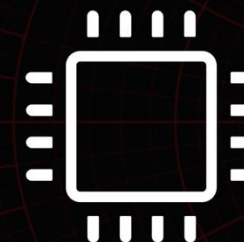


Wireless SoC Evolution

Balancing Compute, Power, Cost, and Complexity

October, 6 2026

2026
tech t|ks
WEBINAR SERIES



**WIRELESS
COMPUTE**

Agenda

- 01** Why requirements are changing
- 02** What process scaling can and cannot change
- 03** Customer decision lenses
- 04** Developer tradeoffs
- 05** Series 2, Series 3, and product fit
- 06** Takeaways and Q&A

The Real Design Question

CUSTOMERS CHOOSE OUTCOMES



Battery life



Cost & BOM



Connectivity



Security & compliance



Product lifetime

**Which
platform
fits the
product?**

DEVELOPERS MANAGE CONSTRAINTS

FIT

Code, RAM & OTA

RF

Timing, RF & coexistence

SEC

Provisioning & security

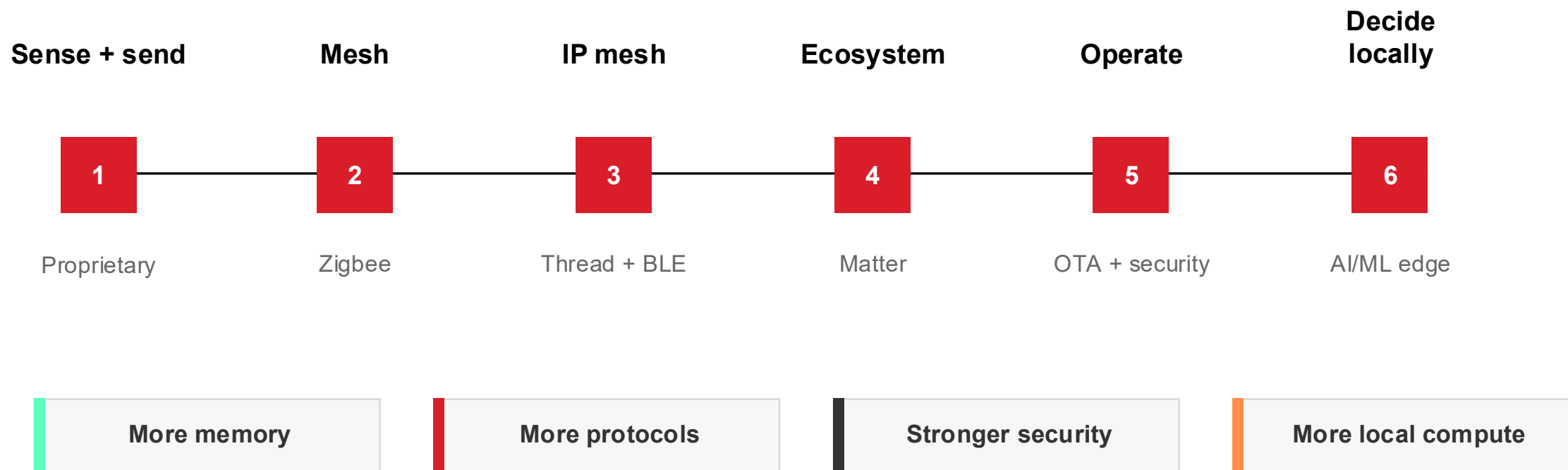
TEST

Test, debug & validation

Process node is one architectural lever - not the customer's purchase decision.

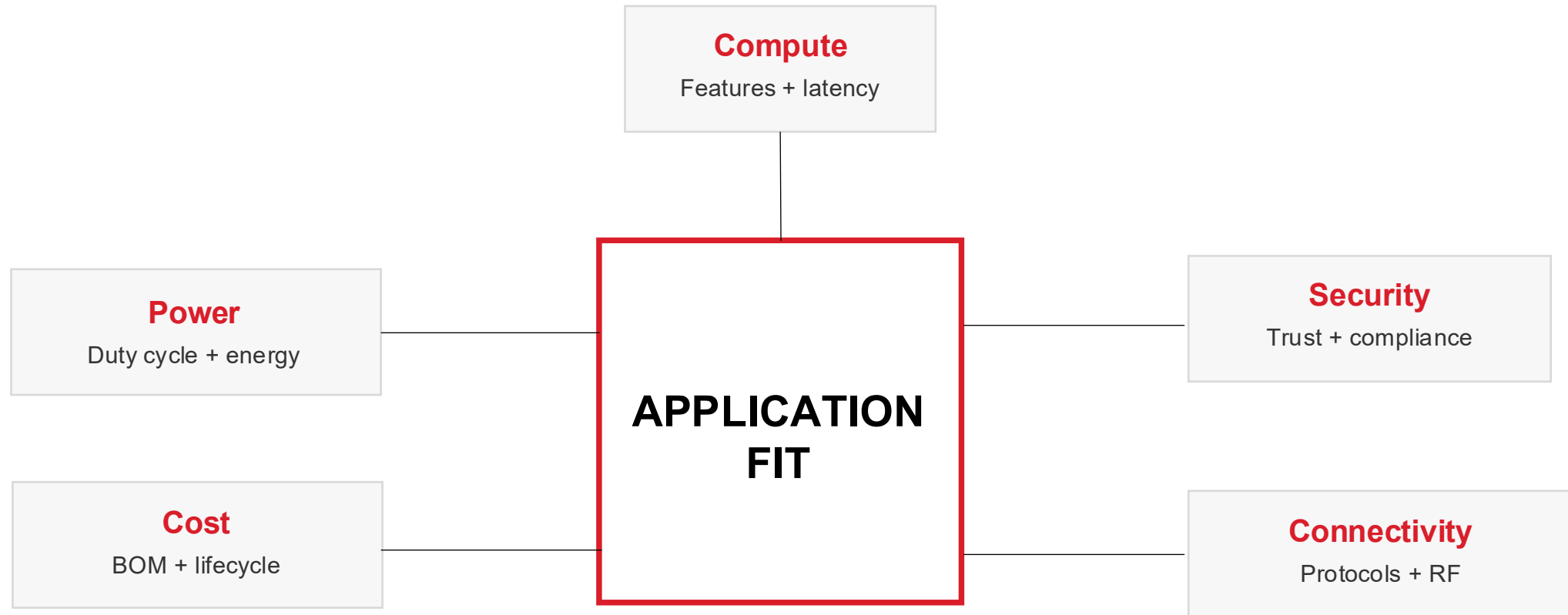
Wireless IoT requirements are broadening

The simple sensor is no longer always simple, but simple end points still matter.



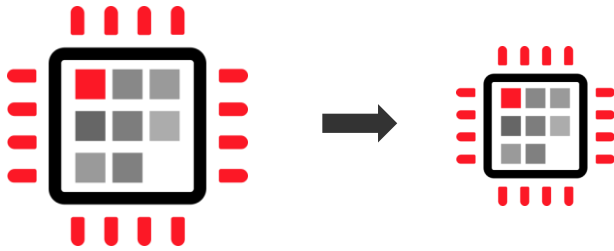
Platform requirements are expanding - not converging on one universal design point.

Every wireless product is a negotiation



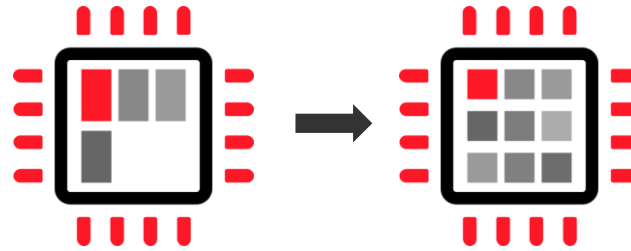
Improving one dimension can move another: more protocols consume memory; more security changes provisioning; more compute can raise peak current and software complexity.

Process scaling is not the same as product value



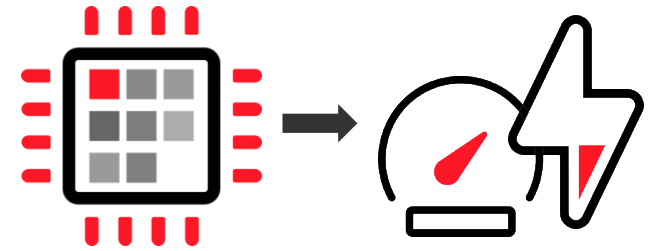
SAME FUNCTION, SMALLER DIE

Potential area or cost benefit - if the product keeps the same function.



SAME AREA, MORE CAPABILITY

Density can be spent on RAM, interfaces, security, accelerators, or redundancy.



NEW OPERATING POINT

Lower capacitance/voltage may reduce task energy, or higher frequency may shorten active time.

What smaller geometry can enable?

D

Density

More logic, memory, accelerators, or security blocks.

CPU

Compute

More performance per area for richer applications.

MEM

Memory

More headroom for Matter, OTA, logs, and models.

E

Active energy

Lower capacitance and voltage can reduce energy per operation.

INT

Integration

Potential to remove external parts or a second MCU.

ARC

Architecture

Dedicated domains, power gating, and concurrency options.

The benefit is option value not a universal guarantee of lower current, lower cost, or longer battery life.

What does not automatically get better?

DIGITAL SCALING OPPORTUNITIES

- Lower switching capacitance
- Lower-voltage operating points
- More logic and memory density
- Higher performance per area
- More power-domain options

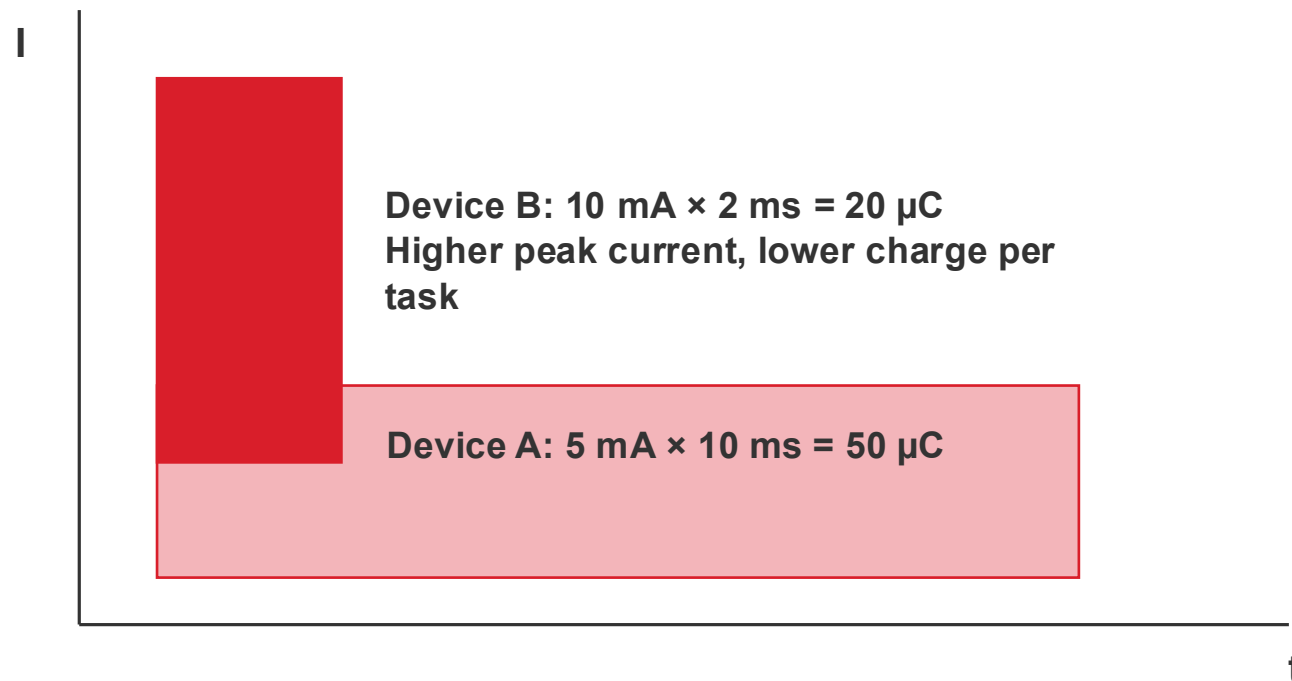
ENGINEERING TRADEOFFS

- Leakage and always-on current
- Analog/RF does not scale like logic
- SRAM, NVM, retention, and XIP choices
- Mask, IP, test, and validation costs
- More capability increases software complexity

Digital logic often scales more readily than RF, analog, nonvolatile memory, and product-validation effort.

Smaller node ≠ automatically lower current

Higher instantaneous current can still use less task energy



Illustrative, equal voltage; not product data.

Sleep-dominated products
may lose the “race”

**99.9%
asleep**

- Leakage paths
- Retention domains
- Always-on blocks
- Unexpected wakeups

$$P_{\text{dynamic}} \propto C \cdot V^2 \cdot f \quad | \quad P_{\text{static}} \approx I_{\text{leak}} \cdot V \quad | \quad E_{\text{task}} = \int V \cdot I(t) dt$$

Measure energy per task and average current across the real duty cycle, not peak current alone.

Power is workload-dependent



BATTERY SENSOR

Dominant: sleep + RF bursts

Watch leakage, polling, retries.



SMART LOCK

Dominant: compute + crypto + RF

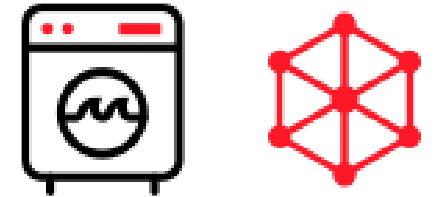
Wake time and secure operations matter.



MAINS-POWERED LIGHT

Dominant: integration + RF + thermal

Battery life is not the top constraint.

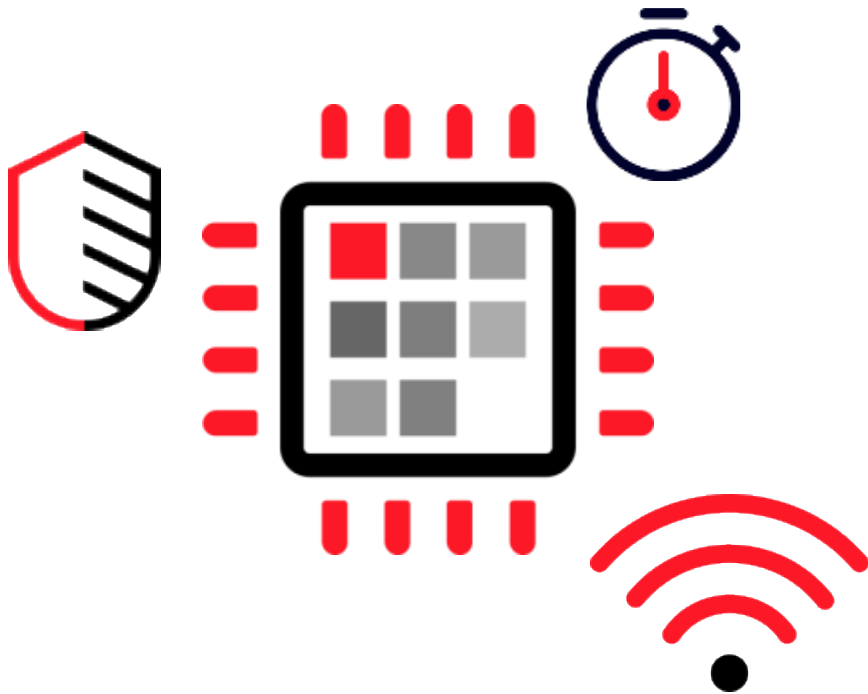


AI EDGE NODE

Dominant: inference + memory traffic

Track energy per inference and model updates.

Mature nodes still matter



Why established platforms remain valuable

- Proven RF and analog behavior
- Known cost and yield structure
- Established software ecosystem
- Well-characterized power modes
- Long lifecycle and field confidence
- Strong fit when sleep, RF, BOM, or simplicity dominate

A mature-node platform is not an inferior choice; it may be the lowest-risk and highest-value solution.

What the customer is choosing

CUSTOMER QUESTION	WHAT IT REALLY REQUIRES
Can I hit the BOM target?	SoC, module vs. chip, external components, manufacturing and test.
Will the battery last?	Sleep current, wake time, RF energy, retries, and the real duty cycle.
Can I support the required ecosystems?	Memory, protocol stacks, certification, and multiprotocol behavior.
Can I reduce SKUs?	Flexible software, dynamic/concurrent multiprotocol, and common hardware.
Can I ship and maintain it?	SDK maturity, examples, OTA strategy, security updates, and headroom.
Can I meet security requirements?	Root of trust, provisioning, secure boot, debug policy, and lifecycle controls.

Customers are comparing total product risk not MHz or nanometers in isolation.

Customer example: battery sensor or switch



Years of battery life

CUSTOMER OPTIMIZATION TARGET

Years of battery life



Optimize for

- Sleep current and wake time
- RF transmit energy and retries
- Cost, form factor, stable stack



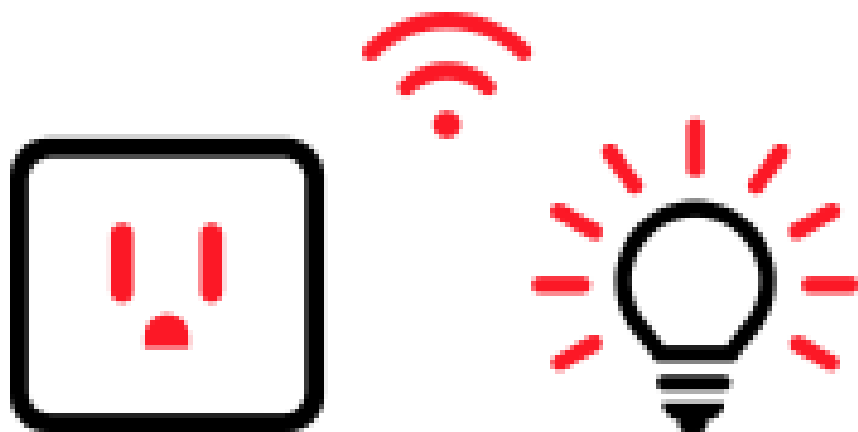
Watch for

- Leakage and retention
- Poll interval and network behavior
- OTA footprint and future margin

BEST-FIT IMPLICATION

A mature, low-leakage, highly optimized wireless SoC may be the best fit when the workload does not need extra compute or memory.

Customer example: lighting or smart plug



Integration, connectivity, reliability, and BOM

CUSTOMER OPTIMIZATION TARGET

Integration, connectivity, reliability, and BOM



Optimize for

- Zigbee, Thread, Matter, and BLE needs
- RF robustness and secure OTA
- LED/control integration and SKU reuse



Watch for

- Thermal and enclosure constraints
- Multiprotocol test matrix
- Software and certification complexity

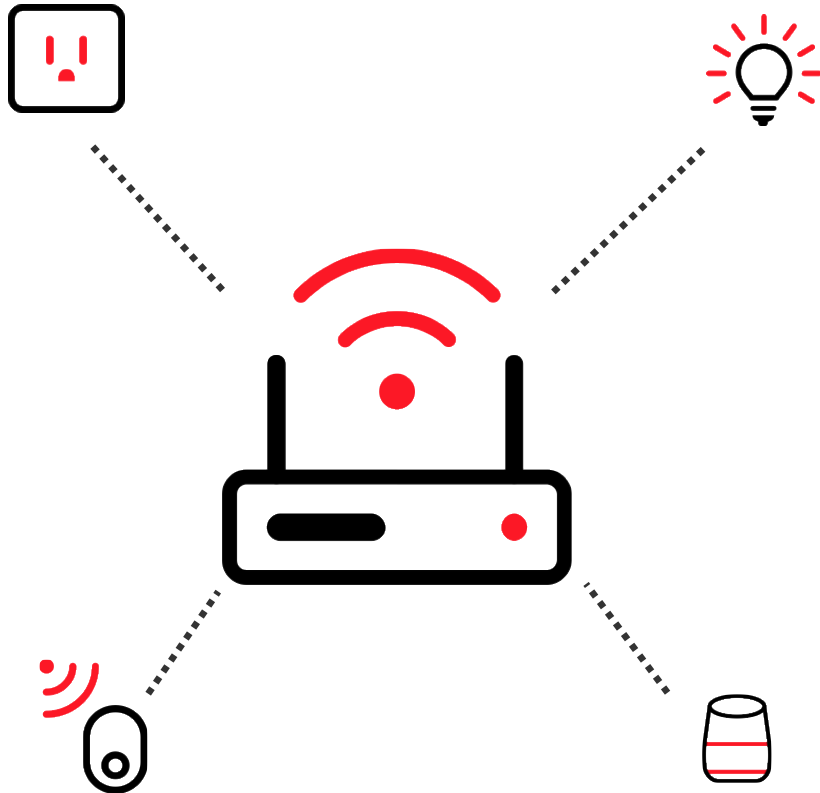
BEST-FIT IMPLICATION

A more integrated, higher-compute platform can create value when it simplifies the product, reduces external parts, or consolidates SKUs.

Silicon Labs publicly positions SiMG301 for line-powered LED lighting, plugs, and switches.

Customer example: bridge, hub, or controller

CUSTOMER VIEW



Software headroom over the product lifetime

CUSTOMER OPTIMIZATION TARGET

Software headroom over the product lifetime



Optimize for

- Matter and multiprotocol stack growth
- OTA slots, rollback, logs, diagnostics
- Security storage and certification



Watch for

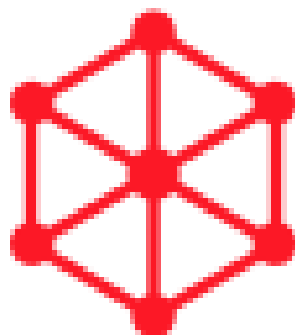
- The second and third year of updates
- Partitioning and recovery strategy
- Future features consuming the margin

BEST-FIT IMPLICATION

More memory and compute headroom can reduce lifecycle risk—but the value is future software capacity, not the node label.

Silicon Labs MG26 public page: up to 3.2 MB flash and 512 kB RAM, with headroom for Matter over Thread.

Customer example: AI-enabled edge device



Useful local intelligence per joule

CUSTOMER OPTIMIZATION TARGET

Useful local intelligence per joule



Optimize for

- Energy per inference and latency
- Model size, RAM, and sensor bandwidth
- Privacy and reduced cloud dependence



Watch for

- Always-on leakage and duty cycle
- Memory movement and model updates
- Secure OTA for firmware and models

BEST-FIT IMPLICATION

Smaller geometry can help when active compute dominates, but the entire sensing, memory, radio, and sleep profile still determines battery life.

Behind every product promise

**Battery life, reliable connectivity, security, and edge intelligence
are delivered through engineering tradeoffs.**

The customer experiences the outcome. The developer manages the constraints.

What the developer feels

“I am out of RAM.”

“The OTA image barely fits.”

“Matter increased the flash requirement.”

“BLE commissioning conflicts with Thread timing.”

“Security provisioning is complicated.”

**“RF works in the lab—
but fails near Wi-Fi.”**

**“The model works on my laptop,
not on-device.”**

**“Power looks good until I
measure the real duty cycle.”**

The process node is invisible to the developer; memory, timing, RF, power, and lifecycle constraints are very visible.

Compute: more is useful, but not free

VALUE

- Application and protocol headroom
- Faster processing and potential race-to-sleep
- DSP/FPU/AI algorithms become practical
- More margin for future updates

COST / RISK

- Higher peak current and thermal bursts
- More ambitious software scope
- More RTOS tasks and timing interactions
- CPU may expose memory, flash, or radio bottlenecks

PROFILE THE
CHAIN



Memory headroom is product headroom

A realistic firmware footprint is more than the application

Future update margin
Logs / diagnostics
AI model / data
OTA rollback image
OTA primary image
Security / NVM
Matter / Zigbee / Thread / BLE
Bootloader + application

+

Why more memory helps

Fewer compile-time compromises, safer OTA layouts, better diagnostics, room for protocol and security growth

!

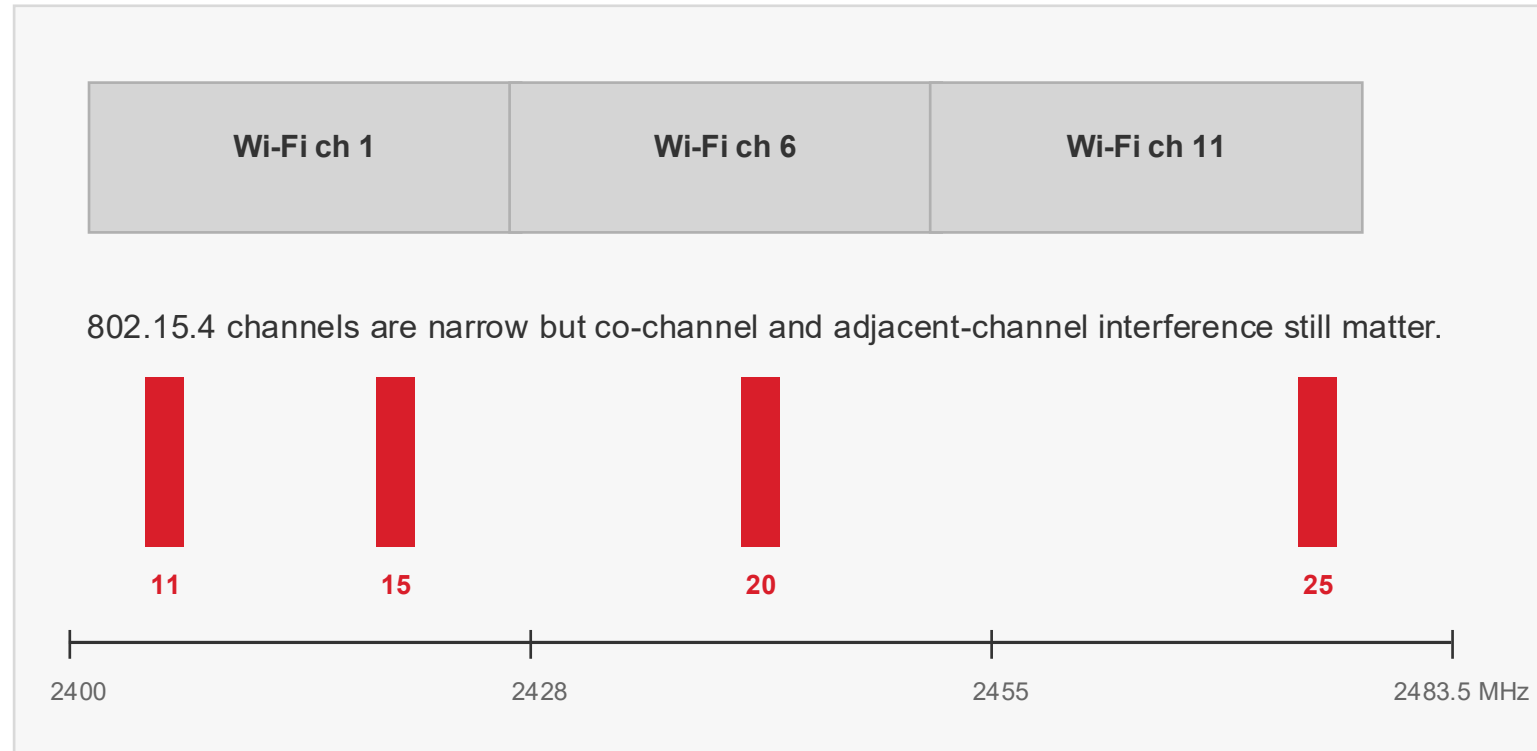
What more memory adds

Larger images, more complex partitioning, secure boot/authentication work, rollback and recovery decisions

The first release is not the only release - reserve explicit margin for the product’s second and third year.

RF and coexistence: a faster chip does not fix 2.4 GHz

Conceptual 2.4 GHz channel map



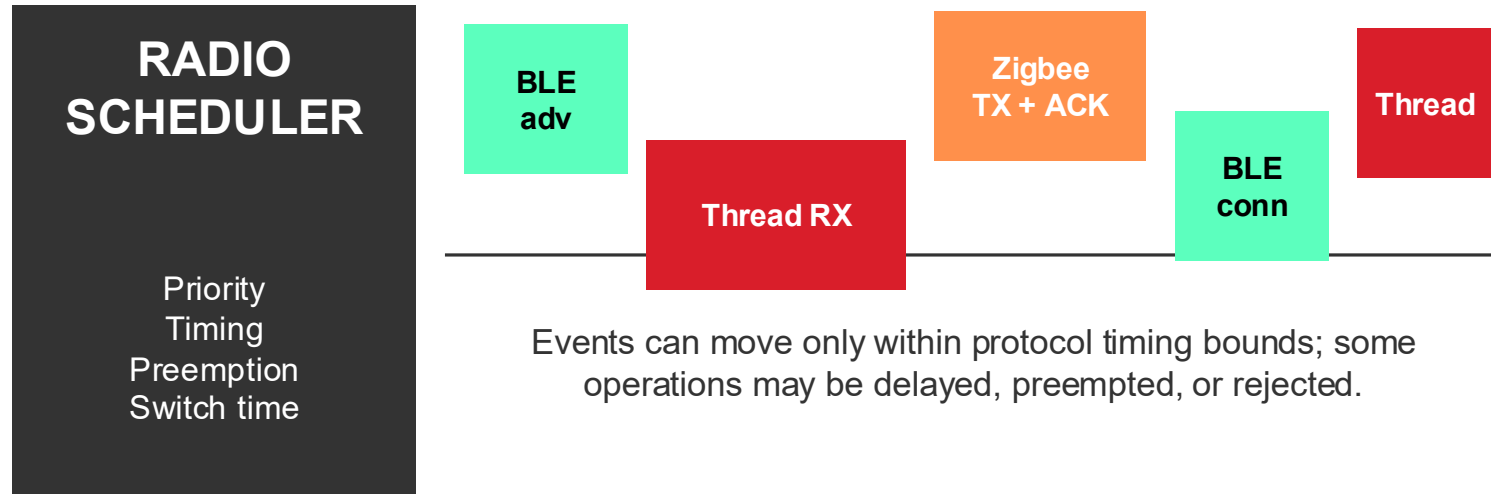
System-level RF still decides success

- Channel planning
- Antenna + PCB layout
- Enclosure and placement
- Coexistence testing
- Radio scheduling + retries

Process scaling cannot remove Wi-Fi interference, antenna loss, CCA behavior, or a poor RF layout.

Multiprotocol improves products and complicates engineering

One radio, multiple protocol deadlines



+ Customer benefit

BLE commissioning • operational mesh • SKU consolidation • smoother user experience

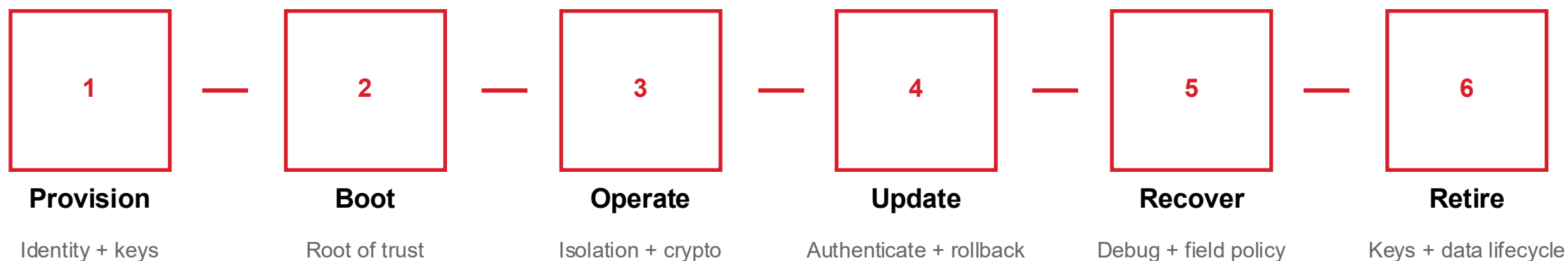
! Developer cost

Memory pressure • timing analysis • power profiling • certification and debug matrix

Multiprotocol is not a checkbox: it is a scheduling, memory, power, and validation problem.

Security is a lifecycle not optional plumbing

Hardware building blocks are necessary; disciplined provisioning and update processes complete the system.



Select Series 2 Secure Vault High devices: PSA Certified Level 3

Series 3 Secure Vault on SiXG301: PSA Certified Level 4

Platform evolution, not platform replacement

MATURE / CURRENT PLATFORMS

- Proven products and workflows
- Known power, RF, cost, and field behavior
- Strong fit for sensors and cost-sensitive endpoints
- Lower migration risk for established designs

SHARED FOUNDATION

Secure wireless
802.15.4 + BLE
Software ecosystem
RF engineering

NEWER ARCHITECTURE OPTIONS

- More compute and memory density
- New security and integration options
- Protocol concurrency and SKU simplification
- Headroom for new workload classes

40 nm-class and 22 nm-class platforms can both be correct - the portfolio expands the design space.

Series 2: strong fit across many current applications



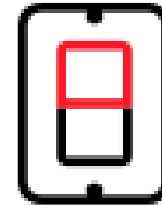
BATTERY SENSORS

Sleep, RF, BOM, and proven stack behavior can dominate.



LOCKS & ACTUATORS

Security, sensing, motor control, and multiprotocol balance.



SWITCHES & ENDPOINTS

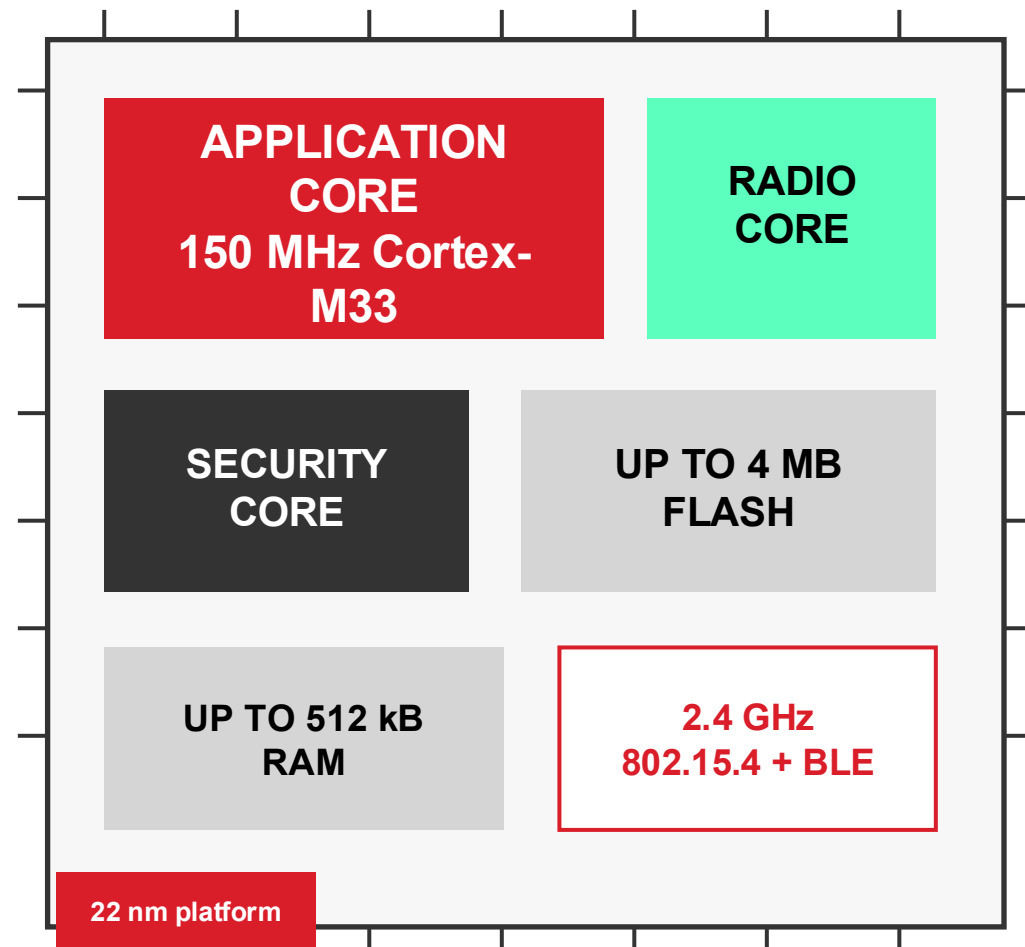
Cost-sensitive Zigbee/Thread/Matter endpoints remain central.



HUBS & CONTROLLERS

MG26 adds substantial memory and AI/ML headroom for complex designs.

Series 3 / SiMG301: new options for new workloads



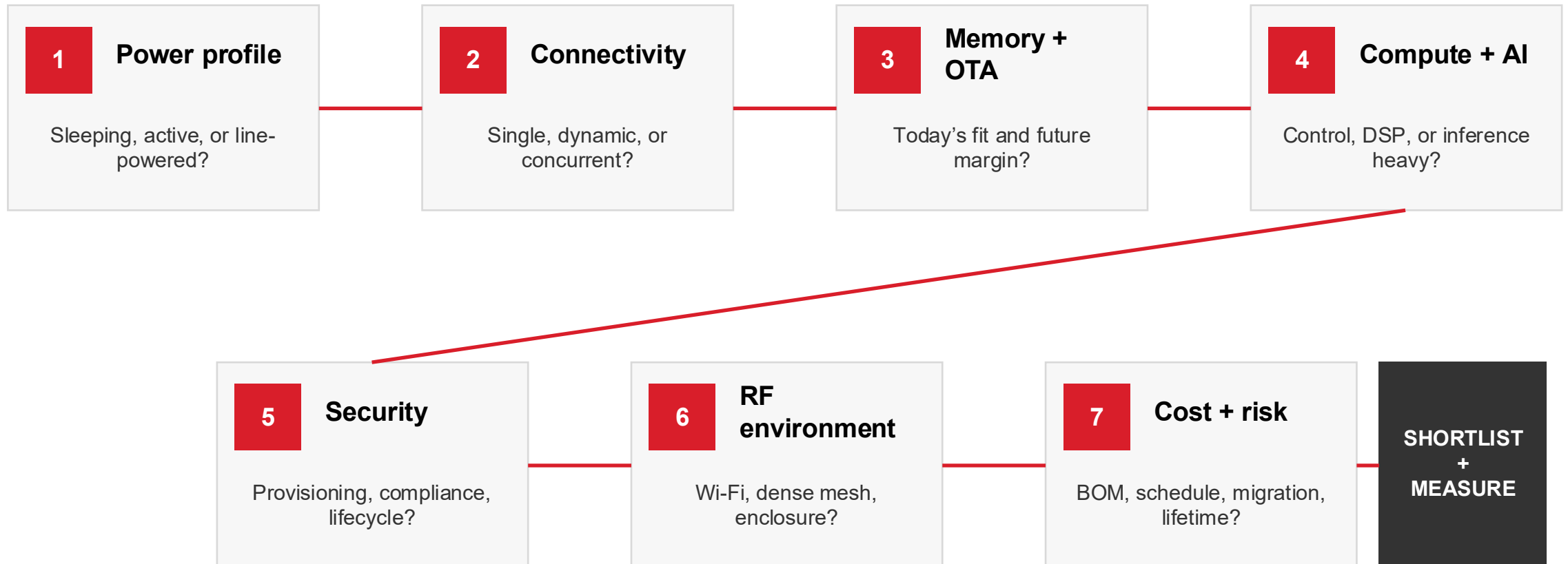
CUSTOMER VALUE

- Line-powered lighting, plugs, and switches
- Concurrent Zigbee + Matter over Thread can simplify SKUs
- Integrated LED features can simplify design and BOM

DEVELOPER VALUE

- Compute and memory headroom
- Dedicated application, radio, and security functions
- Concurrent 802.15.4 channel scanning and Secure Vault

Start with the workload not the node



The last step is measurement on target hardware with representative software and RF traffic.

The same decision seen from two perspectives

DECISION AREA	CUSTOMER ASKS	DEVELOPER VERIFIES
Power	Will the battery last?	What is the actual duty cycle, leakage, wake time, and RF energy?
Cost	Can I hit margin?	Can we remove external parts or a second MCU without adding risk?
Connectivity	Can I support the required ecosystems?	Do the stacks fit, schedule, coexist, and certify reliably?
Security	Can I meet compliance and trust requirements?	How do provisioning, secure boot, OTA, debug, and recovery work?
AI / compute	Can I add differentiated features?	Does the workload meet latency, memory, thermal, and energy targets?
Memory	Can the product survive future updates?	Is there enough flash/RAM for OTA, logs, rollback, and growth?
RF + lifecycle	Will it work in real homes and buildings for years?	What happens near Wi-Fi, metal, dense networks, and future SDK changes?

A good platform choice satisfies both the customer's business constraints and the developer's implementation constraints.

Five principles to carry forward

