

# The Actor Model and why you should be using it

Joe Wirtley

@JoeWirtley

Wirtley   
Consulting LLC

# About Me

Wirtley Consulting LLC

Springboro, OH

Dayton .NET Developer Group

C#, WPF, MVC, Web API

@JoeWirtley



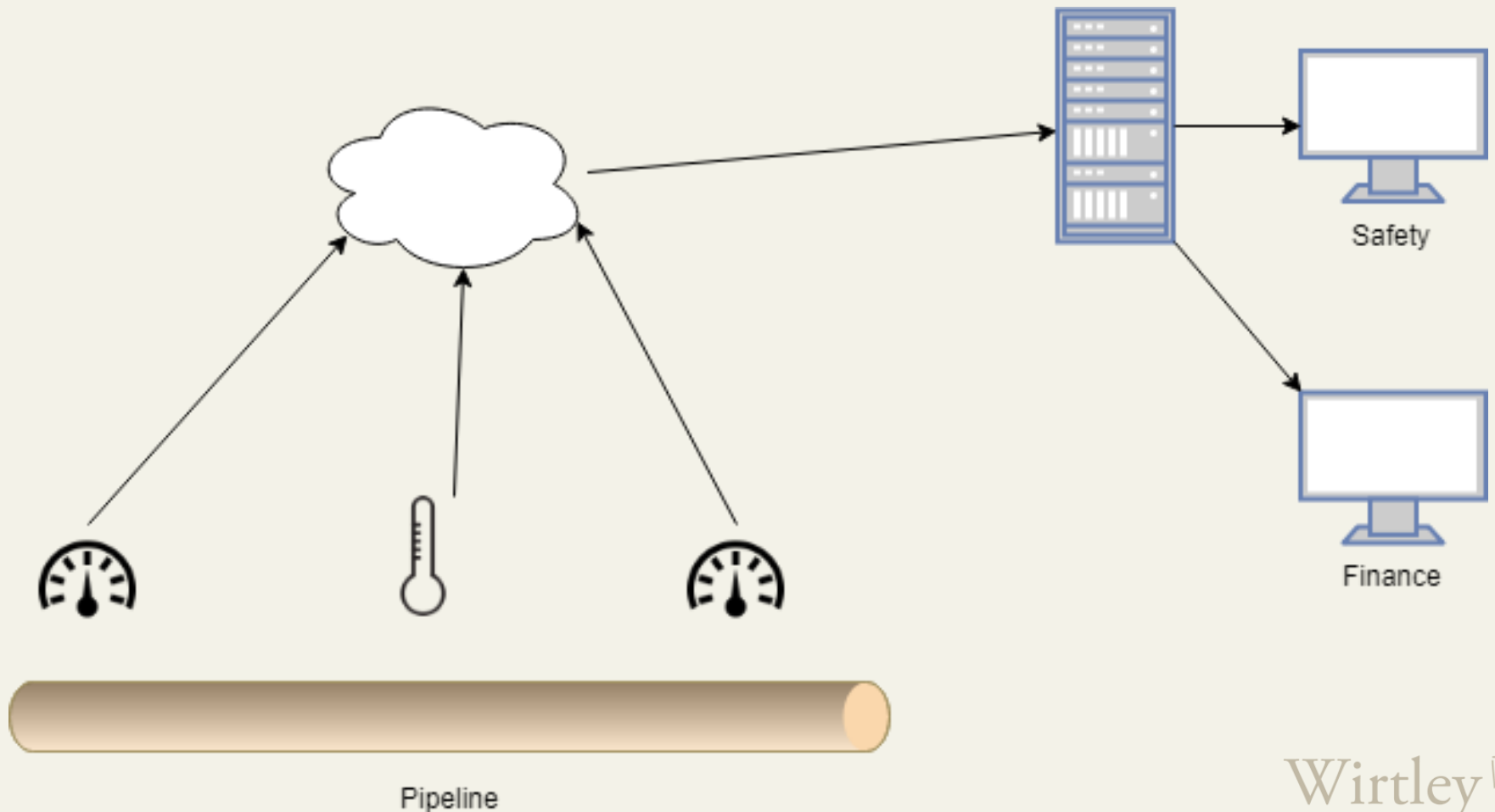
# About You



# Outline

- Problem
- Solution Approaches
- Actor Model description and features

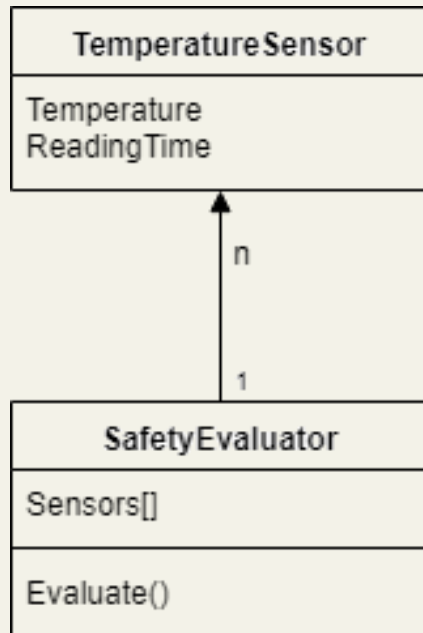
# The Problem



# Solution Approaches

- Paradigms
  - Object Oriented
  - Functional
  - Actor Model
- Considerations
  - Model
  - Concurrency
    - Consistency
    - Blocking

# Object Oriented Approach



- Model
  - Combine state and behavior
- Mutable state
  - Read and write temperature

# Functional Approach

*fx*

SafetyEvaluator

- Model
  - Separate state and behavior
- Immutable State
- Sensor?



# What I want

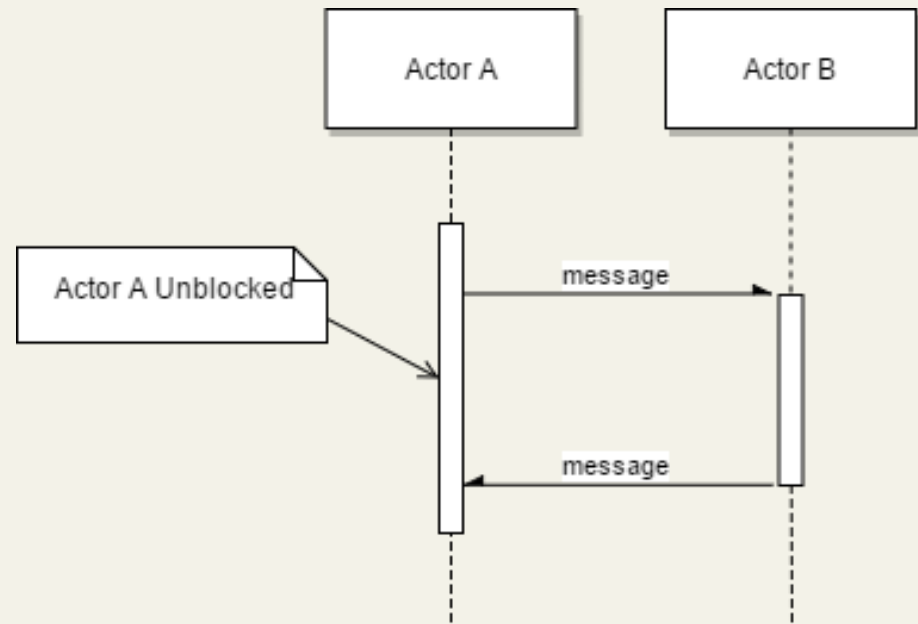
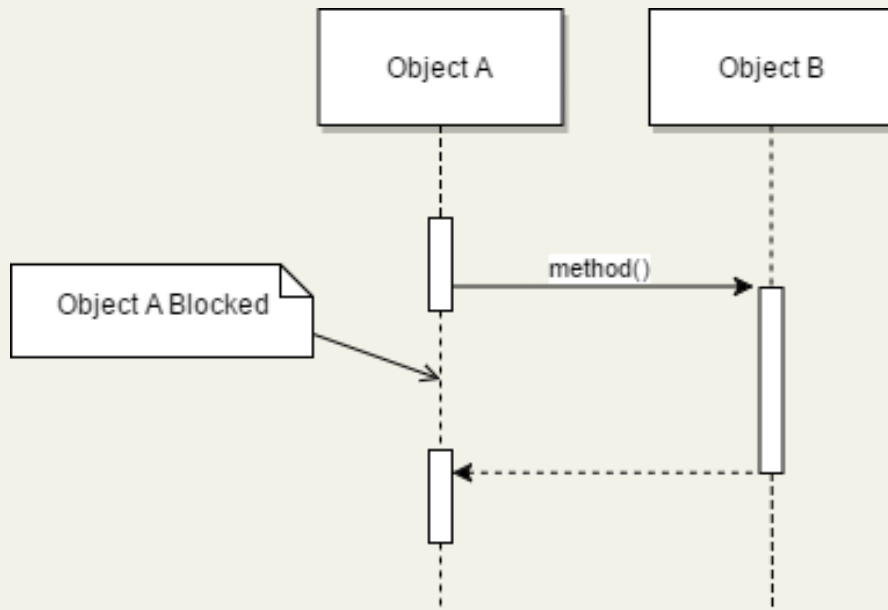
- Modelling of object approach
  - State and behavior together
  - Entities corresponding to domain objects
- Concurrency handling of functional approach

# Actor Model!

# The Actor Model

- Everything is an actor
- Actors can
  - Send and receive messages
  - Create other actors
  - Change behavior based on a message
    - State machine
- Actors combine state and behavior

# Concurrency – Part 1



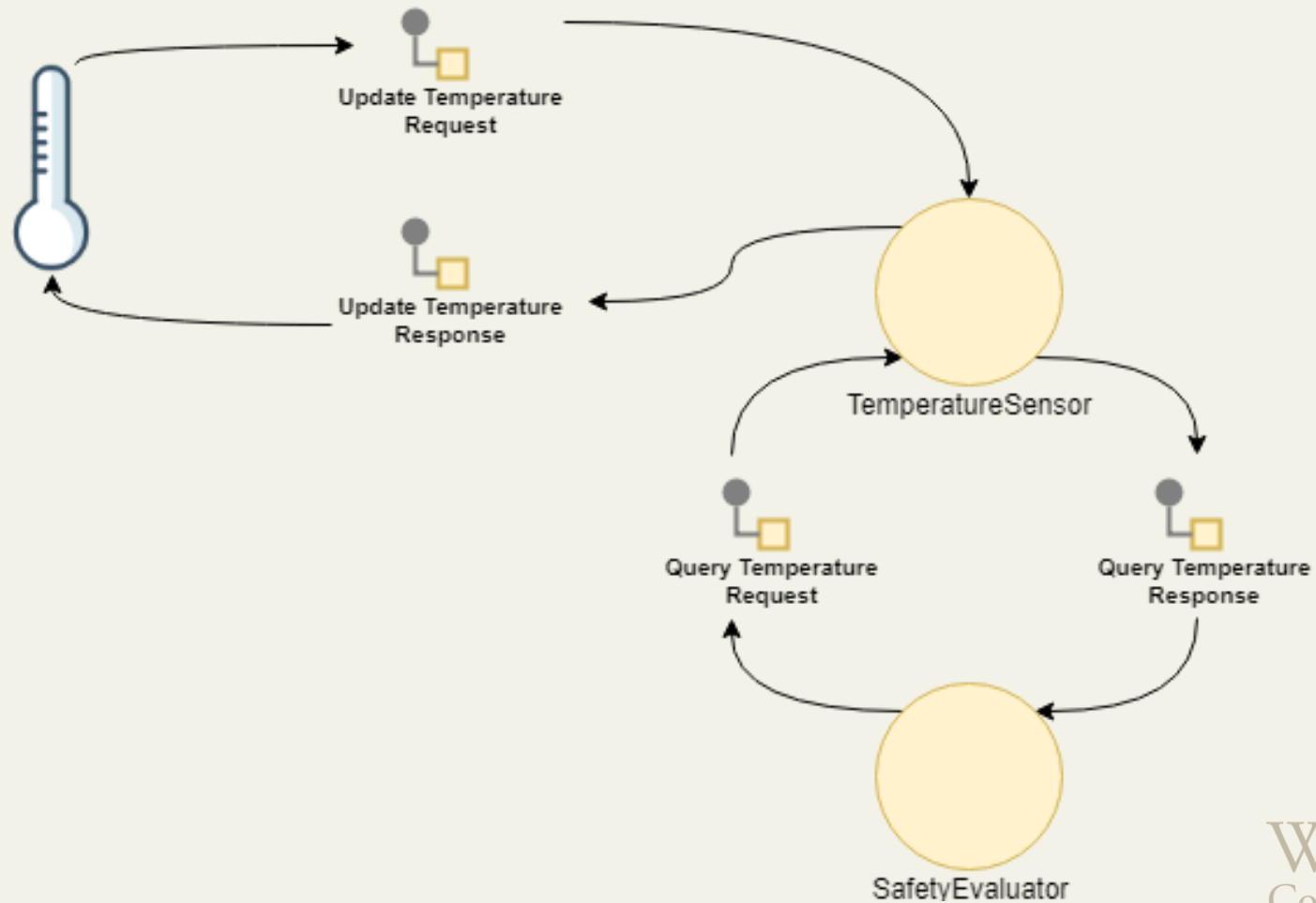
# Concurrency – Part 2



Actor

Thread

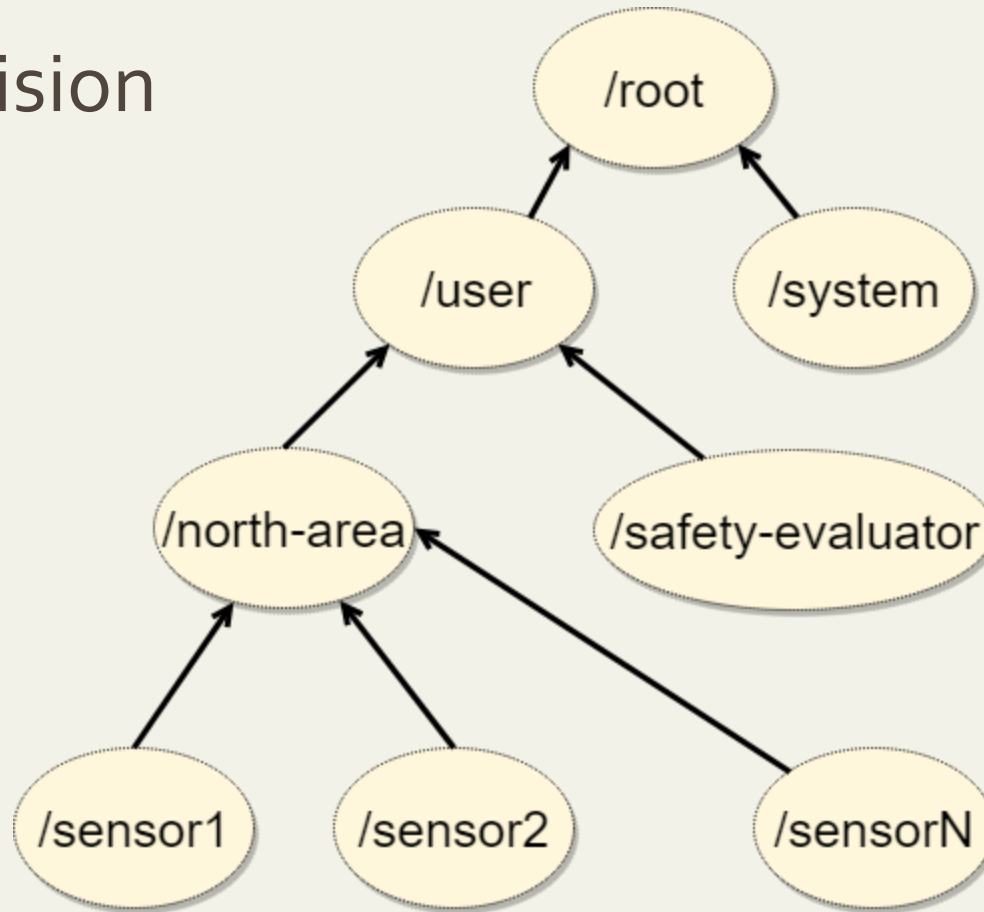
# Actor Model Solution



# Demo

# Fault Tolerance

## Supervision





# Location Transparency

- Local address  
akka://system/user/service-a/worker1
- Remote address  
akka.tcp://system@example.com:5678/user/  
service-b

# Scalability

- Routers
- Clustering
- Akka.NET metrics
  - 50 million messages/sec on a single machine
  - ~2.5 million actors per GB of heap

# Message Delivery

- Default delivery rules
  - At most once delivery
  - Message ordering guaranteed per sender-receiver
- Patterns exist for
  - At least once delivery
  - Exactly once delivery

# Implementations

- Akka
  - Java and Scala
- Erlang
- Elixir
- .NET
  - Akka.NET
  - Orleans
  - Service Fabric Reliable Actors
- Proto.Actor
  - .NET and Go

# Use the actor model when

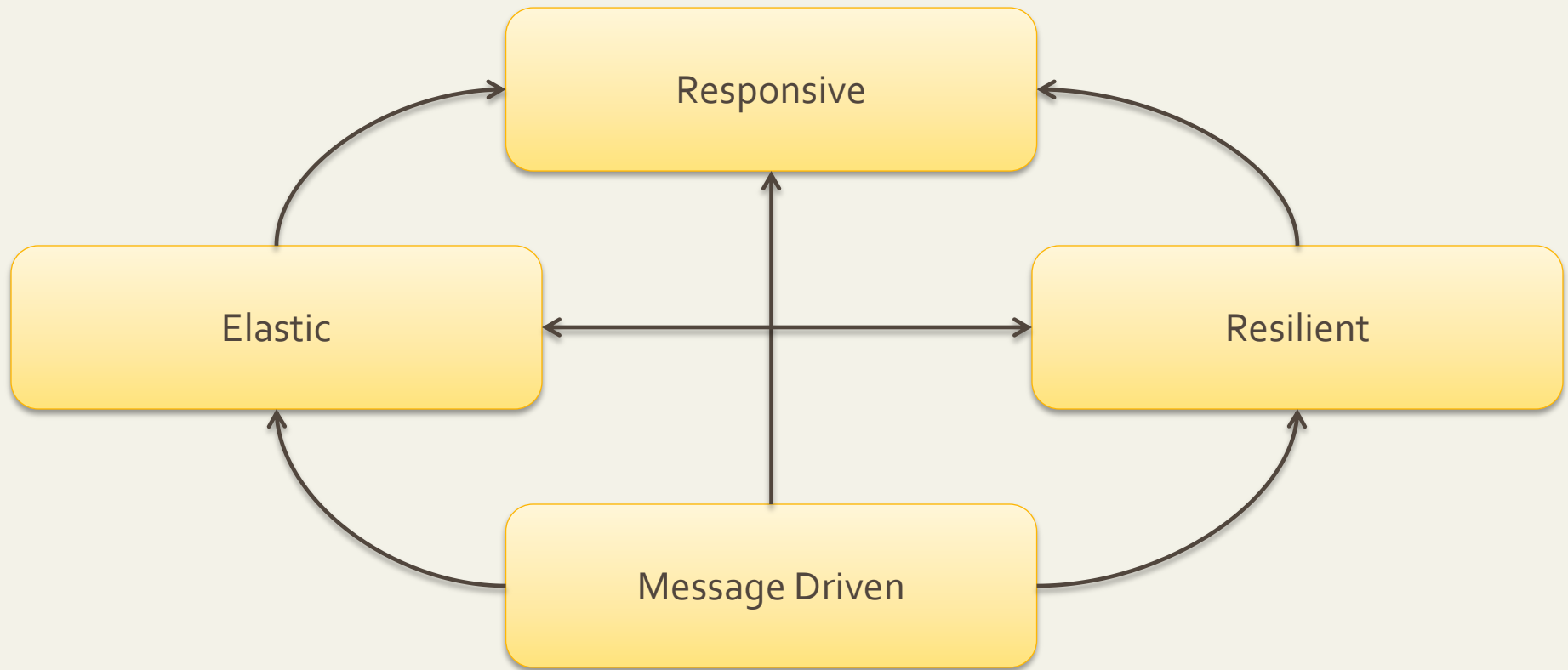
You have tasks that are:

- Long running
- Independent
- Substantial
- Especially when working in an object-oriented language

# Don't use the actor model when

- You need simple background tasks
- You only need asynchronous behavior

# Reactive Manifesto



# Contact Me

Presentation and source on GitHub:

<http://bit.ly/ActorModelPresentation>

[Joe@WirtleyConsulting.com](mailto:Joe@WirtleyConsulting.com)

[@JoeWirtley](#)

<http://WirtleyConsulting.com>