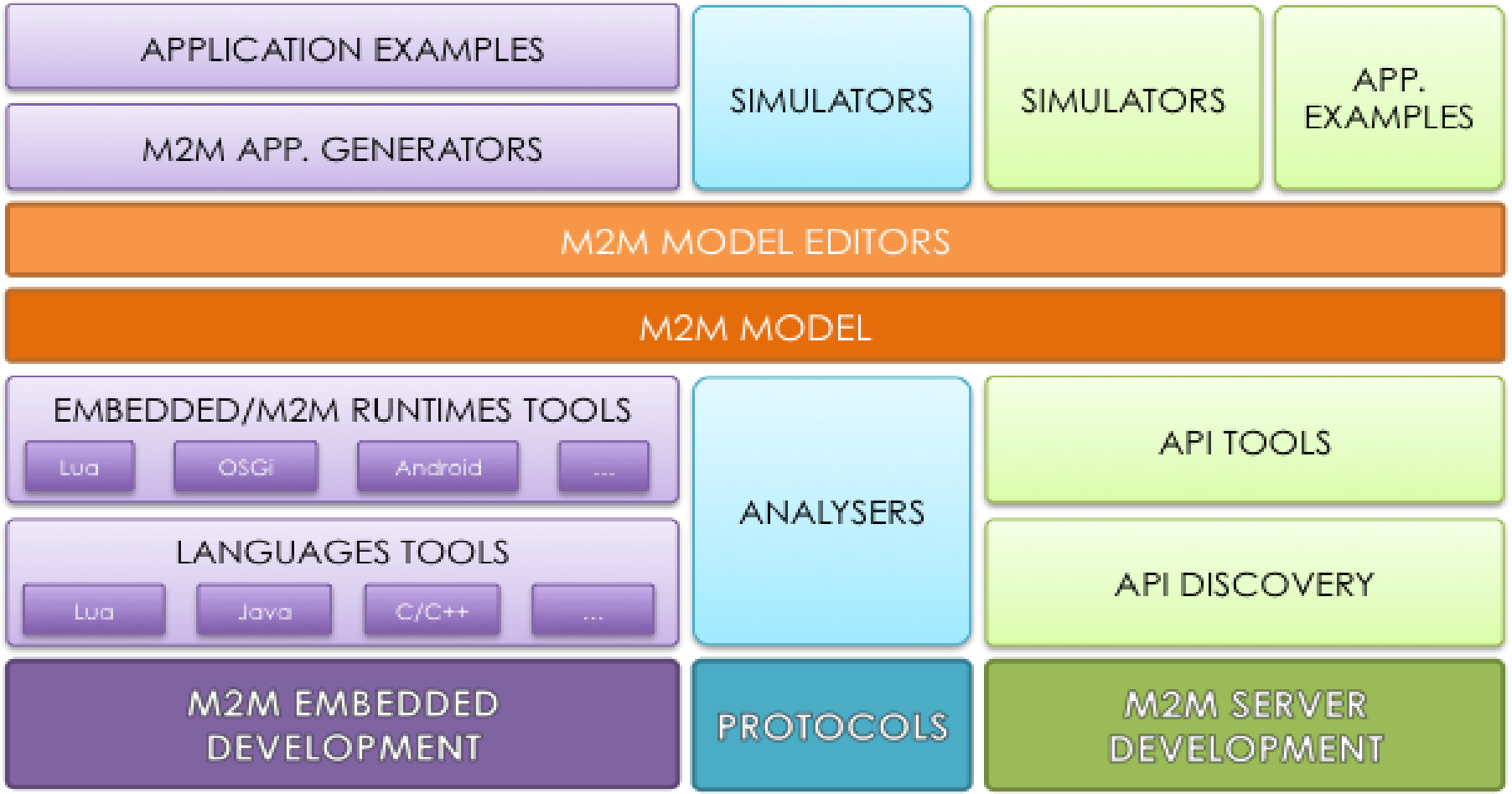


CATMedia

M2M | Vertical Market Scenarios



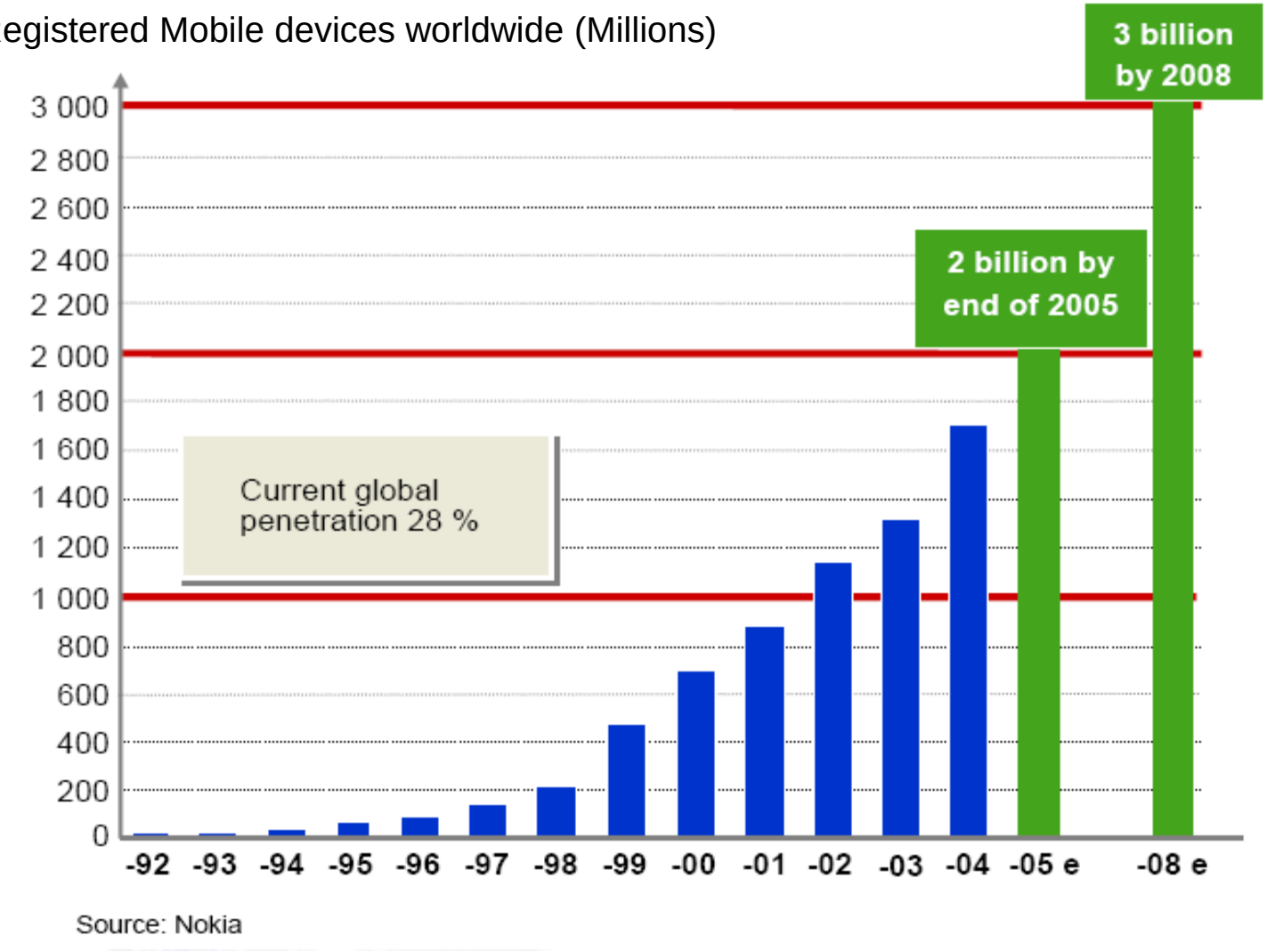
M2M | Tools



NFC

NFC | Stats

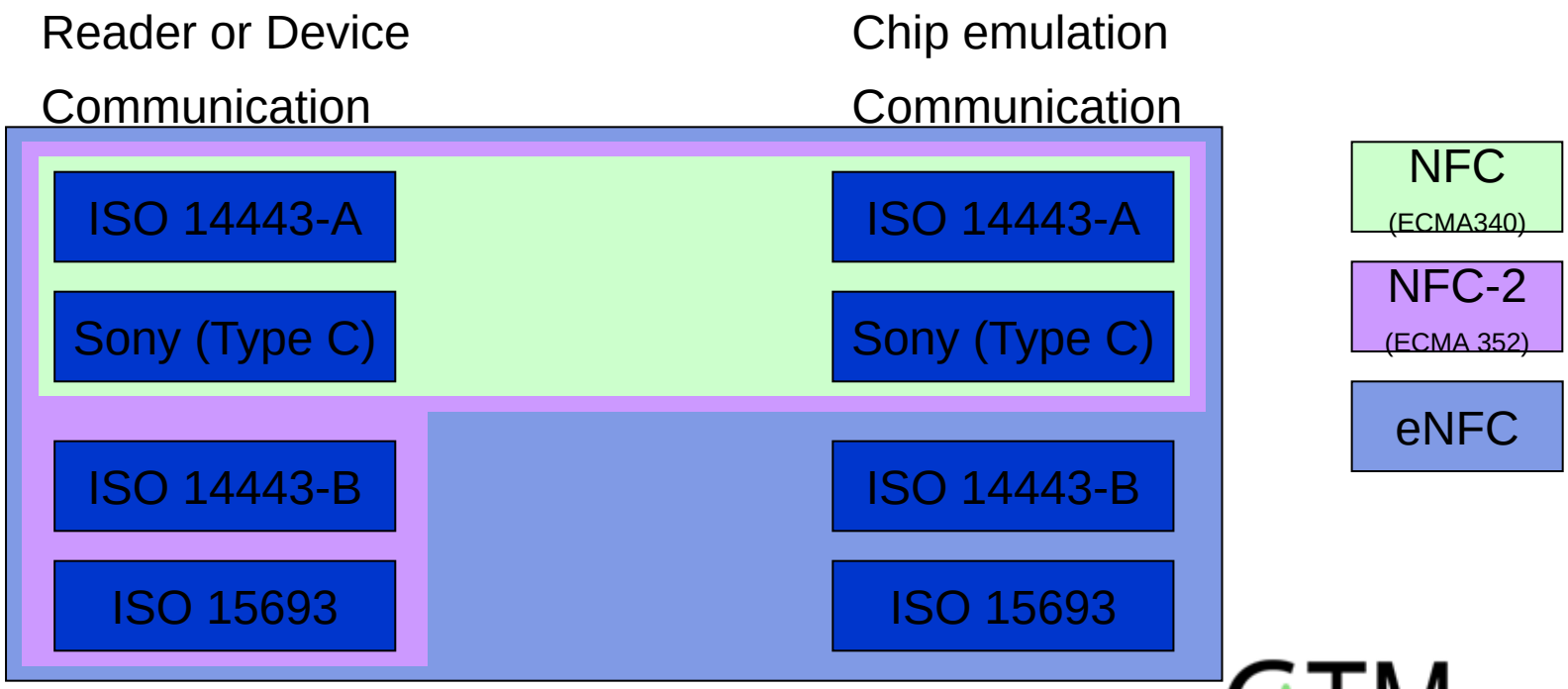
Registered Mobile devices worldwide (Millions)



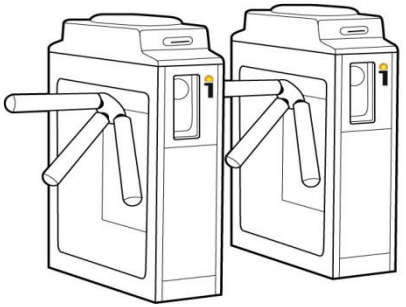
Source: Nokia

NFC | What is eNFC?

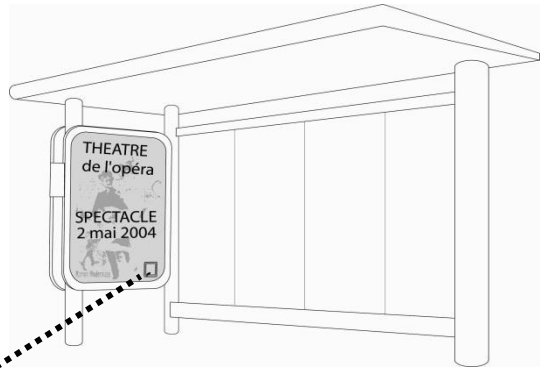
- eNFC (enhanced NFC): Fully compliant NFC technology enhanced by ISO 14443B and ISO 15693 standards on chip emulation side
- eNFC is compatible with all existing and future application using contactless technology



NFC | Use Cases



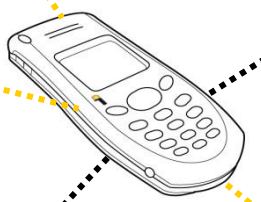
> Pass the gate



> Read a map from interactive billboard

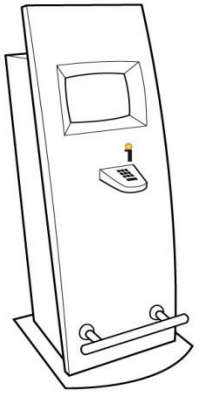
NFC

NFC



> Pay without cash
> Redeem coupons
> Get your Receipt

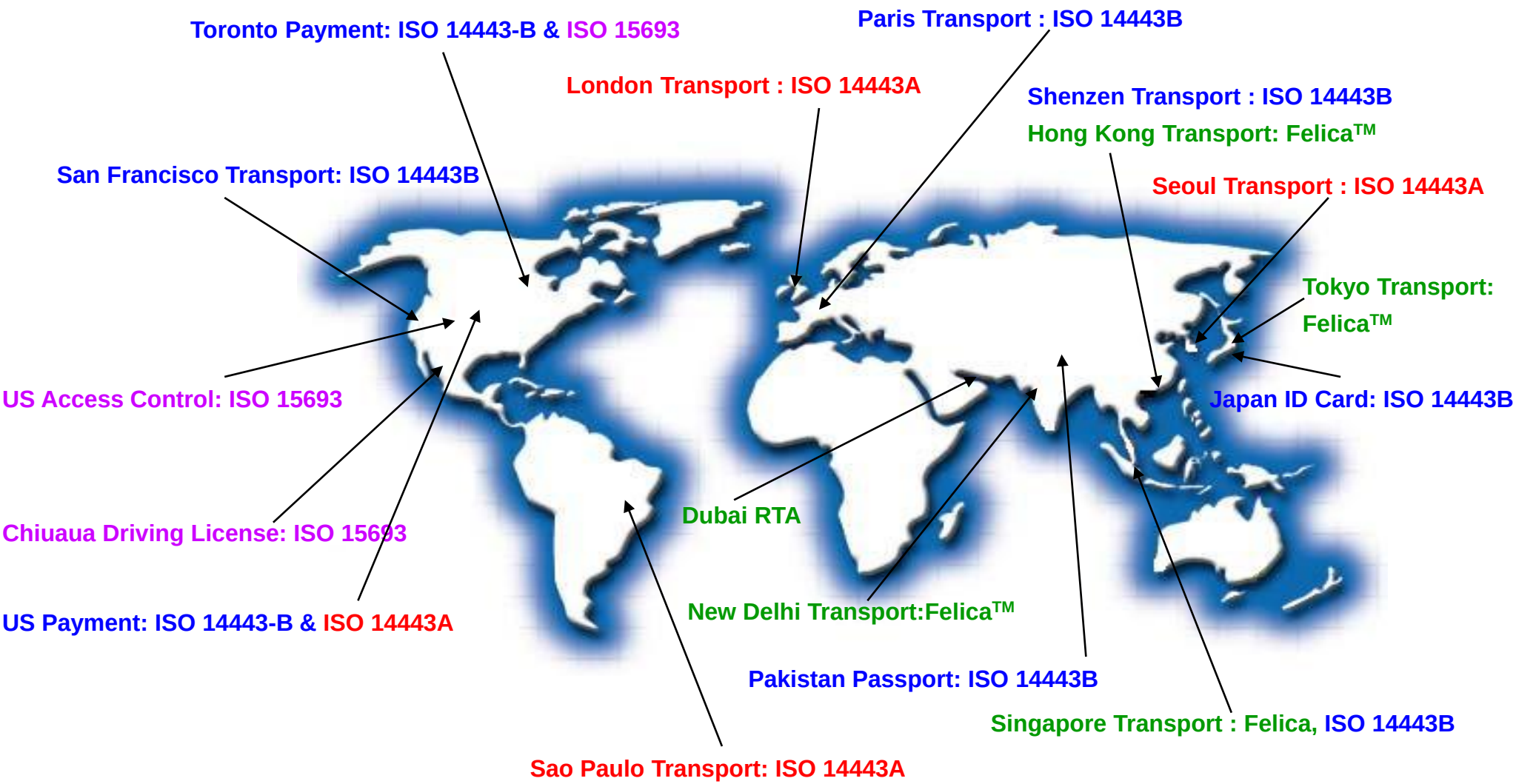
NFC



> Get your e-ticket

> Exchange photos

NFC | Where to use this technology



NFC | Open NFC™

Open NFC interfaces can be classified at different levels, from very high-level interfaces that greatly simplify the usual tasks of NFC applications, to very low-level interfaces that allow fine tuning of NFC hardware parameters for example.

High Level Interfaces:

- NDEF Messages
- Bluetooth and Wi-Fi pairing
- Read / Write to any tag
- P2P
- Virtual Tags

Starting Open NFC 4.3.0, the support for Java porting for JSR-257 devices is discontinued. Older releases of the stack were fully compliant with the JSR-257 standard.

► Android Edition is currently the only one actively maintained with Java Binding!

Security | Possible Usage Scenarios

- Keep close control of software on a system
- Protect kiosk Computers (ATMs..) software from manipulations such as installing a key sniffer
- Strongly identify a machine and its software configuration in online banking or Pizza delivery
- Protect IP in the Cloud

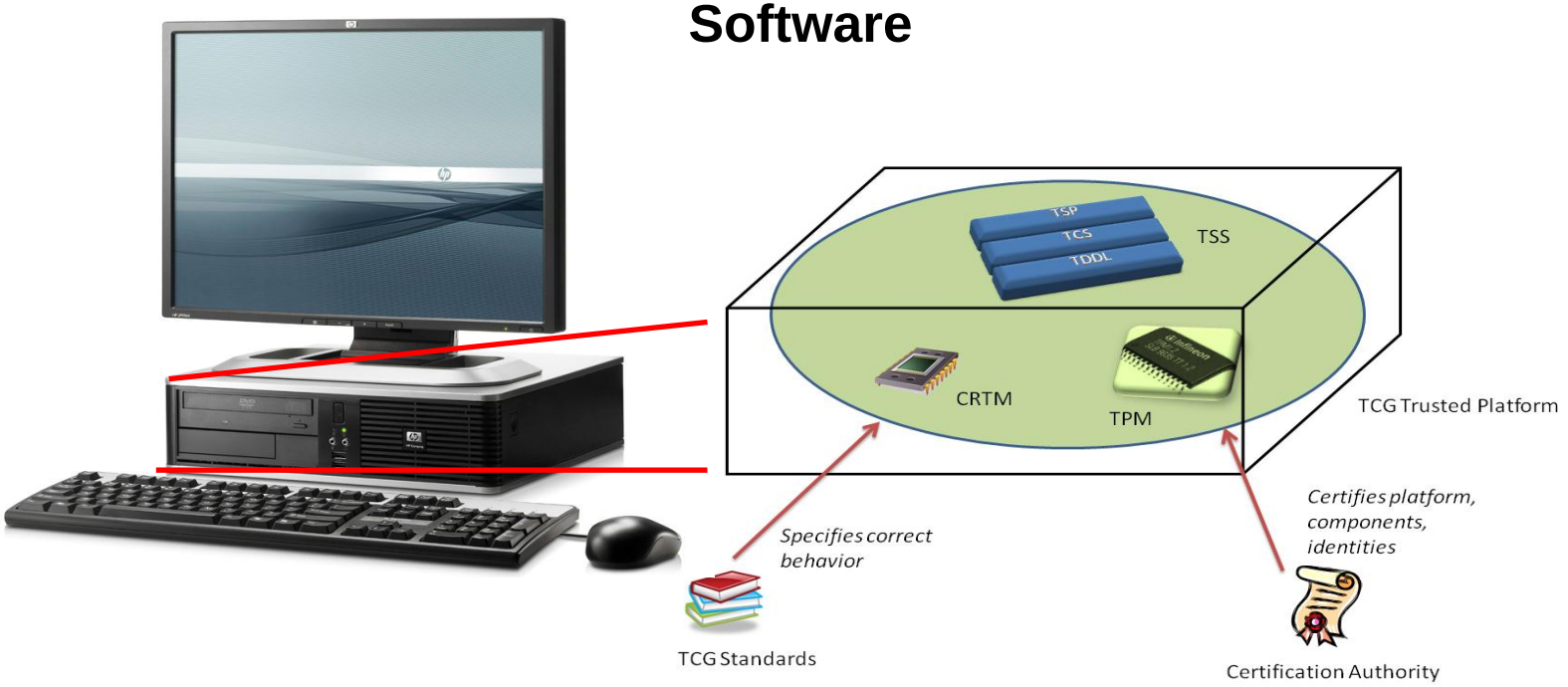
Security | To Catch A Thief

Security | Trusted Platforms

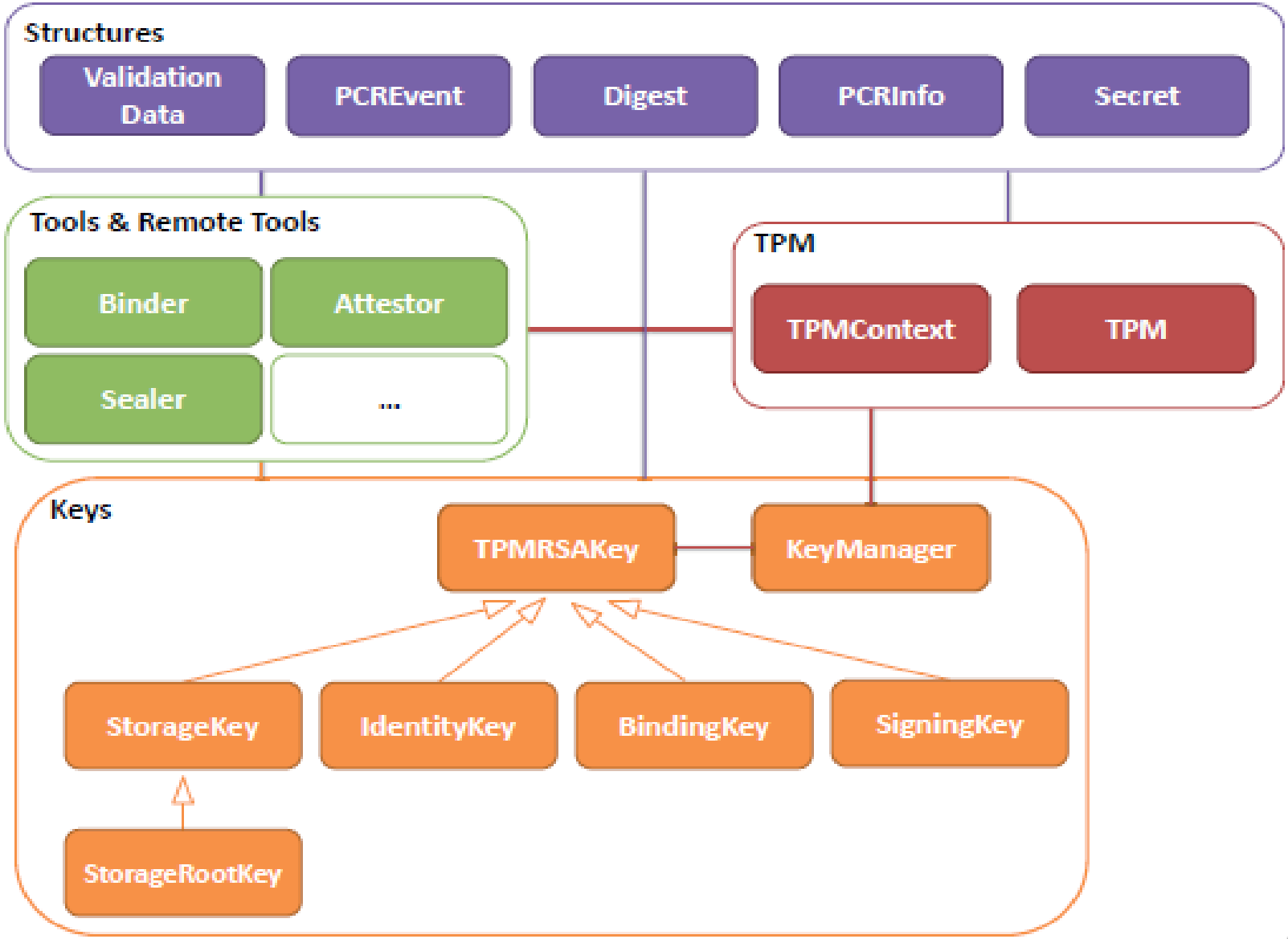
- **Measure** the software executed
- **Store** data securely
- **Report** their status

and feature

- a hardware **TPM**
- an **advanced BIOS** or **chipset**
- a set of **Trusted Computing Software**

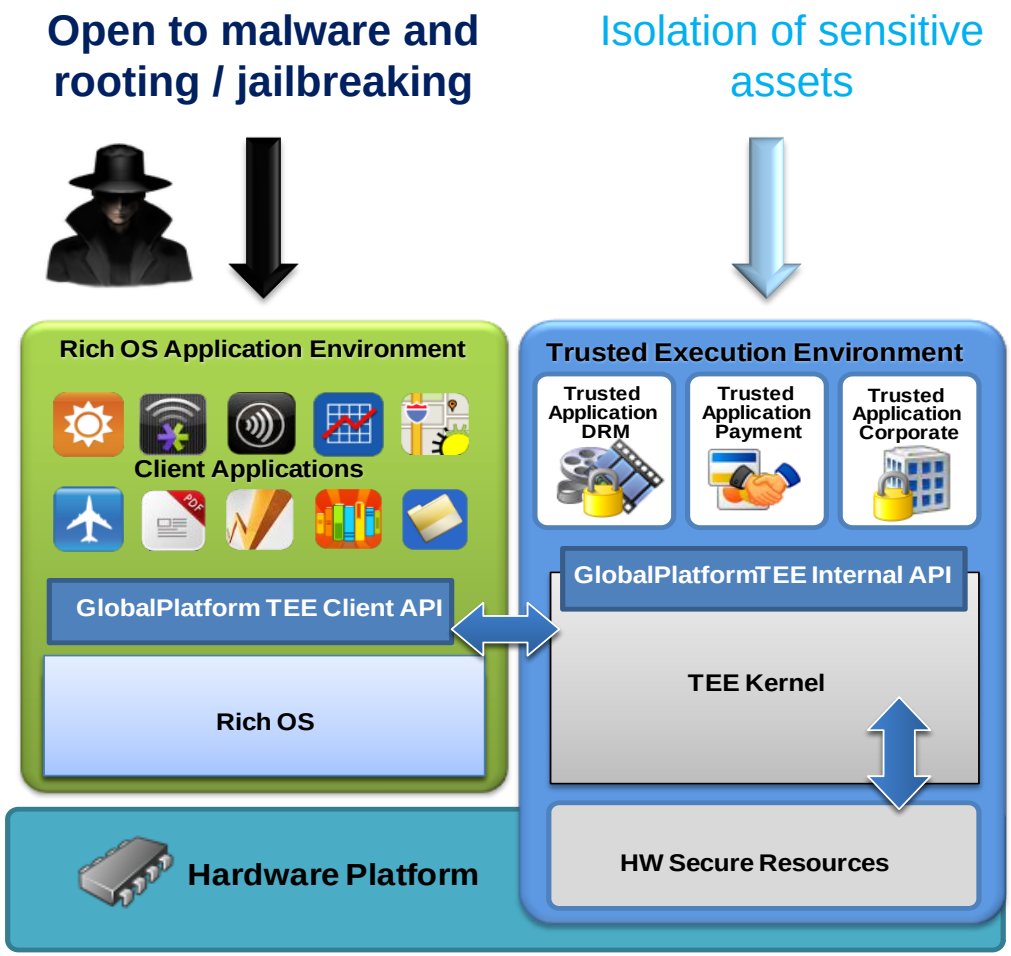


Security | JSR - 321



Security | TEE

What is a Trusted Execution Environment (TEE)?



- TEE provides **hardware-based isolation** from rich operating systems (OS) such as Android, Windows Phone, Symbian, etc.
- TEE runs on the **main device chipset**
- TEE has **privileged access** to device resources **(user interface, crypto accelerators, secure elements…).**

CATMedia

Security | Secure Element

- EMV applications and their data shall be always stored in a secure area of a handset – in a ***secure element***
- Secure element is a smart card chip
- Currently 3 approaches:
  - **SIM-centric:** Secure Element is (in) USIM – payment applications are stored on a USIM card
  - **Embedded secure element** – additional smart card chip integrated in a mobile phone (e.g. Samsung NEXUS S)
  - **External secure element** (e.g. smart card chip integrated in a Micro SD card)
  - Application management ‘over-the-air’

CATMedia

Security | Java Card Technology

Secure, Connected, Versatile

- Interoperable platform for delivery of trusted personal services
  - High, industry-proven security
  - Designed for the smallest silicon hardware devices
  - Runs Java in as little as 4 KB RAM
- Deployed on >5 billion devices
  - Growing at 1.4 bill. Devices p. year
  - SIM Cards, secure elements, eID, payment services

CATMedia

Security | Java Card Technology

Let‘s talk

Q & A

CATMedia

Links

Eclipse – Project UOMo

http://www.eclipse.org/uomo/

Units of Measurement API

http://www.unitsofmeasurement.org

UCUM

http://www.unitsofmeasure.org

CATMedia

Links (2)

Eclipse – M2M IWG

http://www.m2m.eclipse.org

CTMedia

Contact

werner@catmedia.us

or

uomo@catmedia.us

Twitter: @wernerkeil

Hashtag #EclipseUOMo

CATMedia