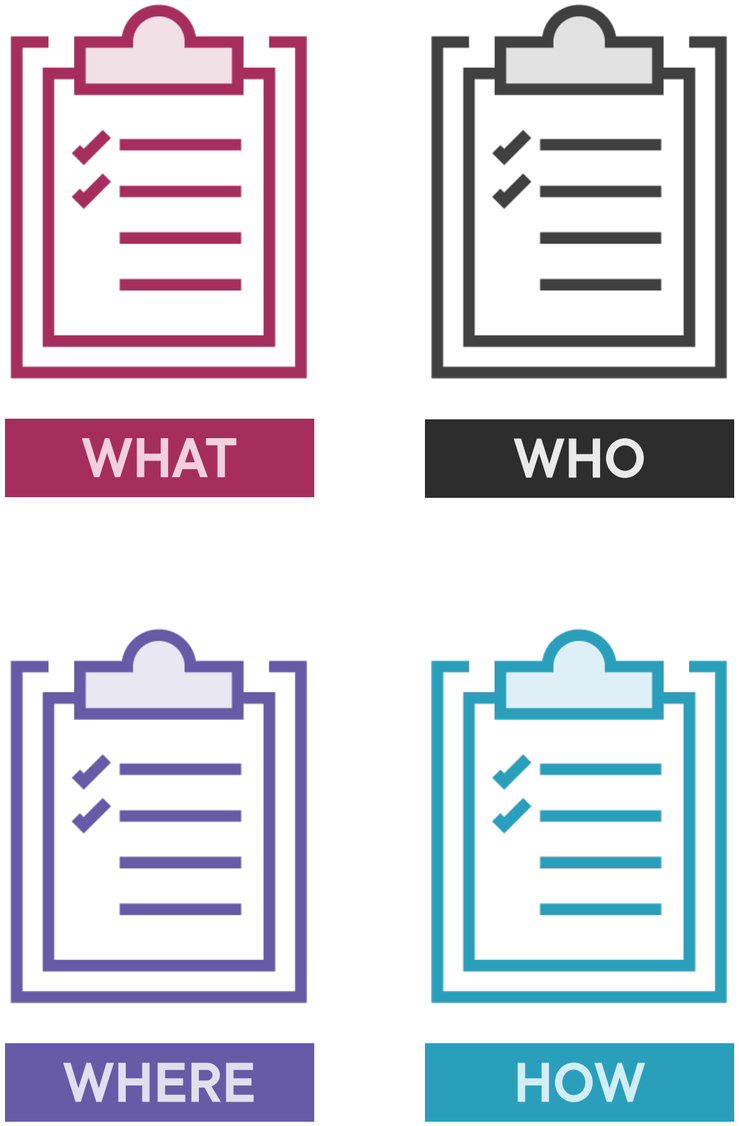
Best Practices for Using Extension Methods

Elton Stoneman
Consultant & Trainer

@EltonStoneman blog.sixeyed.com



Guidelines

Not universal

But mistakes cost

WHAT

- Use sparingly
  - For multiple consumers
  - Otherwise stick to class methods
- Target specific types
  - Narrow the scope
  - Minimize risk of shadowing

WHERE

Separate assemblies

- Logically group extensions
- Minimize dependency graph

Consider the namespace

- Prefer custom namespaces
- But target namespaces work too

WHO

Sponsor class

- Conventions help maintenance
- Not seen by consumers

Method name

- Clear and explicit
- Beware risk of shadowing

HOW

Minimize dependencies

- Causes bloat for consumers
- And versioning issues

Document clearly

- Pops in IntelliSense
- XML summary as a minimum

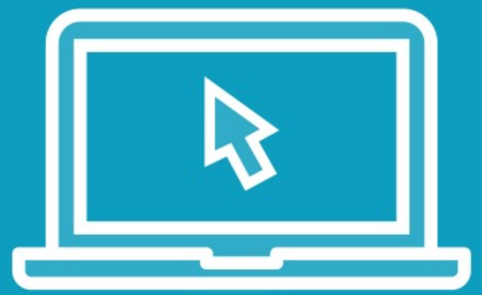
WHAT

WHO

WHERE

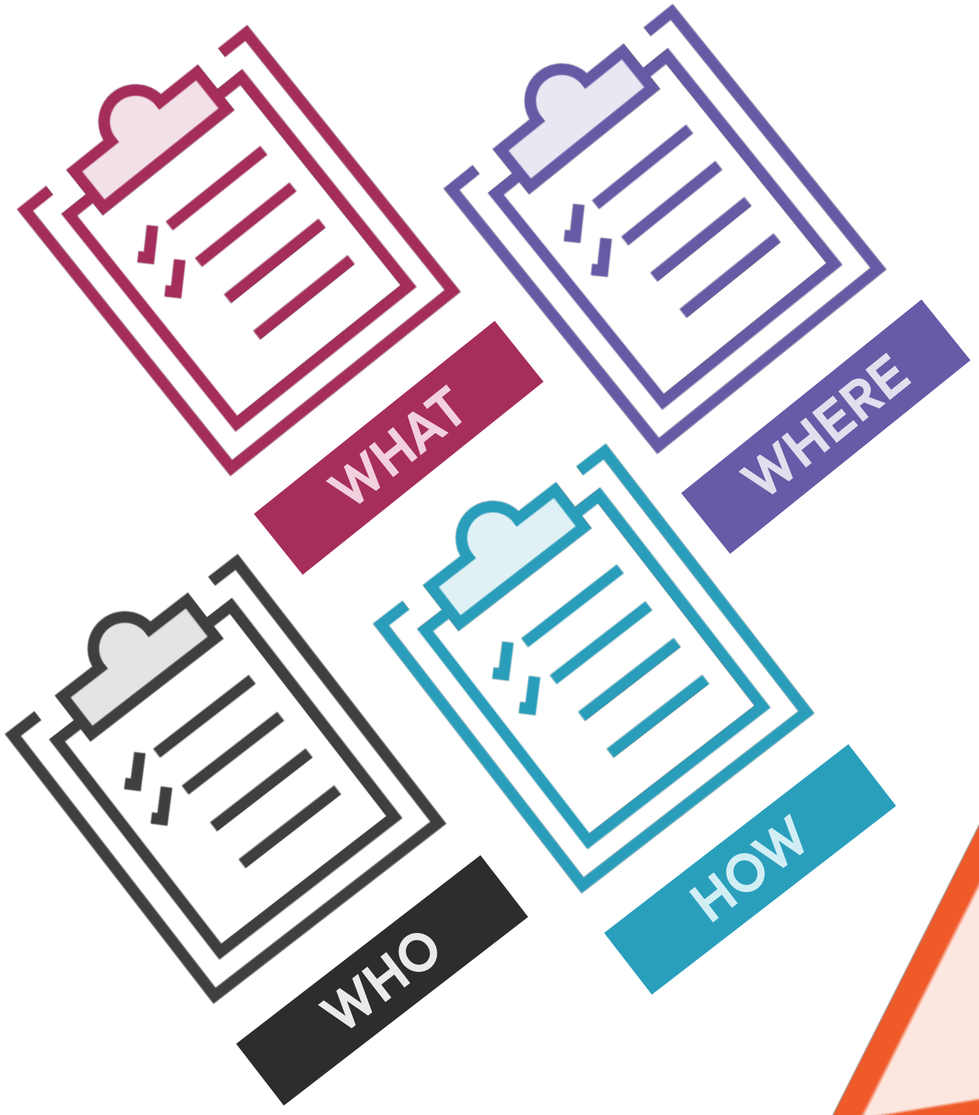
HOW

Demo

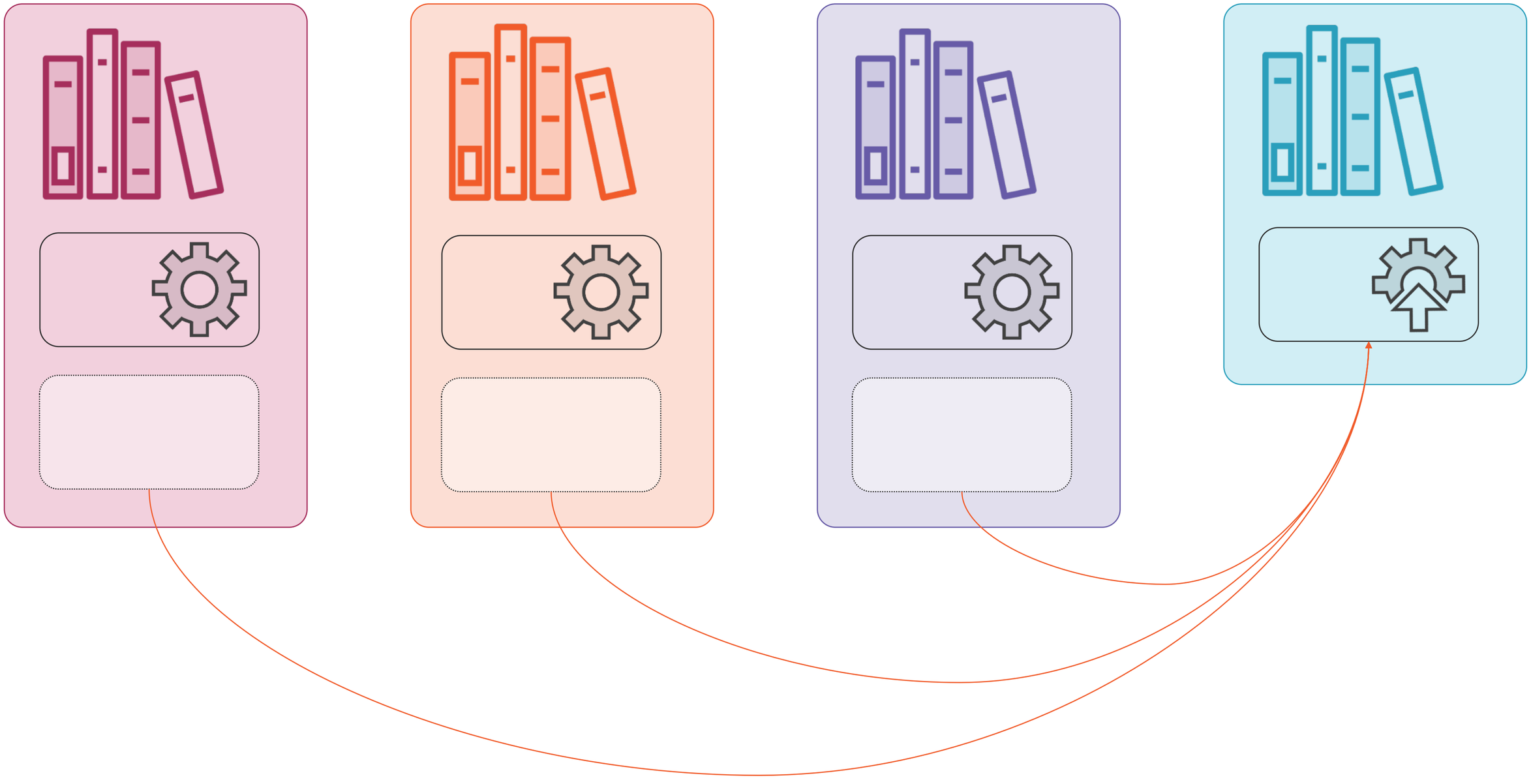


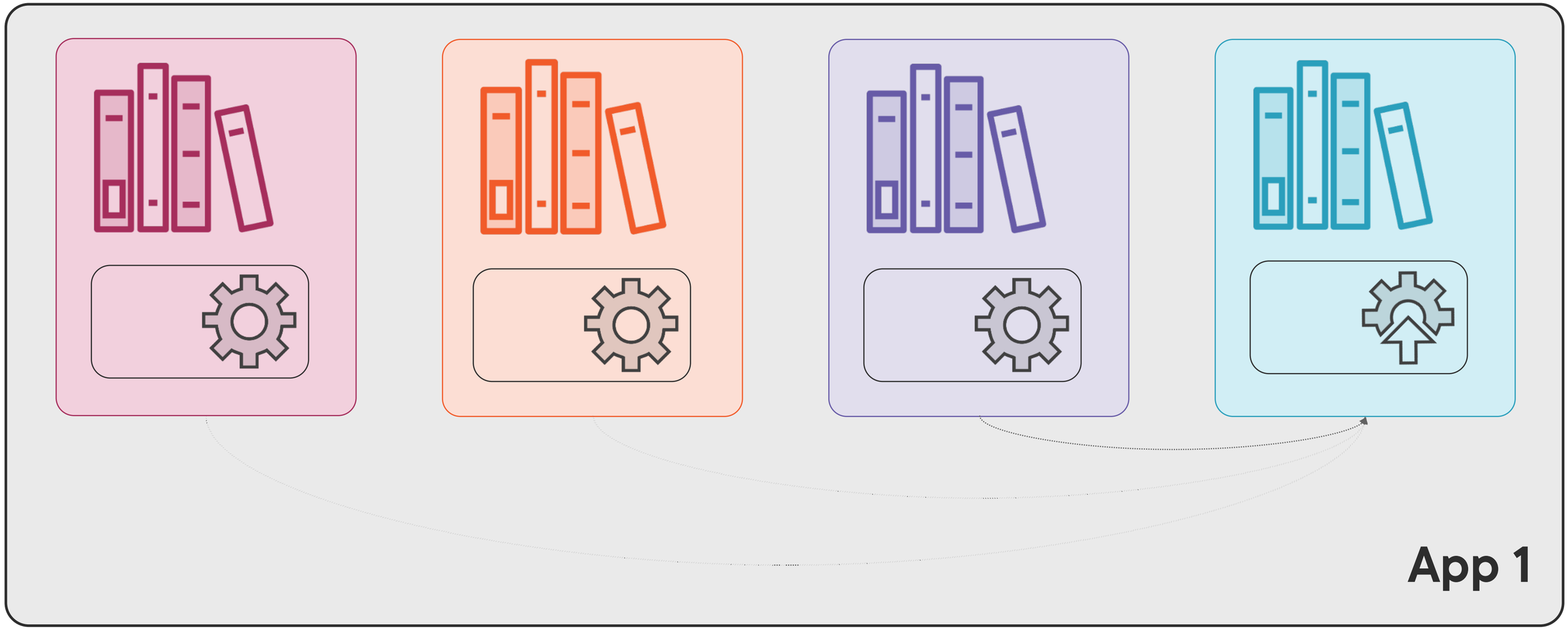
Simple extension methods

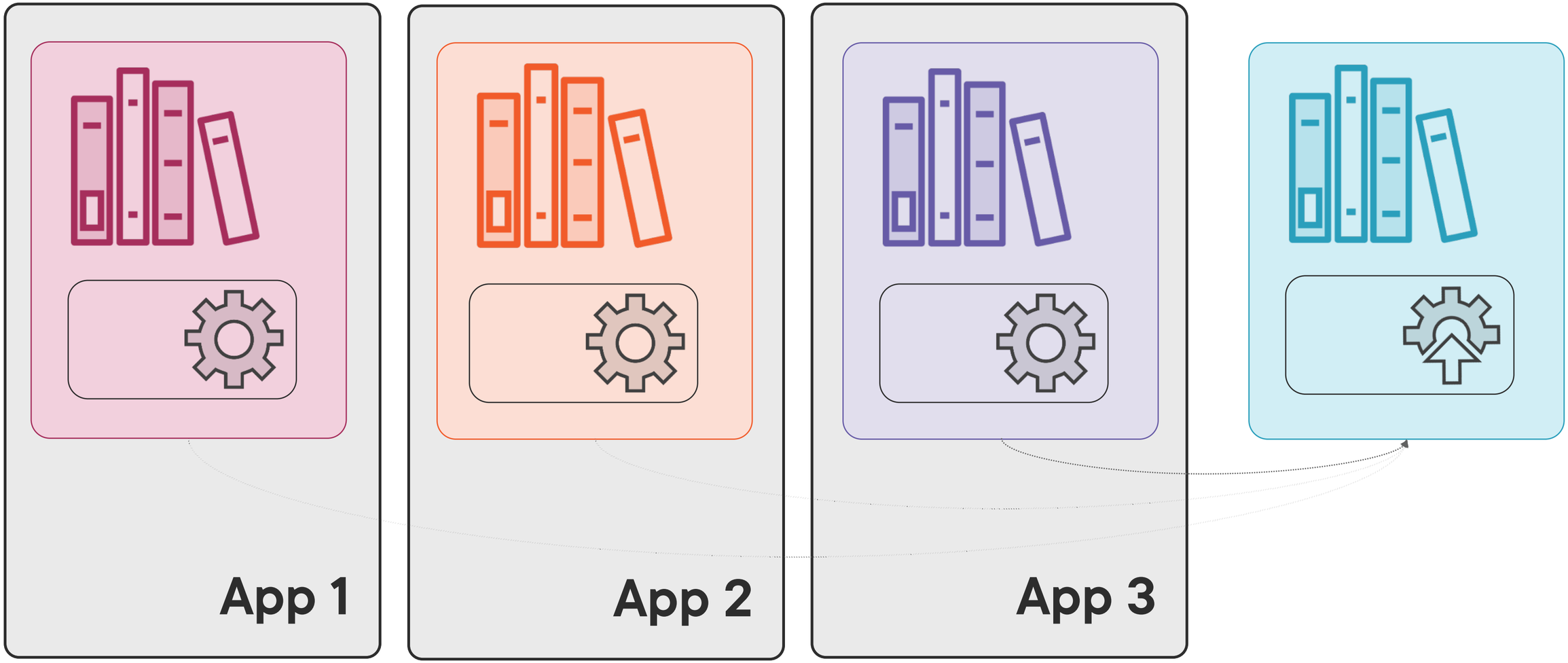
- **Date formatting logic**
- **Problems with bad practices**
- **Value of good practices**

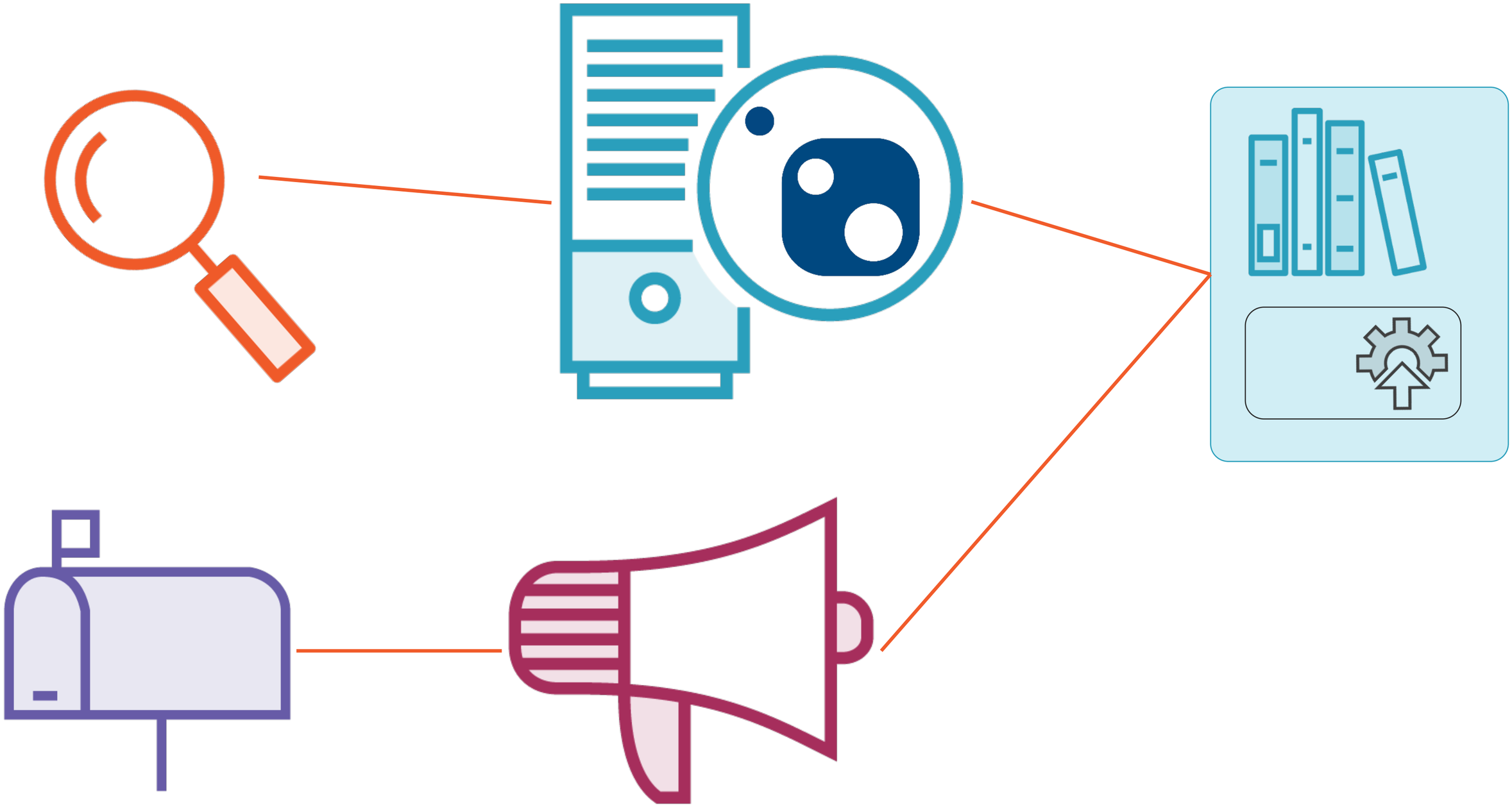


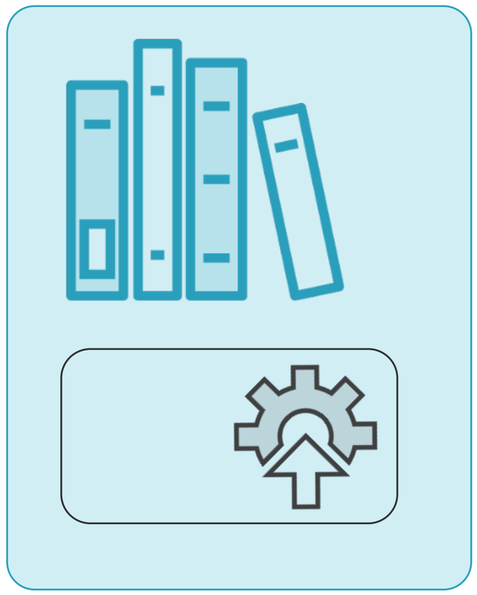
1. Discoverable
2. Deliberate
3. Robust

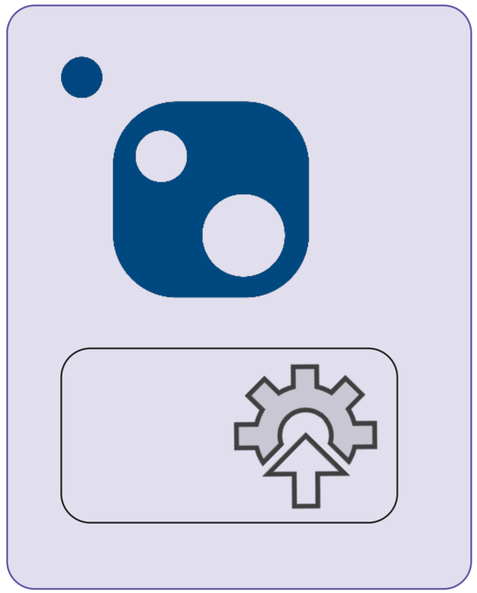



App 1


App 1
App 2
App 3





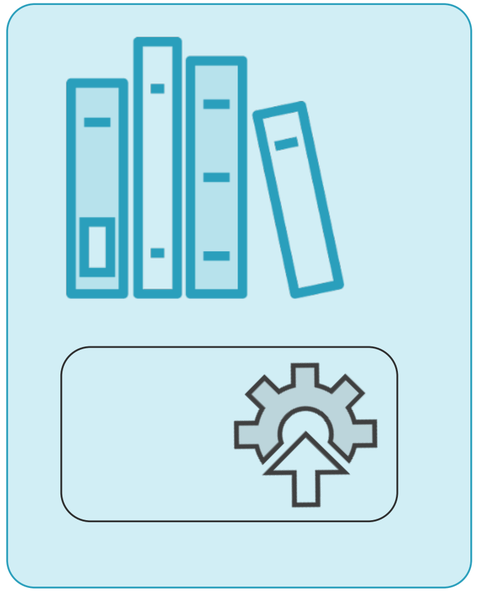


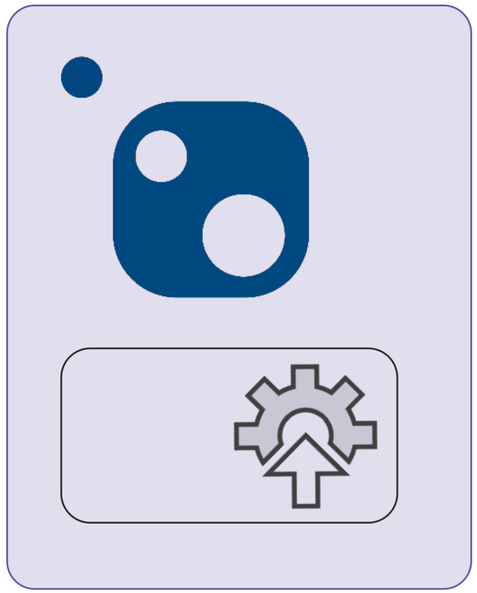
Use different assemblies

Group extensions by scope

Minimize dependencies

Aid discovery



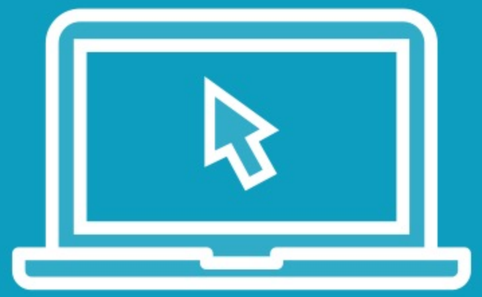


Design for consumers

Adopt target namespace

Target specific types

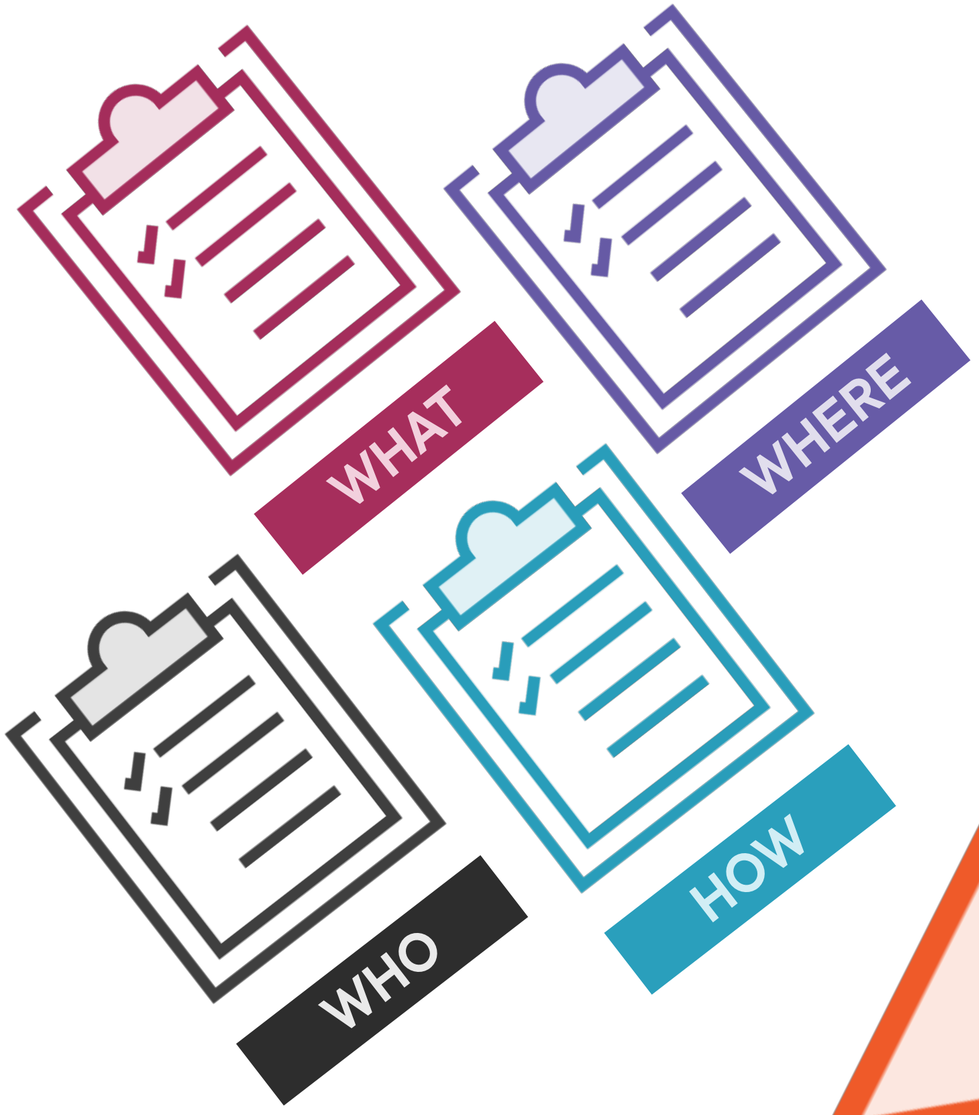
Demo

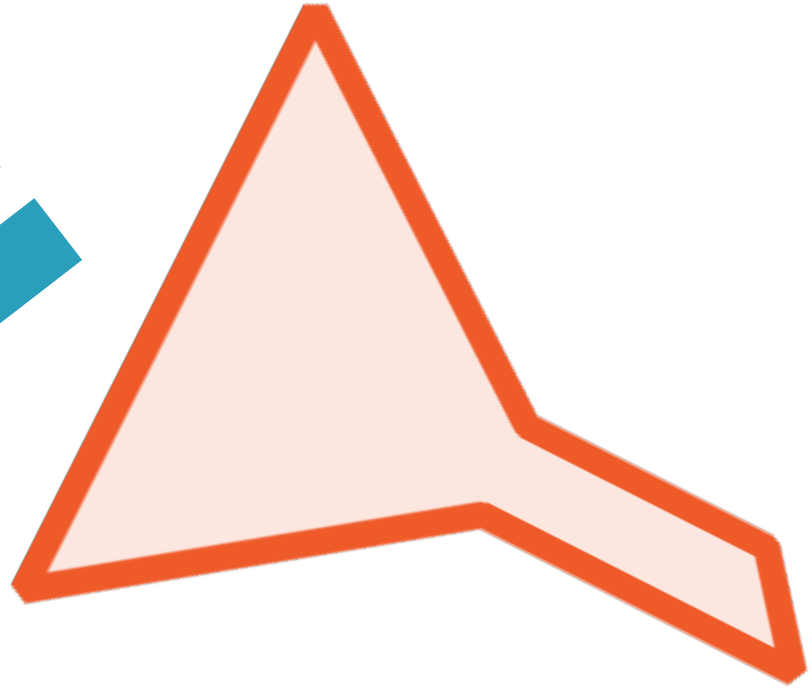


Extension method assemblies by scope

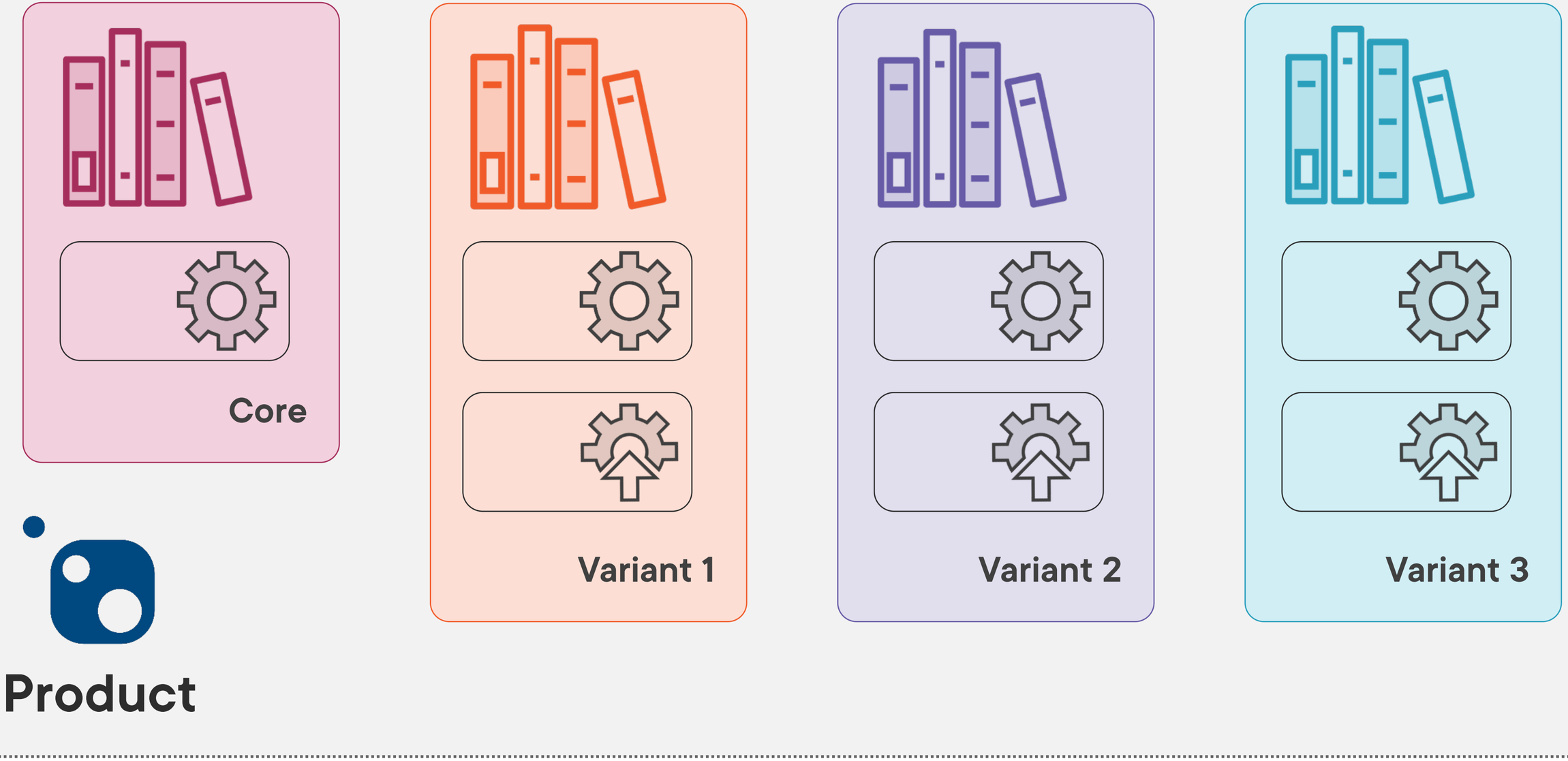
- **Multiple extensions assemblies**
- **Extension methods grouped by target**
- **Best practices for internal libraries**

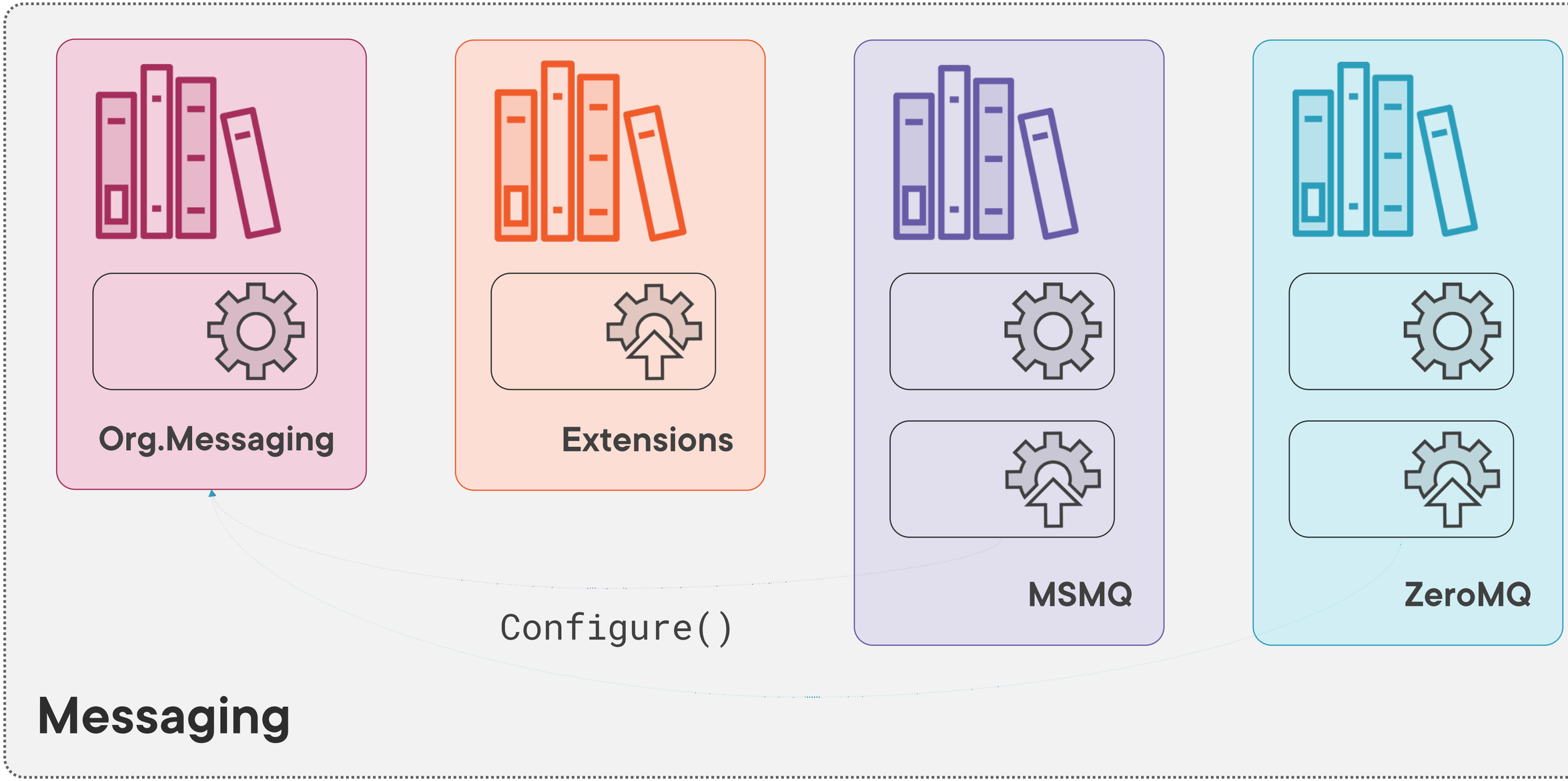
WHAT
WHO
WHERE
HOW



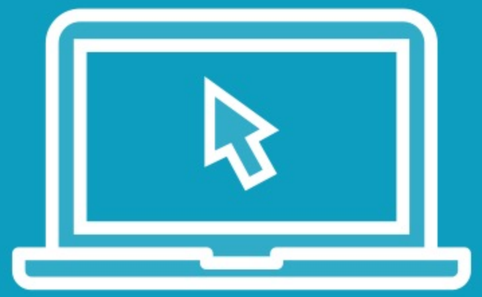


1. Robust
2. Deliberate
3. Discoverable


Core
Variant 1
Variant 2
Variant 3
Product


Org.Messaging
Extensions
MSMQ
ZeroMQ
Configure()
Messaging

Demo



Exploring public extension methods

- **Building a consistent API**
- **Entity Framework Core**
- **Serilog logging framework**

WHAT

WHO

WHERE

HOW

WHAT

Use sparingly

Target specific types

WHERE

Separate assemblies

Consider the namespace

WHO

Convention for sponsor class

Clear method name

HOW

Minimize dependencies

Document clearly

Extension Methods Should Be...

Useful

Easy to discover

Hard to use by mistake

Resilient to future breakages

Up Next:
Building Your Own Extension Method Library