SYNOPSYS®
Silicon to Software™

Processor Solutions for Energy-Efficient IoT Applications

Pieter van der Wolf

MPSoC 2017

July 2 – 7, 2017

Agenda

IoT applications

Sensor processing and voice / audio

Wireless connectivity

Security

Conclusion

From the Edge to the Cloud

IoT Edge Devices
(Smart Devices)

“Things” with sensors & actuators that monitor and control

Aggregation Layers
(Hubs/Gateways)

Connectivity & Interfaces to aggregate the edge data to send to the cloud

Remote Processing
(Cloud Based)

Applications to analyze the data and offer cloud services

IoT Edge Device Market

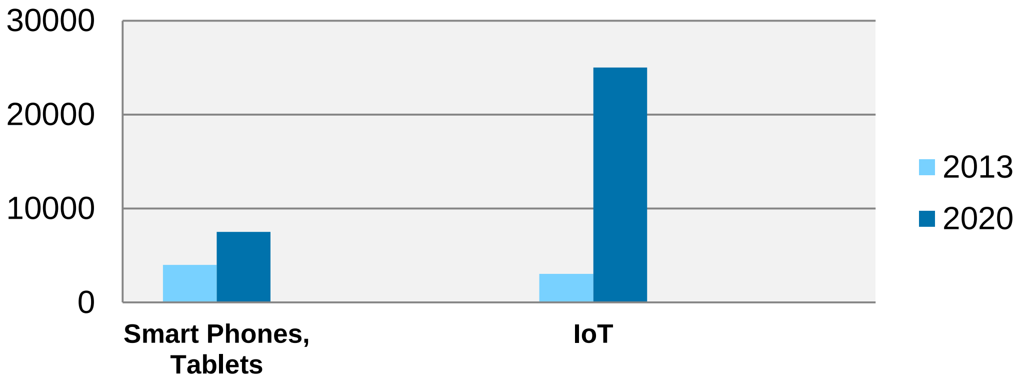
Internet of Things and its Attractive Growth…

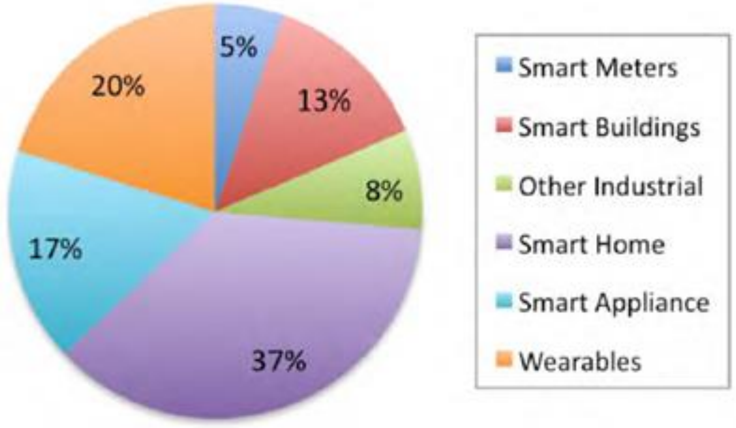
- 5B people connected by 2020 [1]
- 11.5% CAGR through 2022 for IoT Chip Market *($4.6 -> $10.8B)* [2]
- 50%+ Volume driven by Smart Home and Wearables [3]
- 55% Global IoT security market growth through 2019 [4]
- $7.4B and over 887 deals to IoT startups since 2010 [5]

Fragmented Market….. Key Applications Drive Innovation

- Mobile handsets drive interoperability *(WiFi, Cellular, Bluetooth)*
- Regulations & Standards drive security
- Wearables drive energy efficiency
- Drive for low-cost in high-volume markets

Billions of "Things" in use*





2021: 2.2 billion units

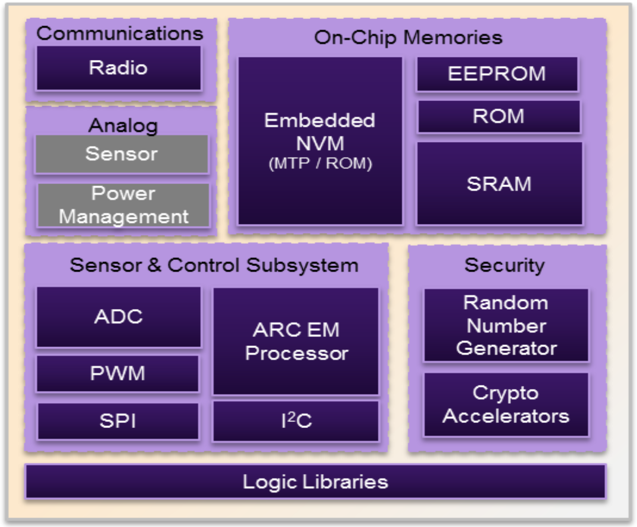
Figure 4-1. IoT unit share by market segment, 2021. We project the consumer-driven smart home and wearables segments to be the largest contributors. (Source: The Linley Group)

Source[1]: World Economic Forum
Source[2]: Market and Markets
Source[3]: Linley 2016 IoT Report
Source[4]: Marketresearchreports.biz
Source[5]: CB Insights

Example IoT SoC Architectures

Corral the Market Fragmentation

Smart Analog Device



Communications: Radio

Analog: Sensor, Power Management

On-Chip Memories: Embedded NVM (MTP / ROM), EEPROM, ROM, SRAM

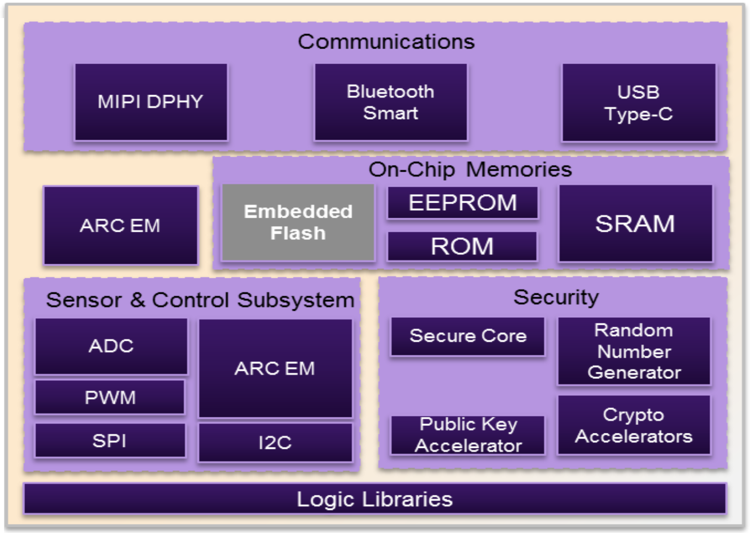
Sensor & Control Subsystem: ADC, PWM, SPI, ARC EM Processor, I²C

Security: Random Number Generator, Crypto Accelerators

Logic Libraries

- Bare Metal
- 180nm some 130/90nm

Low-End Edge Device



Communications: MIPI DPHY, Bluetooth Smart, USB Type-C

ARC EM

On-Chip Memories: Embedded Flash, EEPROM, ROM, SRAM

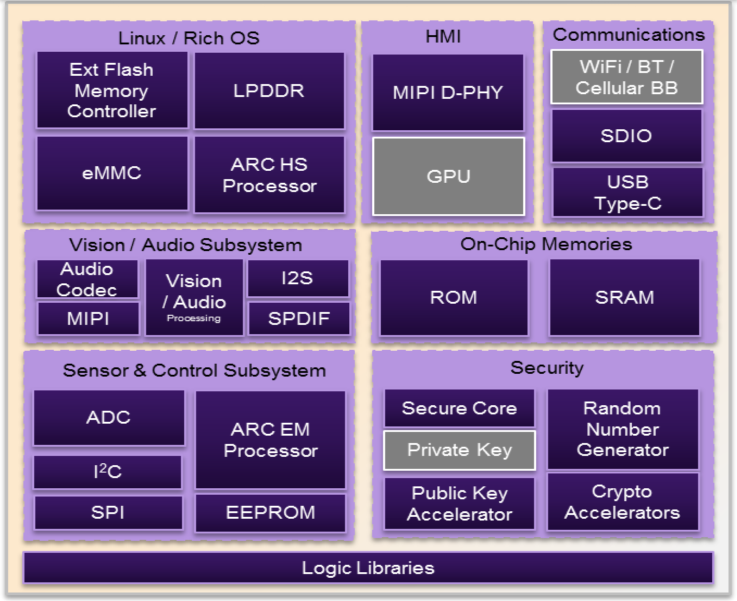
Sensor & Control Subsystem: ADC, PWM, SPI, ARC EM, I2C

Security: Secure Core, Random Number Generator, Public Key Accelerator, Crypto Accelerators

Logic Libraries

- RTOS: FreeRTOS, Zephyr, Rocket, Contiki….
- Integrated Radio
- 90nm → 55nm & 40nm (0.9v)

High-End Edge Device



Linux / Rich OS: Ext Flash Memory Controller, LPDDR, eMMC, ARC HS Processor

HMI: MIPI D-PHY, GPU

Communications: WiFi / BT / Cellular BB, SDIO, USB Type-C

Vision / Audio Subsystem: Audio Codec, MIPI, Vision / Audio Processing, I2S, SPDIF

On-Chip Memories: ROM, SRAM

Sensor & Control Subsystem: ADC, I²C, SPI, ARC EM Processor, EEPROM

Security: Secure Core, Private Key, Public Key Accelerator, Random Number Generator, Crypto Accelerators

Logic Libraries

- Linux, Android
- 65nm → to 28nm-16nm

IoT Innovation Driving Design Requirements

Sensor Processing

How much processing is needed to balance cost & performance?

Wireless Connectivity

What de-facto standards will emerge?

Security

Pervasive security needed but what exactly is required?

Energy Efficiency

Add processing, connectivity & security while extending battery life

Successful strategy for IoT needs to address all of these requirements

Agenda

IoT applications

Sensor processing and voice / audio

Wireless connectivity

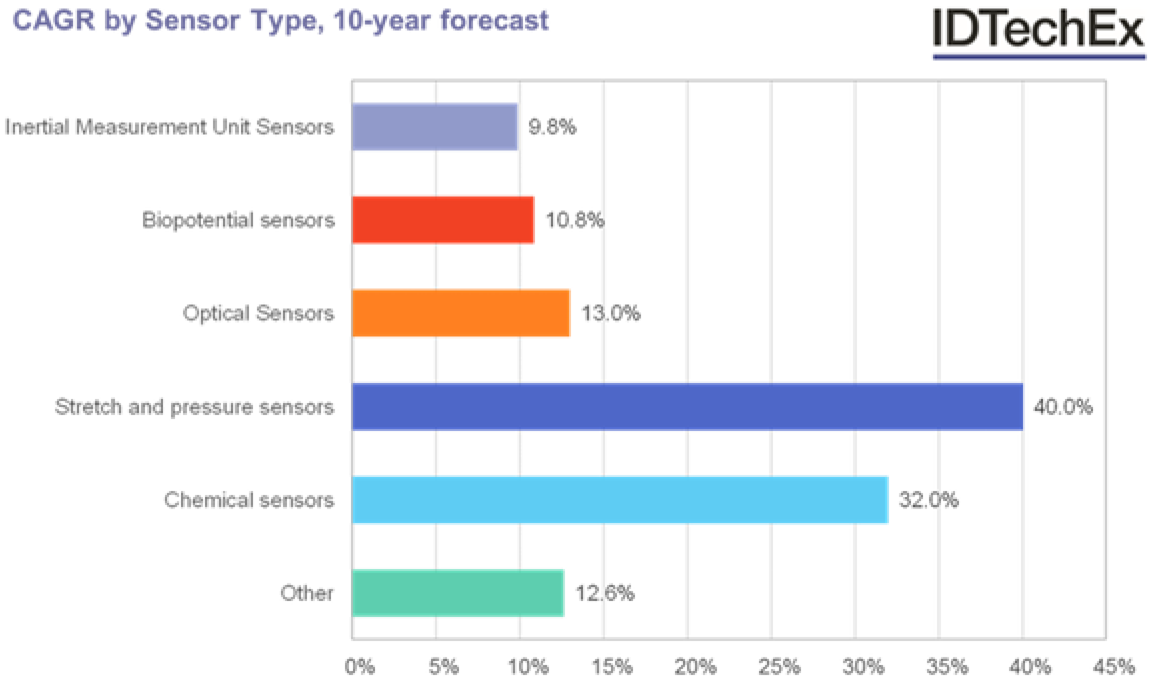
Security

Conclusion

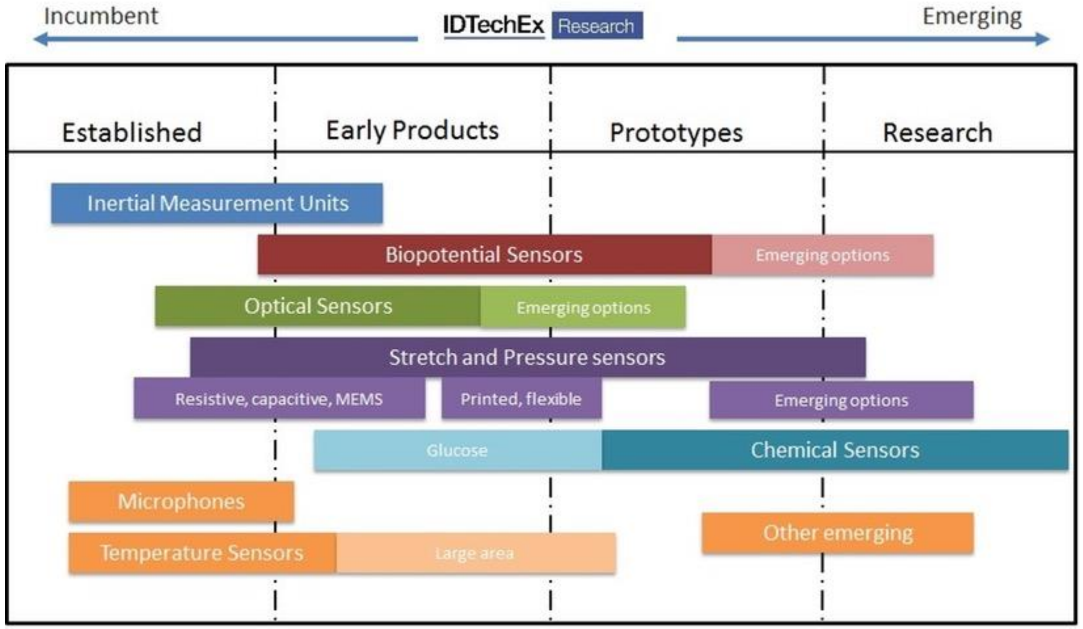
Wearable Sensors

Future Sensors Market Estimations

- “The market for wearable sensors will reach $6.1bn by 2026”



SOURCE: The IDTechEx Research report - “Wearable Sensors 2016-2026: Market Forecasts, Technologies, Players”

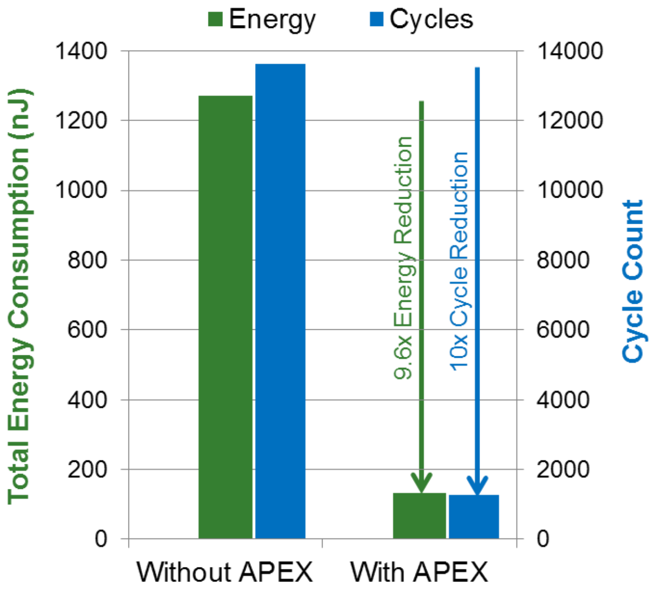


For more information, see the IDTechEx Research report:
Wearable Sensors 2015-2025: Market Forecasts, Technologies, Players
(www.IDTechEx.com/WTSensors)

Low-power DSP for IoT

Key processor features

- Energy efficiency
  - Low power consumption → low µW/MHz
  - High cycle efficiency → low MHz
- Key processor features
  - Processor architecture / ISA
    - For low power consumption and high cycle efficiency for targeted application
    - For small code size
  - Memory architecture
    - For reducing accesses to instruction and data memories
  - Configurability
    - Allow hardware features to be (de-)configured
    - For best balance of efficiency and area / power
  - Extensibility
    - Allow extension with customer-specific instructions
    - For high cycle efficiency

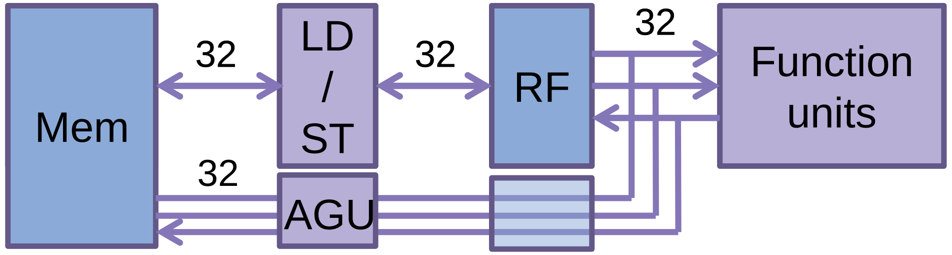


XY for higher DSP performance

And lower I-memory power

C source code

```
q31_t foo(__xy q31_t *b, __xy q31_t *c) {
   q31_t s = 0;
   for (i = 0; i <  N; i++)
      s += b[i] * c[i]
   return s;
}
```



	Non-XY	XY
Performance [MAC/cycle]	0.3	1.0
I-power [B/MAC]	12	4

Key benefits of XY:

- Higher DSP performance
- Lower I-memory power
- Smaller code size

Non-XY assembly

```
          // loop, unrolled 2x
          LP lpend
          LD %r2, [%r0, 4]
          LD %r3, [%r1, 4]
          LD %r4, [%r0, 4]
          LD %r5, [%r1, 4]
          MAC 0, %r2, %r3
          MAC 0, %r4, %r5
lpend: // epilogue for odd sized loops
          …
```

3x performance

XY assembly

```
          // prologue, set-up address gen
          SR ...
          SR ...
          LP lpend
          MAC 0, %agu_u0, %agu_u1
lpend: // no epilogue
          …
```

XY and multi-issue architectures

Code size and I-memory power

C source code

```
q31_t foo(__xy q31_t *b, __xy q31_t *c) {
   q31_t s = 0;
   for (i = 0; i <  N; i++)
      s += b[i] * c[i]
   return s;
}
```

	Non-XY	XY	Multi-issue
Performance [MAC/cycle]	0.3	1.0	1.0
I-power [B/MAC]	12	4	8

XY assembly

```
          // prologue, set-up address gen
          SR ...
          SR ...
          LP lpend
          MAC 0, %agu_u0, %agu_u1
lpend: // no epilogue
          …
```

- Lower I-memory power
- Smaller code size

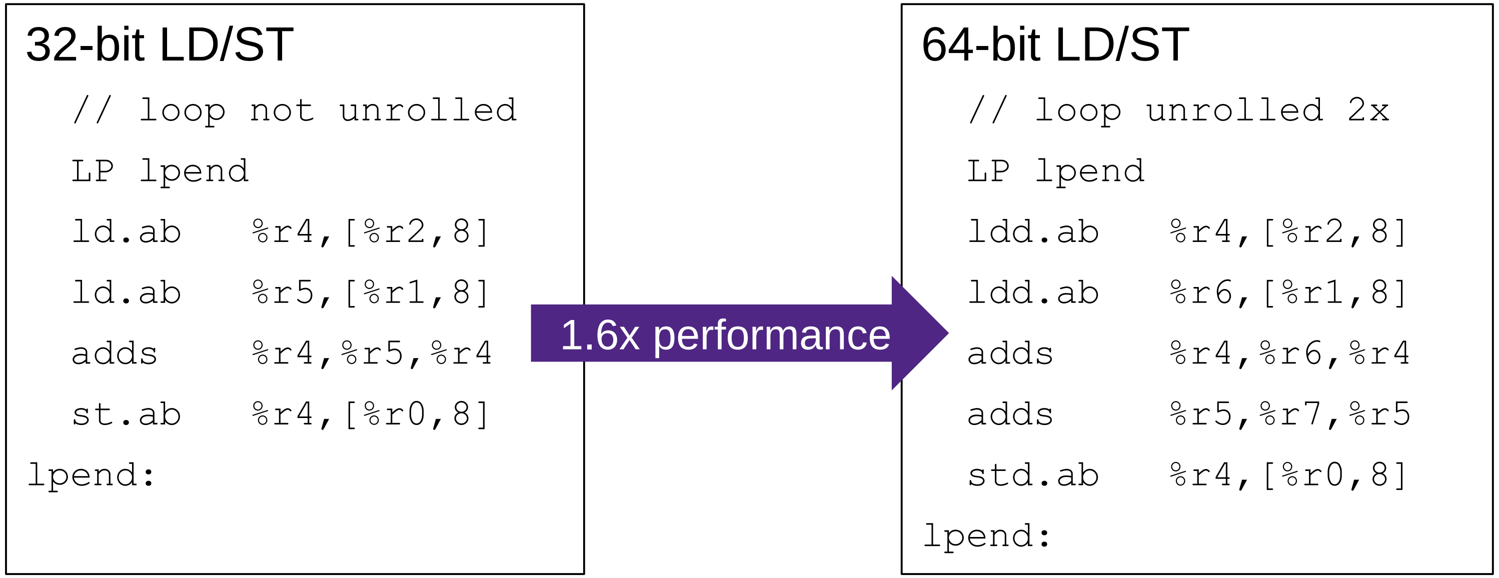
Multi-issue assembly

```
          // Prologue SW pipelined
          LDD %r2, [%r0, 8]  ; 64b vector load
          LDD %r4, [%r1, 8]  ; 64b vector load
          LDD %r6, [%r0, 8]  ; 64b vector load
          // 4x unrolled loop
          LP lpend
          {MAC 0, %r2, %r4; LDD %r8, [%r0, 8]}  ; 32b MAC and 64b load
          {MAC 0, %r3, %r5; LDD %r2, [%r1, 8]}  ; 32b MAC and 64b load
          {MAC 0, %r6, %r8; LDD %r4, [%r0, 8]}  ; 32b MAC and 64b load
          {MAC 0, %r7, %r9; LDD %r6, [%r1, 8]}  ; 32b MAC and 64b load
lpend:    // epilogue if loop count not multiple of 4
          …
```

Reducing memory accesses

Wide memories for reducing I&D memory power

- Wide instruction memory
  - Fetch multiple instructions at once → no need to fetch instruction every cycle
  - Widening I-memory by 2x reduces I-memory power by 30-40% for many applications
- Wide data memory
  - For example, combine 32-bit math with 64-bit LD/ST
  - Compiler performs LD/ST widening



32-bit LD/ST

```
  // loop not unrolled
  LP lpend
  ld.ab    %r4,[%r2,8]
  ld.ab    %r5,[%r1,8]
  adds     %r4,%r5,%r4
  st.ab    %r4,[%r0,8]
lpend:
```

1.6x performance

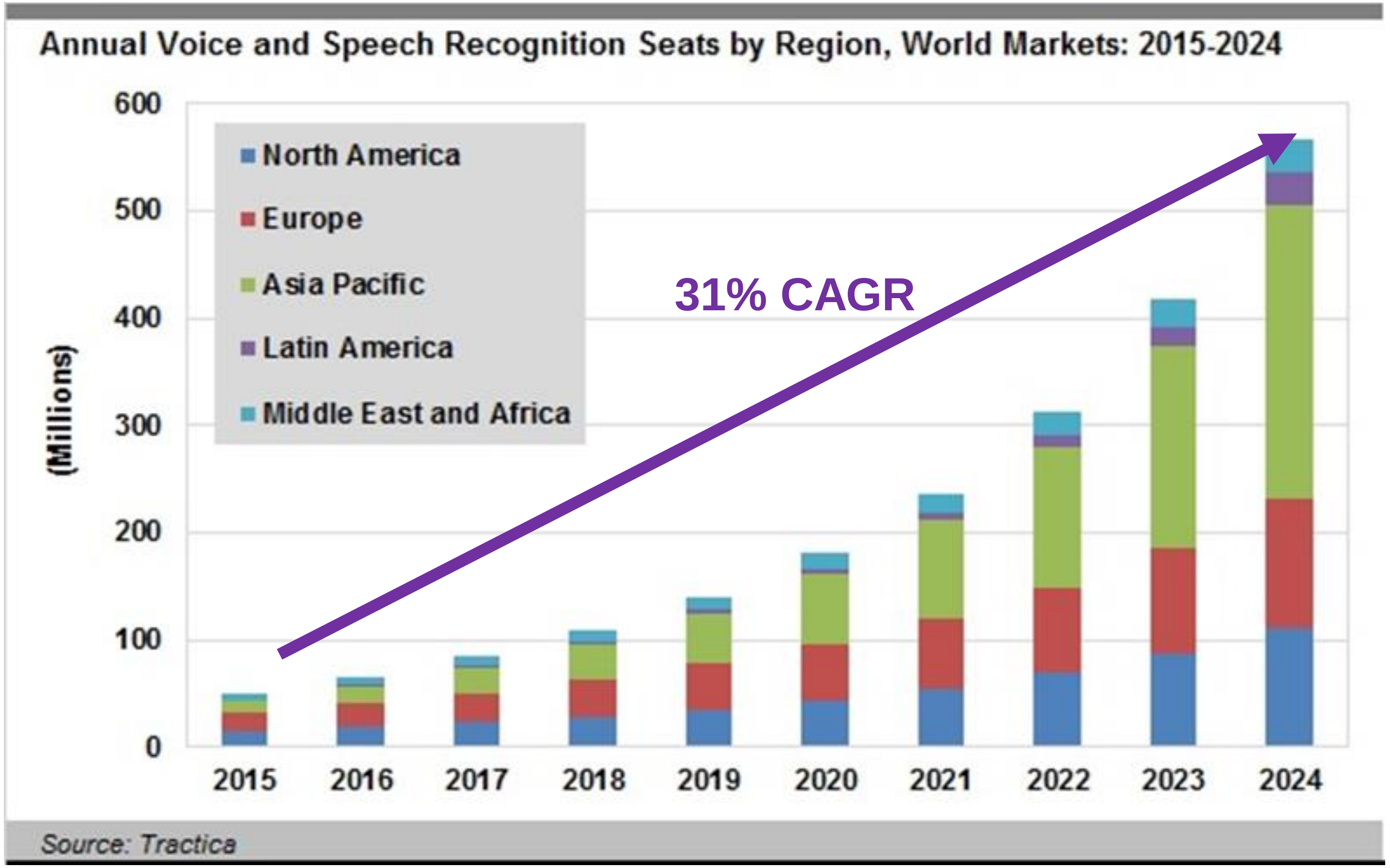
64-bit LD/ST

```
  // loop unrolled 2x
  LP lpend
  ldd.ab    %r4,[%r2,8]
  ldd.ab    %r6,[%r1,8]
  adds      %r4,%r6,%r4
  adds      %r5,%r7,%r5
  std.ab    %r4,[%r0,8]
lpend:
```

- LD/ST widening
  - Loop unrolling
  - Vectorization of LD/ST
  - To use 64-bit LDD/STD
- Higher performance
  - Reduces cycle count
- Fewer memory accesses
  - Reduces memory power

Market Update

Rapid Adoption of Natural Voice Based Human/Machine Interfaces (HMI)



Annual Voice and Speech Recognition Seats by Region, World Markets: 2015-2024

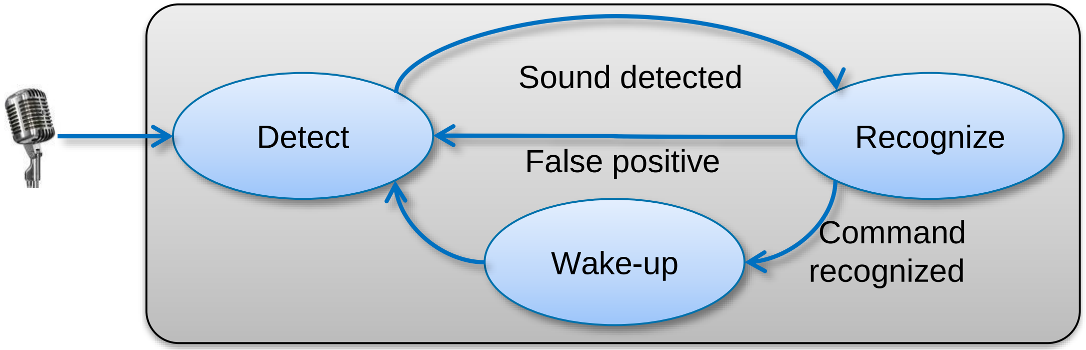
Source: Tractica

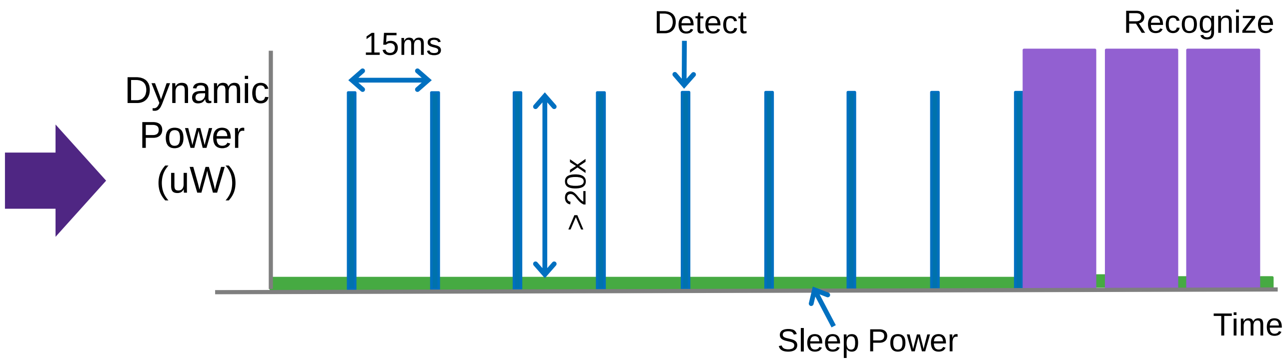
Microsoft

Voice Activation

Sensory TrulyHandsfree^{TM} on ARC EM5D

Example of always-on application: always listening





ARC EM5D TSMC 28nm HPM process	**Frequency Requirement**	**Power Consumption**
Detection Mode *Sensory LPSD function, 16 kHz microphone input*	**0.26 MHz**	**0.9 µW***
Recognition Mode *100% (full cycle) recognition, 16 kHz microphone input*	**7.6 MHz**	**40 µW***

** Logic dynamic power, gate-level simulation with post-layout RC*

Agenda

IoT applications

Sensor processing and voice / audio

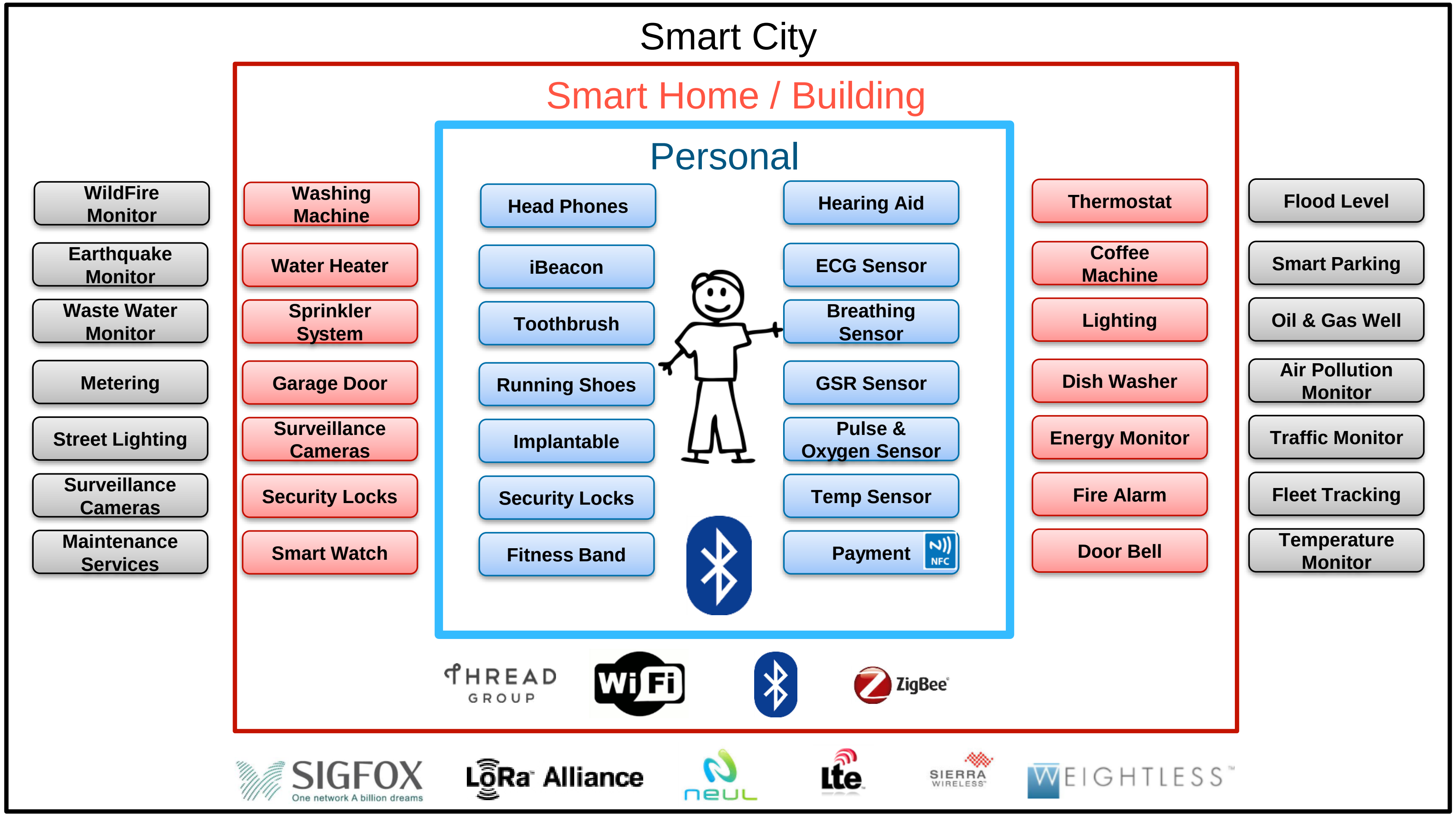
Wireless connectivity

Security

Conclusion

Communications for the Internet of Things

Many applications and protocols



Wireless Market Trends

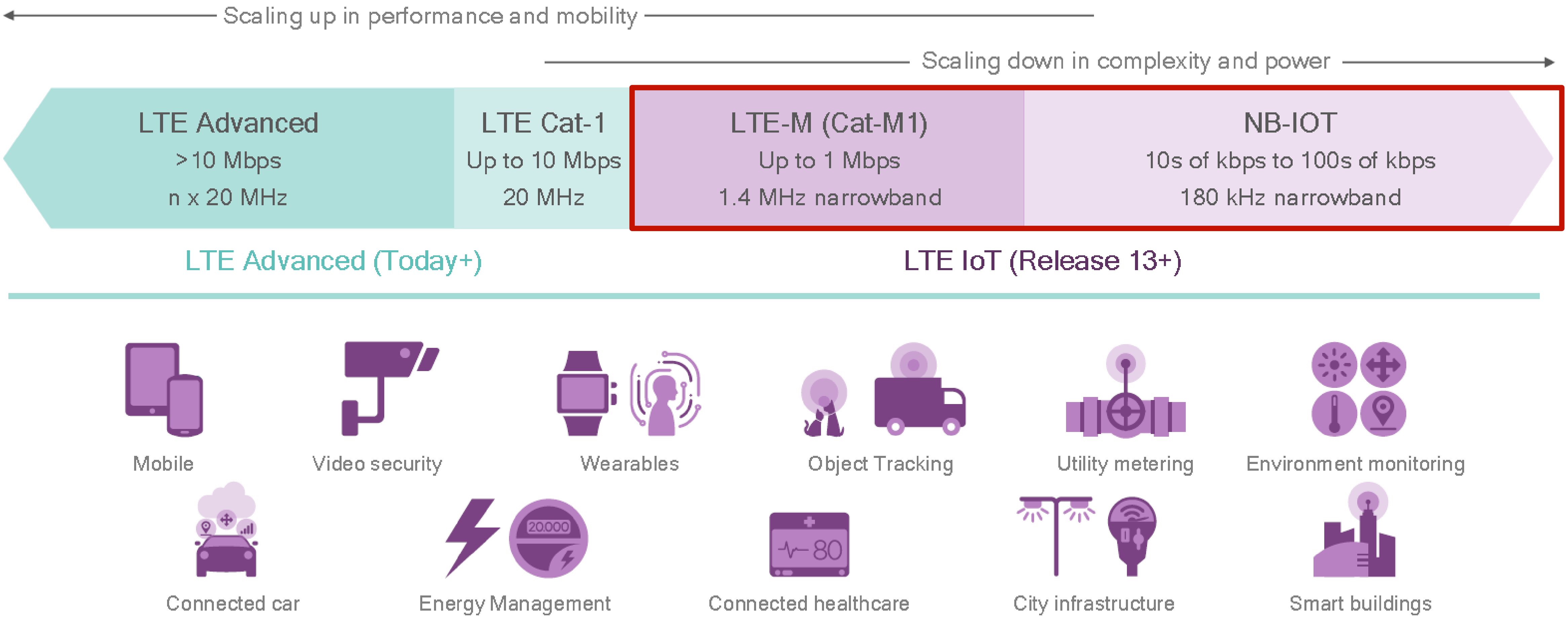
- **5G cellular (ITU IMT-2020)**
  - Support >1 Gbps data rates for mobile devices
  - Support large numbers of connected devices
  - Standard to be finalized
  - Huge investment in know-how, hardware (processor + accelerators) and software

- **Low Power Wide Area Networks**
  - Sigfox: 1st mover, adopted by Samsung, Atmel (ATA8520)
  - LoRa Alliance: Semtech developed radio, supported by 11 cellular providers, Cisco, IBM, Microchip, FSL
  - Weightless: Huawei acquired Neul

- **LTE-M and NB-IoT**
  - Low data rate (<1Mbps) / low power / low cost cellular IoT
  - Builds on installed cellular network infrastructure
  - Supports long-range communication and mobility
  - Automotive, utility meters, tele-health, tracking, vending machines, etc.

Communications for the Internet of Things

Many applications with broad range of data rates and performance requirements



Scaling up in performance and mobility

Scaling down in complexity and power

LTE Advanced
>10 Mbps
n x 20 MHz

LTE Cat-1
Up to 10 Mbps
20 MHz

LTE-M (Cat-M1)
Up to 1 Mbps
1.4 MHz narrowband

NB-IOT
10s of kbps to 100s of kbps
180 kHz narrowband

LTE Advanced (Today+)

LTE IoT (Release 13+)

Source: Qualcomm

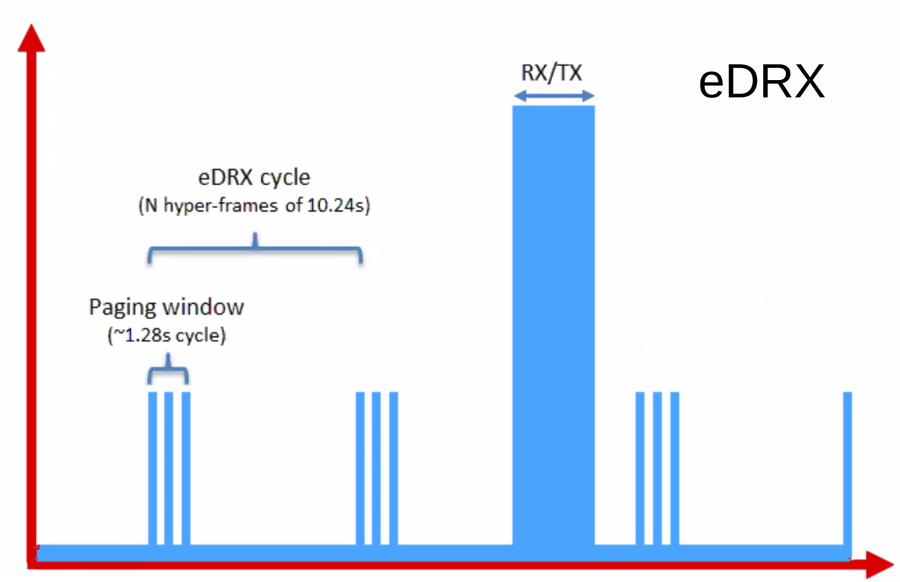
LTE for IoT

Reduced complexity

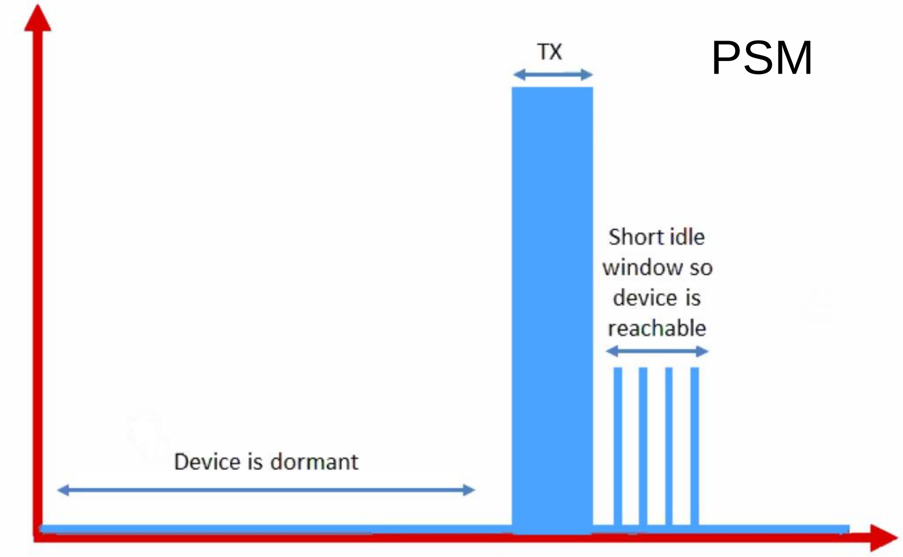
Feature	Cat. 1 (Rel. 8+)	Cat. M1 (Rel. 13)	Cat. NB1 (Rel. 13)	eNB-IOT (Rel. 14)
Bandwidth	20 MHz	1.4 MHz	180 kHz	180 kHz
Deployments	LTE channel	Standalone, in LTE channel	Standalone, in LTE channel, LTE guard bands	Standalone, in LTE channel, LTE guard bands
Full / half duplex	Full duplex (no HD-FDD)	HD-FDD preferred	HD-FDD	HD-FDD preferred
Data rates (peak)	DL: 10 Mbps, UL: 5 Mbps	~800 kbps (FD-FDD) 300/375 kbps DL/UL (HD-FDD)	Less that 100 kbps	Higher than Cat. NB1
Latency	< 1s	~ 5s	<10s	At least the same as Cat. NB1
Coverage	Standard LTE coverage	Improved coverage	Deep coverage to work in cellars	Deep coverage to work in cellars
Mobility	Seamless	Seamless	Connections get dropped on base station switch	More mobility than Cat. NB1
Voice	Yes	Yes	No	No
FEC	Turbo (DL + UL) Viterbi (DL + UL)	Turbo (DL + UL) Viterbi (DL + UL)	Viterbi (DL) Turbo / repetition (UL)	
Encryption	AES-128, SNOW 3G	AES-128, SNOW 3G	AES-128, SNOW 3G	
Power saving	DRX	eDRX, PSM	eDRX, PSM	eDRX, PSM

LTE power saving features

- Extended Discontinuous Reception (eDRX)
  - DRX sleep period of up to 10.24s
  - eDRX allows device to sleep for multiple periods of 10.24s
  - Cat.M1: up to ~40 minutes of extended sleep
  - Cat.NB1: up to ~4 hours of extended sleep

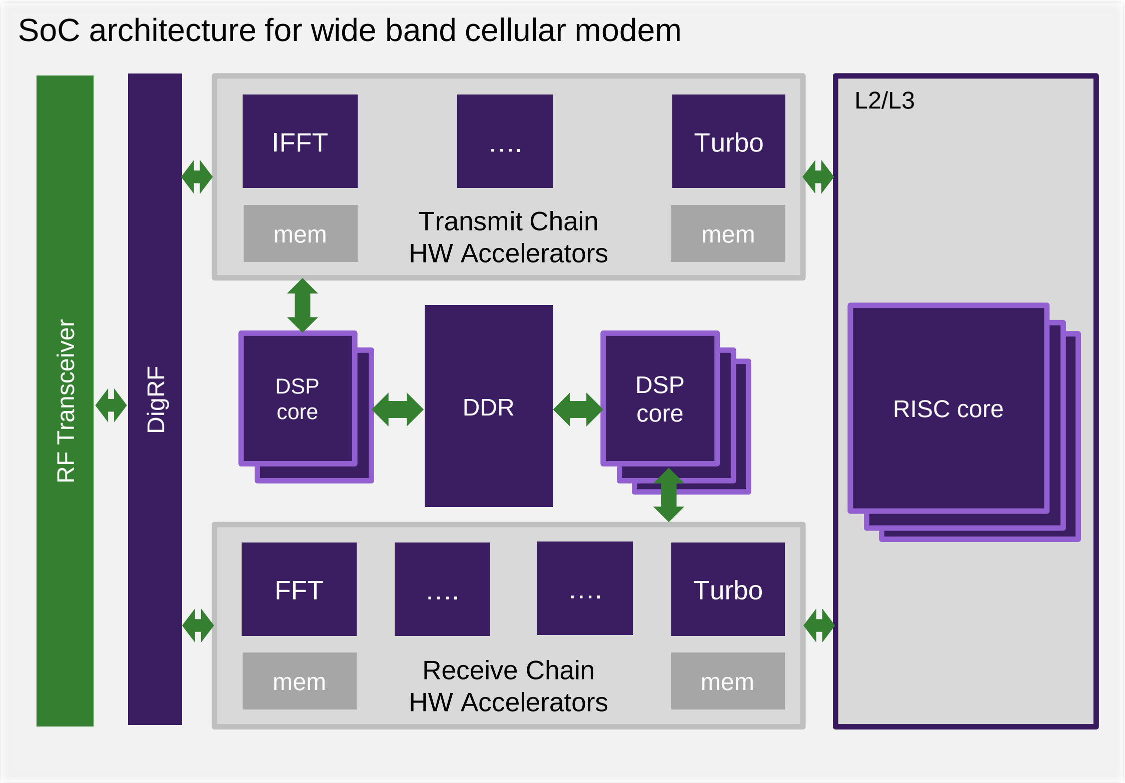


- Power Saving Mode (PSM)
  - User Equipment (UE) can decide to go dormant indefinitely
  - When UE decides to wake up it transmits to the network
  - UE remains in RX mode for 4 idle frames to receive a reply



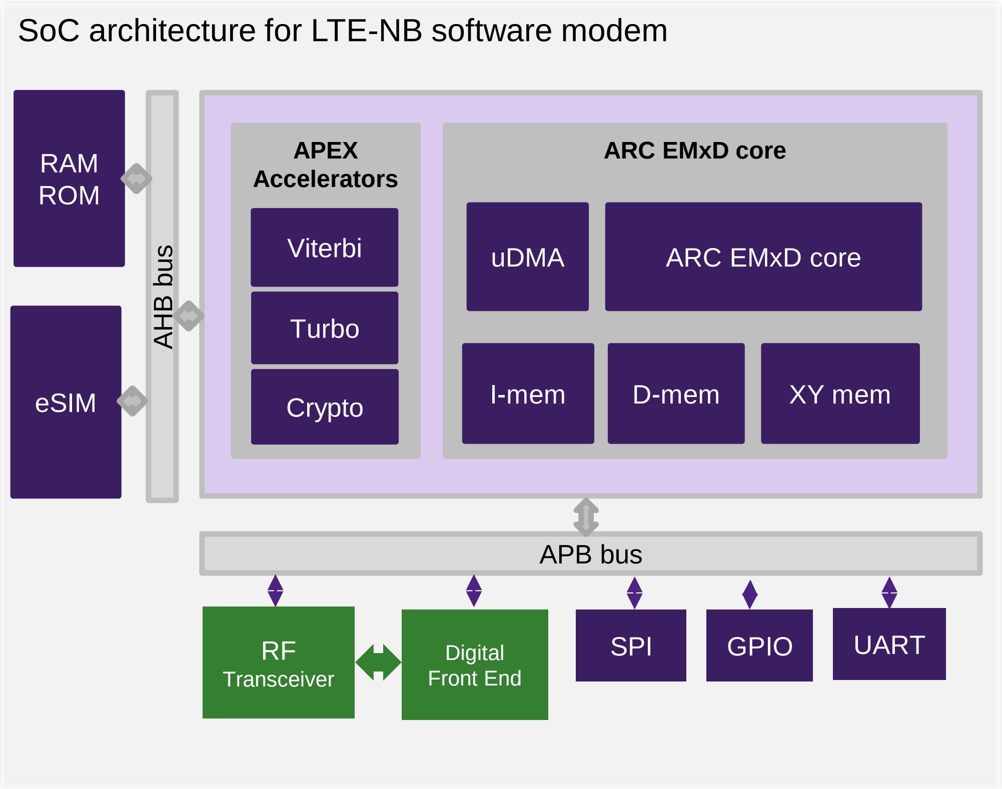
LTE-NB modem architecture

RISC+DSP based SoC with HW accelerators



SoC architecture for wide band cellular modem
RF Transceiver
DigRF
IFFT
….
Turbo
mem
Transmit Chain HW Accelerators
mem
DSP core
DDR
DSP core
L2/L3
RISC core
FFT
….
….
Turbo
mem
Receive Chain HW Accelerators
mem

ARC EMxD with extensions



Hardware Acceleration	Pure Software
Longer development cycle -> slower time to market	Shorter development cycle -> faster time to market
Limited flexibility for changing standards	High flexibility for changing standards
Data transfer overhead, complex synchronization	Simple software synchronization, optimal data exchange
Limited multi standard support	Software standard implementation – can be switched on the fly

Viterbi processor extension

Code example decoder

```
void viterbi_decode(_xy int32_vec3x8b sample[], _xy int32 result[], int frame) {
 // sample: array of 3*8b components of input sample
 // result: decoded result
 // frame:  number of bits in the frame (multiple of 32)
 // store path metric decision bits
 _xy int32 decisions[frame*2];  // infer XY address generation
 // path metric reset, single cycle
 vitrst();
 // compute path metrics and decision bits, two cycles per bit
 for (i = 0; i < frame; i++) {
     decisions[2*i]   = vitacc0(sample[i]);
     decisions[2*i+1] = vitacc1(0);
 }
 // traceback, one cycle per bit
 i = 0; j = 0;
 while (i < frame) {
    for(k = 0; k < 31; k++) {
        vittb(decisions[2*i], decisions[2*i+1]);
        i++;
    }
    result[j++] = vittb(decisions[2*i], decisions[2*i+1]);
    i++;
 }
}
```

```
…
721924:    000002a8 386f4001|vitacc1  agu_u0,0
721925:    000002a4 382f4840|vitacc0  agu_u0,agu_u1
721926:    000002a8 386f4001|vitacc1  agu_u0,0
721927:    000002a4 382f4840|vitacc0  agu_u0,agu_u1
721928:    000002a8 386f4001|vitacc1  agu_u0,0
…

…
928719:    000002ec 3900483e|vittb    0,agu_u1,agu_u0
928720:    000002f0 3900483e|vittb    0,agu_u1,agu_u0
928721:    000002f4 3900483e|vittb    0,agu_u1,agu_u0
928722:    000002f8 3900483e|vittb    0,agu_u1,agu_u0
928723:    000002fc 3900483e|vittb    0,agu_u1,agu_u0
928724:    00000300 3900483e|vittb    0,agu_u1,agu_u0
928725:    00000304 3900483e|vittb    0,agu_u1,agu_u0
928726:    00000308 39004822|vittb    agu_u2,agu_u1,agu_u0
928727:    000002ec 3900483e|vittb    0,agu_u1,agu_u0
…
```

Viterbi processor extension

MHz requirements for LTE-NB

Viterbi decoding	
Path metrics [cycles/bit]	2
Traceback [cycles/bit]	1
Total [cycles/bit]	3
Overhead for tailbiting [%]	50
Overhead per frame [cycles/frame]	10
Total including overhead [cycles/bit]	~5

Standard	Peak data rate	MHz requirements ARC EMxD
Cat.NB1	100 kbps	0.5 MHz
Cat.M1	800 kbps	4 MHz

- Low MHz requirements
- Small code size
- Easy software integration
- No need to move data to/from accelerator

Key features Viterbi processor extension

- Supports constraint lengths K in range [5, 12]
- Supports base coding rates ½, ⅓, ¼; other rates supported through (de)puncturing
- Supports tail-biting
- No need to have hardware accelerator on shared interconnect
  - With associated overheads of programming accelerators and moving data to/from accelerators
  - Instead have custom instructions as extensions to EMxD core

Agenda

IoT applications

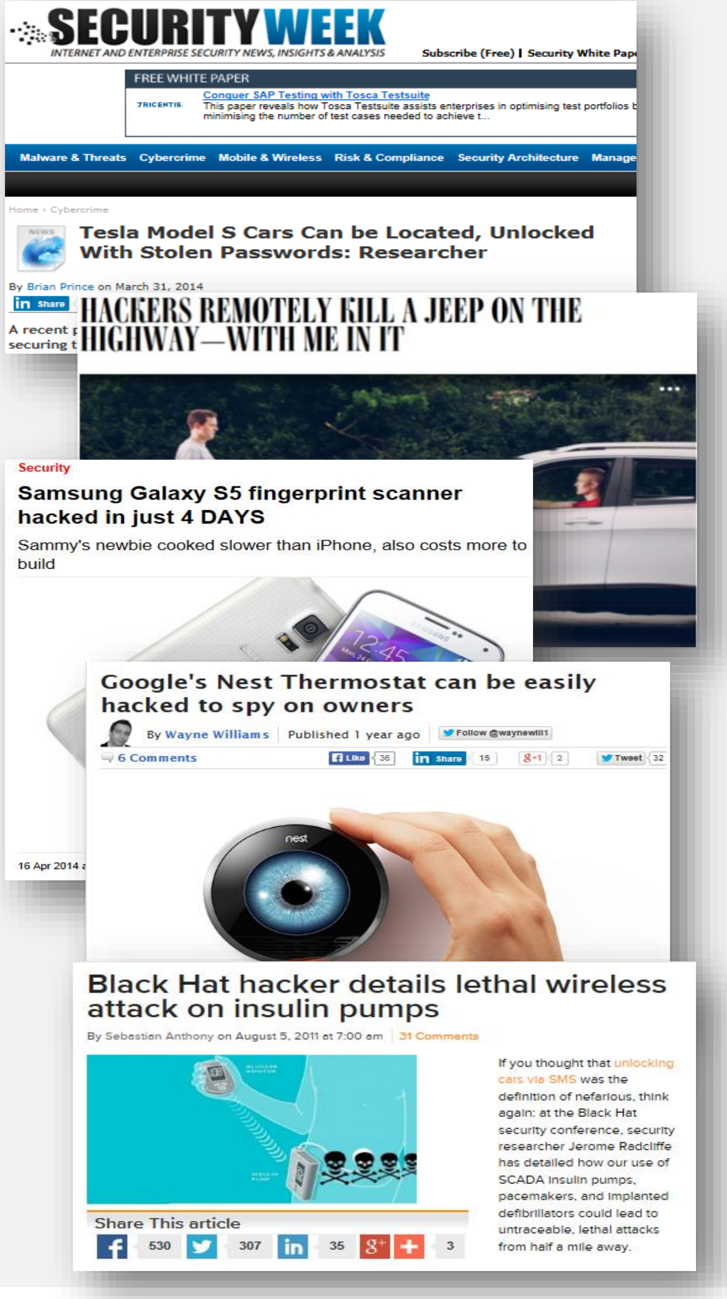
Sensor processing and voice / audio

Wireless connectivity

Security

Conclusion

Attacks on the Rise & Evolving



SECURITYWEEK
INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS
Subscribe (Free) | Security White Pap

FREE WHITE PAPER
TRICENTIS. Conquer SAP Testing with Tosca Testsuite
This paper reveals how Tosca Testsuite assists enterprises in optimising test portfolios b minimising the number of test cases needed to achieve t...

Malware & Threats | Cybercrime | Mobile & Wireless | Risk & Compliance | Security Architecture | Manage

Home › Cybercrime

Tesla Model S Cars Can be Located, Unlocked With Stolen Passwords: Researcher

By Brian Prince on March 31, 2014

A recent p
securing t

HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

Security

Samsung Galaxy S5 fingerprint scanner hacked in just 4 DAYS

Sammy's newbie cooked slower than iPhone, also costs more to build

16 Apr 2014 a

Google's Nest Thermostat can be easily hacked to spy on owners

By Wayne Williams | Published 1 year ago | Follow @waynewill1

6 Comments | Like 36 | Share 15 | +1 2 | Tweet 32

Black Hat hacker details lethal wireless attack on insulin pumps

By Sebastian Anthony on August 5, 2011 at 7:00 am | 31 Comments

If you thought that unlocking cars via SMS was the definition of nefarious, think again: at the Black Hat security conference, security researcher Jerome Radcliffe has detailed how our use of SCADA insulin pumps, pacemakers, and implanted defibrillators could lead to untraceable, lethal attacks from half a mile away.

Share This article 530 307 35 3

Who lost your data in 2015's biggest hacks, breaches ?

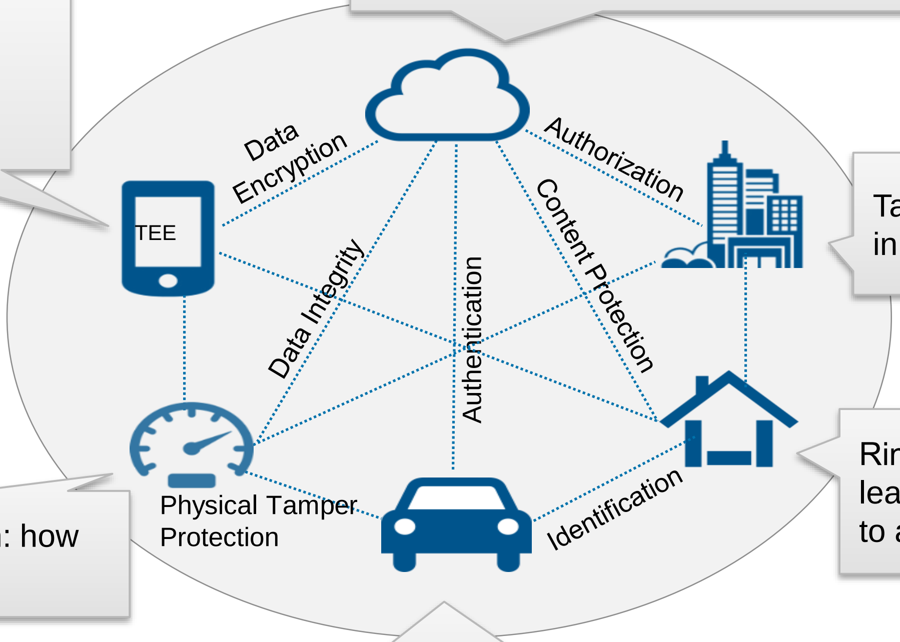
Security flaw affects nearly every Android phone with an "?" chip

Target hackers broke in via HVAC system

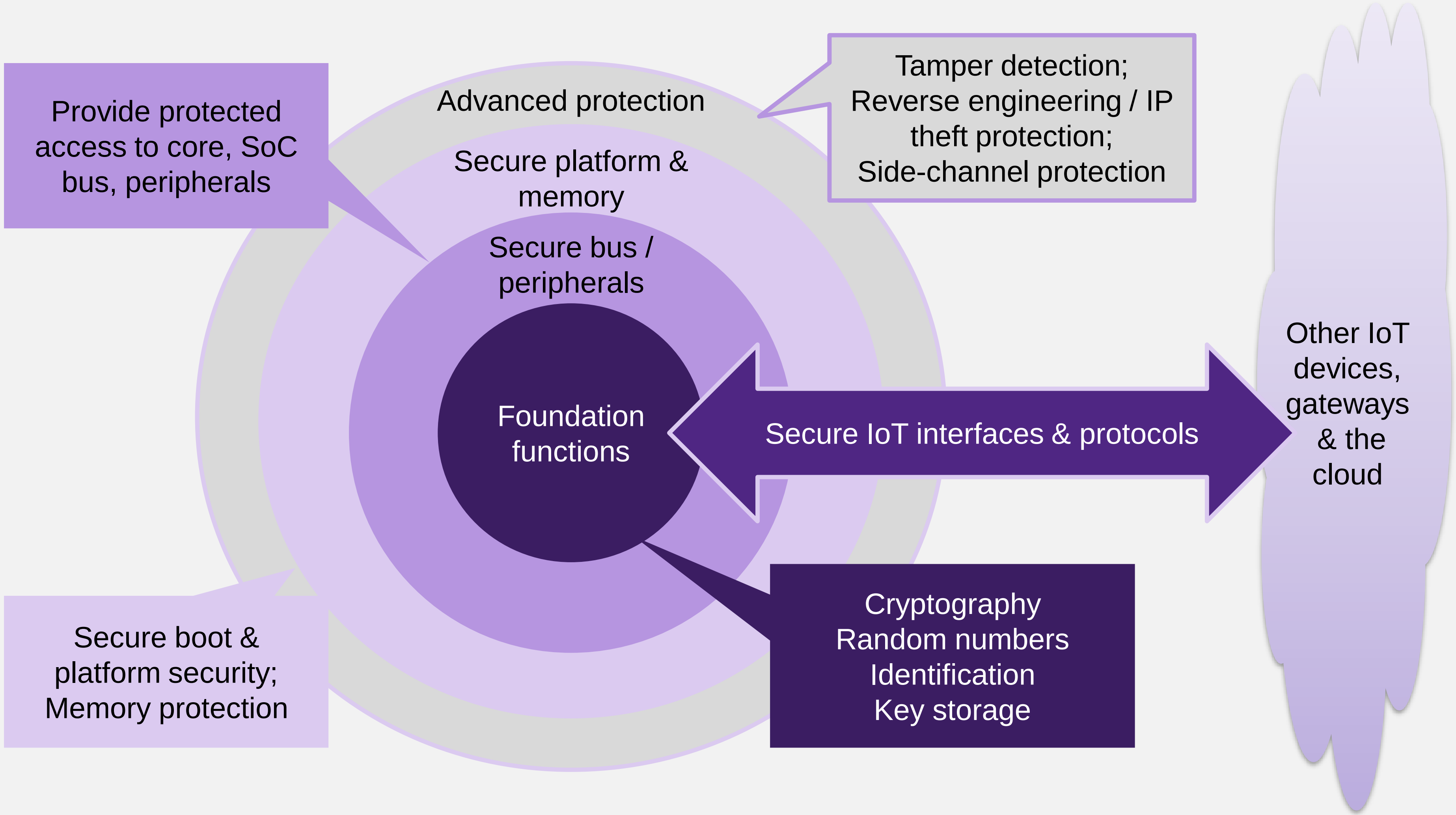
Ring's smart doorbell can leave your house vulnerable to attacks

The meter dash: how to hack parking

Remotely kill a jeep on the way to the highway – with me in it!



Embedded Security Requirements for Processors



Provide protected access to core, SoC bus, peripherals

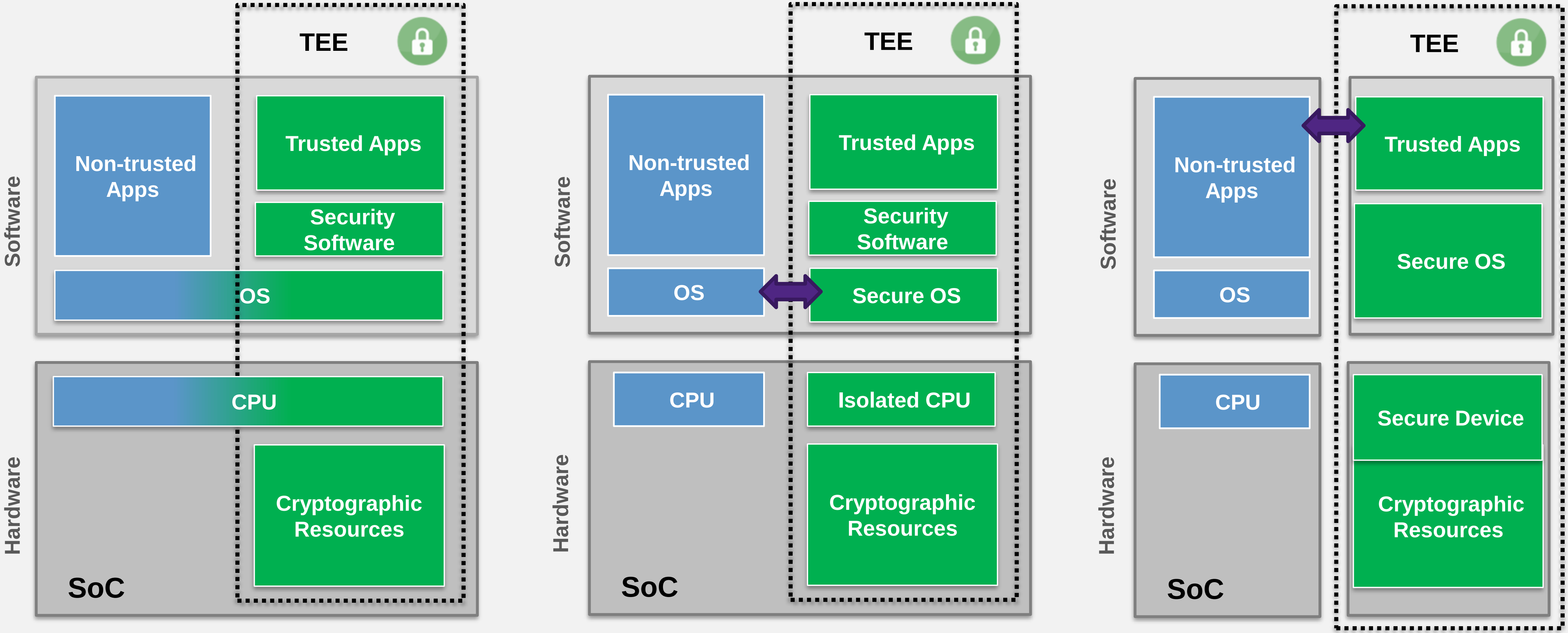
Secure boot & platform security; Memory protection

Tamper detection; Reverse engineering / IP theft protection; Side-channel protection

Cryptography
Random numbers
Identification
Key storage

Trusted Execution Environments

Ensure separation of secure processes. Various implementations.



Single CPU with
HW Separation (Memory, Bus, etc)

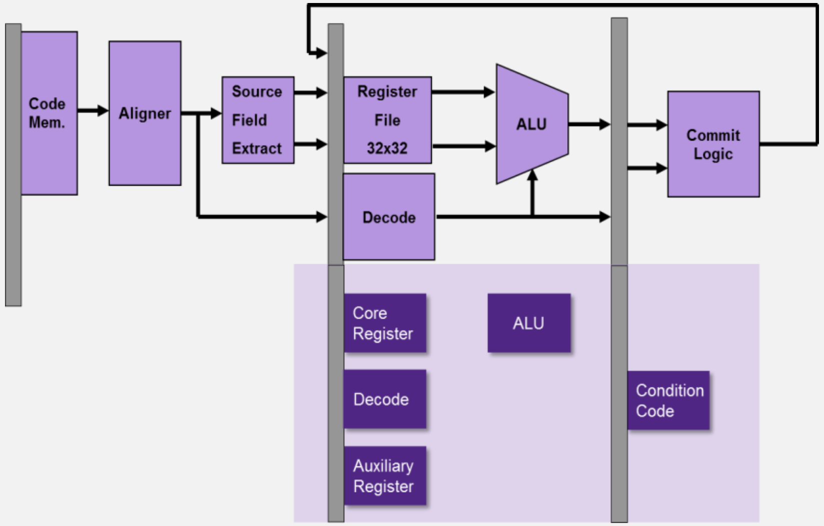
Separated CPU
on same SoC (Open/Closed)

Physically separated
Secure Device (TPM)

Cryptography Implementation Options

Crypto Software

Specialized CPU Instructions

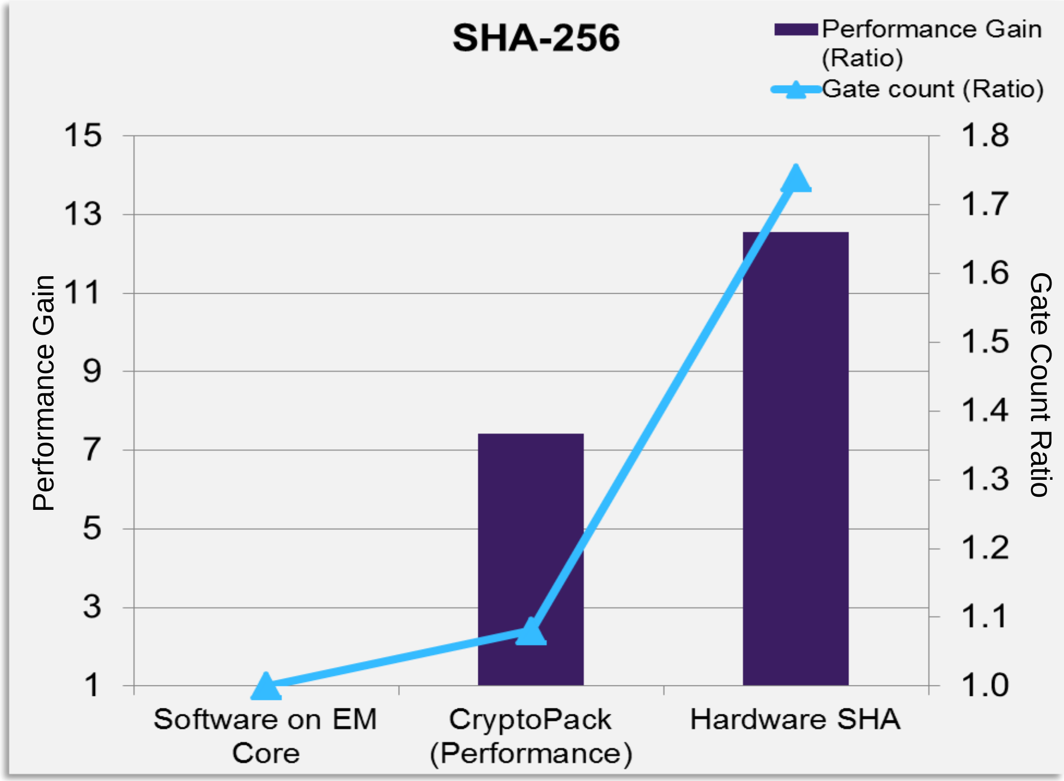


HW Crypto Cores

CryptoPack for ARC EM and SEM Processors

Hardware Extensions to Accelerate Cryptographic Algorithms

- Speeds up software implementations through tested and verified custom instructions
- Also reduces code size
- Area optimized and performance optimized versions
- Support for common crypto algorithms such as AES, 3DES, ECC, SHA-256, and RSA



CryptoPack SHA-256 increases ARC area by 8% but increases performance by 7x and reduces energy by 6.5x

Agenda

IoT applications

Sensor processing and voice / audio

Wireless connectivity

Security

Conclusion

Conclusions

- Great variety of IoT applications
  - High growth in many markets
  - Combine functionalities for sensor processing, connectivity and security
  - Constraints on power and cost will require optimized implementations

- Significant energy savings can be achieved with processors optimized for IoT applications
  - Processors architected for energy efficiency and small code size
  - Memory architecture for reducing accesses to instruction and data memories
  - Configurability for best balance of efficiency and area / power
  - Extensibility for increased cycle efficiency and reduced code size

Thank You