Architectural Snapshot Testing

Andreas Hölzlwimmer

Content

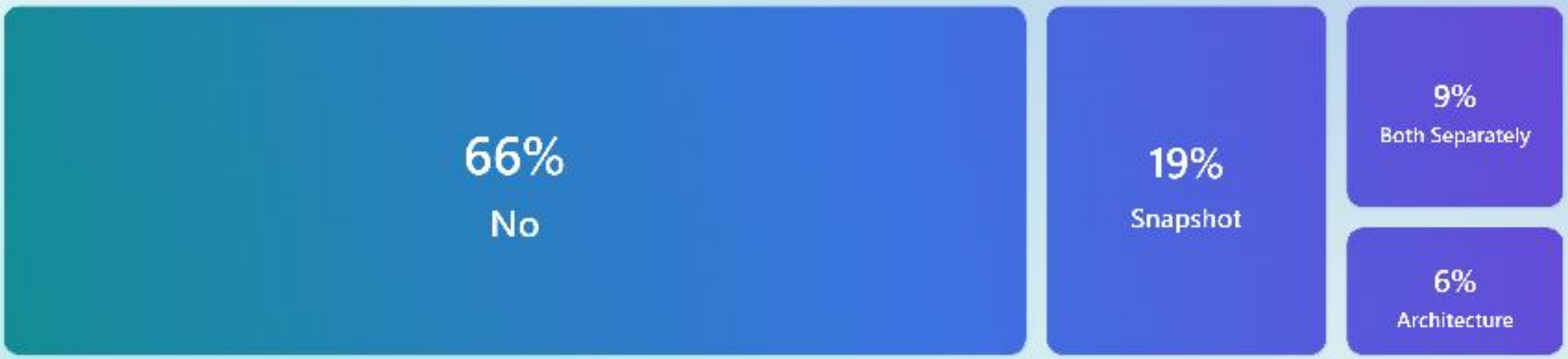
- Motivation
- Poll
- What is Snapshot Testing
- What is Architectural Testing
- Architectural Snapshot Testing
- Real Life Example Output
- Key Take Aways
- Questions

Motivation

- Working at Kontron, Linz
- Project with Fonds Soziales Wien
- Large and growing full stack application
- Changing architectural requirement and technical lead
- Many teams may contribute to codebase
- How do you ensure people conform to the architectural vision of your app?
- “Enshrine” architecture in a ruleset
- Make violations for this ruleset easily visible

32 responses submitted

Have you ever worked with Snapshot or Architecture Testing



Treemap | Bar

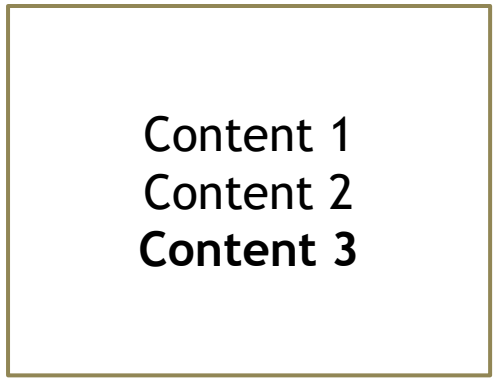
1 of 1

What is Snapshot Testing?

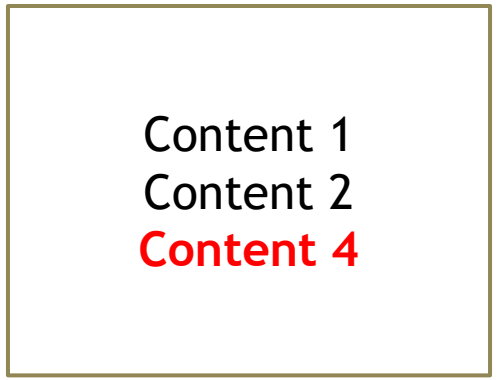
- Produce a state of your software
  - Website State
  - Image
  - Output Text
- Store the produced output
- Compare a later test to your previously recorded output
- Mismatches are easily identifiable
- Mismatches are visual in a local environment

What is Snapshot Testing?

Expected, Verified, Master



Result, Observed



What is Snapshot Testing?

- Create a “Golden Master” or Snapshot
- Objects or Images can be compared
- Objects need to be serializable
- Non-reproducible values are problematic
  - Generated GUIDs
  - Random Values and Image Noise
  - Current Timestamps
- Build Pipelines often do not display diff results

What is Snapshot Testing?

- Example Frameworks
  - Verify
    - https://github.com/VerifyTests/Verify
    - Comparing serialized objects
    - Text output
    - Use your favorite diff tool for test comparison
  - Playwright
    - https://playwright.dev/
    - Testing the UI on top of your .NET backend
    - Allows for screenshot-based comparison between revisions for your UI
    - Built-in Image comparison

What is Snapshot Testing?

- Demo

What is Architectural Testing

- Allows you to extract architectural attributes, e.g.
  - Dependencies
  - Type hierarchy
  - Namespaces
- Define your architecture and layers depending on these attributes
- Define rules for your architecture
- Define tests for each rule

What is Architectural Testing

- https://archunitnet.readthedocs.io/en/stable/guide/#2-quick-start
- Architecture

```
private static readonly Architecture Architecture =
    new ArchLoader().LoadAssemblies(typeof(ExampleClass).Assembly,
    typeof(ForbiddenClass).Assembly).Build();
```

- Layers

```
//use As() to give your variables a custom description
private readonly IObjectProvider<IType> ExampleLayer =
    Types().That().ResideInAssembly("ExampleAssembly").As("Example Layer");

private readonly IObjectProvider<Class> ExampleClasses =
    Classes().That().ImplementInterface("IExampleInterface").As("Example Classes");
```

What is Architectural Testing

- Tests/Rules

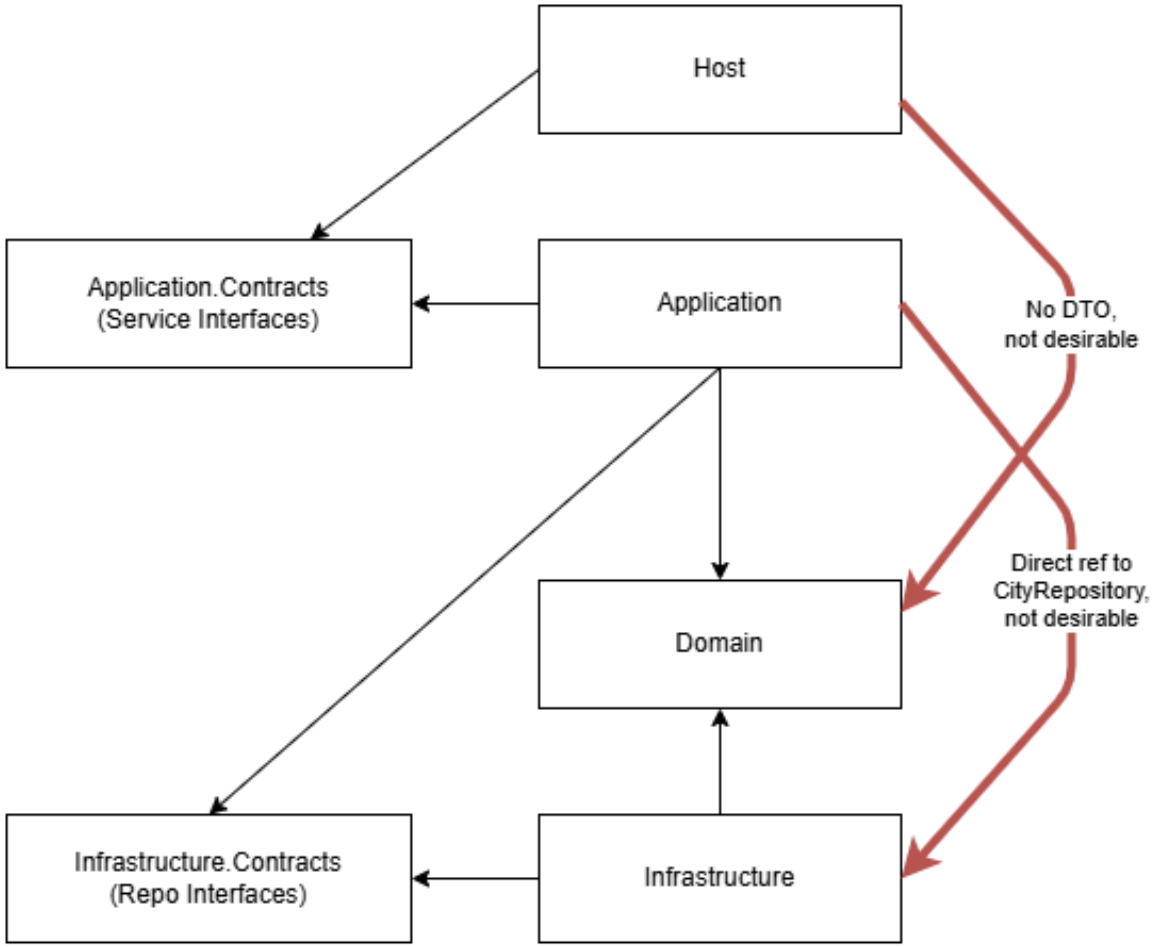
```
IArchRule exampleClassesShouldBeInExampleLayer =
    Classes().That().Are(ExampleClasses).Should().Be(ExampleLayer);
IArchRule forbiddenInterfacesShouldBeInForbiddenLayer =
    Interfaces().That().Are(ForbiddenInterfaces).Should().Be(ForbiddenLayer);

//check if your architecture fulfills your rules
exampleClassesShouldBeInExampleLayer.Check(Architecture);
forbiddenInterfacesShouldBeInForbiddenLayer.Check(Architecture);
```

What is Architectural Testing?

- Ensures that developers are made aware of breaking rules for your architecture
- Ensures that changes to these rules are always made intentionally
- Most effective at the start of a project
- Can require many exceptions in your layering definitions if added to a project without a fully consistent architecture
- Can have issues with generics

What is Architectural Testing?



What is Architectural Testing

- Sample Framework
  - ArchUnitNET
    - https://github.com/TNG/ArchUnitNET
    - .NET fork of java library

What is Architectural Testing

- Demo

Architectural Snapshot Testing

- Define your architectural rules
- Extract Namespaces for layers of your modules
- Define what layers your tested assembly/layer may depend on
- Structured output of your namespaces on a layer-to-layer basis
- Enrich your output with other information

Architectural Snapshot Testing

- Creates a text-based visualization of your architecture
- Can be used as additional architecture documentation
- Synergizes well with some tool support, like “adjust all namespaces”
- Can be used to document technical debt
- Possible to cover both layering and slicing

Architectural Snapshot Testing

- Demo

Real Life Examples

- Verified.json example

Key Takeaways

- Ensure your architectural changes are intentional
- Display your changes in an easily comparable style
- Use your tests as documentation
- Use test results as an architectural ToDo list
- Can support in making targeted changes
- Can show unexpected dependencies (esp. transitive)
- Can require high(er) initial investment

Questions?

Thanks for your attention

- Sample solution on GitHub
  - https://github.com/privatemeta/SnapshotArchitectureTestingSample