CEC Continuing Education Center

DesignNews

Real-Time System Software Architecture Design

DAY 3 : Modeling with UML and the 4C Model

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets

Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.

CEC Continuing Education Center

BENINGO EMBEDDED GROUP

DigiKey

THE SPEAKER

Jacob Beningo

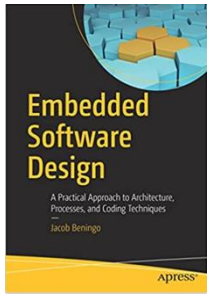
Jacob@beningo.com

jacobbeningo

Beningo Embedded Group – CEO / Founder

Focus: Software Architecture, Processes, and Dev Skills

At Beningo Embedded Group, we believe everyone deserves the skills to confidently advance their careers, meet deadlines, and deliver quality embedded systems. We provide modern strategies, insights, and hands-on training to equip developers and teams with the tools they need to succeed.



EMBEDDED SOFTWARE ACADEMY BY BENINGO

Visit **www.beningo.com** to learn more

This week's topics:

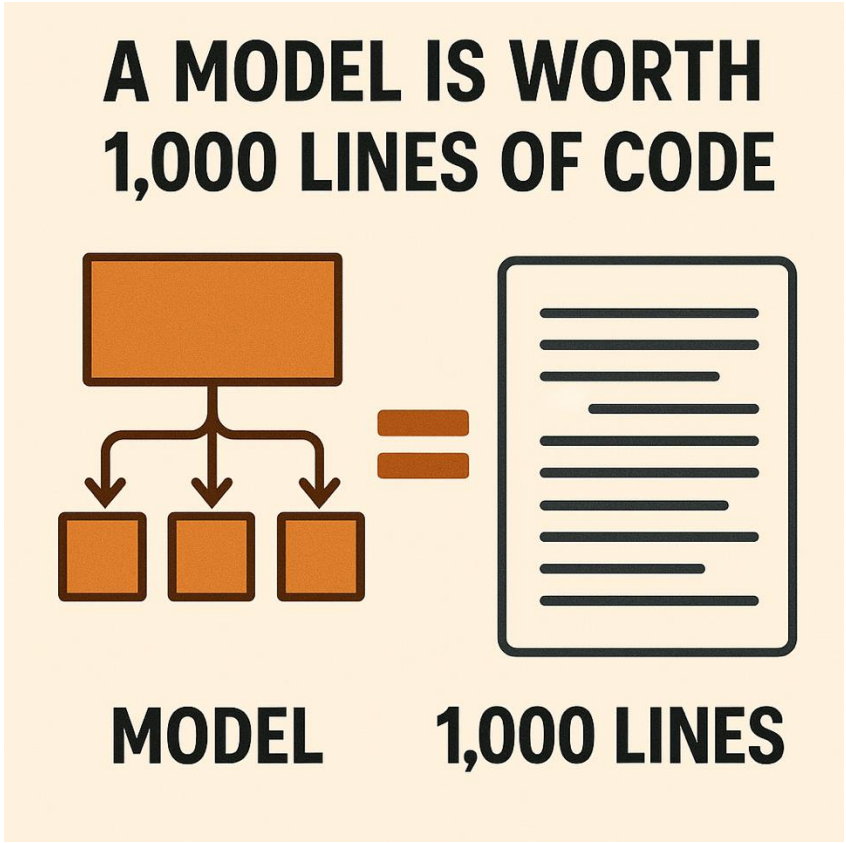
- ✓ What is Software Architecture?
- ✓ Design Philosophies and Principles
- Modeling with UML and the 4C Model
- Data-Centric Architecture Design
- Beyond UML – Data, Isolation, Security

Architectural Modeling

01

Modeling is Thinking

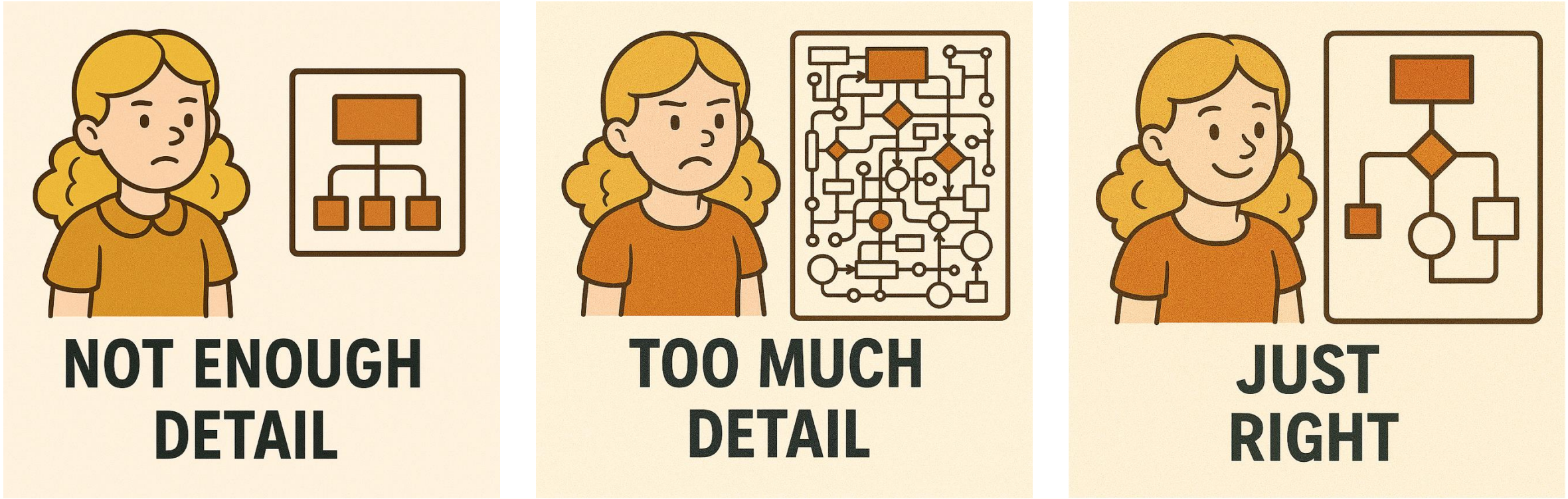
- Architecture isn't just about diagrams; it's about clarity.
- A good model makes invisible decisions visible.
- Helps answer: What are we building? and Why this way?
- Supports communication across developers, testers, and stakeholders.
- Avoids "tribal knowledge" traps and reduces rework.



Common Modeling Pitfalls

- **Too vague:** "It's just a block diagram" with no structure or meaning.
- **Too detailed:** Dives into code structure and loses architectural clarity.
- **No shared framework:** Everyone draws it differently—confusion spreads.
- **Static views only:** Doesn't show runtime behavior or data flow.
- **Neglects embedded realities:** Ignores real-time, memory, and hardware interactions.

The "Goldilocks' Zone"



Audience POLL Question

What characterizes a successful architecture?

a) One that fully specifies every function and module so developers have no ambiguity.
b) One that leaves all design decisions to developers to promote creativity.
c) One that captures critical decisions and system organization while leaving implementation details to developers.
d) One that prioritizes speed of delivery over clarity or structure.

The 4 C Model

01

What is the 4C Model?

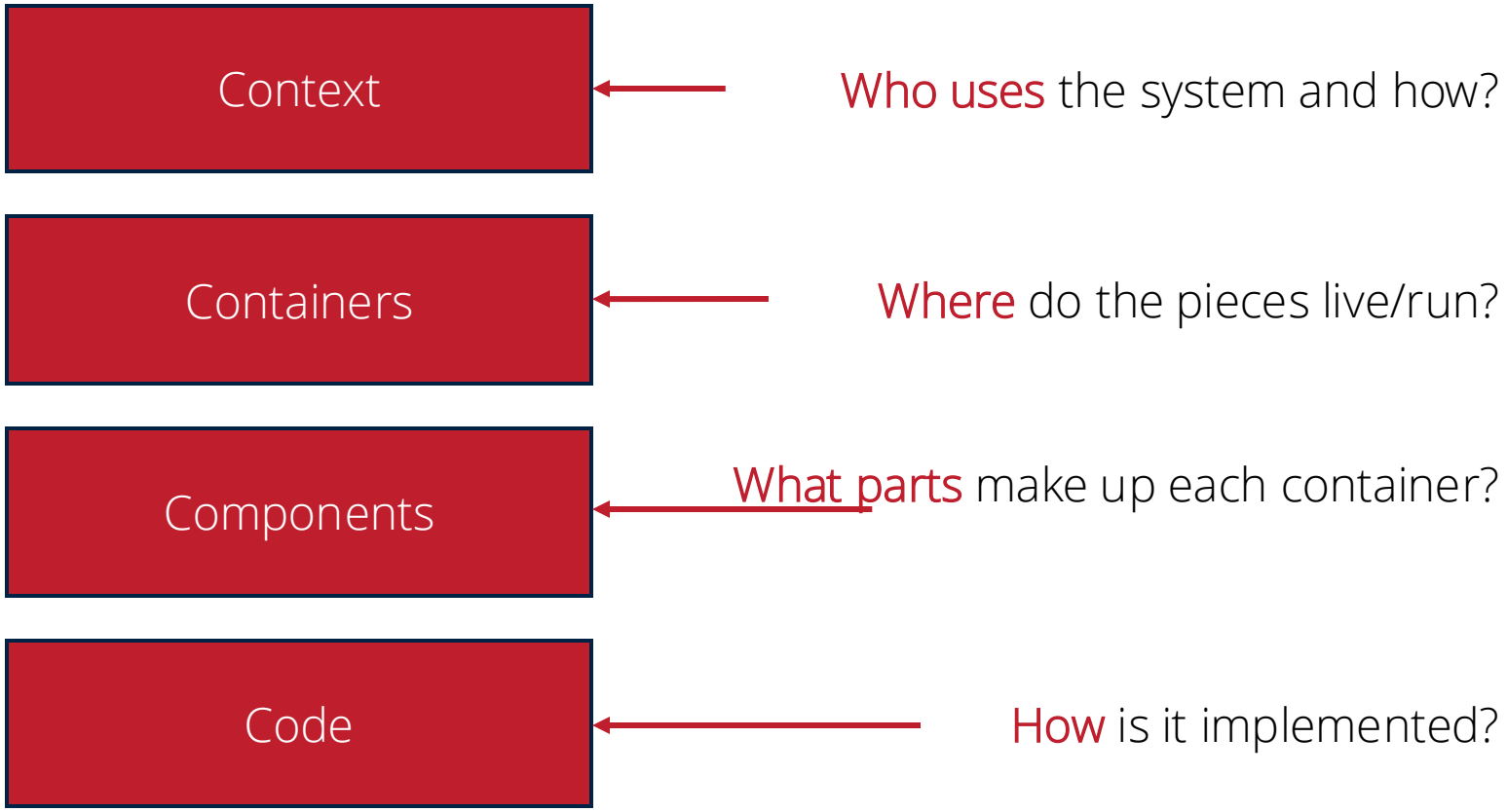
Developed by Simon Brown to bring consistency to architecture diagrams.

Defines four hierarchical views of a system:

- Context – External relationships and system boundaries.
- Containers – Runtime processes and executables.
- Components – Functional building blocks inside containers.
- Code – Implementation details (optional at architecture level).

Each "C" answers a specific question about the system.

The 4C Visual Hierarchy



- Each level drills down into more detail.
- Focus shifts from
  - users →
  - processes →
  - modules →
  - source code.
- Visual consistency ensures everyone is speaking the same architectural language.

Applying 4C to Embedded Sysetms

Context:

- Product use in the real world (technician, cloud, mobile app).
- Physical interfaces like buttons, sensors, BLE, cloud APIs.

Containers:

- Logical processes like: MCU firmware, RTOS tasks, cloud service, mobile app.
- For MCUs, treat containers as threads, tasks, or key binaries (bootloader, app, etc.).

Components:

- Software layers or modules: SensorManager, NetworkStack, HAL, Logging.
- Often match design layers (drivers, middleware, services, apps).

Code:

- Source files, classes, and function organization.
- Use only when needed—focus on architectural boundaries, not syntax.

Audience POLL Question

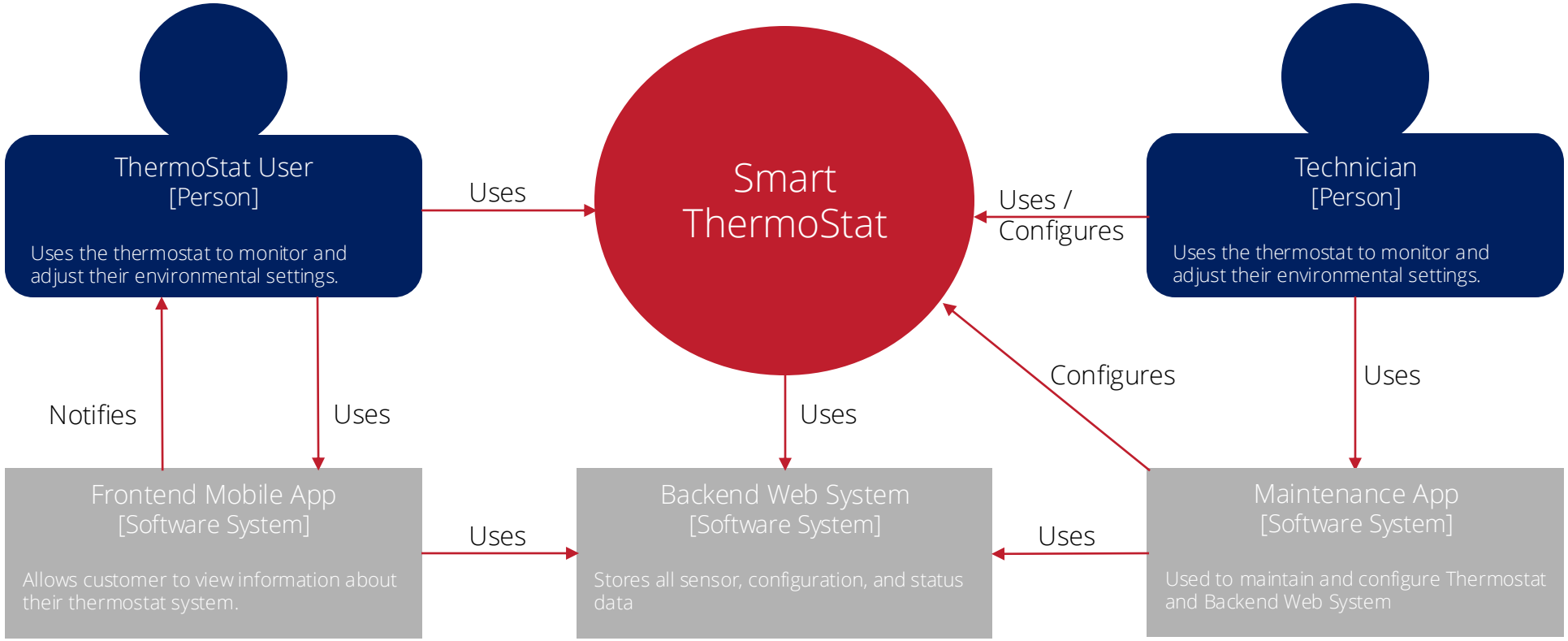
What is the primary purpose of using the 4C Model in software architecture?

a) To document every line of code so developers have no freedom in implementation.
b) To provide just enough structure to capture key architectural decisions and system organization without over-specifying details.
c) To replace all other architectural frameworks with a rigid, one-size-fits-all model.
d) To describe only the hardware components of a system.

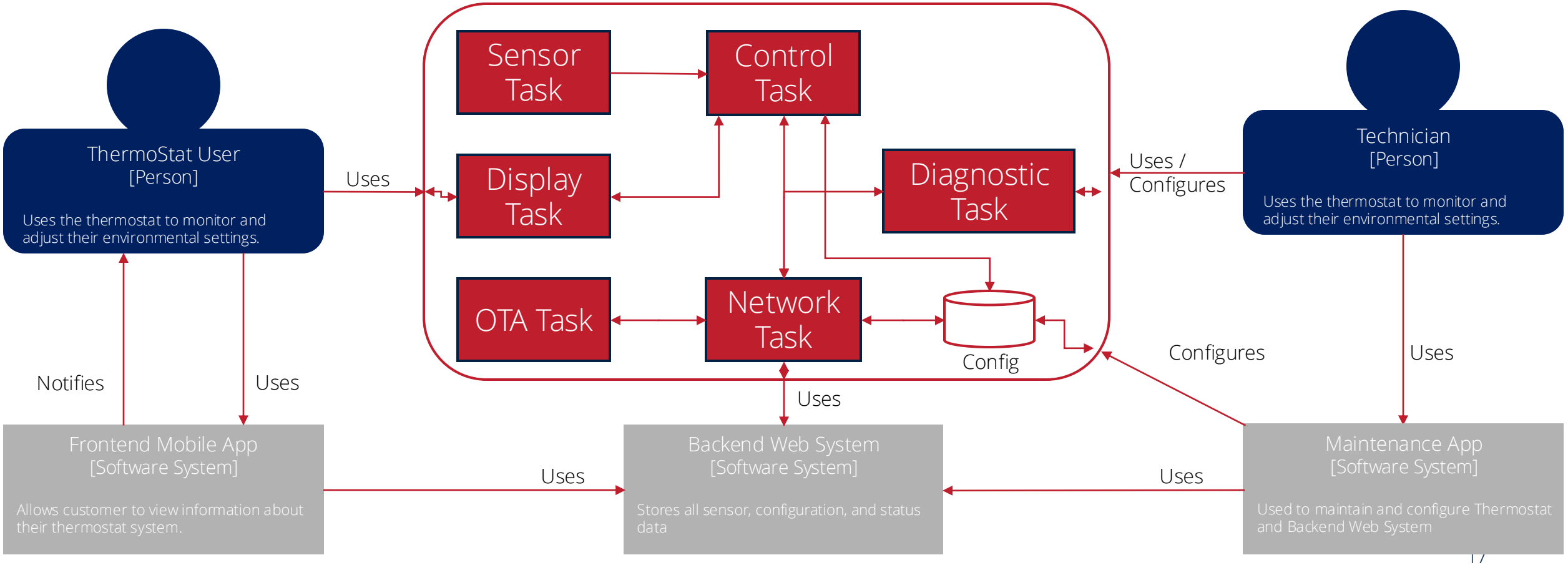
Examples

03

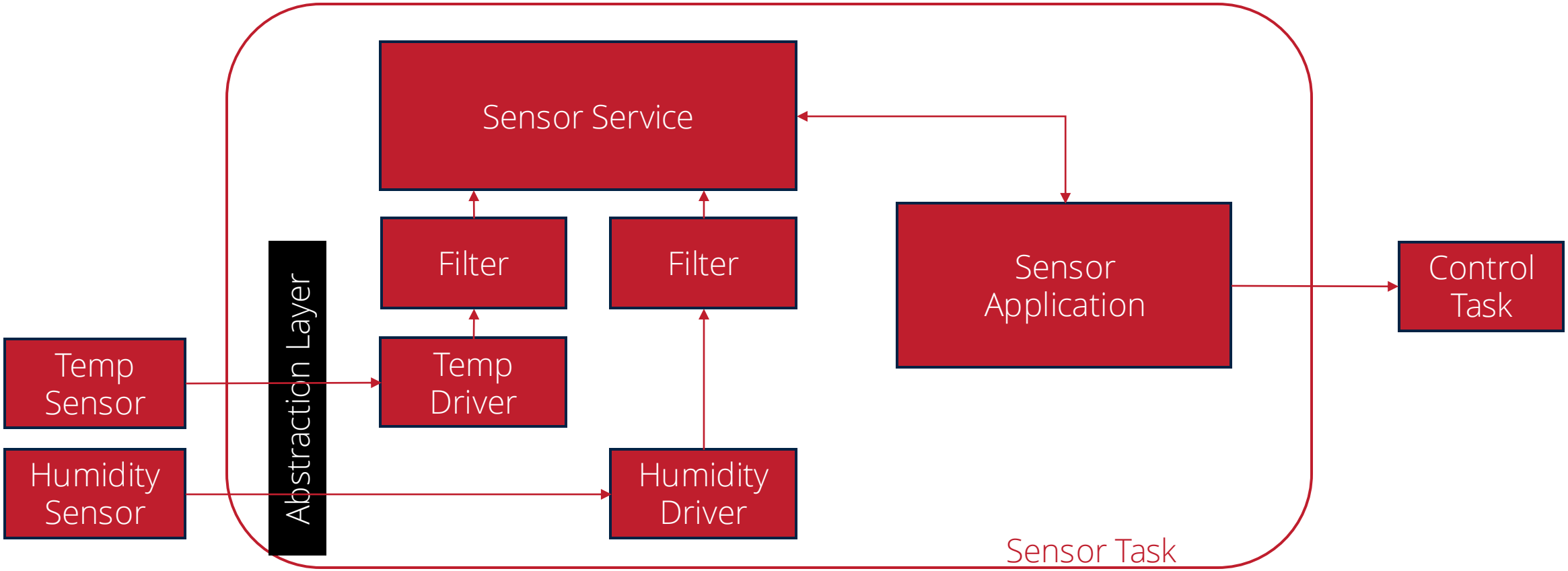
Context Diagrams



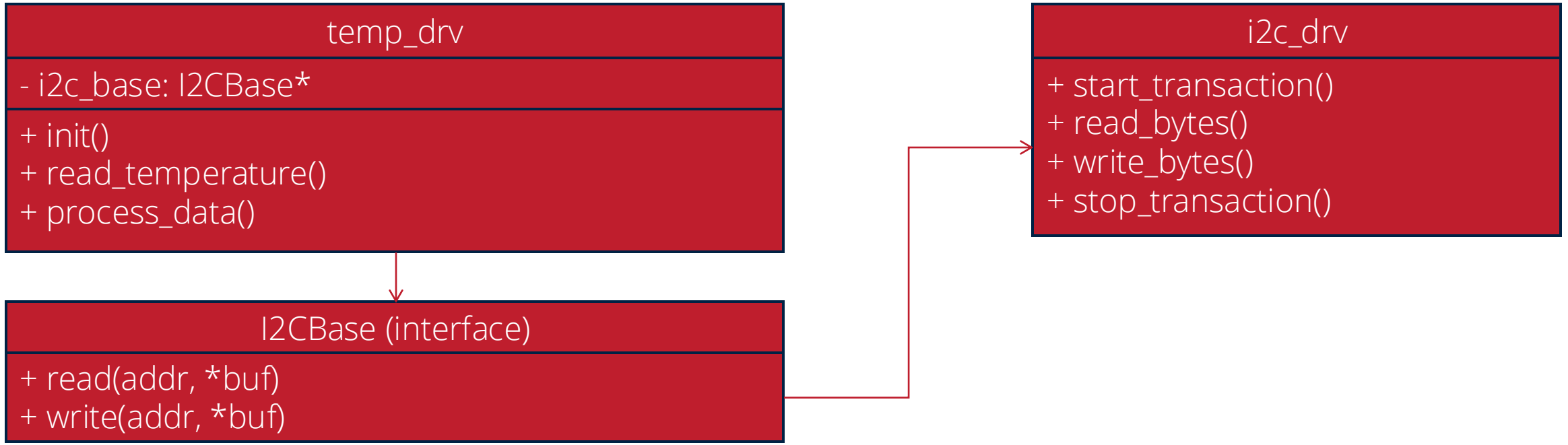
Container Diagrams



Component Diagrams



Code Diagrams



Audience POLL Question

Does this help you better see the structural organization of your system?

a) Yes
b) No
c) Other

Next Steps

04

Going Further

Download the extra resources:

- https://beningo.short.gy/25DNCEC-10-Architecture-Resources

Get Hands-On:

- Analyze an existing projects software architecture
- Draw a new software architecture
- Invest a widget to practice your architecture skills
- Join an Embedded Software Architecture Kata

Downloadable Resources:

- Characteristics Worksheet
- Modern Principles
- Architecture Book List
- ADR Template

Save $100:
https://beningo.short.gy/DNCEC2510

Additional Resources

Please consider the resources below:

- Jacob's Blogs
- Jacob's CEC courses
- Embedded Software Academy

- Embedded Bytes Newsletter
  - http://bit.ly/1BAHYXm

Consulting | Coaching | Training

www.beningo.com

Next Steps

- What is Software Architecture?
- Design Philosophies and Principles
- Modeling with UML and the 4C Model
- Data-Centric Architecture Design
- Beyond UML – Data, Isolation, and Security

CEC Continuing Education Center

DesignNews

Thank You

Sponsored by

DigiKey

BENINGO EMBEDDED GROUP

informa markets