Representing IoT Systems with the Actor Model and Akka.NET

DESIGNING THE ACTOR MODEL SYSTEM

Jason Roberts
.NET MVP

@robertsjason dontcodetired.com

Overview

Course overview

Why use the Actor Model for IoT?

Suggested course prerequisites

A brief Actor Model refresher

- Actors
- Messages

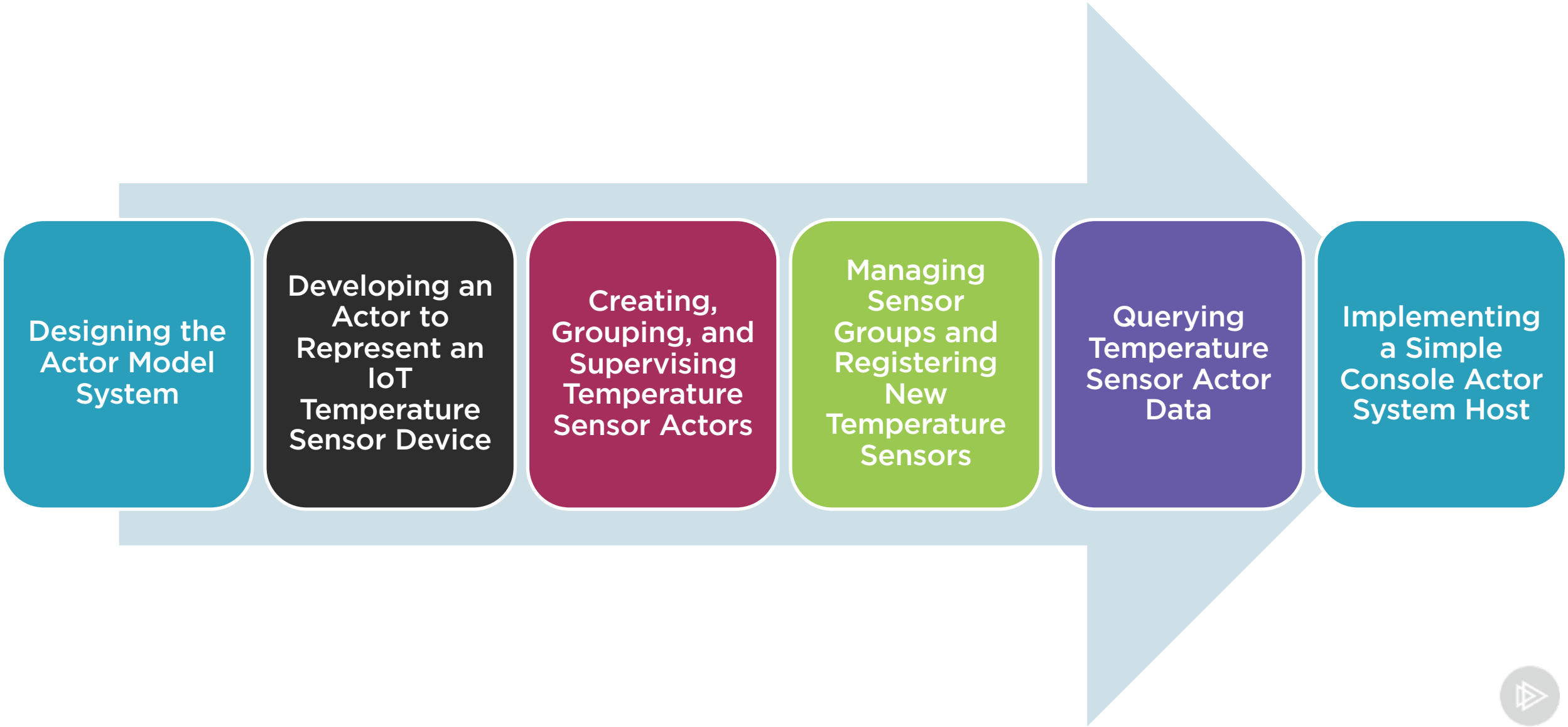
Demo scenario design overview

- Physical
- Actors
- Supervision hierarchy

Get started in Visual Studio

- Create actor model project
- Create actor test project

Course Outline



Why Use the Actor Model for the Internet of Things?

Concurrency

Scalability

Fault tolerance

Lightweight

Network protocol decoupling

Suggested Course Prerequisites

Building Concurrent Applications with the Actor Model in Akka.NET

- Actors, messages, location transparency, supervision, actor lifecycles, etc.

Akka.NET Testing Fundamentals (optional)

- TestKit, test probes, expected messages, mock actors, etc.

A Brief Actor Model Refresher

Actors

Fundamental computation unit

Perform small well-defined tasks

Encapsulate state

Actors can:

- Send and receive messages
- Create additional child actors
- Change their behavior

Supervision hierarchies

A Brief Actor Model Refresher

Messages

Simple POCO classes

Should be immutable

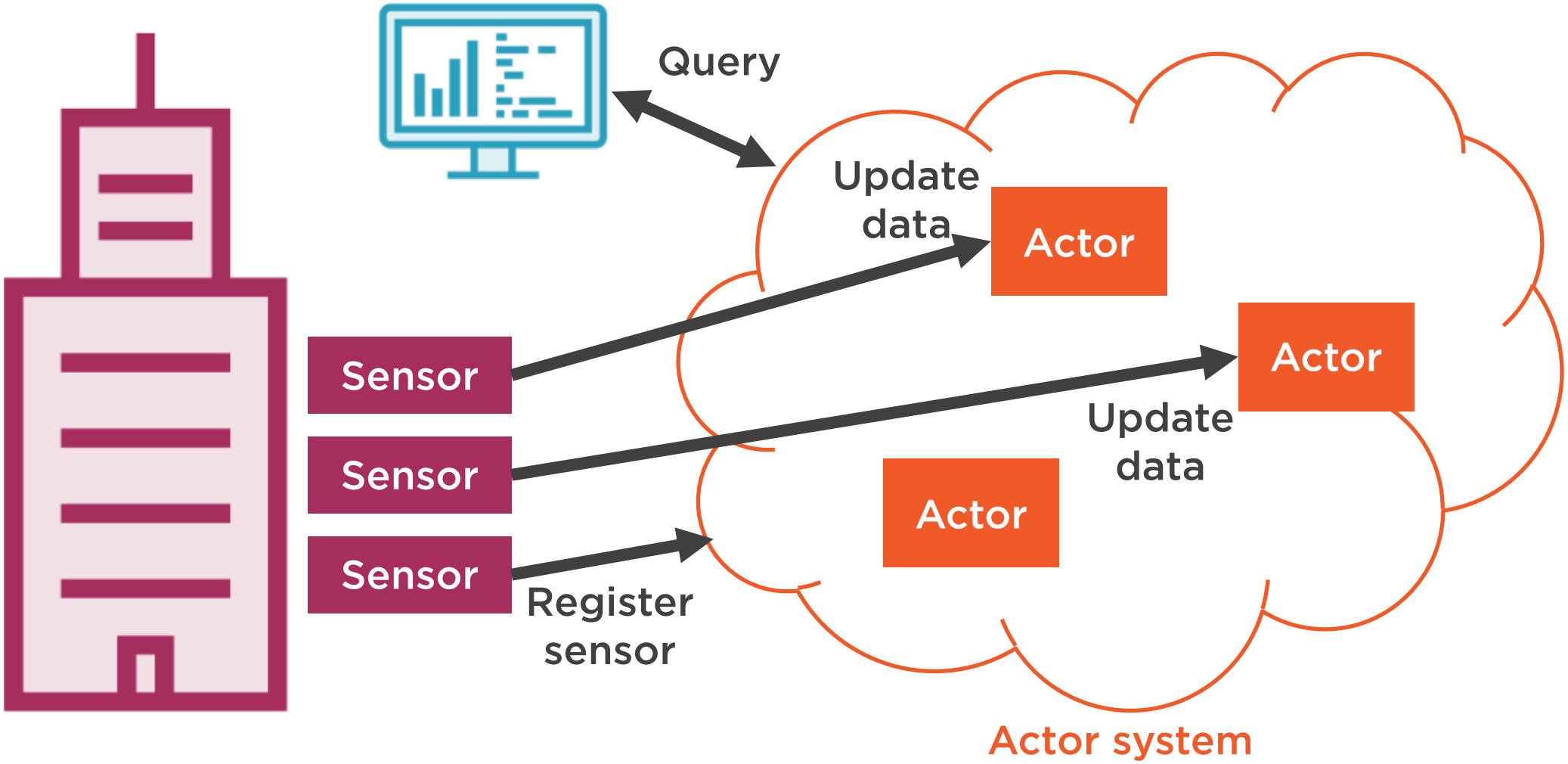
Sent from:

- Actors
- Outside actor system

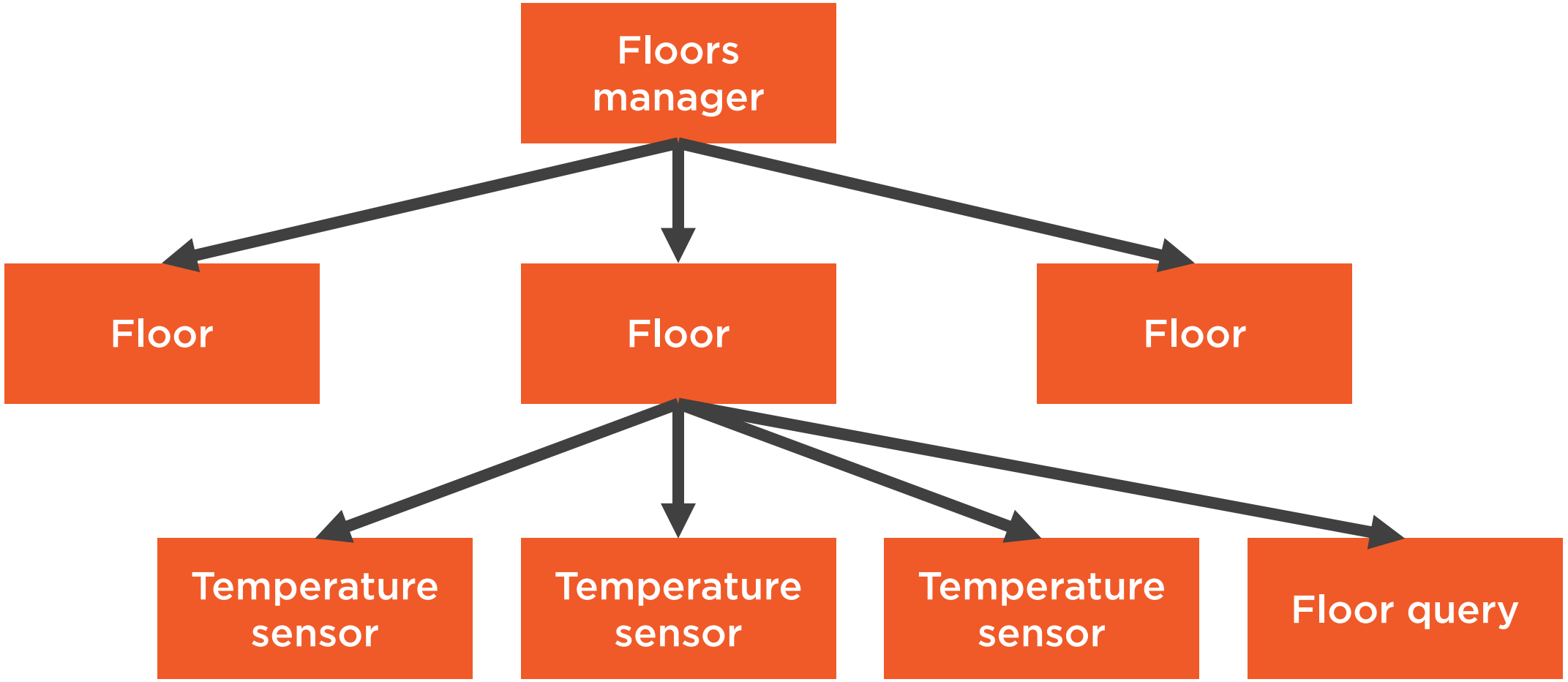
Change actor state

Execute actor functionality

Demo Scenario Design Overview



Supervision Hierarchy



Summary

Why use the Actor Model for IoT?

- Concurrency
- Scalability

Course overview and prerequisites

Actors and messages

Demo scenario design overview

- Floors manager
- Floor
- Temperature sensor
- Query

Created actor model and test projects

Next:

Developing an Actor to Represent an IoT Temperature Sensor Device