HAE Innovations
Pioneering Wireless Innovation

Designing for IoT

HAE Innovations – Pioneering Wireless Innovation

March 2016

Market

- How Integrators think about building IoT devices
  - **Module COGS** $8-$17 for module to $50 for turn-key module + app host
  - **Certification costs:** inverse of COGS. $15K to $175K
  - 80% of the market use the certified module or module turn-key products (Tier II/Tier III below).

	Sales Volume	Module Cost	Design choice	Design Testing	Certification Requirements
Tier I	>500K	$	"chip down"	**Tier II +** **RF/RRM/NAS:**	1. Equivalent to a new certification ~$175K (credit fo BB SW cert at Module)
Tier II		$$	Module+ custom host PCB	**Tier III +** **Software DVT:** **APP-Module Comms issues** (unique to each module model) **Hardware DVT:** Antenna, Power, SIM i/f	1. TIS/TRP ~$20K-$30K 2. Module Cert. paperwork 3. Software review 4. Carrier FOTA, Provisioning
Tier III	<5K	$$$	Turn-key Module + Host PCB: Unit is calibrated, certified, and comes with commercial SIM to work with	1. **Travel:** ~$2K-5K 2. **Airtime:** ~$2K Internet connect, SIM i/f, Security (APP, SIM, TCP/IP)	1. TIS/TRP ~$10K-$15K 2. Module Cert. paperwork 3. Minimal software review 4. Carrier FOTA, Provisioning

Top 7 IoT Check list –

What are the top 7 issues faced and how do enable customers to solve them

Issue
APP-Module COMM implementation issues
Antenna design issues (impedance, i/f)
Power supply design (noise,…)
Module configuration issues by APP
SIM electrical continuity
Carrier certification cost
Security – OTA attacks

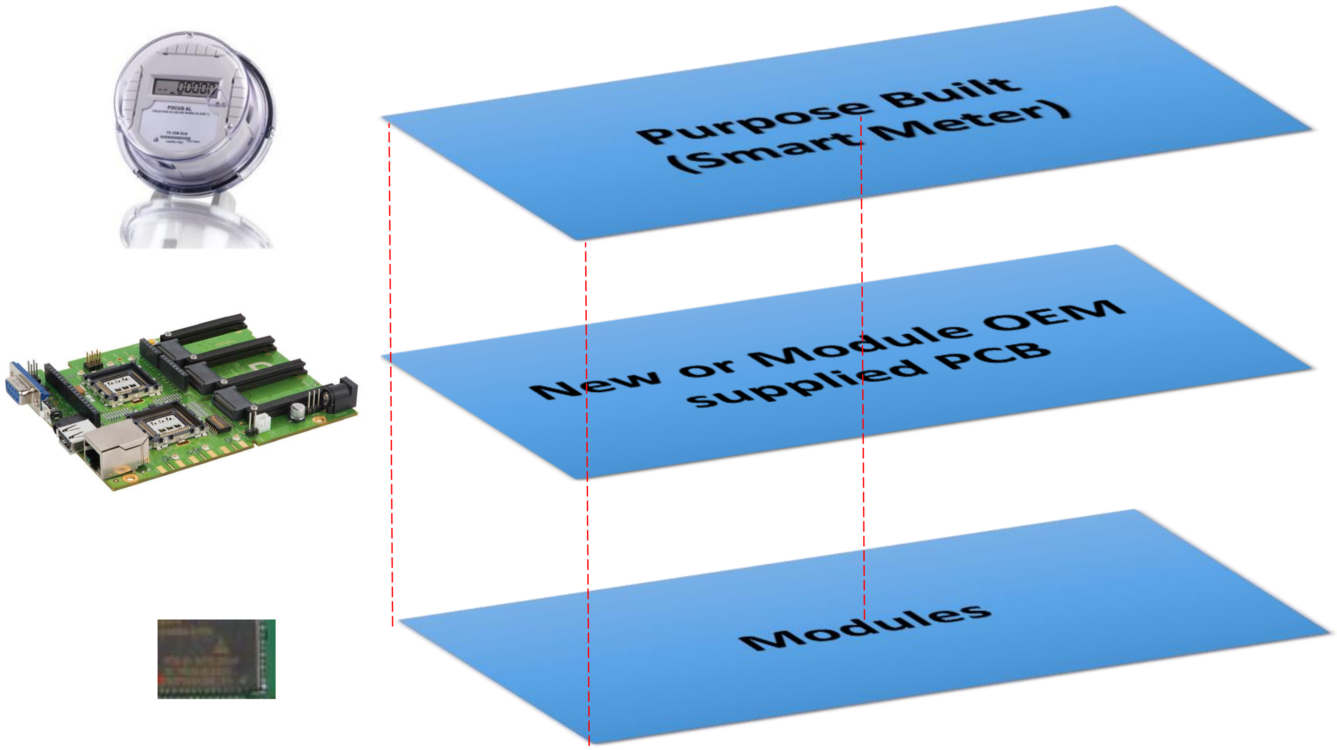
IMPT: Importance to Integrator

ID – implementation difficulty

Security Testing

- Improper mobility handling triggers camping on cells with poor coverage
- Chatty applications that use unnecessary bandwidth impacting load balance
- Requesting more RRB than actually used will waste a lot of network resources
- Poor upper layer (IoT client-IoT Cloud server) security exposes RAN to malware behaving within the rules of the link layer
- SIM spoofing
- Attach procedure floods given link layer security hasn't been established yet
- Paging floods in deployments where devices IP address is NAT'ed by a controller instead of the RAN
- SIP client invite floods
- From a different perspective, RAN's ability to distinguish mission critical (mhealth) from non-mission critical devices could lead to inadvertent shut down of devices solely based on adversarial data behavior.

Construction of an IoT Product

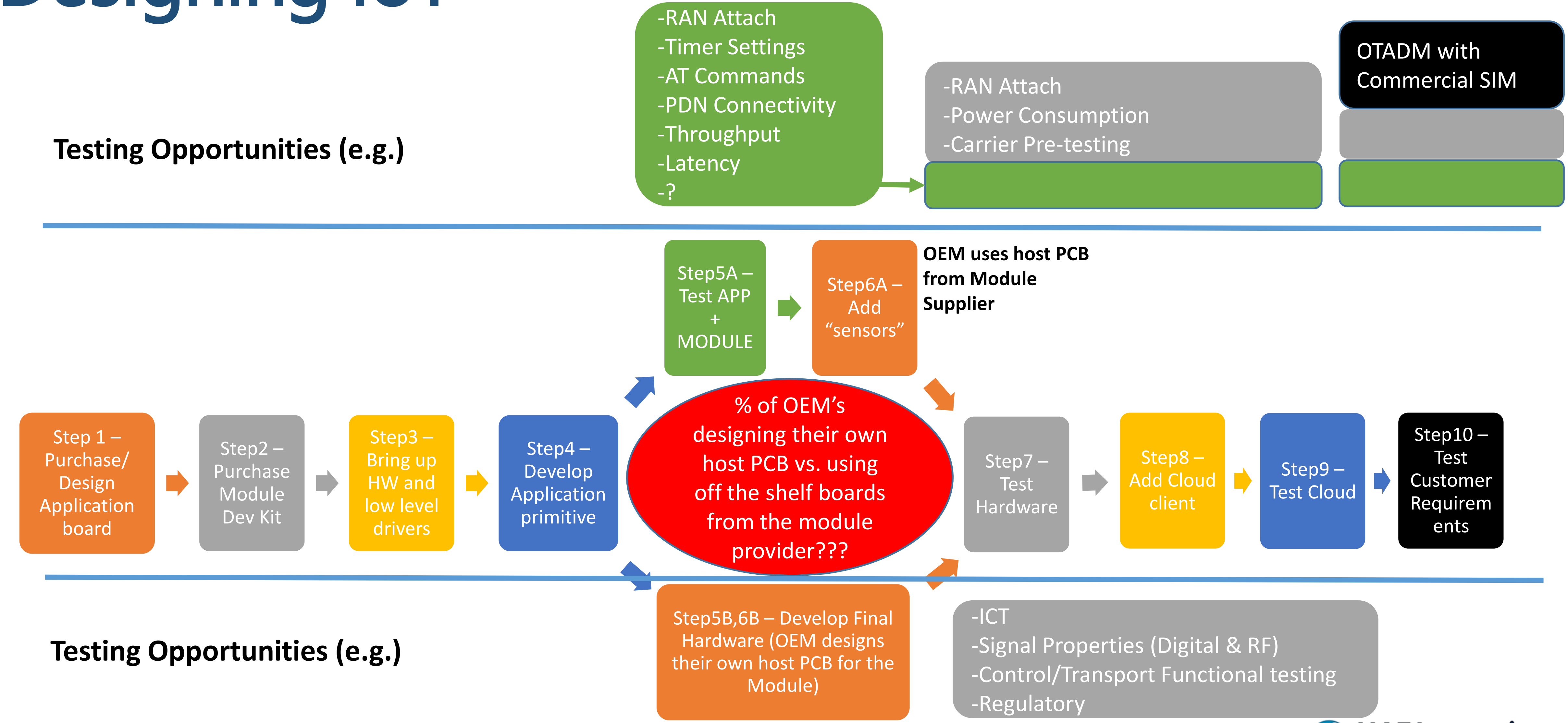


Purpose Built
(Smart Meter)

New or Module OEM
supplied PCB

Modules

Designing IoT



Testing Opportunities (e.g.)

-RAN Attach
-Timer Settings
-AT Commands
-PDN Connectivity
-Throughput
-Latency
-?

-RAN Attach
-Power Consumption
-Carrier Pre-testing

OTADM with Commercial SIM

Step 1 – Purchase/ Design Application board

Step2 – Purchase Module Dev Kit

Step3 – Bring up HW and low level drivers

Step4 – Develop Application primitive

Step5A – Test APP + MODULE

Step6A – Add “sensors”

OEM uses host PCB from Module Supplier

% of OEM’s designing their own host PCB vs. using off the shelf boards from the module provider???

Step7 – Test Hardware

Step8 – Add Cloud client

Step9 – Test Cloud

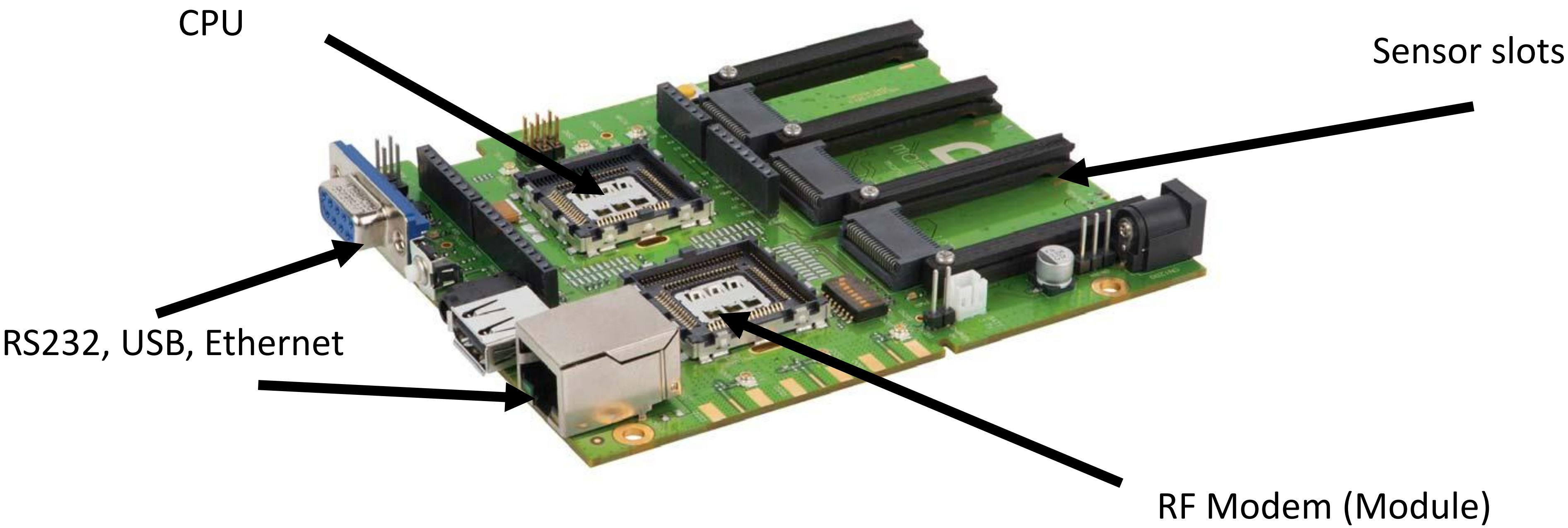
Step10 – Test Customer Requirements

Step5B,6B – Develop Final Hardware (OEM designs their own host PCB for the Module)

Testing Opportunities (e.g.)

-ICT
-Signal Properties (Digital & RF)
-Control/Transport Functional testing
-Regulatory

Step2 – Purchase RF Modem & CPU from Module provider



CPU

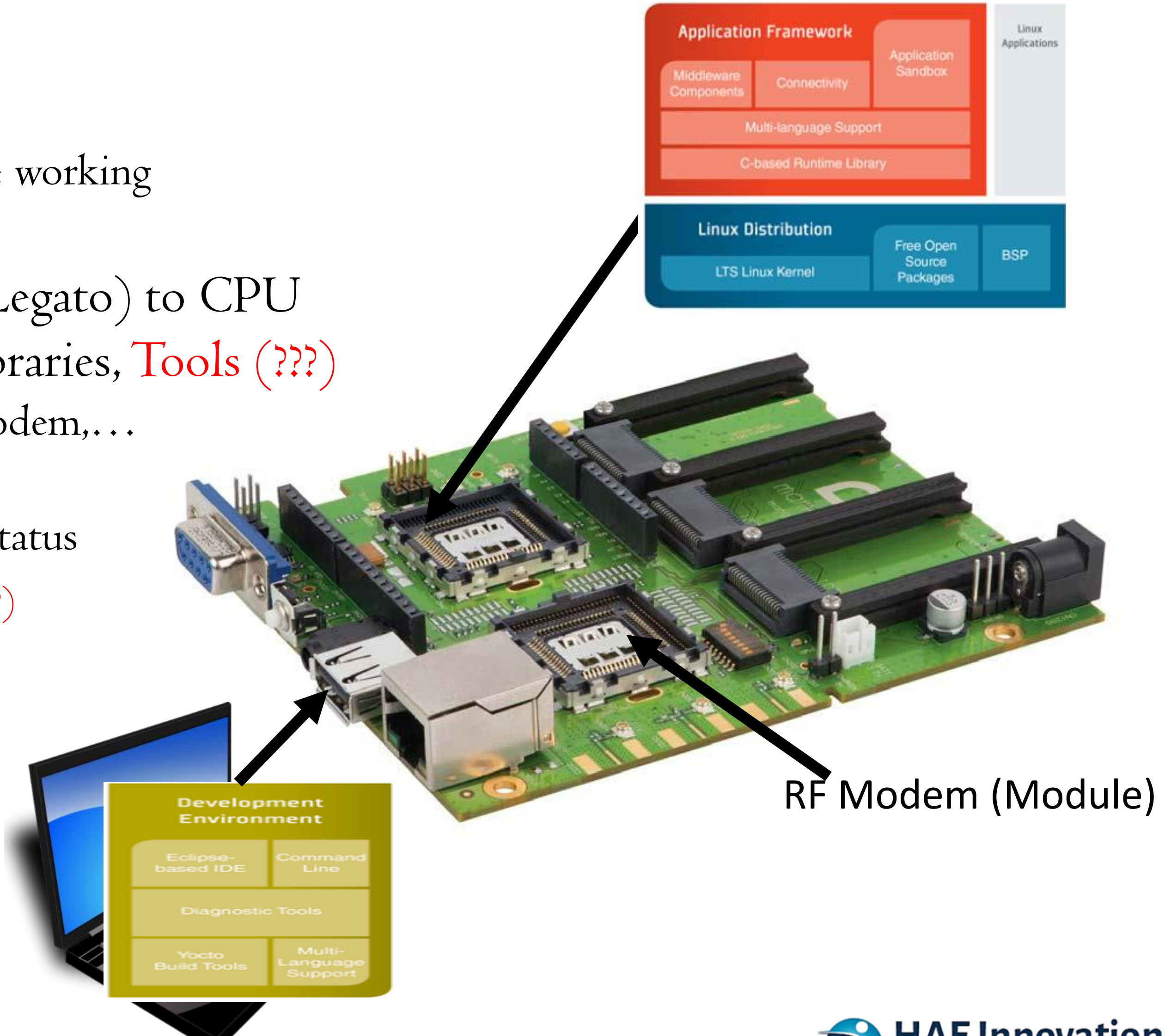
Sensor slots

RS232, USB, Ethernet

RF Modem (Module)

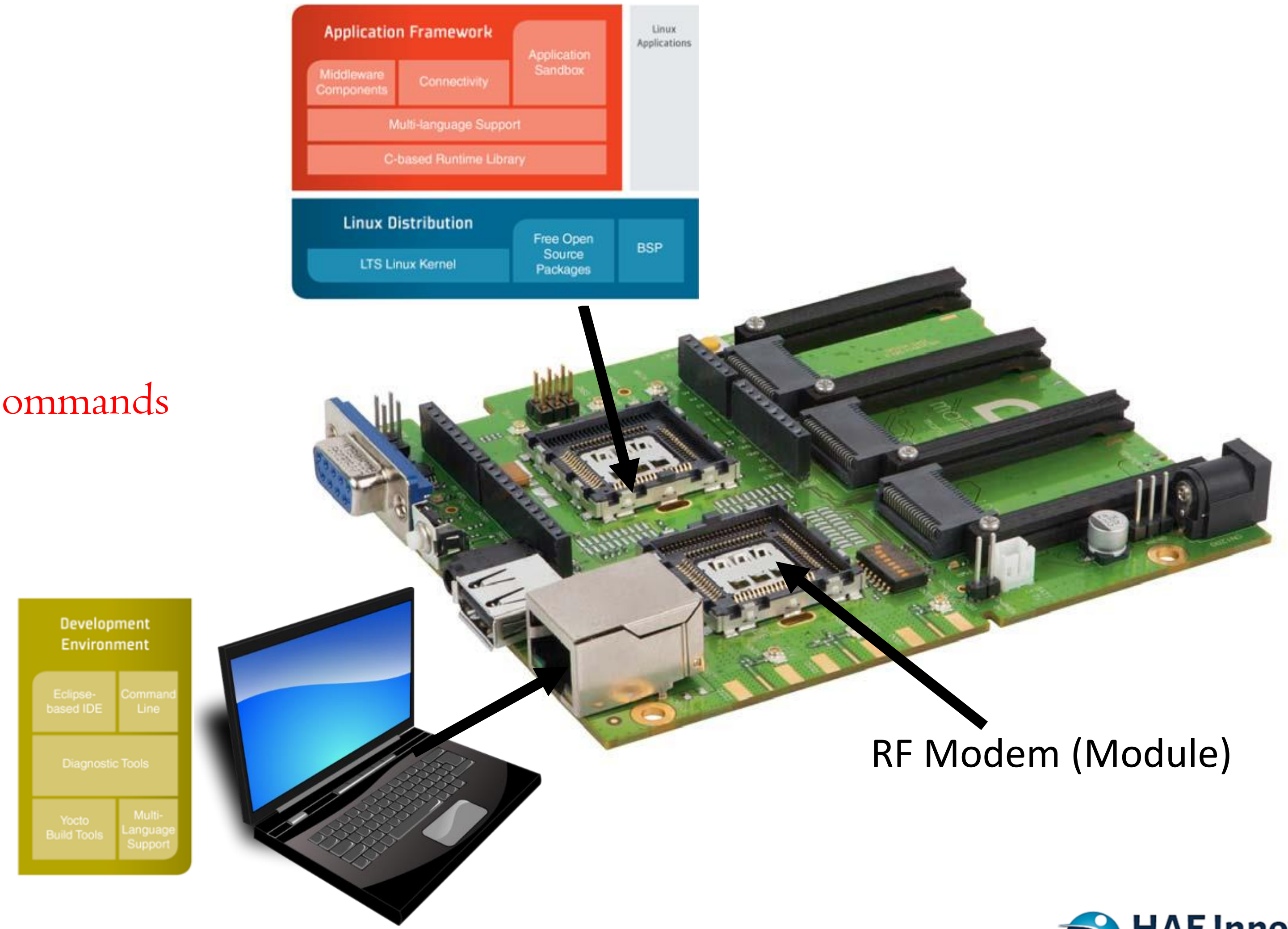
Step3 – Bring up HW and low level drivers

1. Install USB driver to connect PC to PCB
   a) Telenet in PCB to test communications are working
2. Download Linux OS to CPU
3. Download Application Framework (e.g. Legato) to CPU
4. Launch Framework, and test Services, Libraries, Tools (???)
   1. e.g. Services API's Manage GNSS, Cell Modem,…
   2. e.g. Libraries CLI syntax
   3. e.g. Tools cm radio (reports back modem status
5. Verify Module works (Test or Commercial SIM?)
   1. Native (Module OEM) AT Commands
   2. RAN Attach
   3. Timer Settings
   4. PDN Connectivity
   5. Throughput
   6. Latency



Step4 – Develop Application primitive

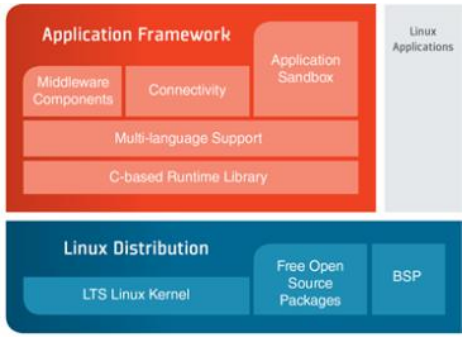
1. Create Component
2. Create App
3. Build App
4. Install App to Target
5. Test App
    1. New (IoT OEM) AT Commands



RF Modem (Module)

Step5A – Test APP + Module (off-shelf host PCB)

I. Required Testing(Test or Commercial SIM?)
   1. New (IoT OEM) AT Commands
   2. RAN Attach
   3. Timer Settings
   4. PDN Connectivity
   5. Throughput
   6. Latency
   7. …?



Application Framework

Middleware Components

Connectivity

Application Sandbox

Multi-language Support

C-based Runtime Library

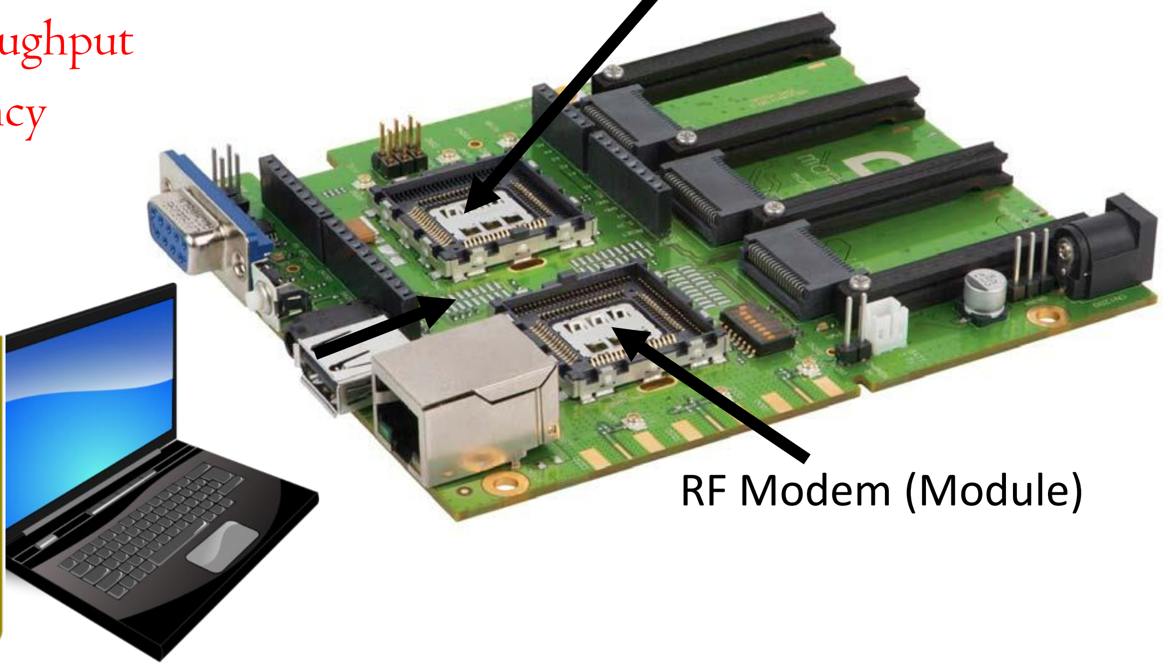
Linux Applications

Linux Distribution

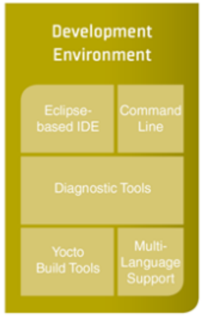
LTS Linux Kernel

Free Open Source Packages

BSP



RF Modem (Module)



Development Environment

Eclipse-based IDE

Command Line

Diagnostic Tools

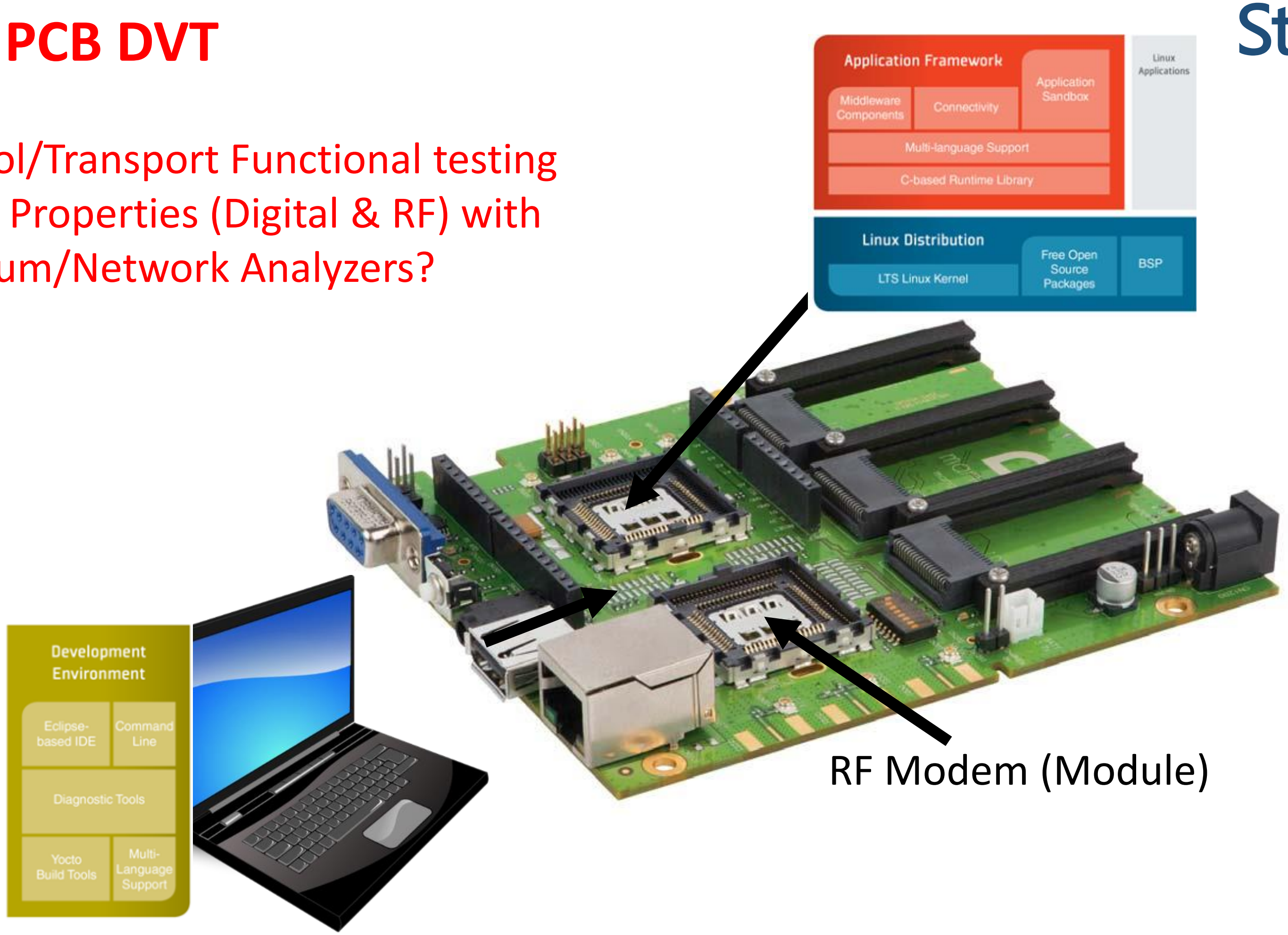
Yocto Build Tools

Multi-Language Support

Step5B – Test Hardware (new PCB host)

New PCB DVT

-ICT
-Control/Transport Functional testing
-Signal Properties (Digital & RF) with Spectrum/Network Analyzers?



RF Modem (Module)

Step6B – Add "Things"

1. Modify App to include
   1. Executables (drivers for each sensor)
   2. Event trigger/responses
   3. Directories
2. Reinstall App on Target

Step7 – Test Final Hardware

I. Test App (Test or Commercial SIM?)

  I. New (IoT OEM) AT Commands

  2. RAN Attach

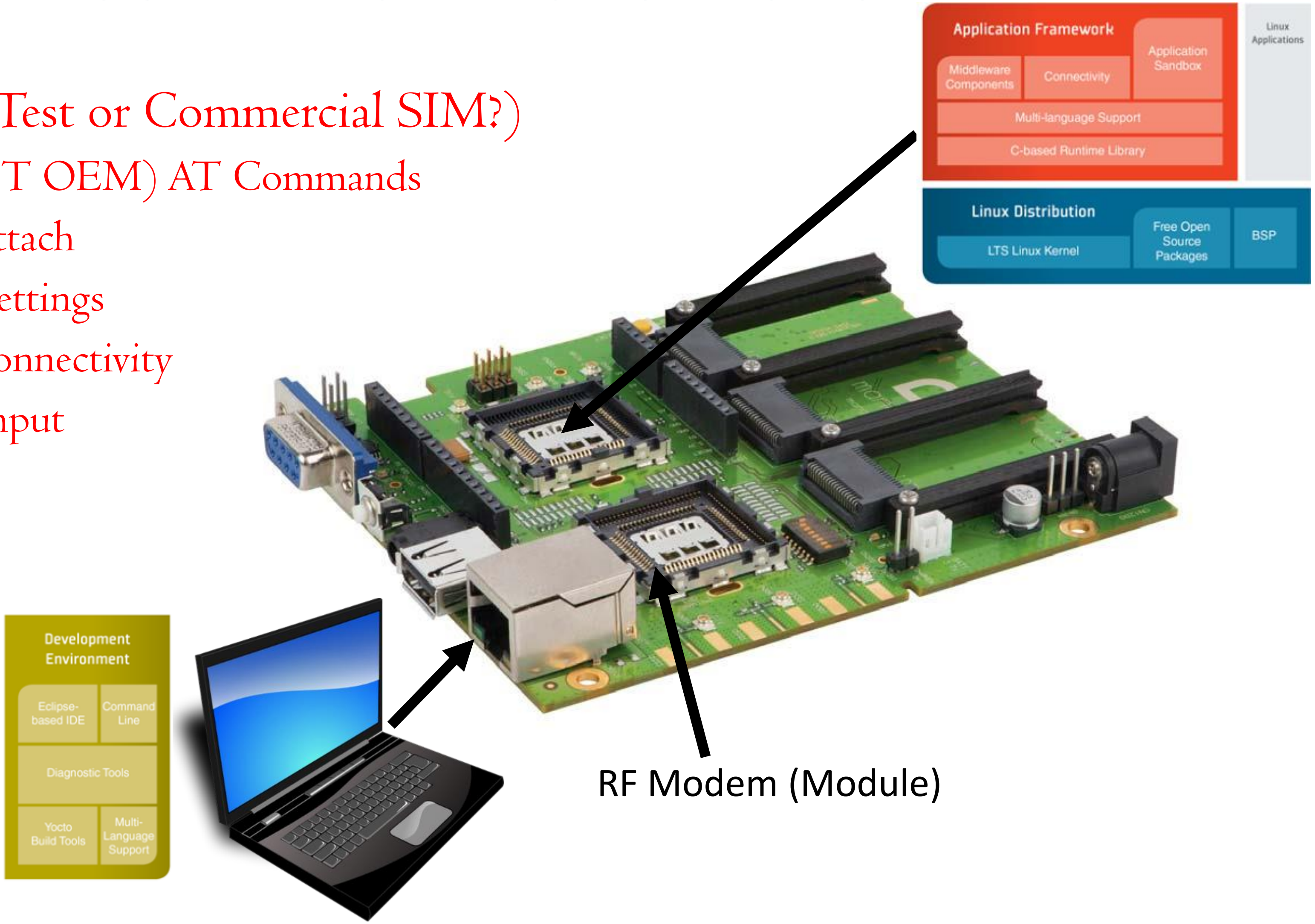
  3. Timer Settings

  4. PDN Connectivity

  5. Throughput

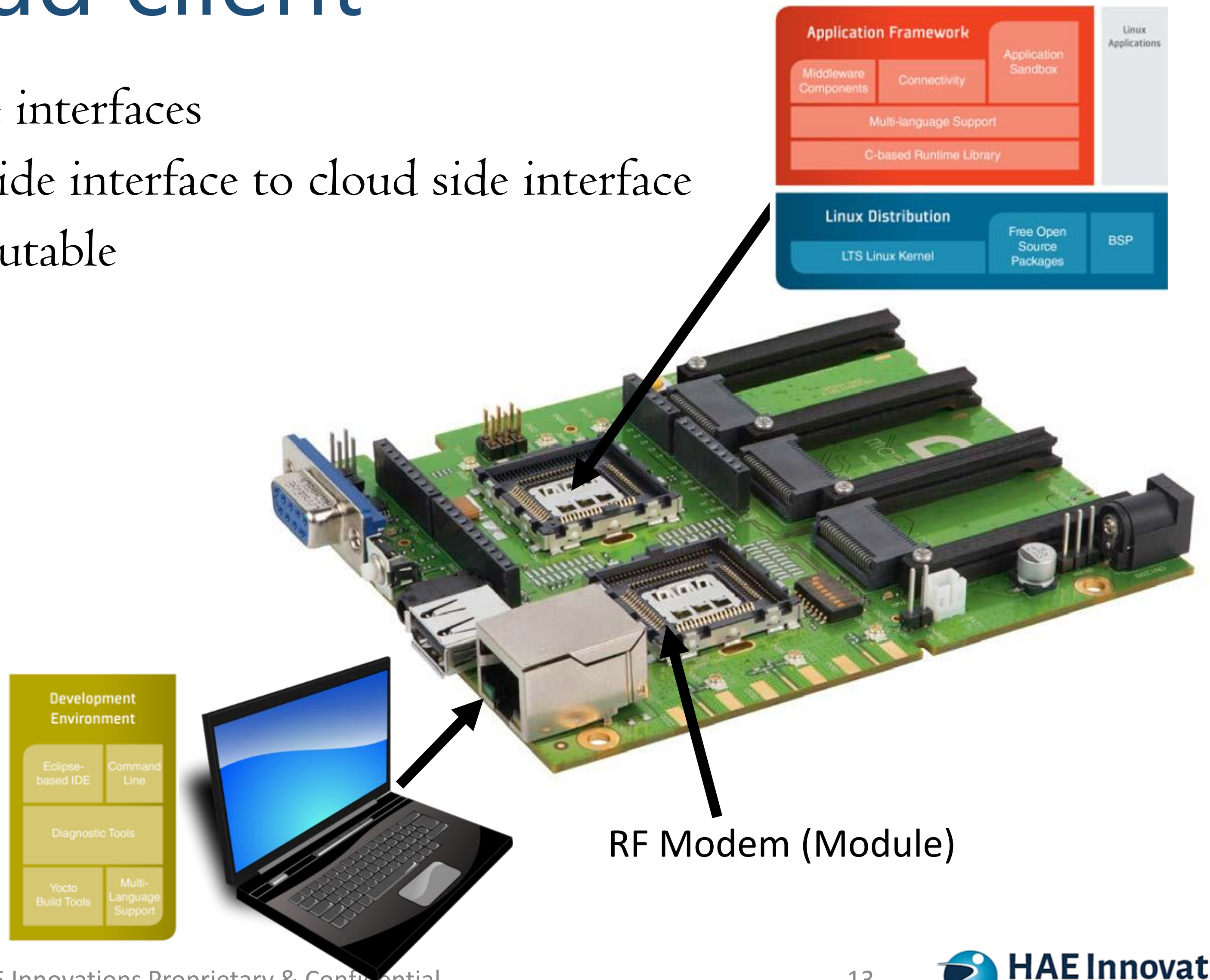
  6. Latency

  7. …?



Step8 – Add Cloud client

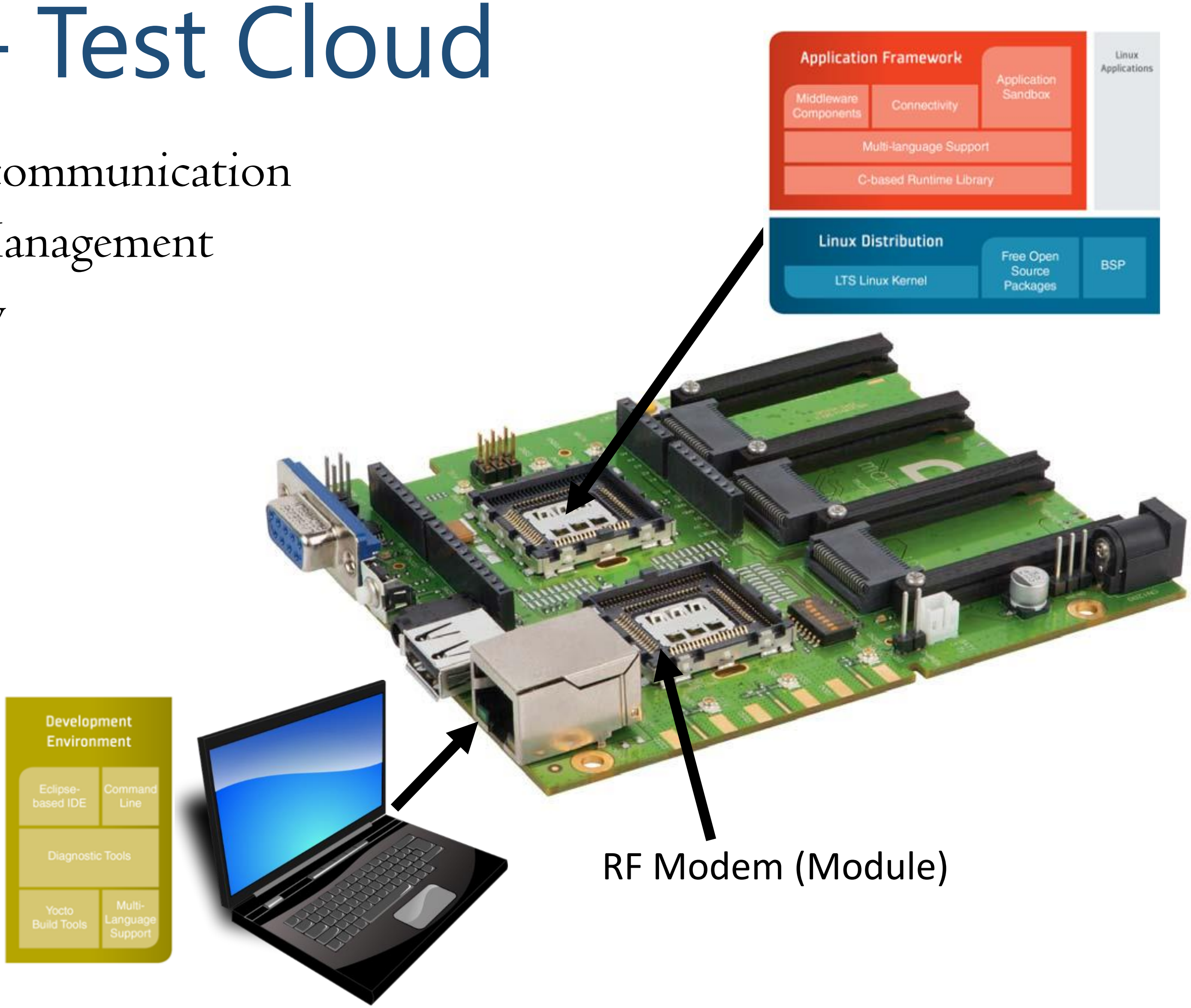
1. Develop client with requisite interfaces
2. Develop app to bind client side interface to cloud side interface
3. Install on target to start executable



RF Modem (Module)

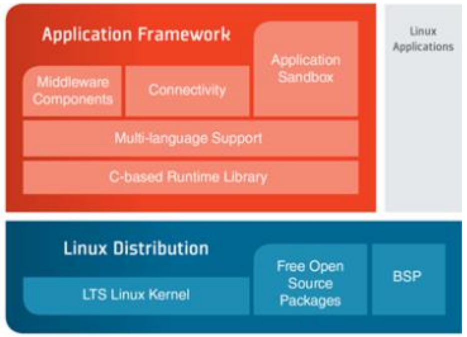
Step9 – Test Cloud

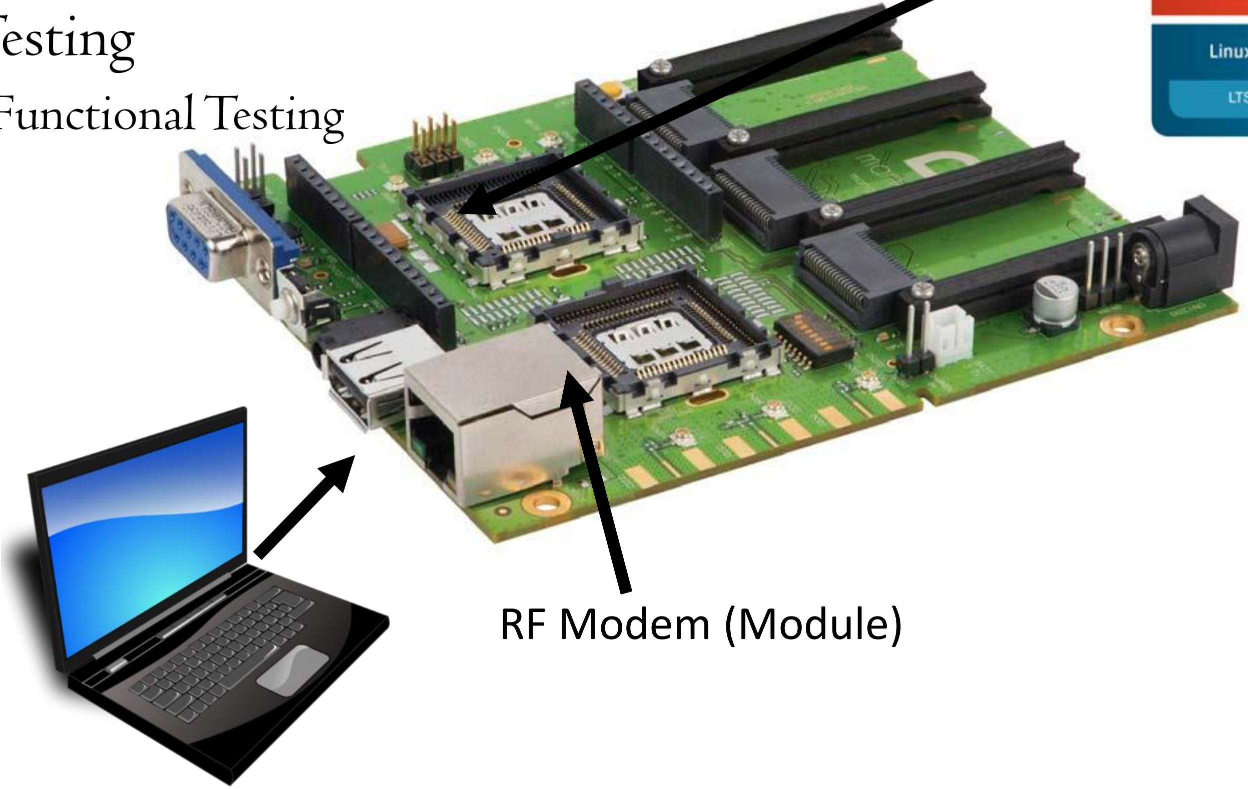
1. Client communication
2. Data Management
3. Security



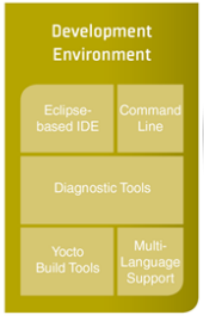
Step10 – Test Customer Requirements

1. Carrier Testing
   1. Informal: test on live network with commercial SIM
   2. Formal: FCC/PTCRB/Carrier Specific, Regulatory
2. End Customer Testing
   1. Market based Functional Testing





RF Modem (Module)



Open Questions for IoT OEM's

- What are IoT OEM's doing to try out the module after they
  - First purchase a development kit
  - Develop their application to control the module
  - Develop their host PCB to ensure they can still get a signal that attaches
- Do OEM's often know what module they require?
- Where in the design cycle do they struggle the most?
- Is carrier approval more than ensuring provisioning and OTADM works?
- What's more interesting to the typical IoT OEM. The LTE network simulator is
  - At their desk
  - Nearby location where they can take their device to test
  - Ship the device somewhere for someone to deal with it (ie get it to attach)
- Post deployment, do OEM's concern themselves
  - with a lot internal regression testing of new software builds (vs. test a few devices and declare it done).
  - Monitoring their devices in the field via third party applications

Open Questions for Module Providers

- How are modules tested during internal development?
- How are module development boards tested during internal development?
- Is the development board a core part of the sale or something that would be attractive to outsource?
- When a carrier refers to a certified module, are they referring to chip on pcb or the development board hosting the chip/pcb + connectors, etc.
- Externally,
  - What kind of support consumes a lot of time from the module supplier? LTE attach? Board layout issues, ….

THE END