

COGS IN THE MACHINE

Roy Osherove
Typemock

AGENDA

What are cogs?

Why are they hard to test?

Integration Testing

Unit Testing

Interaction Excavation

Today's Patterns for cog isolation

Disasters waiting to happen

ENVELOPING SYSTEMS

Silverlight

Sharepoint

Windows Workflow

The CLR and java runtimes

Visual Studio (or any plugin host)

**CREATING A PERIMETER
IMPORTANT QUESTION!**

**Does system provide a way to
replace the system?**

**CLR – You can host your own
envelope with different behavior**

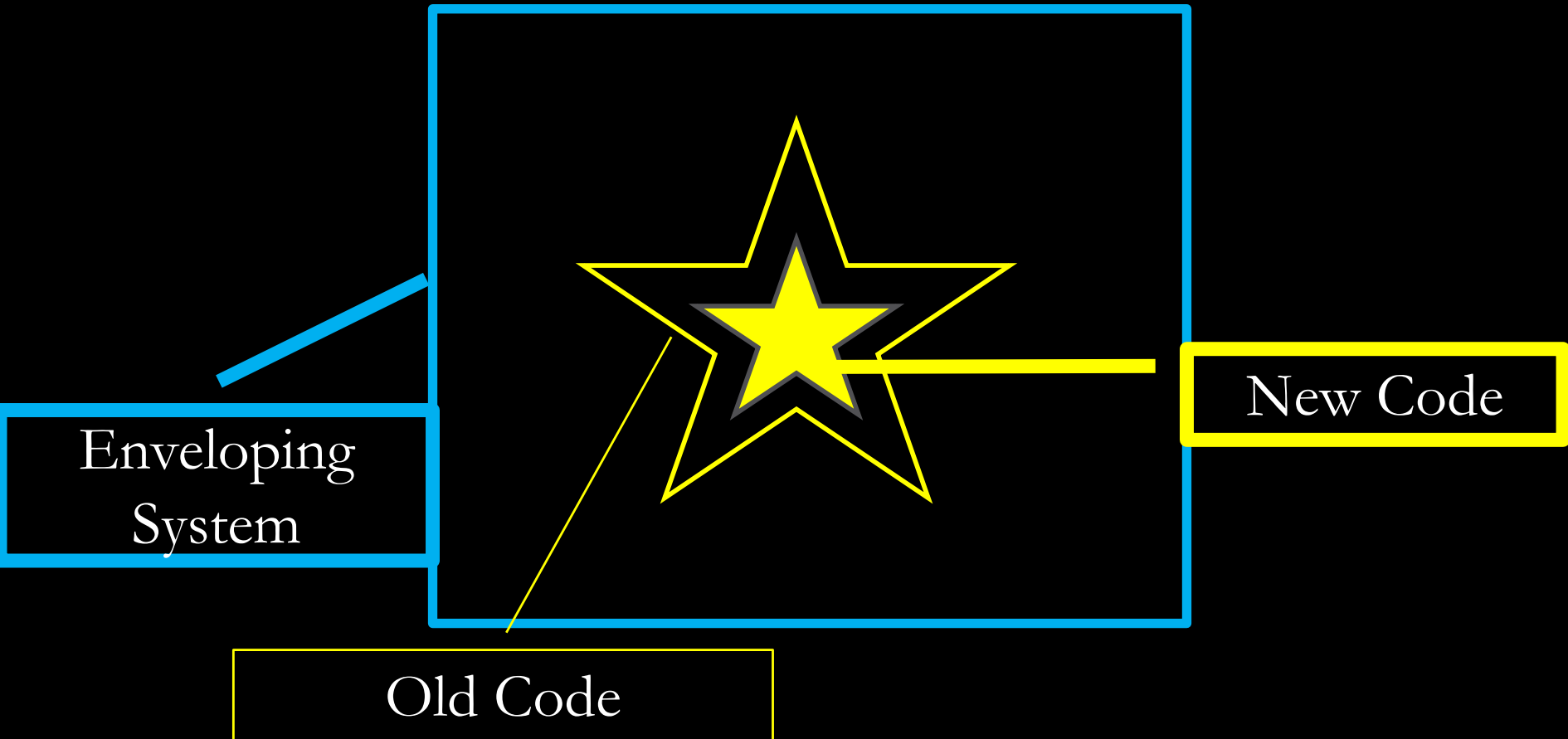
TRANSPARENT PERIMETER PROPERTIES



Enveloping
System

New Code

TRANSPARENT PERIMETER PROPERTIES



Enveloping
System

New Code

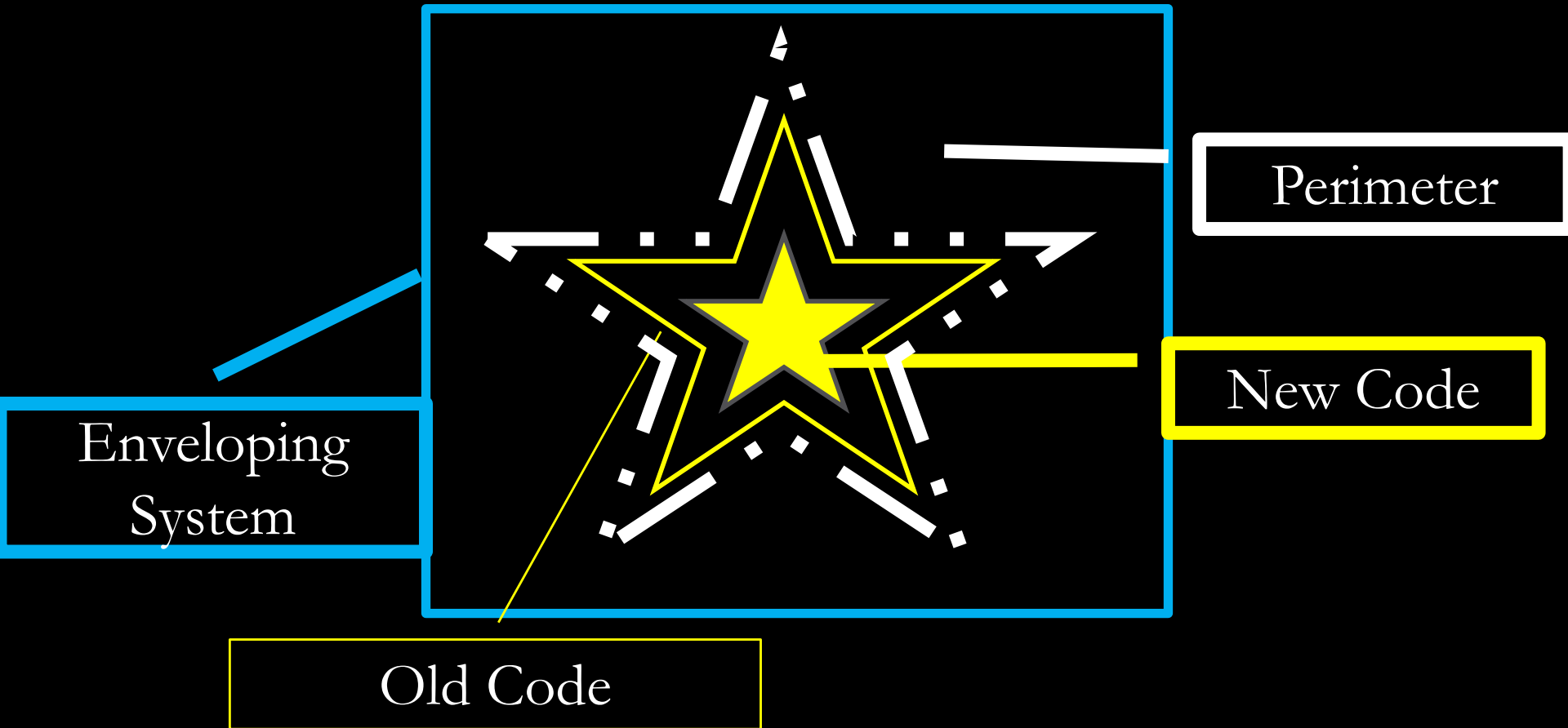
Old Code

SYSTEM IS REPLACEABLE

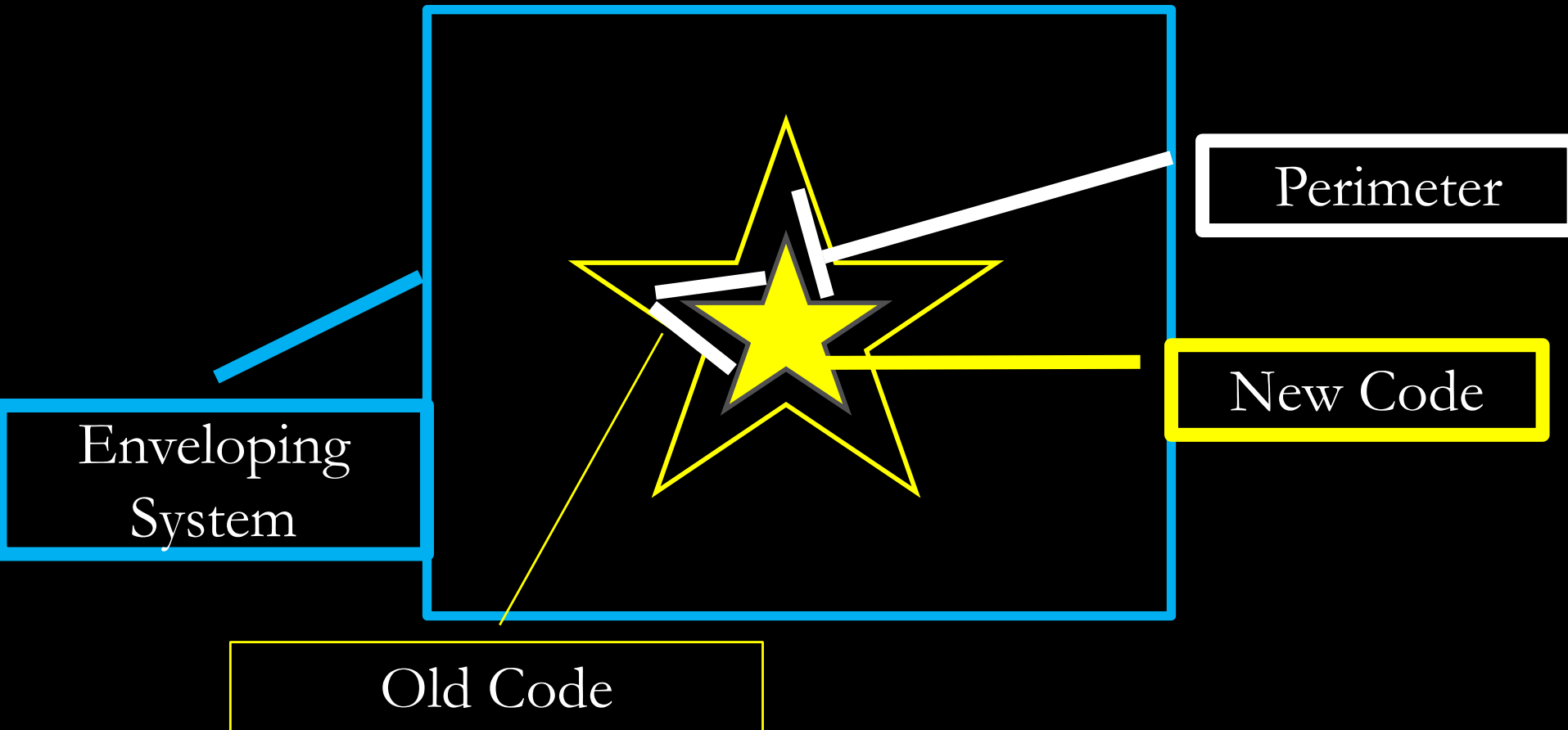
Hand code your own subsystem

**Use isolation frameworks and hand
rolled stubs to change system
boundary behavior**

PERIMETER FOR HAND ROLLED SYSTEM ADAPTER



PERIMETER FOR AD-HOC ISOLATED SYSTEM USING FRAMEWORKS



SYSTEM IS NOT REPLACEABLE

**Surround code with your own hand
coded system adapter**

**Use AOP to intercept and change
system calls**

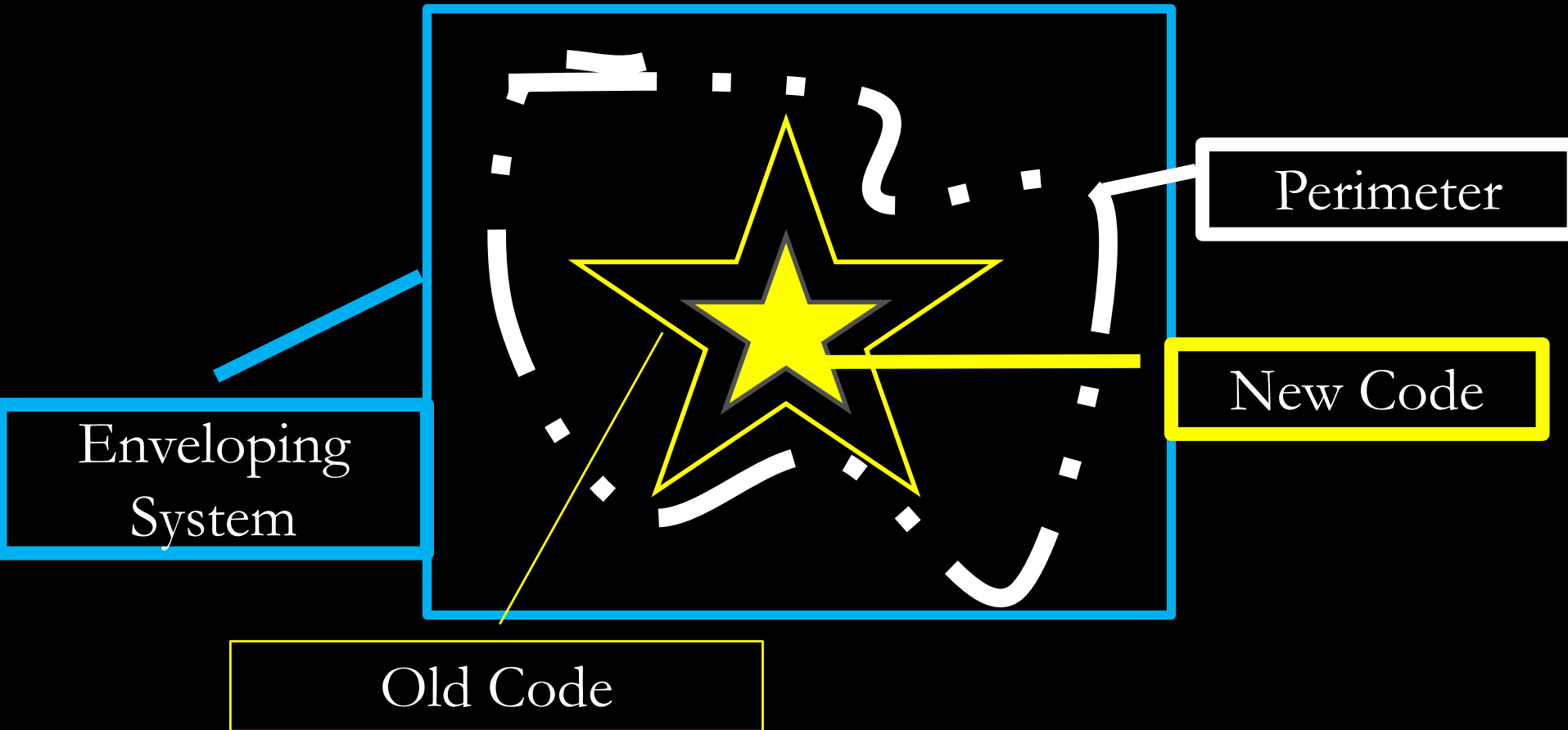
INTERACTION EXCAVATION

Wishful Invocation

Get **dependency exception**

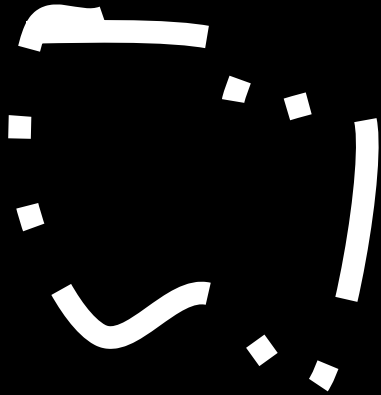
Fabricate interaction

POSSIBLE PERIMETER AFTER EXCAVATION



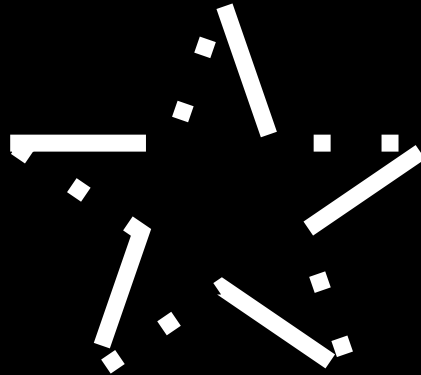
EMERGING PERIMETERS

Deep-Rooted



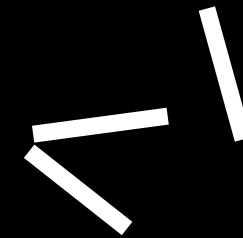
- Was Excavated
- Based on dependencies
- Very far from your own code
- Robust

Role Based



- Was pre-designed or JIT designed with TDD
- Based on roles
- Directly touching your code
- Robust

Interaction Based



- Was Excavated or pre-thought
- Based on Contracts
- Directly touching your code
- Can be brittle

**TOOLS FOR CREATING
PERIMETERS**

Powermock (Java)

Cthru \ Typemock (.NET)

Moles (.NET)